Emulation Learning

A Roadmap to Achieving Human-Like Planning in Creative Tasks

by

Alexander A. Spangher

A Dissertation Presented to the

FACULTY OF THE USC GRADUATE SCHOOL

UNIVERSITY OF SOUTHERN CALIFORNIA

In Partial Fulfillment of the

Requirements for the Degree

DOCTOR OF PHILOSOPHY

(COMPUTER SCIENCE)

August 2025

Acknowledgments

Paths are forged by the million doors that open, some in small ways, others in big. I was not a typical student when I started my academic path. I am only where I am today because of the many people who took a chance on me. I want to thank everyone who took that chance, believing me when I said had something interesting to say and to contribute to the fields of NLP and computational journalism.

Thank you to Emilio Ferrara, who first took an interest in my work and welcomed me to USC – Emilio, you helped me grow with kindness and a patience beyond what I deserved, rescued me from nightmares too many times to count, and will always continue to remind me that we should work on misinformation together. Jonathan May, you helped me feel comfortable in the field of NLP and showed me what exuberant cross-disciplinary work looks like; your effortless way of framing a story will always stick in my mind. Nanyun Peng, you mentored me with a brilliant ease, solving problems in real-time and suggesting approaches whose rightness would take me years to see. Thank you James Hamilton, you gave me a desk, an open door at Stanford and a sense of purpose, *twice*, when I found myself in difficult transitions. You opened the door to an academic pathway, in computational journalism, that has forever changed my trajectory.

To everyone at Bloomberg, you have truly changed my life. The three years of PhD fellowship funding you have given me, combined with a *fourth* year simply *because I asked nicely*, I will never, ever forget; it meant that I could pursue the ideas I would have never been able to pursue otherwise, under mentorship to help me thrive. Gideon Mann, Daniel Preoțiuc-Pietro and Amanda Stent and the rest of the committee who reviewed my fellowship application, you took a chance on me. Mark Dredze and Sebastian Gehrmann, you gave me mentorship and advice through the years when I was stuck; you also showed me the humanity of academia. Mark, the way you run a lab and give advice to students

who work with you is a model I hope to emulate. Lingjia Deng, Tzu-Rung Shiang, Yao Ming and Xinyu Hua, you pushed me on in machine learning when I would have taken "easier" routes.

Thank you to all "my" undergrads at Berkeley and Stanford, who took chances on me as a mentor and helped me grow immensely: Michael Vu, Michael Lu, Yiqin Huang, Ethan Hsu, Aaron John, Ines Bouissou and many more, I hope you got a small fraction out of working with me as I got working with you. To my lab and officemates at ISI through the years — Justin Cho, Alex Bisberg, Bijean Ghafouri, Julie Jiang, Emily Chen, Patrick Gerard, David Chu and Myrl Marmarelis, you kept me sane.

I want to give thanks, too, to those who helped me before I even got to my PhD. Many people deserve thanks here and it's difficult to prioritize. Thank you to Thompson Marzagao and Dan Simpson, who first hired me at *The New York Times*. Thank you to Mark Hansen, who helped me survive and thrive through Columbia Journalism School. Thank you to Jose Muanis and Chris Wiggins who managed me throughout those four years and mentored me as much as I would let you. Chris, your sheer brilliance has always inspired me and working under you will always remain some of the best and most exciting years of my life. What a moment we all lived in, to work at the *Times* when we did and play a role in shaping the course of journalism. Chase Davis, Mike Dewar, Sarah Cohen, Dan Simpson, Dana Canedy, James Robinson, Stuward Ward, Nick Ursa, Erica Greene, Timothy Warnock, you made time for me through countless lunch meetings — James, your way of thinking about data, where to collect it, and what to do with it, has shaped my thinking in ways I am still realizing. Even Dean Baquet and Arthur Sulzeburger made time for me, which I will never forget (although perhaps my saying during our meeting that "the print newspaper will soon thrive again" was not the most insightful comment a mid-20 year old could have made).

Berk Ustun, you deserve special thanks. You took me under your wing through so many years, you have been a consistent friend and a generous mentor. You gave me my

first research experience, and broke the ice on helping me write my first paper; I am forever trying to repay you forward. Muhao Chen, Tuhin Chakraborty and Kristina Gligoric, you gave me invaluable advice on the job market. Kevin Knight, Dan Jurafsky, Luke Zettlemoyer, Noah Smith and Yulia Tzvetkov, I am constantly keeping your advice about how to build a lab and motivate others to me in my head. Kevin, you told me to "find the bottlenecks" and get students interested in *ideas*, not just projects. I hope I will put this advice to good use. Diyi Yang, Sanymi Koyejo, Dan Ho and Dongyeop Kang, you are the most recent to have taken a chance on me. I am excited to continue to grow under and with you and see where we go.

Of course, I need to close by thanking those who have provided me emotional support through the years. Thank you to my family, my mom, dad and brother, Lucas. You housed me through the pandemic far longer than you probably expected. Mom, you are an academic inspiration, a source of scientific integrity and a touchstone that I measure all others against. Dad, you taught me to work diligently and habitually, sitting with me daily and teaching me how to practice piano, shooting basketballs with me in the driveway and attending every concert and soccer game we had even if you had just worked a night shift the night before. Lucas, you have shown me joy and creativity beyond what I could have mustered up myself, and taught me to work smart. Caitlin, you taught me to see the humanity in those around us and not to see the world in black-and-white, even in the most trying times; you do it with a grace and effortlessness that, by emulating, *truly* made me a better mentor and a better human being. To all my friends, those in LA (Julian Spector, Eileen Guo, Johnny Wei, Patrick Gerard, Dan Roman, Magali Gruet, Adam Taylor), New York (Lauren Weiss, Josh Steinberg, Julia Witham, Greg Lubin), and elsewhere (i.e. the Durdy Boyz group chat: Julian Spector, Aaron Krolik and Casey Williams) thank you. Julian, you made multiple lists for obvious reasons. For a good meal, great time, good cinema, and a true friendship, thank you.

Table of Contents

Acknowledgments ii

List of Tables ix

List of Figures xiv

Datasets xix

Abstract xxiv

Chapter 1:
Introduction 1
1.1 Current Approaches to Modeling Creative Tasks 2
1.1.1 Pre-training on self-supervised objectives 2
1.1.2 Tuning with hand-labeled data or hand-crafted rewards 3
1.2 Emulation Learning: Learning from Other Humans 5
1.2.1 *Emulation Learning* in the Cognitive Sciences 7
1.2.2 *Emulation Learning* in NLP: Meaning Hierarchy, Action and Discourse 10
1.2.3 Comparison between *EL* and Other Methods in AI 13
1.3 Outline of This Thesis: *Emulating* 4 Steps in Computational Journalism . . . 19
1.3.1 News Finding — An *Observability* Challenge for Emulation Learning's Inverse Model $q_\theta(a|x)$ 20
1.3.2 Source Finding – Trajectory Planning for Emulation Learning's Policy Model $\pi(a|x)$ 21
1.3.3 Story-Structuring – State Realization for Emulation Learning's Transition Model $P(s_{t+1}|a_t, s_t)$ 22
1.3.4 Story Editing – Increased State-Space Observability 23

Chapter 2:
The Observability Challenge in Emulation Learning 25
2.1 *Newsworthiness Prediction*: A Study in How Information is Prioritized 25
2.2 To Cover an Event or Not to Cover an Event? 29
2.2.1 Linking Function M_ψ Gives Observability 29
2.2.2 Local News Coverage: San Francisco Board of Supervisors 30
2.2.3 Probabilistic Relational Models: A General Linking Function 31
2.2.4 Learning a newsworthiness model 34

2.3 Which stories are *more* newsworthy than others? . . . 43
2.3.1 A Pairwise Comparison Model . . . 45
2.3.2 News Homepages Across the World: Our Dataset . . . 47
2.3.3 Newsworthiness Preference Modeling . . . 51
2.3.4 Newsworthiness Prediction with Homepage Preference Models . . . 58
2.3.5 Summary . . . 59
2.4 Chapter Conclusion . . . 60

Chapter 3:
Learning Action Trajectories via Emulation Learning . . . 63
3.1 *Source-Finding*: A Study in how Information Complements . . . 63
3.2 Identifying Sources in News Articles and Testing Compositionality . . . 68
3.2.1 Source Attribution Modeling . . . 69
3.2.2 Insights from Source Analysis . . . 75
3.2.3 Source Compositionality . . . 78
3.3 Does Pretraining Implicitly Learn $\pi(\tau|x)$ for *Source-Finding*? . . . 84
3.3.1 Press Release Dataset . . . 85
3.3.2 Press Release Coverage as Contrastive Summarization . . . 88
3.3.3 LLM-Based Creative Planning . . . 94
3.4 Hierarchical Planning for Emulating *Source-Finding* . . . 101
3.4.1 Task and Dataset Creation . . . 103
3.4.2 Analysis . . . 105
3.4.3 Discourse in Multi-Document Information Retrieval . . . 107
3.4.4 Experiment Setup . . . 110
3.4.5 Discussion . . . 113
3.5 Examining Discourse Schemas for *Source-Finding* . . . 116
3.5.1 *Schema Criticism* as Latent-Plan Selection . . . 118
3.5.2 Building a Silver-Standard Dataset of Different Possible Plans . . . 123
3.5.3 Comparing Schemata . . . 126
3.5.4 Using Schemata Prediction for Explanations . . . 131
3.6 After Source-*Finding*: A System to *Obtain Information* from Sources . . . 133
3.6.1 Grounding Challenges in Human-LLM Dialogues . . . 134
3.6.2 Dataset Processing . . . 136
3.6.3 Analysis . . . 137
3.6.4 NewsInterview: An Interview Game . . . 141
3.6.5 Discussion . . . 148
3.7 Chapter Conclusion . . . 150

Chapter 4:
State-Space Realization in Emulation Learning . . . 152
4.1 *Story-Structuring*: A Study in How Information is Organized . . . 152
4.2 Controlling the Structure of Generated Text . . . 157
4.2.1 Task Definition . . . 158
4.2.2 Our Approach . . . 160
4.2.3 Additional Methodological Approaches . . . 162

4.2.4 Datasets and Schema . . . 166
4.2.5 Implementation Details . . . 166
4.2.6 Experiments . . . 167
4.2.7 Results . . . 168
4.2.8 Discussion . . . 170
4.3 A Beam-Search Based Approach to Generating Structural Outputs . . . 173
4.3.1 *Structural Summarization* Task and Dataset . . . 174
4.3.2 Method . . . 178
4.3.3 Experiments . . . 180
4.3.4 Implementation Details . . . 181
4.4 Classifier Free Guidance . . . 187
4.4.1 Problem Statement . . . 188
4.4.2 Experiments . . . 191
4.4.3 Cost Analysis of CFG: FLOPs and VRAM . . . 198
4.4.4 Explaining the Success of Classifier-Free Guidance . . . 199
4.4.5 Discussion . . . 201
4.5 Underlying Semantics of Structural Discourse Benefits from Multitask Learning 204
4.5.1 Methodology . . . 206
4.5.2 Datasets . . . 207
4.5.3 Experiments and Results . . . 211
4.5.4 Discussion . . . 215
4.6 Structural Discourse and Computational Law . . . 219
4.6.1 A Legal Discourse Schema . . . 221
4.6.2 Dataset Creation . . . 224
4.6.3 Legal Entity and Relational Modeling . . . 229
4.6.4 Results and Discussion . . . 233
4.6.5 Practical Use Case: Census 2020 . . . 234
4.6.6 How does a discourse approach fit within the broader computational law field? . . . 238
4.7 Chapter Conclusion . . . 240

Chapter 5:
State-Space Observability in Emulation Learning . . . 242
5.1 *News-Edits*: A Study in How Information is Updated . . . 242
5.2 Measuring State-Change: The NewsEdits Dataset . . . 248
5.2.1 Dataset Creation . . . 249
5.2.2 Exploratory Analysis . . . 253
5.2.3 Predictive Analysis on NewsEdits . . . 256
5.3 Mapping an *Action-Space* $\mathcal{A}$ Onto News Edits . . . 266
5.3.1 Learning Edit Intentions in Revision Histories . . . 267
5.3.2 Edit Intentions Schema . . . 267
5.3.3 Exploratory Insights . . . 271
5.3.4 Predicting Factual Updates . . . 272
5.3.5 Question Answering with Outdated Documents . . . 276
5.4 Chapter Conclusion . . . 281

Bibliography 282

Glossary 341

Discourse 356

List of Tables

2.1 Results for Probabilistic Relational Model linking policy items with news articles. . . . 33

2.2 Features used for binary news-finding policy, $\pi(a|x)$. . . . 35

2.3 Words most associated with newsworthy policy proposals, meeting speech and public comment. . . . 36

2.4 Newsworthiness of different topics. . . . 36

2.5 Example prompt for news-finding binary policy learning $\pi(a|x)$ 38

2.6 Demonstration that news-finding policy $\pi(a|x)$ learns longer-term newsworthy trends. . . . 39

2.7 Results of news-finding policy training, $\pi(a|x)$: fine-tuning GPT3 on full and ablated versions of the prompt. . . . 40

2.8 Human evaluation of our news-finding policy $\pi(a|x)$ and linking function $q_\theta(a|g)$. . . . 41

2.9 Error analysis of bounding box detection methods for news homepage analysis. 50

2.10 F1 scores for predicting pairwise newsworthiness preference between articles learned via homepage analysis. . . . 53

2.11 Pairwise newsworthiness preference judgments across a sampling of different outlets. . . . 54

2.12 Newsworthiness prediction using homepage models applied to city council policies. . . . 57

3.1 Illustration of the different informational sources used to compose a single news article. . . . 64

3.2 Example sentences from different articles where sources are implicit. 70

3.3 Modeling results for two steps in *source attribution*: *detection* and *identification*. 71

3.4 Corpus-level statistics for *source-attribution* training, test, and silver-standard datasets. 74

3.5 Results for *Source Prediction*, broken into four canonical news topics and 'other.' 77

3.6 Examples of press releases (left) and news articles that cover them in the *PressReleases* corpus. 87

3.7 Efficacy of our document-level NLI classifiers to *capture critical coverage* in news articles covering press releases. 90

3.8 Correlation between doc-level NLI labels and the # sources in the article. . . 91

3.9 Correlation between doc-level NLI labels and the creativity of planning steps journalists took. 91

3.10 Correlation between the level of contradiction between a news article and press release and the types of sources used in the news article. 91

3.11 Results comparing $\pi^{(llm)}$ for source-finding with human decisions π^*. 95

3.12 The 5-point creativity scale that we used to evaluate decisions made while covering press releases. 96

3.13 Distribution of Discourse Types in News Articles. 105

3.14 Results of running different hierarchical retrieval strategies. 109

3.15 Example of different informational sources synthesized in a single news article and possible explanations for their inclusion. 117

3.16 Classification F1, macro-averaged, for each of the 8 schemata. 123

3.17 Results of comparing the schemata against each other, in terms of *conditional perplexity* and *posterior predictive*. 125

3.18 Distribution over source-types with different *Affiliation* tags, by newspaper section. 130

3.19 Top keywords associated with articles favored by stance or affiliation. 132

3.20 Proportion of our validation dataset favored by one schema. 132

3.21 Discourse-Level alignment of LLM-generated questions with human interview questions. 137

3.22 Performance of LLMs as Interviewers, with Ablations. 146

4.1 Illustration that the discourse structure of *Ovid's Unicorn*, an early famous GPT2 generation, is not human-like. 159

4.2 Sample document generated according to our sequentially controlled process.165

4.3 Results from different runs of our sequentially controlled generation process.169

4.4 Overall counts of different categories of data in our dataset. 176

4.5 Statistics on the news article to summary graph, showing the number of edges between post types. 176

4.6 An example news article, an example structural sequence inputted by the user to guide summarization, and an example summary generated. 181

4.7 Comparison of structurally-controlled summarization strategies across different metrics. 184

4.8 Results of general natural language benchmarks of CFG applied to NLP models. 193

4.9 CFG's impact on code-generation benchmarks. 196

4.10 Explaining CFG: correlation between CFG vs. Instruction-Tuning perplexities.199

4.11 Illustration of CFG's effect on re-ording the logit distribution. 200

4.12 Percent increase in sentiment and toxicity under different guidance regimes. 201

4.13 Details about datasets used for multitask discourse classification. 209

4.14 F1-scores of individual class labels in VD2 and Macro-averaged F1-score (Mac.) and Micro F1-score (Mic.). 211

4.15 LR coefficients (β) for each dataset show the effects of each dataset on overall multi-task prediction. 213

4.16 How our legal discourse schema handles edge-cases and extensions. 222

4.17 The prevalence of different discourse units across our annotated dataset. . . 226

4.18 Types of relations common in our legal discourse corpus. 227

4.19 F1 scores shown for span-identification for our 6 primary legal discourse elements. 229

4.20 Relation Detection and Classification F1 score for legal discourse analysis. . 233

5.1 A comparison of revision-history corpora, their size and composition, and the intention of their release, to situate *NewsEdits*. 249

5.2 Illustration of three challenging examples of sentence-matching, showing how our algorithms help us track information change across sentences. . . . 251

5.3 F1 scores on validation data for different sentence-matching algorithms. . . 252

5.4 Summary statistics, after running sentence-matching algorithms, of state-space changes. 253

5.5 % Additions, Deletions or Unchanged sentences that contain Events or Quotes, or have news discourse role. 255

5.6 Selection of top event extracted from edited sentence pairs across article versions. 256

5.7 Baseline model performance for document-level edit-prediction. 260

5.8 Baseline model performance for sentence-level edit-prediction. 261

5.9 Baseline model performance for next-version edit-prediction task. 261

5.10 Predictability of edit patterns for $y^{(2)}$ on documents grouped by topic. 263

5.11 Predictability of $y^{(2)}$ by growth rate. 263

5.12 F1 scores (%) for edit-action prediction. 269

5.13 Counts of coarse-grained semantic edit types, broken out by syntactic categories (for fine-grained counts, see [654]). 272

5.14 Distribution over update-types, across CNN section classifications. 272

5.15 Results for the factual update prediction task. 274

5.16 Linguistic cues characterizing factual updates. 275

5.17 A small sample of sentences in the high-likelihood region of $p(l|s_i, D)$. . . . 275

5.18 LLM Abstention Demonstration. 277

5.19 LLM-QA Abstention Accuracy. 277

5.20 Likelihood of abstaining in the three test cases. 278

5.21 Sample of the most likely fact-update sentences. 279

List of Figures

1.1 Overview diagram of *Emulation Learning*. 5

1.2 *Emulation Learning* View of the State-Action Trajectory. 6

1.3 Stick-building experiment from [88]'s classic study on end-state observation in children. 8

1.4 Hierarchy of linguistic meaning and *emulation learning*'s focus in NLP. 10

1.5 Overview of machine learning methods related to *emulation learning*. 13

1.6 Overview of the main body of this thesis. 20

1.7 The Story Production Pipeline in journalism. 22

2.1 Overview of *computational journalism* focus in Chapter 2: *news-finding*, or discovering newsworthy stories to write about. 25

2.2 State-action trajectory showing observability of the *news-finding* task. 26

2.3 News-finding observability occurs by discovering policy items written about in news articles. 30

2.4 Probabilistic Relational Model for linking policy items with news articles. . 32

2.5 Number of words spoken per meeting for newsworthy policies versus non-newsworthy policies. 37

2.6 Diagram showing that newsworthiness decisions go beyond binary policy decisions. 43

2.7 Diagram showing homepage placement communicates newsworthiness signals. 44

2.8 Summary statistics for the *NewsHomepages* dataset: countries, coverage areas and languages. . . . 48

2.9 Comparison of Kendall's τ rank correlation (on newsworthiness judgements) and SBERT cosine similarity (on articles) across news outlets. . . . 56

3.1 Overview of *computational journalism* focus in Chapter 3: *source-finding*, or discovering sources to support stories. . . . 63

3.2 State-action trajectory showing observability of the *source-finding* task. . . . 65

3.3 The number of sources in an article as it gets republished, based on *NewsEdits* datset. . . . 76

3.4 Diagram illustrating *source-predictability* probes, designed to test whether source action trajectories τ are composable. . . . 78

3.5 Diagram illustrating how we probe implicit policies $\pi^{(llm)}$ for *source-finding* learned via pretraining. . . . 93

3.6 Creativity evaluation results across models and match status. . . . 98

3.7 An overview of our planning-executor process for retrieving sources. . . . 101

3.8 Proportion of sources within each discourse role that occupy High, Medium or Low Centrality in their stories. . . . 105

3.9 Retrieval accuracy scores, broken down by different discourse types. . . . 110

3.10 Diagram illustrating the *conditional perplexity* and *posterior predictive* metrics we introduce for comparing unobserved latent schemas explaining source selection. . . . 118

3.11 Label-sets for different "latent" source-planning schemata compared in our experiments. . . . 121

3.12 Diagram illustrating a walkthrough of the the NewsInterview game; a sandbox for training information-obtaining policies, $\hat{\pi}$. . . . 134

3.13 Comparison of discourse types, between LLM-generated interviews and human interviews, *throughout* the interview. . . . 138

3.14 Distribution of Discourse Roles in Questions, Across Different Prompting Strategies. . . . 139

3.15 Comparison of `gpt-4o`'s performance across different persona types. 147

3.16 Comparison of news-interview rewards gained over time across language models. 148

4.1 Overview of *computational journalism* focus in Chapter 4: *story-structuring*, or synthesizing information into longer narrative forms. 152

4.2 Discourse structure [25] of articles generated via humans or LLMs. 153

4.3 State-action trajectory showing observability of the *story-structuring* task. . . 154

4.4 Diagram showing how discourse analysis can be used to guide the structured of stories. 157

4.5 Diagram illustrating our sequentially controlled *generation* and *editing* processes. 162

4.6 Discriminator performance of our controller, by position. 166

4.7 Comparison of different structural-control methods across different pipelines and hyper-parameters. 170

4.8 Effect of editing across different pipelines and hyper-parameters. 171

4.9 Diagram illustrating our structurally controlled summarization task: summaries across different social media platforms. 173

4.10 Mean Reciprocal Rank (MRR) scores from human preference evaluations of summary quality 185

4.11 Levenshtein Distance and Longest Common Subsequence (LCS) of structurally controlled summaries. 185

4.12 Toy example showing how CFG with *negative prompting* might be used to guide a state-transition. 190

4.13 CFG's impact on chain-of-thought prompting (GSM8K dataset). 194

4.14 CFG's impact on HumanEval code generation. 197

4.15 Evaluator preference for CFG's negative prompting, across guidance strengths.197

4.16 Diagram showing our multi-task sentence-Level classification model, used for different discourse schemata. . . . 208

4.17 Optimal loss coefficients (α) for multi-task training. . . . 212

4.18 Comparison of class-level accuracy vs. label count for three models. . . . 212

4.19 Accuracy of different multi-task approaches. . . . 215

4.20 Confusion matrix for different multitask approaches. . . . 215

4.21 Illustration of discourse roles in legal text. . . . 219

4.22 Illustration of hierarchical discourse structure in legal text. . . . 220

4.23 The conditional likelihood of a target discourse class, given a source discourse class, in our legal discourse corpus. . . . 227

4.24 Sitemap for our website, `statecensuslaws.org`, used by journalists to study discourse laws. . . . 236

4.25 A heatmap of the state of Tennessee, showing laws we discovered would no longer apply based on population counts. . . . 237

5.1 Overview of *computational journalism* focus in Chapter 5: *edit-prediction*, or tracking edits through versions. . . . 242

5.2 Two versions of a news article covering a coup in Myanmar . . . 243

5.3 State-action trajectory showing observability of the *edit-prediction* task. . . . 244

5.4 Three different forms of social learning, pictures from [647] . . . 245

5.5 Number of versions per article, by outlet, in the *NewsEdits* dataset. . . . 248

5.6 Sentence-level changes – Edit, Addition, Deletion and Refactor – between two versions of a news article (merges and splits are a special cases of Edits). 250

5.7 Dynamics of state-space changes across article version number and across the article body. . . . 254

5.8 Architecture diagram for the model used for edit-prediction tasks. . . . 259

5.9 Diagram illustrating edit *actions* that can be inferred after inferring *state space* changes. 266

5.10 Discourse schema for edit *actions* $\mathcal{A}$ across news edit versions. 268

List of Datasets Introduced

Chapter 2: The Observability Challenge in Emulation Learning

- **SFChron Article Corpus**: (∼202,644 articles) Deduplicated *San Francisco Chronicle* articles (2013–2023) used as goal states g for linking and policy learning; collected from Common Crawl, parsed to text, domain/time filtered, de-duplicated, boilerplate removed; *unlabeled* (links produced downstream by PRM). (Section(s) 2.2.2, 2.2.3)
- **SFBOS Policies & Meetings**: (13,089 policies; 27,371 discussions across 410 meetings) San Francisco Board of Supervisors policy items and meeting proceedings with agenda metadata, WhisperX transcripts with diarization, and *Public Comment*; scraped from official portals, audio fetched and transcribed; features (counts, durations, lexical) engineered for modeling. (Section(s) 2.2.2, 2.2.4)
- **NewsHomepages**: (363,340 homepage snapshots; 3,489 outlets) Twice-daily snapshots of homepages for each outlet from 2019–2024; scrapers run via GitHub Actions and actively maintained by a community of 35 activists, journalists and developers. Outlets are selected from 32 countries and 17 languages. We developed a novel layout parser to detect bounding boxes for all articles and generate *weak* pairwise prominence labels (size/position) with a small human-labeled validation set. (Section(s) 2.3, **??**)

Chapter 3: Learning Action Trajectories via Emulation Learning

- **PressRelease** $\leftrightarrow$ **Articles**: (250,224 press releases linked to 656,523 news articles; 1,100 gold-labeled press release–article pairs). We performed two-way hyperlink mining (forward links: article$\rightarrow$press release; backlinks: press release$\rightarrow$article) to link articles with press releases and construct a press release–news graph. We labeled a subset

with manual gold labels for whether the news article *covered* and *challenged* the press release; we developed algorithms to identify 3,000 press release/article pairs with these coverage/challenge patterns. (Sections: 3.3, 3.3.2, 3.4)

- **Source Attribution, Hand Labeled**: (Gold: 1,032 train / 272 test; Silver: 9,051 docs). 10k news articles were sampled from the *NewsEdits* dataset (Section 5.2). Each news article was hand-labeled on the sentence level with: a *source attribution label*, an *information channel* from one of **16 information channels**: (e.g., Quote, Statement, Published Work, Legal filing, Observation). **Gold** human labels were applied by two primary annotators for detection+identification. **Silver** labels were auto-labeled by best model. (Sections: 3.2, 3.4, 3.3)

- **NewsSources**: (600 news articles; 4,922 hand-labeled sources; 8 discourse schemata (3 novel)). Sampeld from *NewsEdits* corpus, sources and their attributable sentences were extracted [1]. Articles were hand-annotated for **Affiliation**, **Role**, and **Identity** schemata – with a small extra set (100 sources in 25 docs) for the other schemata. 2 annotators (former journalist and an undergraduate). **Affiliation** (which group a source belongs to); **Role** (participation of the source in story's main event); **Identity** (reader-identifiability, such as Named vs. Unnamed individual); **Stance** (source's opinion relative to the headline/topic); **NLI** (source's factual relation to the headline/topic); **Argumentation** (argument components, e.g. statistic, testimony, etc.); Van Dijk's **Discourse** (narrative functions such as background, analysis, expectation, history, etc.); Retrieval (information channel, etc.) (Section: 3.5.).

- **Source Retrieval Sandbox** (∼400,000 source snippets) We sample 60,000 news articles from *PressRelease* dataset. For each article, we use methods in Section 3.2.1 to extract all sources. We build a dataset of source "cards", where each card represents a full *packet* of information associated with that source, and we embed these cards in a retrieval database. In addition, we label each "card" with 1-of-8 discourse roles; a centrality

measure {High/Medium/Low}; and narrative purpose. In addition, from linked press releases, we derive initial starting queries. (Sections: 3.4.1.2, 3.4.3)

- **NewsInterview** (487,310 raw transcripts, 45,848 cleaned transcripts; 1 discourse schema with 8 roles). Collected NPR-Media and MediaSum transcripts; removed non-interviews via keyword filtering (e.g., "Sunday Puzzle," ads, commentary), enforced a two-speaker constraint; used `Llama-3.1-70B` to retain only informational interviews. Discourse schema comprises Starting/Ending Remarks, Acknowledgement Statement, Follow-Up Question, Verification Question, Topic-Transition Question, Opinion/Speculation Question, Challenge Question, Broadening Question; developed over ∼50 interviews through three conferencing sessions, achieving $\kappa = .6$ on 10 blind co-annotations; corpus-scale labels produced with `Llama-3.1-70B`. 3 human annotators for schema design; 1 professional journalist validator; 50-interview role-assignment check.

Chapter 4, State-Space Realization in Emulation Learning

- **DiscoSum**: (20,811 news articles; 103,788 platform summaries (66,030 Instagram posts; 18,275 Facebook posts; 8,977 Twitter posts; 10,506 Newsletters); 45,195 article→summary links). A cross-platform news summarization corpus paired professionally written summaries (on Facebook, Instagram, Twitter, newsletters) to news articles from 23 outlets across 10 countries. Summaries are sentence-segmented and labeled with a 5-role clustered schema (Introductory Elements, Contextual Details, Event Narration, Source Attribution, Engagement Directive). *Construction*: we scraped two years of social media feeds to collect full posts, archived newsletters, and retrieved article HTML via the Wayback Machine;. Article↔summary links were formed with SBERT top-k retrieval followed by strict LLM pairwise verification. Newsletter text was segmented into article-level blocks by LLM prompting with iterative verification (>95% accuracy on audits). (Sections: 4.3, 4.3.1.2, 4.3.2, 4.3.4, 4.3.4.1)

- **LegalDiscourse Corpus**: (100,000 state-level laws; 602 hand-labeled laws; 5,386 total annotations (3,715 discourse spans; 1,671 relations)) A large legal-text dataset for analyzing structural discourse in law. Collection comprises 100k state-level laws gathered via 26 custom scrapers for scraping state-level law sites; one scraper for crawling *Justia*. A subset is hand-annotated by 4 annotators with span-level legal discourse segments (SUBJECT, OBJECT, TEST, CONSEQUENCE, EXCEPTION, PROBE, CLASS, DEFINITION) and inter-sentential relations. (Section(s) 4.6)

- **Multitask News-Discourse (Van Dijk–variant)**: (50 articles). A sentence-level labeled news dataset introduced for *multitask* learning over a variation of Van Dijk's discourse schema. Labels are normalized to a consistent taxonomy and paired with auxiliary supervision to study transfer between structural skills and related tasks; intended for training/evaluating inverse models $q_\theta(a \mid g)$ and probing cross-task generalization of structural signals. (Sections: 4.5)

Chapter 5: State-Space Observability in Emulation Learning

- **NewsEdits**: ($\sim$1.2M articles; $\sim$4.6M versions; 22 outlets) Versioned news histories (2006–2021) with sentence-level alignments and state-change ops ADDITION/DELETION/EDIT/REFACTOR; built from periodic snapshotting/archival fetch, HTML normalization, sentence segmentation, asymmetric similarity alignment, and an emission estimator for ops; includes a **gold** sentence-matching set for thresholding/evaluation. (Sections: 5.2, 5.2.1.3, 5.2.2)

- **Edit-Intentions**: (9,200 annotated sentence pairs across 502 news articles) Sentence-pair labels mapping observed changes to the action/intent ontology $\mathcal{A}$ (Factual/Style/-Narrative subtypes); sampled revision pairs from NewsEdits, schema developed in pilot rounds, annotations by trained annotators; final schema used for inverse-model training/eval. (Section(s) 5.3.2.1, 5.3.1)

- **Silver-Labeled Intentions (corpus-wide)**: (full NewsEdits coverage) Large-scale silver intentions produced by running the best inverse model over NewsEdits; filtered by confidence; used for semantic EDA and section/topic cross-tabs. (Section: 5.3.3)

Abstract

We are interested in modeling how humans perform *complex*, *creative* tasks — tasks that occur over multiple steps and have poorly-defined rewards. These tasks are difficult to learn under current paradigms (e.g. *imitation learning* paradigms like language model pretraining demonstrably fail to encourage long-range coherence or learn complex planning; *reward-based learning* requires either large preference datasets or clearly defined rewards, which we lack). Yet, humans are able to infer the rewards, methods and *goals* of other humans *simply through partial observations of their actions and outputs*. This is known, in the cognitive sciences, as *emulation*. We take inspiration from this and introduce a new machine learning new approach, called *emulation learning*. In *emulation learning*, we assume human creative processes progress via trajectories (i.e. $\tau = (\mathbf{a}, \mathbf{s})$) consisting of *actions* (i.e. $\mathbf{a} = a_1, a_2, \ldots$) and *states* (i.e. $\mathbf{s} = s_1, s_2 \ldots$); we assume that we only have observability into the *final, goal state* of a complex human process (i.e. $s_n = g$, e.g., a published news article). Emulation learning progresses in two main steps: (1) *backwards modeling*, where *actions* are inferred via an inverse function (i.e. $q_\theta(\mathbf{a}|g)$), and rewards are inferred via inverse reinforcement learning [2]; and (2) *trajectory modeling*, where a policy function (i.e. $\pi(a|s_0)$) and transition function (i.e. $P(s_{t+1}|s_t, a_t)$) is learned from inferred actions (i.e. $\tilde{a}$) or rewards (i.e. $\tilde{r}$).

We focus our exploration of *emulation learning* to primarily the domain of *computational journalism* and introduce four novel computational approaches to journalistic tasks. In *journalism*, *process* data is scare but *outcome* data is plentiful; decisions made by humans are normative yet difficult to explain, making it an ideal testing ground for *emulation learning*. In Chapter 2, we introduce *news-finding*: how journalists select events to cover. We explore constructing the *inverse function* and confront *observability challenges*: observable goal-states are distant from starting states (i.e. $s_0 = x$), we must construct *observation channels* to make inferences about latent actions. In Chapter 3, we introduce *source-finding*: how

journalists find sources to support their stories. We explore modeling the policy function and confront challenges around composibility, hierarchical modeling, and comparing latent action spaces. In Chapter 4, we introduce *story-structuring*: how journalists assemble facts into narratives. We explore modeling the *state transition* function, *realizing* a sequence of *actions* into a state space (e.g. converting an outline, or document plan, into a final document). Finally, in Chapter 5, we introduce *story-editing*, how stories get updated with new facts. We use observed data about partial state-spaces (i.e. article versions that we can observe) to infer more temporal dynamics about trajectories. *Emulation learning* contains a number of challenges, as we will see. But we are in an era where (1) the need is present for assistive tools (e.g. news deserts exist across the world) and (2) large models can help us make progress in areas towards more sophisticated forms of social and behavioral learning. *Emulation learning* is not only a necessary approach to learning how to perform more sophisticated tasks, it is also a tantalizing approximation of the very *human* process of studying each other and learning from each other's works. In understanding our processes, we might be able to learn more about ourselves.

Preface

Let me start, first, with what motivates me. There are a few moments that stick out in my mind as setting the course of my academic mission, *to understand how humans perform creative tasks and build tools to assist in their workflows – with a specific focus on journalism*.

The first was a lunch meeting I had in 2012 with a mentor, Robert Neer. He was a graduate student at Columbia University, when I was an undergraduate, and he described how fun it was to work for *his* student newspaper, *The Harvard Crimson*. My interest was piqued, and that summer I landed an internship at *Huffington Post*. It was a fun internship, and I was becoming more interested in the pace and energy of newsrooms – magical places, it seemed, where wildly passionate writers came together to practice their craft. One afternoon, I was sitting in the newsroom reading an article published by *The New York Times*[1]. This article was about how, between 2003-2005, Walmart Mexico bribed Mexican officials to build Walmart superstores on historic sites. It contained shocking details and damning interviews — I was dumbfounded and furious. So, it turns out, were other readers. Within days, the governments of both the United States and Mexico announced investigations, Walmart's CEO had stepped down, an internal investigation was launched. Justice, it seemed, had clearly and unequivocally prevailed. Indeed, I would learn, this is not just anecdotal: research has found that newspapers causally reduce government and corporate corruption [3, 4, 5]; $1 spent by a newspaper yields $1,000 in social benefits [6].

Feeling the power of the story, and seeing how it righted a wrong through the simple elegance of words stirring collective action, I was convinced to devote my life to this. In

[1]How Wal-Mart Used Payoffs to Get Its Way in Mexico, by David Barstow and Alejandra Xanic von Bertrab, published Dec. 17, 2012. https://www.nytimes.com/2012/12/18/business/walmart-bribes-teotihuacan.html

2014, after tons of practice and lots of good luck, I got my dream job at *the Times*[2], working as both a journalist and data scientist. Every day for the next 4 years, I walked through those grand doors at 620 8th avenue, past the steel "*The New York Times*" logo and into the glistening glass newsroom; I rode the red elevators and overheard the crisp conversation of reporters and editors. On the quiet 10th floor, I would show visitors the cathedral of Pulitzers and letters written by heads of state, attesting to the skill and importance of our work. Meeting rooms carried names of reporters who lost their lives while reporting – in the wars, famines and strife they covered. Every day felt like a mission to save the world.

Yet, it was becoming impossible to ignore what was happening in the broader landscape of journalism. The *Times* was flourishing[3], but the outlook for most news outlets and magazines around the world was incredibly bleak. Revenues that had historically sustained news outlets were being eaten by Craigslist, Google, Facebook and other large internet companies [6] – the news industry had lost as much as 80% of its advertising revenue to tech giants since the 2000s [7]. This was having devastating consequences. By 2020, approximately two-thirds of newspaper journalists had lost their jobs, and about one-third of newspapers had closed [8]. By 2016, half of U.S. counties had a single local newspaper *at most*, and by 2024, 203 counties had *no* local news outlets at all [9, 10]. Misinformation and propaganda were filling the void. I left the *Times* in 2018 to start my PhD, convinced that something had to be done (and that *I*, specifically, could help). My hope was that we could find tools and techniques to reduce costs and raise revenues. And indeed, there were technologies emerging in 2018, in both artificial intelligence (AI) and Natural Language Processing (NLP) that looked promising. If each tool help a little bit; taken together, I hoped, it could be enough to help newspapers become profitable again, expand into local communities and revitalize the decimated news landscape. Now, at the end of my PhD, we are in a curious moment – AI has progressed farther and faster than seemed conceivable in

[2]I am a, now, 6 year resident of Los Angeles and a proud reader of *the Los Angeles Times*. Readers of *the LATimes* also call it by the abbreviated title *the Times*. Do not take my short-hand reference to *the Times* as municipal favoritism; I just do not want to waste a single word of my readers' attention.

[3]Mainly on strength of it's subscription business.

2018. Almost no one needs convincing that AI tools can, indeed, save newsrooms time and money, but everyone needs convincing that they can do it *well*.

What does this mean to *do journalism well*? Is it to publish the article that will topple the government, drive the clicks, win the prize? Sometimes, it is more subtle than that. Let me start with a motivating example. A recent story published by the *New York Times* tells how Saudi Arabia donated two leopards to the Smithsonian Zoo in Washington DC[4]. It is interesting, light and yet revelatory, weaving in the personal dimension of politics. The author, in an interview, said that he found the story at the end of a White House press release[5] buried in between massive economic deals ($600 billion investment commitment) and defense agreements ($142 billion defense sales). What makes it stand out, though, are the sources used to tell the story. He used an eclectic mix: Brandie Smith, director of the Smithsonian National Zoo (she was directly involved in the negotiations); Roger Stone, a former presidential advisor (for insights into Trump's thinking); the Holy Bible (to provide cultural context – "mountains of leopards", in Song of Solomon 4:8); and then, Joseph Maldonado or Joe Exotic, subject of the documentary *Tiger King* (for expertise on big cats). This is clearly a *good* article: enjoyable, memorable, well-crafted. What makes it good? Imagine we want to build a system to help journalists *find stories* and *find sources to support these stories* (two tasks that we will consider in depth in this thesis). Would this system have ever found *this* story, buried in the press release, or recommended *these* sources? Can traditional quantitative metrics explain why these sources together create a *good* story (e.g. diversity, factuality [11, 12, 13, 14, 15, 16])? These are not frivolous academic questions — they are at the core of what it means to support human creative activities.

[4]Leopards on the Potomac! Trump Is Delighted by Deal With Saudis for Rare Cats. By Shawn McCreesh, published June 4, 2025. https://www.nytimes.com/2025/06/04/us/politics/leopards-trump-saudi.html

[5]https://www.whitehouse.gov/fact-sheets/2025/05/fact-sheet-president-donald-j-trump-secures-historic-600-billion-investment-commitment-in-saudi-arabia/

Creativity, Symbolic Systems and Norms

In this thesis, we seek to study exactly these kinds of *complex, creative* tasks – of which the creation of news is just one. These tasks are intensive and multi-step; they are subjective (i.e. it's unclear when a creative output is good or bad) and *humanistic* (i.e. they are associated more with communicative social processes than physical or technical processes). Examples of ways AI can aid in such tasks include: in news, as mentioned, a system that detects a *newsworthy* story in a press release and *finds relevant sources* [17, 18]; in writing, an assistant that helps write *well-structured* tweets [19]; in music, a generative *music* model that helps composers ideate with different songs [20]; in law, a patent analyzer that establishes an idea's *novelty* (or lack thereof) [21].

Newsworthiness, *well-structuredness*, *musicality* and *novelty*: these are all abstract cultural *metrics*, or *norms*, driving creative tasks, and they resist simple definitions. What are they and, importantly for our purposes, how are they created and understood? Take *musicality*: Susan Langer, in her seminal 1942 work *Philosophy in a New Key* [22], posits a process by which musicality arises within a culture. "All of our sense-data is symbolic", she declares – we interpret the world using symbols and express these interpretations to each other, forming shared symbolic vocabularies. Composers write songs, she writes, following a process: once a symbolic vocabulary is established (e.g. combinations of tones, rhythms and dynamics), composers then choose sequences of symbols from this vocabulary (e.g. themes, melodies), and decide, via higher-level *actions*, how to string these sequences together (e.g. adhering to compositional forms). Accumulations of symbolic rules, or *norms*, established within the composers' social group, or culture, inform *musicality*. How *musical*, or "good", a composition is, she writes, depends how well it's symbolic sequences capture communicative intent while acknowledging these norms.

Do these observations apply to our other examples? Galtung and Ruge, in their 1965 work *The Structure of Foreign News* [23], similarly identify a symbolic process by which the *newsworthiness* of events are established (i.e. an event is newsworthy when it is an

"infrequent event", "meaningful event", "national-level event", etc.). Journalists, they write, *select* events containing these symbols; writers and editors *construct* a story that emphasizes these symbols. Newsrooms "follow steps in the news chain [from journalists → writers → editors] where each step anticipates the reaction of the next step in the chain," and interprets and re-expresses the symbolically newsworthy aspects of the event. As Stuart Hall writes in *Writings on the Media*, "'News values' are one of the most opaque and deep structures of meaning in modern society. All 'true journalists' are supposed to possess it: few can or are willing to identify and define it." [24]. We can see the same observations being made by theorists of *novelty* and writing *structure*: Van Dijk in *News as Discourse* [25], identifies structural elements in writing (e.g. "lead", "background", use of "data"), how they arise over time, are combined in news stories, and perceived by readers.

So creative acts, and their associated *metrics* (or norms) are symbolic generative processes, based on shared, emergent vocabularies. By focusing on symbolic processes, we frame *creative work*, here, not as the product of sudden, inexplicable sparks, but *quasi-linguistic processes*. We open the door to studying creative work like linguistics has been studied: at large scale, observationally and computationally, with the modern machinery of language modeling. Even the wild creativity that yields a story about leopards and the White House, sourcing Tiger King himself, can be computationally understood and supported, if we understand how it came to be.

Emulation Learning: An New Approach to Studying Creative Processes

Can we understand creative symbolic systems and the norms that govern their usage, even if these systems are largely unobservable [24]? And if these norms and symbols are not fully known to a composer, a journalist, or a writer, how might we hope to build models that understand them? This thesis endeavors to establish framework to answer these questions, which I call *emulation learning*. I will introduce emulation learning more formally in Section 1.2, but, on a high-level, we'll seek to (1) study finished creative *works* – or the *end-state* of a

human creative process, (2) infer the process of creation, including inferring the *unobserved actions* that gave rise to the finished work and (3) use these inferences to understand human norms, rewards and decision-making. *Emulation learning* is, at the core, concerned with *norm-finding* (as opposed to *norm-breaking*) and considers, first-and-foremost, the actions of the creator (as opposed to their thoughts, intentions and directives). We will close this section discussing some tensions at the heart of our study creativity.

Firstly, the focus on norm-finding might strike many readers as odd. Norm-finding involves learning and applying the symbolic rules that guide creative works in a culture — for example, mastering the inverted pyramid in journalism [26] or Sonata-form in classical music [27]. To many readers, creativity might seem to lie in *newness* of the creative work and how it deliberately deviates from those rules: indeed, the *Tiger King* example is interesting because of how it *stands out* and subverts our expectations. Should we not be interested in developing systems that *break* norms, rather find and adhere to them? My focus is on norm-finding, for several reasons. Modeling norm-finding is the more tractable: works that follow norms are vastly more present than works that breaks them (as we will show repeatedly throughout this thesis — in Sections 2.2.4, 3.2.3, 3.4, 5.2.3, 5.3.4 — *creative actions are predictable* and many creative acts do *not* break norms). Moreso, insights from norm-finding can later *inform* norm-breaking. Norm-breaking operates within a field of shared reference: one cannot meaningfully break a norm without knowing (and signaling that one knows) the norm being broken or which norms are stable enough to break.

Secondly, *emulation's* focus on inferring creator's *actions*, first, rather than their internal monologues, intentions and, deeper still, subjective experiences might also seem misguided. These deeper influences doubtlessly impact creators' work and explicitly modeling them at the outset may improve emulation. In practice, because the actions we infer are *unobserved*, *emulation* mixes reasoning and action together, like other modern frameworks [28]. Theoretically, though, emulation's approach is intentionally *behavioral* [29, 30] — we posit that *actions* are more observable and predictable than intentions; and, like classic

production theories for communication [31, 32], modeling actions yields more stable pathways towards inferring higher-level intentions. Emulation is inspired, too, by advances in computational language modeling, where evidence similarly suggests that predicting actions (e.g. the next word chosen) can give us *bases (base models)* [33] to more explicitly model rewards, intentions and motivations.[6]

Finally, emulation's emphasis on studying the *end-state* of the creative work, rather than seeking to observe the process of creation, might seem misguided as well: much research in Human-Computer Interaction (HCI) seeks to observe end-to-end human processes with the awareness that many steps might be *hidden* from the final output [34, 35, 36, 37, 38, 39, 40]. However, end-state analysis is a key method of social learning that has a long basis in cognitive sciences research and, we believe, is actually *understudied* in computational domains. We will discuss *emulation* as a social learning paradigm extensively in Section 1.2.1, but here, we will make our point by returning to the study of creativity and creative processes. Getzels and Csikszentmihalyi, in their study of the problem-finding aspects of creativity [41], identify a *crucial* creative stage to be the one in which creators define the *end-goal*, or what is worth working on.[7] By focusing on the end-state, or the goal-state, and inferring the decisions that lead to it, emulation seeks to understand how creators navigate the space of possibilities to define, select and pursue problems. This focus provides a

[6] Kevin Knight, my academic grandfather, once described a scene capturing the early incredulity around language modeling. Kevin and another professor were in a DARPA meeting. The other professor was an "old school grammar guy", in Kevin's words, and the topic came up about whether a predictive model for language could be built. "We're supposed to be mathematically capturing what is and isn't a legal sentence of English [Chomsky]," the MIT professor said, "we're not supposed to be predicting the next word that's going to come out of someone's mouth [Skinner]". Then, in front of two program officers dressed in military gear, he took off his shirt and said: "DARPA is funding mind-reading! Mind-reading is impossible! Why are you doing this, DARPA?" This action was, apparently, so inexplicable as to prove the futility of predicting thought, behavior and language. Clearly, our modern language models beg to differ. The same emotional reactions, I find, surround discussions about computational creativity. By explicitly casting creative acts as symbolic processes, I hope we can one day make the same kind of progress that we have made with language modeling.

[7] In more detail, Getzels and Csikszentmihalyi studied artists painting still-lives in studios. They found that some artists (a) took considerably more time than others to place which objects in their drawings (b) indicated, in post-interview comments, a more searching attitude in their work and (c) took longer to have their basic concepts become clear. After following these two groups of artists through their careers, they found that the artists that spent more effort *defining the problem* they wished to draw produced art judged to have more merit and had more professional success in life.

principled way to model creativity in terms of culturally grounded choices.

Before I end this preface, I want to step away from theorizing and return to the practical, to explain why this work is important. It's not obvious why we would need AI systems to help journalists, writers and composers. Are these jobs not already in danger of disappearing? Are there not already many humans wanting, hoping, dreaming of doing these jobs well? To return to the question posed in the beginning of this preface: *should we use AI models to assist humans?* I believe the answer, in many cases, is "yes". Human creative processes are formed and emerge via similar processes and, as I will show repeatedly in this thesis (e.g. in Sections 2.2.4, 3.3, 4.2), are poorly understood by current models. A unified approach to approaching these problems, and a way of casting them in the same framework, can help us learn more about the symbolic systems that drive our worlds, the human behaviors that create them, and the tools we need to advance.

Chapter 1

Introduction

I am interested primarily in answering the following question: can we model *complex, creative tasks* with high enough performance that we can build practically useful tools? I defined complex, creative tasks in the Preface, but to recap: I define a *complex* task as a task that involves multiple steps (e.g. like investigating a claim, composing a piece or writing a news article). Let's define a *creative* task as a task with a poorly-defined goal or output, that is usually culturally determined and/or clarified by the human executing the task.

More formally, let S be a space of possible states and A a space of possible actions. A task is specified by an initial state $s_0 \in S$ and a finite sequence of actions $(a_1, \ldots, a_T) \in A^T$ which, under transition dynamics $s_{t+1} \sim p(s_{t+1}|s_t, a_t) \quad \text{for } t = 1, \ldots, T,$ produces the trajectory $(s_1, \ldots, s_T)$. We equip the task with a reward function $R\colon S \to \mathbb{R},$ which assigns a scalar payoff $R(s_T)$ upon completion. We call the task *complex* if $T > 1$, i.e. it requires multiple steps to reach its terminal state s_T. We call the task *creative* if **neither the goal state,** g (i.e. the desired outcome(s) $g \subseteq G$ of acceptable terminal states) **nor the reward function** R **assessing the quality of terminal states, are fully specified.** Taken together, a complex, creative task is a task that requires a multi-step action sequence $(a_1, \ldots, a_T)$ driving the system from s_0 to s_T, and has poorly defined rewards $R(s_T)$ and goals G. Our modeling goals for creative tasks are either to: (a) learn a policy model $\pi : S \to A$ that, at each state s_t, chooses action $a_t \sim \pi(a_t|s_t)$ so as to approximate the human creative strategies. Or (b) to recover the reward model $R : S \to \mathbb{R}$ – which encodes latent human preferences over

terminal states – or the goal set $G \subseteq S$ of acceptable outcomes. A learned policy π enables us to build multi-step workflows, while a learned reward function R or goal set G provides explicit insight into the characteristic of creative tasks.

1.1 Current Approaches to Modeling Creative Tasks

Can complex, creative tasks be modeled? As discussed in the Preface, framing these tasks as symbolic processes [22] allows us to connect them more explicitly to computational advances in language modeling. I now briefly summarize the two dominant methods that researchers in NLP are using to model human symbolic processes.

1.1.1 Pre-training on self-supervised objectives

Firstly, researchers seek to implicitly model human creative processes by modeling observed text using *pre-training* objectives: they train large models on huge corpora using relatively simple self-supervised learning (SSL) tasks. A standard SSL task in natural language processing, my primary domain of study, is *next-token prediction*, which is defined as follows. Next-token prediction seeks to train a large language model (LLM) to predict the sequence of tokens observed in a large corpus, $\mathcal{D}$. Formally, let $\mathcal{V}$ be a finite vocabulary of tokens, and let $D = [(x_{1,1}, x_{1,2}, \ldots, x_{1,T_1}), (x_{2,1}, x_{2,2}, \ldots, x_{2,T_2})...]$ be a sequence of tokens in a sequence of documents such that $x_{ij} \in \mathcal{V}$ for all x_{ij}. The next-token prediction task is defined as learning a conditional probability distribution $P_\theta(x_{ij} \mid x_{i,<j})$.

The *next-word* prediction task implicitly benefits from modeling unseen thoughts, intentions and actions taken by humans while generating $\mathcal{D}$ (i.e. modeling what the writer intended and what the writer did while writing – their *thoughts* and *actions* – can help to predict the next word in a document). Researchers have shown that knowledge of many complex, creative tasks *is* learned via pretraining [42, 43], and these can be elicited with the right prompt. Prompting has been shown to elicit research-type workflows, like browser-

aided search and citation generation [44] and research tools [45, 46]; interactive systems for creative writing assistance [47, 48, 49, 50, 51]; simulated agents [52, 53]; even social norms and moral judgments have been learned [54, 55, 56, 57]. Prompt-based approaches have also emerged in more domain-specific tasks, like those related to journalism [58, 59].[1].

And yet, I present evidence in this thesis that, for the tasks we study, pretrained models underperform fine-tuned models (and, interestingly, humans performing the same tasks) to such a degree, that they show scant evidence of having modeled the tasks' underlying norms and goals. This accords with an emerging consensus shows that prompt-based approaches *alone* have limits across a range of norm-driven creative tasks [60, 61, 62, 63]. This is unsurprising: while some creative action sequences might help the SSL objective, there are likely others that are too complex, diffuse, or present in a small section of the training corpus, and are swamped out by many clearer sources of predictive signal (e.g. topic, syntax, word-distributions).

1.1.2 Tuning with hand-labeled data or hand-crafted rewards

Secondly, researchers seek to model complex, creative tasks *post-training*, a combination of techniques that include: supervised fine-tuning (SFT) on hand-crafted training datasets, distillation, inference-time techniques (e.g. test-time steering) and, most importantly, reinforcement learning (RL). These techniques rely on access to one of two things: a high-quality hand-crafted dataset or a feedback model (i.e. reward model).

In more detail, RL-based approaches typically redefine the next-token prediction model as a *policy* model $\pi_\theta(a_{t+1} \mid a_t, s_t)$, where the token to sample, now, is labeled as an *action*, a_{t+1} and the *state*, s_t contains the previously generated tokens, $a_1, \ldots a_t$ [64, 65]. A complete generated output corresponds to a trajectory $\tau = ((a_1, s_1), (a_2, s_2), \ldots, (a_T, s_T)$, sampled from the model's autoregressive distribution $\pi_\theta(\tau) = \prod_{t=1}^{T} \pi_\theta(a_t \mid a_{t-1}, s_{t-1})$. We maintain

[1]The publisher of the *Palm Springs Post*, a small newspaper in Palm Springs, California, has described running large parts of the reporting process using prompts to detect newsworthiness. `https://www.fastcompany.com/90954997/how-local-news-is-using-ai-to-tell-better-stories-and-hold-leaders-accountable`

the predictive framing of the last section, where a refers to a token, but we note that "token" here often refers to more than just words – a refers to actions, thoughts and other generations from the language model. After generating a full trajectory, a scalar reward $R(\tau) \in \mathbb{R}$ is assigned and the training objective is to maximize the expected reward over trajectories: $\mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)]$. In this way RL frameworks allow the model to improve its behavior based on a broad variety of feedback signals, even in the absence of gold labels. Researchers have found great promise in this direction for tasks with enough *paired preference data* that reward models can be trained, or tasks with *verifiable rewards* (e.g. math or coding) where reward models are simpler. Reward-based learning has *also* shown great promise in inducing greater *reasoning* (i.e. latent variable modeling [66, 67]) capabilities in language models [68, 69, 70] leading to some tantalizing demonstrations of human-like behavior, like the famous "a-ha" moment observed in Deepseek-R1's reasoning threads [71]. Beyond simply improving performance on downstream tasks, reasoning threads might capture deeper representations of creative workflows [72, 73] recreating generative linguistic processes [74, 75, 76] (although some have argued against this interpretation [77, 78]). Reward-based learning has subsequently been applied to many more creative and open-ended tasks (e.g. in search [13, 79, 80, 81], web-browsing [82, 83, 44] and writing [84, 85, 86]). And yet, the applicability of reward-based approaches faces fundamental limitations for most creative tasks. As I will show in more detail in the body of this thesis, many of the tasks we will consider lack fundamental components that make reward-learning possible. *Lack of reward function:* Creative tasks, by definition[2], have poorly defined or subjective goals, and as such defy simple, heuristic reward functions that are typically used for RL training. So, it is difficult to specify a-priori a clear reward function to determine what makes a creative work good or bad. *Lack of hand-labeled training data*: While data can be labeled for individual tasks, it is prohibitively expensive to label enough data for *all* creative tasks.

[2]By the *problem-finding* definition of creativity [41], the creative act is the act of defining the goals of the task, or the problem to be studied (or, as Einstein says in *The Evolution of Physics*, "The formulation of a problem is often more essential than its solution which may be merely a matter of mathematical or experimental skill").

1.2 Emulation Learning: Learning from Other Humans

To recap, the two approaches typically used to adapt LLMs into creative workflows fall short. The first, *pre-training* – training models on self-supervised learning objective like next-word prediction – often fails to capture higher-order factors in the creative process, like intent, norms, and actions. The second, *post-training* – training models using labeled data or hand-crafted reward models – fails due to lack of training data or unclear rewards.

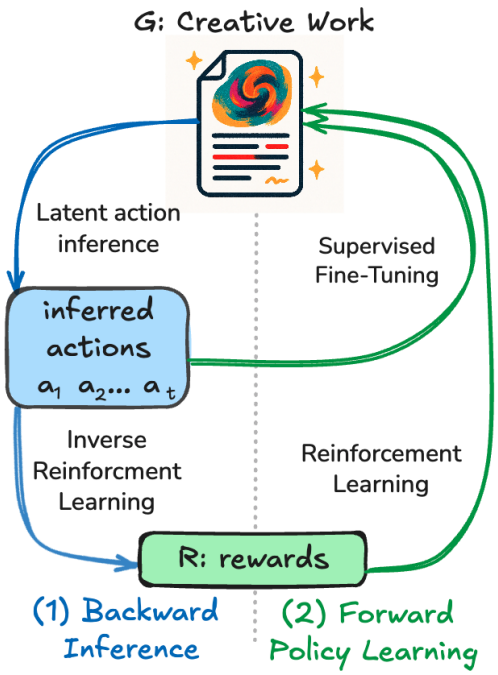


Figure 1.1: *Emulation Learning*: First, a creative work is analyzed to infer the actions that generated it, which can then be used to learn a reward function. These are then used to train models to aid in creative workflows (either via supervised fine-tuning or reinforcement learning).

I will now introduce *Emulation Learning* (EL) a novel and generalist framework for modeling complex creative tasks. EL takes as the object of its study *finished creative works* (i.e. news articles, musical pieces, creative stories) and performs *latent action inference* to infer the actions or steps taken in producing it (i.e. the process of creation: actions, thoughts and reasoning performed by the creator, forming each creative step). It then uses these inferences as training data, either to directly supervise a policy model or to learn the reward function (i.e. the overall guiding norms, motivations and intentions driving the process). Finally, it uses these data and/or reward function to drive learning – either through supervised fine-tuning or reinforcement learning. This flow is shown in Figure 1.1. Formally, let a creative task be modeled as a Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ is the space of possible states a creative work could be in (e.g. *Story idea*, *First Draft* as shown in Figure 1.6), $\mathcal{A}$ is the space of creative actions (e.g. *Call source A*, *Find Background*), $P(s' \mid s, a)$ is the (possibly unknown) transition

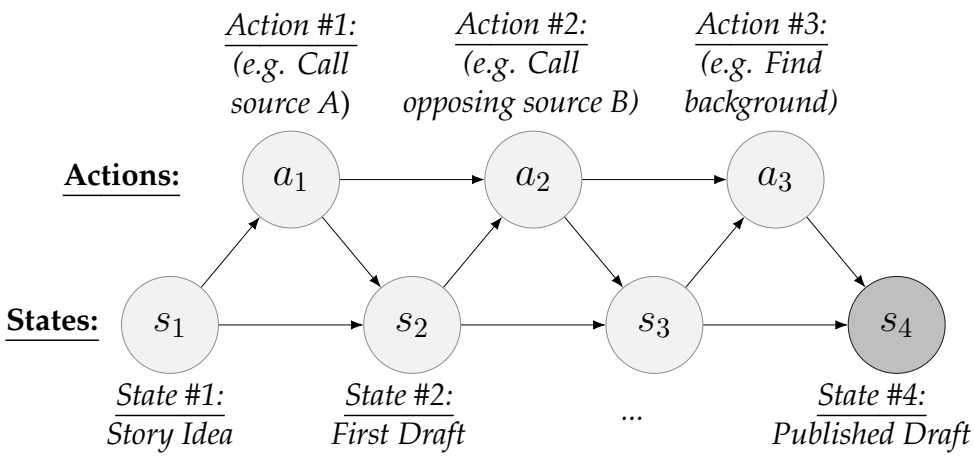


Figure 1.2: *Latent Action Inference* in *Emulation Learning*: A hypothetical state-action trajectory, showing a plausible inferences for the actions that would have generated the final state. Right now, a broad *news production* trajectory τ is shown. In Chapters 2-5, we will return repeatedly to this figure and framing and refine it. Crucially, in this notation, only *gray* states (i.e. the final state, here) is observed; all *white* states (i.e. all other states and actions, here) are unobserved.

function describing how actions transform states, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is an unknown reward function encoding the norms, motivations, and intentions of the creator. In standard RL, we observe trajectories $\tau = (s_0, a_0, s_1, a_1, \ldots, s_T)$ with associated rewards r_t, and learn a policy $\pi(a \mid s)$ that maximizes expected return. In *Emulation Learning* (EL), we instead observe only *goal states* $G \subset \mathcal{S}$, where each $g \in G$ corresponds to a finished creative work. The underlying action sequences and rewards that produced each g are unobserved.

Each g arises from an unobserved creative trajectory $\tau_g = (s_0, a_0, \ldots, s_T = g)$ under some expert policy π^* and reward function r^*. The objective of EL is *latent policy inference*: recover $\hat{\pi} \approx \pi^*$. Practically, we decompose the learning process into two stages:

1. **Stage 1: Backwards Inference:**

   (a) *Latent Action Inference:* Given goal-state g, reconstruct a plausible trajectory $\tilde{\tau}_g$ via an *inverse model* $q_\phi(\tau \mid g)$ (possibly leveraging structural markers in the observed

creative work g, e.g. discourse schema for news, Section 1.2.2).

(b) *Inverse Reinforcement Learning:* Building off Stage 1a, take inferred latent trajectories $\hat{\tau}_g \sim q_\phi(\tau \mid g)$ and infer a reward function $\hat{r}$ consistent with $\tilde{\tau}$.

2. **Stage 2: Trajectory Modeling:**

(a) *Imitation-based policy learning* and *Reward-based policy learning:* Building off Stage 1a, take inferred latent trajectories $\hat{\tau}_g \sim q_\phi(\tau \mid g)$ for each goal g, then fit $\hat{\pi}$ via behavioral cloning on the inferred $(\hat{s}_t, \hat{a}_t)$ pairs. Building off Stage 1b, take inferred reward function, $\hat{r}$, optimize $\hat{\pi}$ by maximizing $\mathbb{E}\left[\sum_{t=0}^{T} \hat{r}(s_t, a_t)\right]$.

(b) *Learn other components of the trajectory*: Train other models, like a *state transition* model $P(s_{t+1}|s_t, a_t)$ (i.e. to *realize* actions in the state space, e.g. for generation).

The learned $(\hat{\pi}, \hat{r})$ can then be used to produce new creative works, or in broader agentic pipelines. In this way, EL learns policies from *final-state* data through latent action and reward inference. *Emulation*, I will show, is a practically useful process that allows us to addresses the bottlenecks associated with previous approaches for modeling complex, creative tasks. Final-state data of completed creative outputs is abundantly available online – from completed news and science articles, to videos, songs, manuals or any other creative task. By more *explicitly* modeling the latent actions performed by humans while achieving these end-states, we can start to collect voluminous data specific to our tasks and explicitly model human processes. We will show how EL is fundamental cognitive learning paradigm, allowing us to *learn* reward functions, goal states and symbolic systems in the same manner as humans. Ultimately, EL can help us understand more about ourselves.

1.2.1 *Emulation Learning* in the Cognitive Sciences

Research in cognitive psychology offers a useful grounding for Emulation Learning (EL), giving it cognitive basis and demonstrating how humans engage in similar processes.

Psychologist David Woods, in 1988 [87], studied children learning how to perform tasks with teachers. Some, he noticed, were directly *imitating* their teachers, or *copying their teachers' actions/motor-movements*. Others, he noticed, sought not to copy actions but to *emulate* them. *Emulation*, he defined, was the learner's study of the teacher's goal states, G; rewards, R; and, to a lesser-extent, actions a; and a synthesis of these that allows the learner to not only reach the same goal-state g through *novel* action sequences $\hat{a}_1, \hat{a}_2, ...\hat{a}_t$ but *even improve on* g. Other researchers have solidified these insights and extended them.

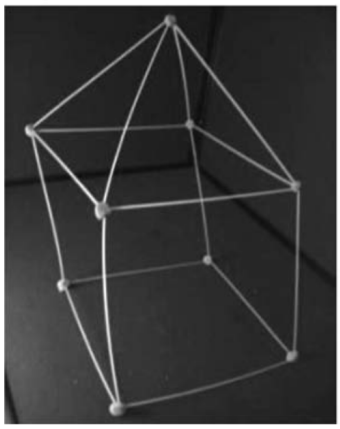


Figure 1.3: *Emulation Learning in cognitive studies*. [88] showed that children could emulate based just on observing the goal-state g of a completed house (the children were shown a picture of g, shown above). They (1) performed latent action inference and (2) learned a reward function to more deeply understand the building process. They used these to build bigger houses.

Observing that *emulation* involved learning from the goal-state g of the teacher, researchers have progressively tested how well learners could learn when shown less and less *action* information, $a_1, a_2, ...a_t$. Lydia M. Hopper and colleagues, in 2008, [89] conducted a series of "ghost" experiments, where the environment changed without an agent performing observable actions – in their experiments, a sliding door moved to reveal a reward without any visible agent in the "ghost" condition. In other words, the state transitions, $s_1, s_2, ...s_t$ were visible, but actions $a_1, a_2, ...a_t$ were not. Learners, they observed, inferred the actions necessary to achieve the reward.

Christine A. Caldwell and colleagues, in 2012, went further, studying how children learned from end-state, or goal-state demonstrations *only*. They showed children *images*, shown in Figure 1.3, of completed house-like stick-structures (i.e. the final states G of a creative process of house-building). This is a complex creative task: there were no well-specified reward functions of what made "good" or "bad" houses, nor were there any instructions about the kinds of actions (e.g. putting tickets together with a marshmallow) that could yield intermediate states (e.g. a floor, a

wall, or a foundation). However, they observed, children were able to not only imitate the stick houses, but build taller, more stable and more intricate houses than the demonstrations. They concluded that *emulation* is a cognitive process that involves *the explicit decoding of implicit actions*, viable *state-space transitions* and *rewards* from end-state demonstrations. This process should feel familiar to us all – we are often told by our supervisors: *"if you want to become a better scientist, read more articles"* (in journalism school, a teacher[3] explicitly told me: *"The best way to become a better journalist is to read more journalism."*). Within the context of emulation learning, this means: (1) study the final-states of other's creative processes. (2) Infer the actions they took, even if they are only implicit, and understand why they took them. (3) Understand how to recreate, combine and add to these actions to reach aligned, or advanced goal states G. Indeed researchers in many fields have pointed to end-state observation as crucial not only in learning but *also* the creation of cultural norms and the advancement of creative cultures, in: toys [90, 91] and design [92, 93]; language [94], artistic transmission [95, 96] and music [97, 98]; and in science [99, 100, 101] and journalism [102, 103, 104, 105, 106, 107].

I want to return now to and an earlier argument and give it a cognitive dimension: if EL is simply the act of learning from end-state observation, is not *pre-training*, or self-supervised learning (SSL), already achieving this objective? Researchers in cognitive science give us a basis for rejecting pre-training as a form of EL, beyond the evidence given in Section 1.1.1 (i.e. how it has been shown to not reliably infer implicit actions or learn deeper, implicit rewards). Numerous works have further clarified the distinction between *end-state* and *tacit* knowledge [108, 109]. End-states often under-specify the skills needed to reproduce them [108]: in many domains a degree of *tacit* knowledge must be acquired by a *learner* before they can perform *action inference*. This *tacit knowledge* (relational, somatic, collective) must be held by learners to process knowledge that artifacts *contain* but not immediately reveal. Classic distinctions between *knowing-that* and *knowing-how* reinforce this point [110, 111,

[3] David Hadju, `https://journalism.columbia.edu/faculty/david-hajdu`

112]; expertise research similarly indicate that enculturation and feedback are required to move to *contributory* skill [113, 114, 115]. In other words, simply reading a book might help us learn many things (e.g. language, ideas, and events) but, without any knowledge of the craft of writing (or a powerful enough *inverse model* $q_\theta(\tau|g)$), reading alone will not give us the tools to *emulate* the writer.

1.2.2 *Emulation Learning* in NLP: Meaning Hierarchy, Action and Discourse

As discussed, *emulation learning* sits alongside learning methods introduced in Sections 1.1.1 and 1.1.2: *imitation* (i.e. learning to *replicate* observed action sequences $\mathbf{a} = a_1, a_2, \ldots$), and reward-driven learning (i.e. prespecified constraints, rewards or directives, r, including *search*, or complete, open-ended exploration towards goal-states g)[4]. Now, let us explore how *emulation learning* intersections, specifically, with fields specific to linguistics and NLP.

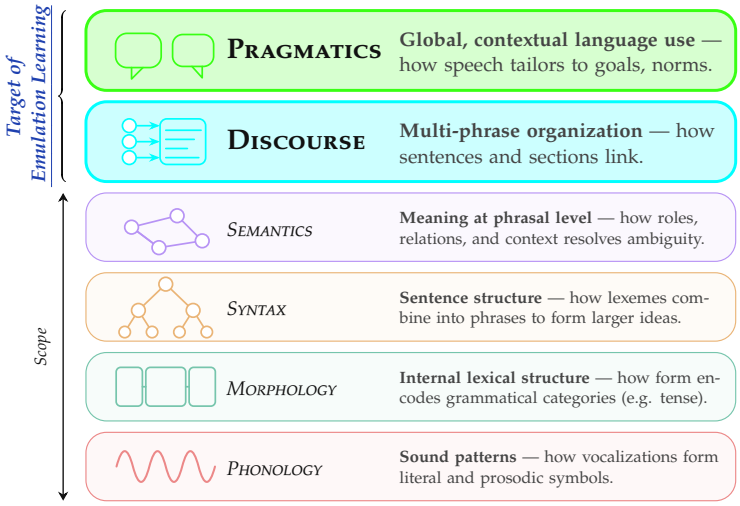


Figure 1.4: In the hierarchy of linguistic meaning construction, *emulation* sits in the intersection of *Discourse* and *Pragmatics*.

We will see, frequently, in this thesis, a focus on *textual discourse and pragmatics analyses* as the primary levels of *linguistic* emulation learning. Let me first define *discourse and pragmatics*, and then I will justify its use in *emulation learning*. Discourse and pragmatics are the studies of

[4]We will continue to see, in this thesis, how emulation learning draws from and contrasts with many of these directions: we compare with *emulation* with distributional imitation [116] (i.e. pretrained LLMs) (Sections 2.2.4, 2.3); we allow for search and exploration (Sections 3.4, 4.3; and we apply constraints and other guidance (Sections 4.2, 4.4).

structure and intentionality in communication; they study language use *above* the level of the sentence and how phrases, sentences and paragraphs are *joined* to organize meaning (Figure 1.4). Discourse and pragmatics have a long history in the computational study is linguistics, which I will briefly discuss now. *Discourse* emerged, in the 1970s, to explain *local* coherence between adjacent clauses and sentences [117, 118]. *Penn Discourse Treebank* (PDTB) codified some of these observations: in PDTB, *spans* of text were annotated with *discourse connectives* specifying how they related (e.g. *temporal*, *comparison*, etc.) [119, 120]. *Rhetorical Structure Theory* (RST), operationalized by the RST Treebank [121], similarly modeled texts as hierarchical trees of *elementary discourse units* (EDUs) linked by rhetorical relations (e.g. *evidence*, *contrast*, *elaboration*), but explicitly targeted paragraph- and document-level organization [122]. The field's center of gravity has since broadened from *structural analysis* to *meaning, context and intentionality*, traditionally the domain of pragmatics [123]. Segmented Discourse Representation Theory (SDRT) emerged in 2008 to connect structure with interpretation [124]. Inspired by linguist Teun Van Dijk [125, 126, 25], a number of computational works in the 2010s expanded discourse analysis into new domains [127, 128, 129], including journalism [130]. These works inherit the strategies of *discourse analysis*, which involve *structural, categorical and relational analysis of text*, and more often then not they approximate pragmatic phenomena: they encode functional, intention-bearing relations and document-level planning, even if most operationalizations stop short of full speaker–hearer modeling, as is common in pragmatics research [123]. As such, I will henceforth use *discourse* as a stand-in for *discourse and pragmatics*.

What is the purpose of *discourse analysis* in *emulation learning*? Writing-process and discourse theory primarily treat *discourse* as the level of linguistic structure where the writer is *intentional and hierarchical* [131, 132, 133, 134, 135, 136] while lower-level forms of meaning (e.g. syntax, semantics, see Figure 1.4) are assembled incrementally: speakers'/writers' local decisions are driven by immediate accessibility, priming, and information-theoretic pressures [137, 138, 139, 140, 141, 142, 143, 144]. In other words, writers *plan* global

communicative goals, allocate information, and manage attention *before* articulating sentences; while writing the sentence they primarily generate. Thus, *a text's structure* is the level of analysis at which to search for the writer's *latent intentions and actions*; if we wish to *emulate* writers, then we would do well to perform *discourse analysis*. In more detail, a standard *discourse analysis* approach involves the construction of a *discourse schema*, which is a *low-dimensional schema* annotating categories $d_1, \ldots d_k \in \mathcal{D}$. Each category captures some aspect of textual discourse structure for the specific structural or intentional phenomena being studied. Then, a model is trained to label sentences for their discourse relations [128, 145]. As we will see in Sections 3.2, 3.4, 4.1, 4.3 and 5.3, a discourse schema (e.g. "Background", "Claim", "Counterargument") can be easily written as an *action vocabulary*, $a_1 \ldots a_k \in \mathcal{A}$ (e.g. "Introduce Background", "State Claim", "Raise Counterargument") and a discourse model can be the *inverse model* $q_\theta(\tau|g)$: in other words, focusing on the *structure* of g can help us infer actions a and plans [146]. It is fair to again ask whether imitation, or next-word prediction (i.e. distributional imitation **??**) can model these intentions and actions. Next-word prediction (Section 1.1.1) trains base models [147] to imitate human language with high fidelity, reliably capturing phonology-to-semantics regularities [148] (Figure 1.4). Imitation can also implicitly capture some discourse regularities — indeed, humans sometimes show "over-imitation" of surface actions, unintentionally capturing higher-level actions [149] — but does not specifically capture plans [150, 76]. We draw a distinction: token-level imitation resembles fast, automatic processing [151, 152, 153, 154]; discourse-level planning resembles deliberate control [155, 156]). Our claim is not that imitation fails — indeed, imitating behavior is an important part of human-social learning [157, 158, 159, 160] — but explicitly modeling discourse-level actions better matches how writers plan and improves hierarchical control and transfer.

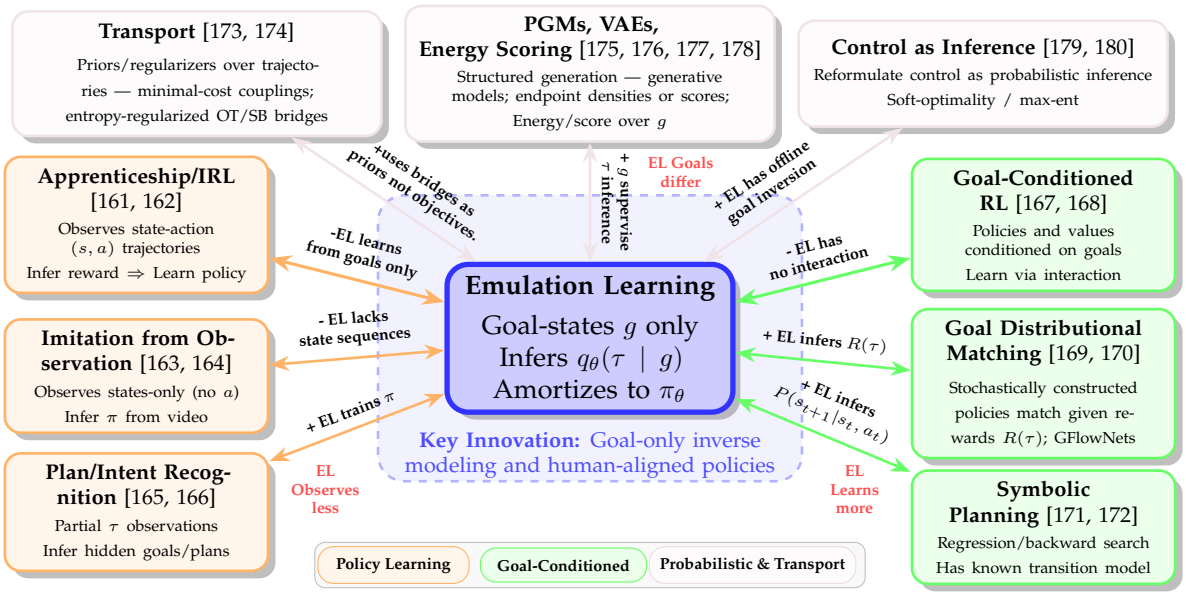


Figure 1.5: **Overview of related methods**: Comparing existing methods (i.e. *Policy Learning*, *Goal-Conditioned* learning methods, *Probabilistic and Transport*-based inference methods.) with *emulation learning*.

1.2.3 Comparison between *EL* and Other Methods in AI

Emulation Learning, as we just discussed, has been broadly studied in the cognitive sciences, art and philosophy. *Why has it not yet been formalized, as a learning paradigm, in computational sciences?* EL indeed shares similarities with and is inspired by a number of several paradigms in control, planning, and probabilistic modeling, in addition to those already discussed (i.e. self-supervised *pre-training*; and reward or supervision-based *post-training*). I will go through other related areas of artificial intelligence now, and then hypothesize why EL has not yet been formalized.

1.2.3.1 Policy-Learning: demonstration-driven learning and behavioral inference

EL is closest to methods that infer *policies*, *rewards*, and *intentions* from demonstrations. *Apprenticeship*, first proposed by Peter Abbeel and Andrew Ng in 2004 [161] formalizes a two-stage pipeline: (1) learn a reward function via inverse reinforcement learning

(introduced by Ng et al in 2000 [162]), then (2) optimize a policy under that reward [162, 161, 181]. Similarly to our framework, *apprenticeship* assumes that true reward functions are complex and cannot be prespecified; only by first learning them from actions can more nuanced policy functions be learned. Apprenticeship has thus far been mainly applied to more classical tasks in control, like robot movement [2] – EL is explicitly concerned with subjective, creative tasks, where reward functions are likely even more nuanced.

A more fundamental difference between EL and apprenticeship is that apprenticeship assumes that full action-state trajectories are fully observable/accessible. EL targets domains where only *finished creative artifacts* (goal states) are observed and puts special emphasis on the *latent action/trajectory inference* process, $q_\theta(\tau|x)$, that must occur before policy learning. In this way, EL relates to *imitation-from-observation* (IfO) problems, which assume that actions are unobserved and instead infer policies from state-only (often video) demonstrations [163, 164, 182, 183]. EL, our framework, goes beyond IfO in that we assume even less visibility into state-space transitions – we assume only the goal-state g is visible and thus need more robust inverse model, unlike IfO which typically assumes full *state sequences*. In EL, trajectory inference can be inferred (e.g., via inverse dynamics or structured latent variables) to enable either (i) implicit behavior cloning from inferred trajectories or (ii) reward inference followed by RL, thereby unifying IfO-style action inference with apprenticeship/IRL-style reward inference. Finally, EL is also related to plan and intent recognition, which infer likely goals or plans from partial observations [165, 166]. EL generalizes this spirit to creative domains with *goal-only* evidence and pushes beyond recognition: the inferred latent trajectories and/or rewards are subsequently *used to train a policy* for generation.

1.2.3.2 Goal-States as Supervision, Symbolic Reasoning and Goal Distributions

Several families of methods make deliberate use of goal states, or endpoint-constraints, to enhance learning. Concretely, given a set of desirable end states $G \subseteq \mathcal{S}$ (e.g., finished

creative works), these approaches either (i) condition policies on desired goals (ii) reason symbolically from goals back to preconditions, (iii) construct flows that connect an initial distribution to $p(g)$ under transport principles. *Emulation Learning* (EL) is adjacent to all three, in that it also learns from goal-states, but is distinct in that it uses these states to infer latent processes (trajectories and/or rewards) with the explicit aim of policy learning.

The goal-conditioned methods *most* related to the previous section on policy-learning are *goal-conditioned RL (GCRL)* methods, which (1) aims to learn *policies* over action sequences (2) explicitly incorporate information about goal statesin their learning process. GCRL parameterizes policies and value functions with an explicit goal, g as input. Formally, with a goal space $\mathcal{G} \subseteq \mathcal{S}$ and goal distribution $p(g)$, GCRL learns $\pi_\theta(a \mid s, g)$ (and optionally $Q_\theta(s, a, g)$) when r_g is sparse (i.e., $r_g(s, a) = 1\,[s \approx g]$). Researchers have explored using universal value function approximators to share structure across goals [167]; Hindsight Experience Replay improves sample-efficiency by relabeling goals with achieved outcomes in off-policy data [168]. EL is complementary: rather than *interacting* to learn $\pi_\theta(\cdot \mid s, g)$, EL *derives* a goal-conditioned policy from goal-only observations by first inferring latent trajectories and/or a reward model that render the observed endpoints likely, then optimizing a policy consistent with those inferences. An interesting related method, *G enerative Flow Networks (GNets)* aims to learn stochastic *construction policies* that assemble a *composite goal state* – a state built incrementally by composing primitives (e.g., molecular graphs, program trees) – through a sequence of steps through partial states [169]. (We will study compositeness in Chapters 3 and 4, most directly in Sections 3.2.3, 3.4, 3.6 and 4.1.) Endpoint supervision in GNets enters via a scalar terminal reward $R(g)$ on a finished goal-state g; the target is to learn a policy whose terminal distribution satisfies $P_\pi(g) \propto R(g)$. Training enforces *flow-matching* constraints or uses the *Trajectory Balance* loss [170], $\mathcal{L}_{\text{TB}}(\tau) = \big(\log P_F(\tau) - \log P_B(\tau) - \log R(g)\big)^2$ where τ terminates at g, and P_F, P_B are forward/backward path probabilities. EL is similar in treating finished objects as informative endpoints, but it does *not* assume an externally provided $R(g)$; instead, it (a)

infers plausible trajectories from endpoints and then (b) either clones a policy from those trajectories or learns a reward model consistent with the observed goals.

Symbolic goal-based reasoning methods also aim to reach composite goal-states, but do so with a more classical approach. This class of methods includes *means–ends analysis* [171] and *regression planning* [172]. Both methods reason backward from a goal $g \in G$ to subgoals by applying inverse operators under a known transition model and typically in a closed-set search space; generic *backtracking* and *heuristic search* [184, 185] supply the algorithmic mechanisms that traverse the state space, prune branches, and guide expansion using heuristics. While such methods can reduce search and produce efficient plans when an explicit state transition model is known, EL replaces explicit backward symbolic search with statistical inference over latent trajectories and then *amortizes* the result into a parametric policy usable without test-time search. Finally, *optimal transport (OT)*-based methods generalize the focus on goal-states to goal distributions: OT constructs "flows" between starting and goal distributions. Static OT finds a coupling $\pi \in \Pi(\mu_0, \mu_T)$ minimizing $\int c(x, y)\, d\pi(x, y)$ between an initial distribution μ_0 and a target (goal) distribution μ_T for a cost c [173]. In dynamic form, one minimizes a kinetic-energy functional subject to the continuity equation that transports μ_0 to μ_T [173]. A related method, *Schrödinger bridges (SB)*, solves an *entropy-regularized* analogue: among path measures $\mathbb{P}$ on trajectories that match fixed marginals (μ_0, μ_T), choose the one minimizing $\mathrm{KL}(\mathbb{P} \,\|\, \mathbb{P}_0)$ relative to a reference diffusion $\mathbb{P}_0$ (e.g., Brownian motion), yielding the most likely bridge consistent with endpoints [174]. Thus, "transport or entropy-regularization objectives" refer to, respectively, minimizing transport cost (OT) or minimizing pathwise relative entropy (SB) under endpoint constraints. EL differs in aim: it uses endpoints to *infer* latent decision variables (trajectories, rewards) that generated them in a creative MDP, and then *learns a policy*, rather than merely constructing a minimal-cost or most-likely flow between fixed marginals. Nonetheless, OT/SB can serve as priors or regularizers over the family of latent trajectories considered by EL.

1.2.3.3 Latent-variable probabilistic modeling and program synthesis

Probabilistic inference has, personally, been hugely inspirational to how I look at the world: my earliest research experiences have been working on probabilistic graphical models (PGMs) with Dr. David Blei at the *New York Times* [186, 187, 188]. The starting-point for each PGM is the "generative story": a story that describes how observed data "came to be". This directly inspires EL's focus on inferring latent actions that "generate" observed goal-states g. Although many learning approaches for probabilistic models (e.g. EM [189], VAEs [175, 190], VI [176] and HMMs [191, 192]) typically perform backwards (observed $\rightarrow$ latent variable) and forwards (latent $\rightarrow$ observed variable) passes over the same architectures (i.e. that make the same probabilistic assumptions), wake–sleep style amortized inference [193] separates backwards architectures from forward (which they call *recognizer* and *generator* models). EL follows this structure, as our inverse model $q_\theta(\tau|g)$ for inferring latent actions needs not be connected with other parts of the EL process.

This family of methods parameterize the posterior over *process* variables (actions, intermediate states) conditioned on goals. Control-as-inference reformulations view RL, too, as probabilistic inference under maximum-entropy/soft-optimality criteria [194, 195, 196, 180, 179]. EL is aligned with this perspective but differs in emphasis from most PGM's goals: EL's *policy learning* objective uses these posteriors to supervise or to define control-as-inference surrogates and treats control variables (actions, rewards) as *latent causes* of observed creative outcomes and uses inference over those causes to *learn the policy itself*. Related to PGMs, *energy- and score-based generative models* also define endpoint densities without trajectories. Energy-based models posit $p_\theta(g) \propto \exp(-E_\theta(g))$ for finished objects g and are typically trained by estimating the gradient of $\log p_\theta$ via contrastive or likelihood-gradient methods and sampling with MCMC [177, 197]. Score-based models instead learn the score $\nabla_g \log p(g)$ by denoising-score matching across a noise (or diffusion/SDE) schedule and then sample with Langevin dynamics or reverse-time SDE solvers [178]. In both cases, supervision is entirely *endpoint-level*: the learner fits a distribution over

Recap: Emulation Learning

Introduced in Section 1.2, *Emulation Learning (EL)* is a novel computational learning approach introduced in this thesis for learning complex, creative workflows where *limited data* or *reward functions* exist. EL studies goal-states g resulting from human state-action trajectories $\tau^* = (a_1^*, s_1^*), (a_2^*, s_2^*), \ldots$. EL *infers* these trajectories, $\tilde{\tau}$, via an inverse model $q_\theta(\tau|g)$, then learns a policy model, $\hat{\pi}(\tau|x)$ from starting-state x, via direct supervision from $\tilde{\tau}$ or after inferring a reward function R.

completed works g without explicit actions. EL can borrow these parameterizations (e.g., as priors over goals or as components of trajectory posteriors), but its optimization target is a *control policy* $\pi(a \mid s)$ that maps states/contexts to actions; endpoint modeling is thus a means, not the end.

Program synthesis illustrates an even more explicit "process-as-latent-structure" view, where observed solutions are explained by discrete programs that compose primitives; wake–sleep style library learning amortizes search [198, 199]. EL is analogous in that it posits discrete/structured *creative workflows* as latent generators of finished works, but its endpoint is a *reactive policy* rather than an explicit executable program. Finally, diffusion-based planners and offline decision-making methods (e.g., Diffuser and Decision Diffuser) explicitly model trajectory distributions and then condition/guidance-sample plans that satisfy goals [200, 201]. EL can incorporate such trajectory models as flexible priors for its latent-action inference step; the distinctive ingredient remains the *use of completed works as observational evidence* to infer policies (with or without explicit reward modeling).

1.2.3.4 Summary

Across these literatures, *emulation learning* is positioned: (i) like Imitation from Observation (IfO) and Inverse RL (IRL), EL is *inverse* in nature and seeks to understand human behaviors, but it assumes more limited observability into human processes (EL assumes goal-states g are observable); (ii) like goal-conditioned RL and other goal-supervised methods, EL *operationalizes* goals, but EL *aligns* action sequences to *human behaviors* rather than allowing

open-ended discovery of *any* action trajectory; and (iii) like latent-variable/Bayesian approaches, EL frames creative processes as latent structures to be inferred, yet EL goes further by *using* those inferences to *learn deployable policies*.

A question remains: if EL is so fundamental in the cognitive sciences and practically useful in modeling human tasks, why has it not yet been formalized so far as a task in machine learning? My strongest assumption is that we simply did not have effective inverse models $q_\theta(\tau|g)$ for inferring latent actions from goal states. Prior to the current age, inverse models had to be carefully constructed [202] through curated datasets. While this is still a large area of research, pretrained language models are *finally* demonstrating promise as inverse models. I expect that the coming years of research will enable much more powerful approaches to emulation learning.

1.3 Outline of This Thesis: *Emulating* 4 Steps in Computational Journalism

Journalism is the creative domain that will be our primarily focus in this thesis, specifically, *computational journalism* (i.e. the application of computational techniques to routines and workflows in newsrooms [204, 205]). Each task that we discuss will have the purpose of further exploring core questions in *Emulation Learning*.

The main body of this thesis will progress in 4 Chapters, each introducing one task in a journalistic workflow, shown in Figure 1.7. They are organized around the steps of the *journalistic pipeline* — or, the professionalized process by a news event is found, produced and published [206]. The four steps are shown along the top of Figure 1.6: (a) *story finding*, the process by which events become news, (b) *source-finding*, the process by which sources are found and added to a news article, (c) *story structuring*, the process by which facts are organized into a cohesive story and (d) *news editing*, the process by which factual and stylistic changes are made and event updates are incorporated. I describe each part in turn.

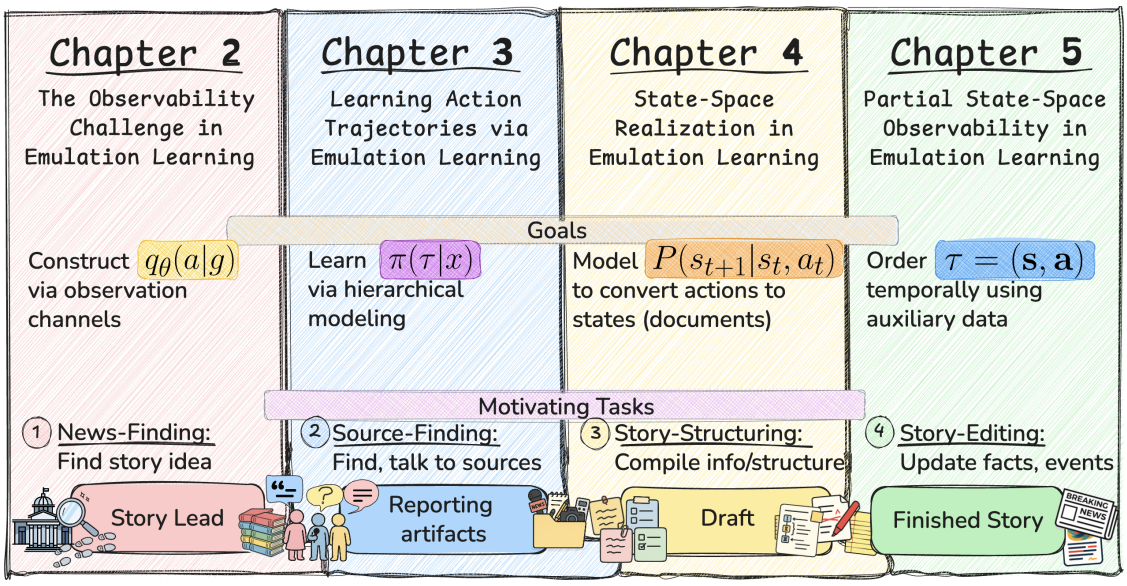


Figure 1.6: **Overview of the main body of this thesis.** I outline the goals of each chapter. In terms of *emulation learning*: Chapter 2, we construct the inverse function, $q_\theta(a|g)$; Chapter 3, we explore ways to learn the *policy function*, $\pi(\tau|x)$; Chapter 4, we turn to learning the *state transition function*, $P(s_{t+1}|s_t, a_t)$; Chapter 5, we increase observability into the state-space (i.e. "ghost conditions" [203]). Motivating tasks shown at bottom and Figure 1.7.

1.3.1 News Finding — An *Observability* Challenge for Emulation Learning's Inverse Model $q_\theta(a|x)$

In Chapter 2, The Observability Challenge in Emulation Learning, I will focus on the first step in *producing news – finding a story*, or event, to write about. When is an event *newsworthy*? Which information is *prioritized*? I introduce *newsworthiness prediction*, the task of learning a policy $\pi(a \mid x)$ where x is an event[5] and a is a score indicating its likelihood of coverage. As a practical task, consider a *newsworthiness recommendation engine* that recommends potentially newsworthy events to a journalist — this could help journalists navigate today's information overload, saving time and surfacing more stories.

To *emulate* newsworthiness, we will train $\pi(a|x)$ on *previous newsworthiness judgments*: we fit an inverse model $q_\theta(a \mid g)$ that infers latent editorial actions a from observed artifacts g, yielding pseudo-judgments $(\tilde{a}, \tilde{x})$. We face two *observability challenges*. In Section 2.2, g

[5]For example, a policy that a city council is trying to pass, the outcome of a political race, a mention of two snow leopards being donated by Saudia Arabia in a White House fact sheet

are news articles and the action space is $a \in \{0, 1\}$, where $a = 1$ means an event x *should be* covered and $a = 0$ means it *should not*. Because g exists only for *covered* events $(a = 1)$, our data lie on a restricted support: $\text{supp}(x)$ s.t. $\exists\ g \subsetneq \text{supp}(x)$. Estimating $\pi_\phi(a \mid x)$ from (x, g) pairs alone, thus, would not reveal how journalists would judge uncovered events. In Section 2.3.3, we expand beyond this binary view of a. Even among covered events, some are judged more newsworthy than others: a is ordered and potentially continuous $a \in \mathbb{R}$ based on intrinsic and extrinsic factors (e.g. world context C). Here we study *news homepages* as g, which give us richer information about C *and* relative newsworthiness of events x, x'. To address both challenges, we introduce *observation channels* $\{\sigma\}$, each extracting observations $y_\sigma = f_\sigma(g)$. Assuming channel-specific emission models $C_\sigma(y_\sigma \mid a)$, we show different ways of constructing the inverse model $q_\theta(a \mid g)$ through auxiliary steps.

1.3.2 Source Finding – Trajectory Planning for Emulation Learning's Policy Model $\pi(a|x)$

In Chapter 3, Learning Action Trajectories via Emulation Learning, I examine the *next* step in covering the news: *finding sources* to confirm, broaden, and contextualize a news event. What *roles* do different sources play, and how do they *complement* one another? Can we *retrieve* the right sources for a story? I formalize a new task, *source-finding*, as learning a policy $\pi(\tau \mid x)$, where $\tau = \langle (a_1, s_1), (a_2, s_2), \dots \rangle$ is a trajectory of actions a_t and intermediate states s_t. a_t is a *Get Source* action, and s_t denotes *the set of sources* retrieved at time t.

To *emulate* source-finding, we first train $q_\theta(\tau|g)$ from *labeled* (τ, g), where g is news articles. We assume fewer observability challenges than in Chapter 2. Our core challenge is to *reason about longer action sequences* $\mathbf{a} = a_1, a_2, \dots$. In Section 3.2 we probe whether $\mathbf{a}$ is compositional, a prerequisite for modeling it effectively. Section 3.3 explores whether explicitly training $\pi(\tau \mid x)$ is necessary or whether large language models' implicit policies $\pi^{(\text{llm})}(\tau \mid x)$ suffice. In Section 3.4, we observe that sources fulfill specific *discourse roles*, $d \sim d(a_t); d \in \mathcal{D}$, within a narrative and propose a hierarchical planner–executor:

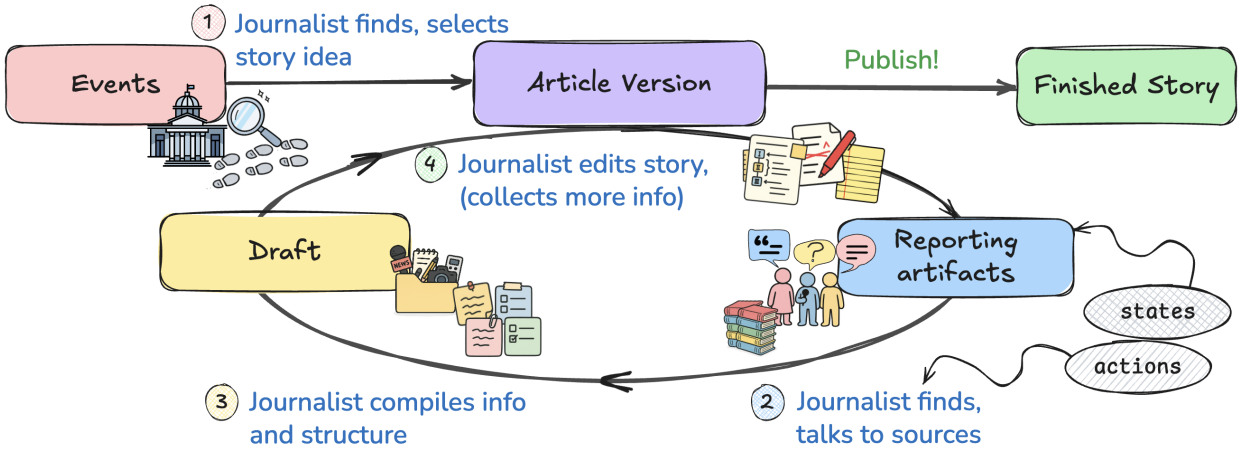


Figure 1.7: **The Story Production Pipeline**: the process by which journalists: (1) find leads to write about (*News-finding*, Chapter 2), (2) find sources to confirm, broaden and narrate their stories (*Source-finding*, Chapter 3), (3) structure and produce their stories (*Story-structuring*, Chapter 4) and (4) edit/update their stories (*Story-editing*, Chapter 5). *States* are shown in rectangles, *actions* are shown atop arrows. Illustrates the iterative nature of story production, showing how *article versions* are generated during each reporting cycle. In Chapter 5, we will use these *article versions* to obtain intermediate states.

$\pi(a_t \mid x, s_t) = \pi_e(a_t \mid d(a_t))\ \pi_p(d(a_t) \mid x, s_t)$, where π_p chooses discourse roles and π_e selects sources based on them. This raises questions about choosing an appropriate schema $\mathcal{D}$ (we address in Section 3.5). Finally, we take steps toward reward-based preference learning by creating a *virtual interviewer* sandbox (Section 3.6).

1.3.3 Story-Structuring – State Realization for Emulation Learning's Transition Model $P(s_{t+1}|a_t, s_t)$

In Chapter 4, State-Space Realization in Emulation Learning, I will introduce the *last* step in producing stories: *writing* the story. How do all collected facts *fit* together cohesively, and tell a story that is well-structured (e.g. inverse pyramid [207])? I introduce *structured generation*, where $\mathbf{a} = a_1, a_2, \ldots$ are a sequence of structural markers (e.g. an outline, or discourse labels: "Write Background", "Introduce anecdote") and $\mathbf{s} = s_1, s_2, \ldots, g = s_n$ is the *realization* of those markers (i.e. the current draft and surface-form text corresponding to each action).

Our primary focus in this Chapter will be *not*, as in Chapters 2 and 3 on selecting a via the policy model, $\pi(\tau|x)$, but *instead* on how a gets *realized* into the state-space, s. In *emulation learning*, this is known as learning the *state transition model* $P(s_{t+1}|s_t, a_t)$. $\hat{\mathbf{s}}$ will be evaluated by how much it *looks* human, i.e. how well text we generate is structured like g^*, an ideal human article. In doing so, we will confront *control* challenges. I introduce methods in Sections 4.2 and 4.3, where *discourse* is again used as a measure of structure *and*, as per our state-action definitions, illuminates actions. I will show how we can use the inverse model $q_\theta(\mathbf{a}|g)$ to *steer* generation to be more structured. Section 4.4 will ask: how can we explore more *generalized realization* mechanisms. These again raise questions about the "rightness" of latent outlines, in Section 4.5 we will probe this question by exploring how much correlation exists between discourse schemes. Finally, in Section 4.6, we will explore how structural awareness can play a role in *information comprehension*.

1.3.4 Story Editing – Increased State-Space Observability

In Chapter 5, State-Space Observability in Emulation Learning, I introduce a task that occurs *throughout* the *news production* process: *story editing*. An *edit* is any action a writer makes *during* the writing process. Edits reflect fact and event updates; structure changes, as a news article progresses from an immediate news alert to a fully-fledged article; and stylistic changes. In *story editing*, we examine trajectories $\tau = (a_{1,1}, s_{1,1}), (a_{1,2}, s_{1,2}) \ldots, (a_{2,1}, s_{2,1}), (a_{2,2}, s_{2,2}) \ldots$ where each a_{ij} is a single *update action* (i.e. any action made during the writing process, including actions studied in Chapters 3 and 4) and each s_{ij} is an edited state of an article. Crucially, when examining *edit* τ, *we have greater observability into the state space*. We introduce novel revision histories datasets for news, which give us observability into *starting*-states $s_{1,1}, s_{2,1}, s_{3,1}, \ldots$ for each subsequence in τ.

Our focus this Chapter is exploring how this greater observability can be used to probe temporality in action sequences and improve *emulation* overall. If $s_{1,1}, s_{2,1}, s_{3,1}, \ldots$. is observed, then we can impose partial ordering on actions $a_{1,1}, a_{1,2} \ldots < a_{2,1}, a_{2,2} \ldots$

occuring between revisions. In Section 5.2, we present our revisions-histories datasedt, *NewsEdits*, and show how *atomic* state-space changes can be deduced and *predicted*. In Section 5.3, we build inverse models $q_\theta(a|s_{i,t}, s_{i,t+1})$ to help us infer actions in sequences. I anticipate this work in edits opens a crucial door into using revision histories for behavioral sequence data, thus opening a new door in *emulation* and leading to more precise tooling, interventions and behavioral understandings.

Chapter 2

The Observability Challenge in Emulation Learning

2.1 *Newsworthiness Prediction*: A Study in How Information is Prioritized

Journalists make decisions on whether or not to write stories about *events* based on qualitative, case-by-case assessments of whether the event meets criteria for being noteworthy, interesting and relevant enough to cover [23]. Collectively, these criteria are called "news values" – they are poorly defined and hard to articulate norms, making them a challenging task to study with traditional machine learning methods; yet journalists share broad agreement on what they are, indicating that they are a learnable task [24].

Figure 2.1: In the *journalism pipeline* outlined in Section 1.3, we focus now on the first step: *newsworthiness prediction*, or predicting the news value of events in order to discover stories to write about. This task requires us to learn a policy model $\hat{\pi}(a|x)$, which gives a score indicating whether event x should be written about. Newsworthiness prediction requires us to learn to capture complex human judgments about events and their salience.

We will focus on this process, *news-finding*, in this Chapter. The practical goal we will center around is a new task, *newsworthiness prediction*, that seeks to predict whether a story *should* get covered. Imagine a tool that would function as a recommendation engine, surfacing story ideas (e.g. a recent city council policy, an interesting line in a press release) to journalists. This system will need to understand the intrinsic newsworthiness factors journalists look for in events (e.g. per Galtung [23]: "relevance to a community", "involves persons of interest") as well as the extrinsic factors in their environment (e.g. "major international event occurred earlier today"). Despite much qualitative analysis of the factors informing newsworthiness [23, 208] has found that many very little quantitative work has attempted to analyze: (1) *what* stories get covered, (2) *why* have they been covered? Not only could such work increase our understanding of coverage patterns and informational salience perceptions [209], but it could empower newsrooms to discover more stories [210]. This task will introduce us to some of the core concepts and challenges in emulation learning. Let us formalize this now.

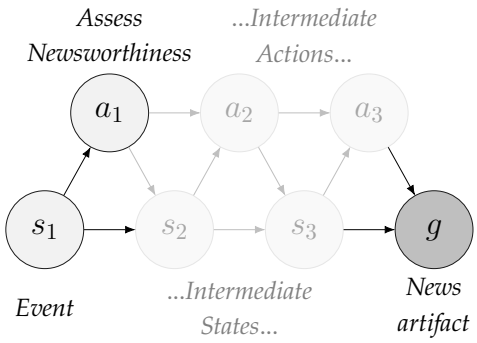


Figure 2.2: *Observability of the newsworthiness-prediction task:* We assume that the first action, a_1 assess the newsworthiness of s_0 ="*Event*". Only *artifacts* related to the news article (i.e. the article itself, Sec. 2.2, or it's placement on the Homepage, Sec. 2.3), g, are observable.

Newsworthiness Prediction as Emulation Learning: The goal of newsworthiness prediction is to learn a policy model $\pi(a|x)$, that will take, as input, an event s_1 or x (more specifically, a *textual description* of the event) and assigns a *newsworthiness judgment*, a_1, to the event. The higher a_1 is, the more newsworthy that event and, as shown in Figure 2.2, the more likely we are to perform further actions τ to write the story (explored in later Chapters). These progress us towards the goal state, g, which is some observable news artifact: either the article itself (Section 2.2 or it's placement on the homepage (Section 2.3). The newsworthiness task gives us an excellent starting point to outline some concepts

Cheat-Sheet: Emulation Learning for *News-Finding*

From finished articles and homepages, we infer both which events *have* and *have not* been covered, as well as their relative importance.

- **a** a_t **(action)** — one-hop decision to consider the newsworthiness of a story idea, x. $a \in \{0, 1\}$ for *cover* vs. *not* (§2.2) and a continuous utility for (§2.3.1).
- **s** s_0 **(state)** — initial state $s_0 = (x, c)$ with c the extrinsic context (captured explicitly by competitor set C on homepages) (§2.2, §2.3.1).
- x x **(starting context)** — Input/event, or the candidate lead to be evaluated as newsworthy or not (primarily SFBOS policy proposals). (§2.2.2, §2.2, §2.3).
- g g **(goal state)** — The published news article (§2.2) or the homepage (§2.3.1).
- q $q_\theta(a|g)$ **(inverse model)** — Recovers latent *newsworthiness* decisions from observables: $q_\theta(a \mid x, g)$ via the linking channel $M_\psi(x, g)$ (§2.2, §2.2.3) and $q_\theta(a \mid x, C, g)$ via pairwise preferences $p_o(x > x')$ (§2.3.1, §2.3.3.1).
- π $\hat{\pi}(a \mid x)$ **(policy model)** — predicts coverage for new events or ranks articles by relative prominence given contemporaneous competitors C (i.e. $\hat{\pi}(a \mid x, C)$). (§2.2.4, §2.3.1, §2.3.4).

in *emulation learning* more concretely. Newsworthiness prediction is in some ways an *easy* task to emulate and in some ways a *challenging* task. It is easy because, as shown in Figure 5.3, typically only a single action, a_1, is needed to assess an event's newsworthiness.[1] So, it allows us to explore *emulation learning* without necessarily considering long action trajectories $a_2, a_3, \ldots$ (e.g. as in Section 3). However, *inferring* newsworthiness poses a significant *observability* challenge. Simply collecting easily accessible newsworthiness signals from observed artifacts g will give us too much *positive*, *intrinsic* signal, and not give us enough information to learn a robust policy model $\pi(a|x)$ that (1) covers a wide space of non-newsworthy events x and (2) considers extrinsic confounders. In other words, many events "look" newsworthy: determining events journalists *should* cover also requires determining events they *should not* cover. The core *emulation* focus in this Chapter will be to explore the *observability* of the newsworthiness spectrum. I will introduce two ways of calculating the inverse function, $q_\theta(\tau|a)$.

[1] For many stories, the decision-making process can often be instantaneous – many events, to experienced journalists, are *clearly* newsworthy. Conversely, for some stories, more actions need to be performed to *assess* the newsworthiness of an event (e.g. to verify information, or "dive deeper" to understand "if there is a story there"). We will not consider these cases in this Chapter.

Chapter 2 Overview

In Chapter 2, *The Observability Challenge in Emulation Learning*, we will approach a core challenge in emulation learning: how can we train policy models $\hat{\pi}(a|x)$ that are (1) sufficiently generalized across a broad space of newsworthy and non-newsworthy events, x and (2) responsive to extrinsic confounders (e.g. fluctuations in daily news volumes)? Relying solely on inferences from an inverse function $q_\theta(\tau|g)$ that considers one artifact, g, at a time, we will see, is not sufficient to overcome these challenges. This section will unfold as follows. First, in Section 2.2, we address the first challenge, ensuring $\hat{\pi}(a|x)$ is robust across newsworthy *and* non-newsworthy events. We simplify a to be binary: $a = 1$ means to *cover* this event, and $a = 0$ means *do not cover*. When $a_1^* = 0$, we see, no artifact g results; so, we introduce a linking function $M_\psi(x, g)$, described in Section 2.2.3, that labels $\tilde{a_1} = 0$ when $M_\psi(x, g)\ \forall\ g \in G$. This allows us to train a more robust policy model $\hat{\pi}(a|x)$ Then, in Section 2.3.1, we will address the second challenge, ensuring $\hat{\pi}(a|x)$ considers the presence of extrinsic factors. We expand a to be a real-valued variable; the higher a is, the more likely we are to cover x. We introduce a pairwise function $p_o(x\ >\ x')$ to compare *pairs* of inputs and judge which one is *more* newsworthy, allowing us to rank inputs across a wide spectrum.

Works Discussed:

- Spangher et al. (2024)". Tracking the Newsworthiness of Public Documents". *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Spangher et al. (2025)". NewsHomepages: Homepage Layouts Capture Information Prioritization Decisions". *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

2.2 To Cover an Event or Not to Cover an Event?

We start with the simplest case of newsworthiness prediction: whether an event x *should* or *should not* be written about in a news article. This allows us to simplify newsworthiness into a binary classification problem: our newsworthiness policy model $\pi(a|x)$ becomes $p(a|x)$, where $a = 1$ if x is written about and $a = 0$ otherwise. This setup is clearly limited – not all articles are equally important: some have outside effects while others are simply routine coverage. We will revisit this simplifying assumption in Section 2.3. Now, though, we explore this simplified problem and how EL can be useful.

2.2.1 Linking Function M_ψ Gives Observability

As outlined previously, we frame *newsworthiness prediction* as a minimal, horizon-1 instance of EL. We start with an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$. Each episode begins at state $s_0 = (x, c)$, where x is a textual description of event x (e.g. "city council passes policy" or "Saudi Arabia donates leopards") and c is factors external to the event x (e.g. the context of the newsroom, other news that is being covered, prior events related to x). Here, we assume c is constant. The action set is binary: $a = 1$ means *cover* the event; $a = 0$ means *ignore*. The episode terminates after a single decision and transitions are deterministic: taking $a = 1$ produces a finished creative work g (a published article) in the goal set G, whereas taking $a = 0$ yields a null terminal $\varnothing$ (no article). The reward $r^*(x)$ is an unknown *newsworthiness* utility and the expert's latent policy π^* maps events to coverage probabilities. Thus, in this one-step EL formulation, a trajectory is simply $\tau = (x, a, c)$[2]. *The primary applicability of EL here results from limited observability*: we do not see trajectories or rewards, only a collection of events X and a collection of goal states $g \in G \cup \{\varnothing\}$.

To calculate our inverse function $q_\theta(\tau|g)$ here we learn *linking (alignment) model*, $\mathcal{M}_\psi(x, g)$,

[2]While it might seem pedantic to use the language of trajectories and policies for a horizon-1 task, we aim to maintain consistency with upcoming sections.

between x and g. $\mathcal{M}_\psi(x, g) = 1$ indicates that g *covers* event x (i.e. $a = 1$, the positive case). $\mathcal{M}_\psi(x, g) = 0$ over *all* $g \in G$ indicates x was not covered by *any* article (i.e. $a = 0$, the negative case). We treat the mapping from latent action to observed goal as an *observation channel*: even if $a = 1$, a discoverable article may fail to appear in our corpus with some small probability (e.g. archives and web-crawls are imperfect). It is therefore useful to think of the channel's *recall* $R(x)$, the probability that a true coverage decision for event x yields a detectable article in the data. The higher the recall $R(x)$, the stronger the evidence that non-detection reflects a true non-coverage decision rather than a missed article. It is also useful to think of the linking model as generating a posterior over the a, given x (and external factors, c, assumed constant). When an article is detected for x, the posterior tilts towards $a = 1$; when no article is detected for x, the posterior tilts toward $a = 0$; ensuring that both observed coverage and the lack thereof enter coherently into EL's inverse step. Here we choose to learn a policy $\hat{\pi}(a \mid x)$ explains these inferred actions[3]. The immediate aim of inferring actions here is precisely to learn this simple policy model for a *new* event x—a newsworthiness estimator that generalizes beyond the observed corpus.

Figure 2.3: A policy item, x, in purple, is covered by a news article, g, in yellow. $M_\psi(x, g) = 1$; the policy is covered by the news article.

2.2.2 Local News Coverage: San Francisco Board of Supervisors

To restrict our area of focus, we restrict ourselves to the following scenario: a local journalist is covering their local city council. The universe of events, x, are policies published by the city council. This is shown in Figure 2.3. We focus on a specific local government, the San Francisco Board of Supervisers (SFBOS), and a specific

[3]We will see that fitting a binary predictor $p(y{=}1 \mid x)$ by cross-entropy coincides with maximum-likelihood estimation in this one-step MaxEnt-IRL view: the predictor's log-odds act as an affine proxy for a reward function $\hat{r}(x)$, and applying a monotone link yields the policy $\hat{\pi}$.

newspaper, the *San Francisco Chronicle* (SFChron), that has a robust local news section. We start by gathering HTML of all SFChron articles published between 2013–2023 and via the Common Crawl[4]. We parse article text[5] and deduplicate based on text, and ultimately are left with a set of 202,644 SFChron articles[6]. We also scrape the public meeting calendar on the SFBOS website[7] to collect all SFBOS meetings between 2013-2023[8] and then collect the proposal text for 13,089 SFBOS policy proposals[9] that were discussed a total of 27,371 times in 410 public meetings. Each policy is, on average, discussed in 3 separate SFBOS meetings.

2.2.3 Probabilistic Relational Models: A General Linking Function

A naive approach to applying EL to newsworthiness would be to construct a inverse function $q_\theta(\tau|g)$ based on what is most observable: published articles, or goal states, $g \in G$. We might collect articles, g, and seek to *extract* details about the event x in the article. However, such an approach fails for two reasons: (1) the representation of the event, x in g, is biased based on how it was portrayed in the article. This is not insurmountable — as we will see in other sections, we can make inferences to correct these biases. However, more importantly, (2) this *only gives us information about positive newsworthiness*, or the events that *did* get covered (i.e. $\pi(a = 1|x)$), not those that did *not* (i.e. $\pi(a = 0|x)$).

A core challenge in training policy models $\pi(a|x)$ (recall, a is a binary action-set where $a = 1$ means that x is covered and $a = 0$ means that x is not) is that $\pi(a = 1|x)$ is often not enough to learn robust policies [213, 214, 215]. We address by first learning M_ψ, a linking function that helps us infer not only what policies *were* covered, but also what policies *were not*. Without M_ψ, our models will lack information about the universe of

[4]We search for all URLs matching wildcard pattern `https://www.sfchronicle.com/*`

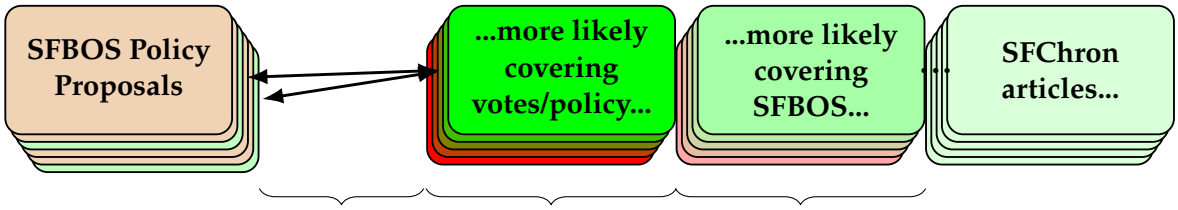
[5]Using `https://github.com/codelucas/newspaper`.

[6]We release the full list of URLs `https://github.com/alex2awesome/newsworthiness-public` and extended data collection here `https://github.com/alex2awesome/explainable-controllable-newsworthiness`, as well as scripts to replicate our collection process.

[7]`https://sfgov.legistar.com/Calendar.aspx`

[8]Example meeting: `https://sfgov.legistar.com/MeetingDetail.aspx?ID=1108038&GUID=8B3A2668-90A9-43E9-A694-8747176617F4`

[9]Example of a policy proposal: `https://sfgov.legistar.com/LegislationDetail.aspx?ID=6251774&GUID=420031B2-94DE-440F-AB74-25FF091F2D61`



$$p(l|g,x) = \sum_{h_3}\sum_{h_2}\cdots\ p(l \mid g,x,h_2,\ldots)\ p(h_2 \mid g,x,h_1,\ldots)\ p(h_1 \mid g,x\ldots)\ldots$$

Figure 2.4: Our probabilistic relational modeling (PRM) process for whether an article g covers a city council proposal, x, i.e. are linked, l. PRM works by introducing auxiliary marginal variables $h_1, ...h_n$ that refine the link model, $p(l|g,x)$ through conditioning. In the diagram, moving from right-to-left, each step shows another variable h_i being applied in the PRM-chain: e.g. h_2 =*"covering SFBOS"*, h_3 =*"covering SFBOS votes and policy"*. h_2, h_3, etc. can be learned separately, and we learn supervised models for each step.

policies that *seem* newsworthy, on the surface, but were not covered by journalists for, likely, important reasons. Determining that a policy[10] was covered in media, as shown in Figure 2.3, is a challenging task. Unlike related tasks, like *citation prediction* [216] or *cross document event-coreference* [217], determining policy coverage requires us to establish links between documents in two different linguistic domains, with no pre-existing labels. Our first challenge is to establish when a news article references a specific local policy document, i.e. to *link* them, allowing us to make inferences about policies that *were* covered and policies that *were not*.

We discover that, despite lacking a labeled dataset of policies labeled as *covered* or *not*, we can nevertheless learn $M_\psi(x,g)$ by breaking this problem down into a chain of decisions, each simple enough that a language model can make a reliable inferences. Eventually these inferences, when conditioned on the previous ones[11], give us high confidence that a coverage link does exist. This is an application of probabilistic relational modeling (PRM) [218] that, we show, helps us outperform other retrieval-based baselines.

More formally, we seek to model the likelihood a link l exists between an article, g, and

[10] A local government policy item is a motion of gov.: a proposal, bill, amendment, settlement, law, etc.

[11] Shown in Figure 2.4, i.e. "article covers local politics" → "article covers city council meetings" → "covers past meeting" → "covers *this* past meeting"

PRM-Chain	TF-IDF	SBERT	OpenAI Embeddings
$p(l\|a,p)$, base	16.0	32.1	30.3
$\sum_{h_1} p(l\|g,x,h_1)p(h_1\|g,x)$	28.5	33.9	37.5
$\sum_{h_1,h_2} p(l\|g,x,h_1,h_2)p(h_2\|h_1,g,x)...$	55.3	48.2	53.5
$\sum_{h_1,h_2,h_3} p(l\|g,x,h_1,h_2,h_3)p(h_3\|h_1,h_2,g,x)...$	**68.2**	55.6	62.6

Table 2.1: Results from training PRM chains, using different sentence embeddings to calculate l. l is defined as a mapping between News article $a \leftrightarrow$ Policy mapping p. We establish a score-threshold for $p(l|g,x)$ for each trial using our gold-labeled dataset, $S_{gold,train}$ and report f1-scores using $S_{gold,test}$. TF-IDF is defined [220]. SBERT uses the **all-MiniLM-L6-v2** model [221]. OpenAI uses the **text-embedding-ada-002** model.

a specific policy item, x, or $P(l|g,x)$. In PRM, we learn conditional attributes $h_1, ...h_t$ of either the article, policy, or both and marginalize over them:

$$P(l|g,x) = \sum_{h_1} \dots \sum_{h_t} p(l|g,x,h_1,\dots,h_t) \dots p(h_1|g,x) \tag{2.1}$$

where, as shown in Figure 2.4, h_2 might be "covers SFBOS", and h_3 might be "covers SFBOS votes/policy."[12] (Note that the model $p(h_i|g,x) = p(h_i|g)$ if the attribute h_i is *only* dependent on the article, g.) Not all politics articles are about SFBOS, and not all SFBOS articles cover policy. Such variety confounds unsupervised models, but is solvable when broken into easier-to-supervise subproblems. This is not dissimilar to Chain-of-Thought (CoT) [219], where language models decompose complex reasoning tasks.

Our attribute-based model, as shown in Table 2.1, helps us retrieve $(g,x) \in S_{gold}$ with 68% F1. We show via an ablation experiment that each attribute h_i is important for our final prediction: Table 2.1 shows how F1 drops from 68% to 16% when we remove h_i-conditioning steps. Surprisingly, using PRM with TF-IDF outperforms different embedding methods like SBERT [221] and OpenAI embeddings [222]. We suspect that specific technical phrases are important for this task, which unsupervised embeddings might ignore; training a supervised retrieval architecture like Dense Passage Retrieval (DPR) might help represent

[12]Because no natural linking information exists (i.e. hyperlinks in the article body), we typically model l_* on the text of the article and/or policy proposal.

these phrases in the embeddings, but as reported by [223] requires 100-1000 times more data than we have collected. Our PRM approach also outperforms retrieval-specific methods like BM25 [224]. Overall, these results indicate that attribute-specificity of PRM is crucial[13]. We note that our PRM approach can seen as a supervised variation of CoT reasoning [219] (albeit with a wide beam). As language models become cheaper and more scalable, more directly applying CoT-style approaches to either identify hidden attributes to train auxiliary classifiers, or directly link articles and policies, could be a viable approach.

Despite our positive results, we acknowledge that our approach is limited in several ways. First, as mentioned above, our identification of hidden attributes was based on manual error analysis and, ultimately may not a scale to new domains. Secondly, another limitation we face is that if there is no lexical overlap between g and x, we would not discover a link even if there were one. Also, we might be more exposed to this risk than the results show: in constructing S_{gold}, our annotators might have also faced a similar bias depending on the retrieval mechanisms (e.g. search) they used. A more comprehensive evaluation set would be generated by journalists *as they are working* on stories.

2.2.4 Learning a newsworthiness model

Next, having established links, we seek to learn π^*, in other words, we seek to learn the expert policy that determines $\pi(a = 1|x)$, if a *new* policy x will get covered ($a = 1$). We use our linked dataset $\{(g, x)\}$, described previously, and treat this problem as a prediction problem where:

$$\pi(a|x) = \begin{cases} 1, & \text{if } x \in \{(g, x)\} \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

Our goal is twofold: (1) Learning a good policy model π can show us which features of events x lead to coverage. (2) Performing this task well at inference time takes us steps

[13]To implement BM25, we index g and use x as a search query. We use the `retriv` Github package: `https://github.com/AmenRa/retriv`.

Policy Features Analyzed
text of proposal # prior meetings proposal has been discussed # prior news articles linked to proposal
length of time proposal is discussed in meeting transcribed text of city-council member's policy discussion # public commenters discussing the policy summary of public commentary

Table 2.2: Summary of features for each policy item. Top section is generated via (a, p). Bottom section is generated via SFBOS video transcriptions.

closer to building tools that will be useful for surfacing potential stories.

Previously, *newsworthiness* has been addressed as a feature-detection problem, as in [225], where engineered-features measured specific criteria[14]. Researchers examined combinations of features to find newsworthy items but could miss items if their newsworthiness did not fit the measurements. The *emulation* learning approach, though, dictates that, having *inferred* volumes of actions from our PRM model, we can now formulate our task as a prediction task and learn a far more complex pattern of newsworthiness norms. We extract features from the linked (g, x) pairs derived in the first section to construct our training corpus. As shown in Figure 2.3, in the news article, there are remarks: "After 14 months of delay", "widespread debate" that seem to indicate that there aspects of this policy that are *not* solely related to its topic that made it newsworthy.To capture some of these features, we include SFBOS meetings where these policies are discussed. We download audio for all meetings in our corpus[15] and we use the WhisperX package [227] to transcribe and perform speaker-diarization. See [17]'s Appendix for more about aligning transcripts. We associate each (g, x) with a specific meeting if: (1) x is discussed in the meeting and (2) g was published within a month of the meeting occurring.

Finally, in every SFBOS meeting, there is a special time for members of the public to speak, called "Public Comment". Since good newswriting is emotional [228], we hypothesize

[14]E.g. "statistically anomalous" [226], "sentiment=happy"
[15]Example: `https://sanfrancisco.granicus.com/player/clip/43908`.

Δ Word Distributions for Newsworthy vs. Non-Newsworthy Text							
Policy Text				Meeting Speech		Public Comment	
authorizing	-0.41	housing	0.35	supervisor	1.98	budget	0.40
county	-0.30	health	0.31	think	0.89	philippines	0.16
grant	-0.26	board	0.30	know	0.82	solar	0.15
lawsuit	-0.25	ordinance	0.29	want	0.78	medical	0.15
bonds	-0.23	covid	0.28	people	0.76	covid	0.14
settlement	-0.22	department	0.23	like	0.58	caltrain	0.14
contract	-0.21	cannabis	0.22	need	0.43	rooms	0.13
expend	-0.19	election	0.21	president	0.37	amendments	0.12

Table 2.3: Most likely words associated with newsworthy policy proposals, meeting speech and public comment, measured by $p(w|Y(x) = 1) - p(w|Y(x) = 0)$, where $p(w|.)$ is based on observed word counts. Also shown in the left-most column is the *least* likely words (negative-valued). Colors shown are a heatmap for easy viewing.

City Lawsuits	Tax/Revenue	Basic Services	Environment	COVID-19	Hearings
francisco	<number>	department	planning	ordinance	health
san	exceed	grant	code	tax	hearing
city	city	housing	findings	tent	case
county	contract	program	environmental	hotel	commission
lawsuit	authorizing	health	street	emergency	filed
settlement	bonds	services	section	covid-19	board
district	revenue	resolution	plan	business	federal
filed	services	california	act	election	supervisors

Table 2.4: Selection of top topics obtained by running LDA with $k = 10$. Color-coding shows the likelihood of a newsworthy city council meeting minute containing a topic, with **green** being more likely and purple being less likely. Titles are inferred topics.

that "Public Comment" might offer an additional lens on a policy's newsworthiness. We determine which speakers are members of the public using diarization to identify speakers that *only* spoke during "Public Comment"[16]. Then, we calculate the lexical overlap between their speech and the policy text. For more details about "Public Comment" and other meeting sections, please see [17]. Features used for newsworthiness prediction are shown in Table 2.2.

[16]We infer the sections of the transcript like "Public Comment" using time-stamped agendas, see [17]'s Appendix for more detail.

2.2.4.1 Newsworthiness Descriptive Insights

Before showing results from the predictive modeling, we show descriptive results. Our main takeaway from this section is that policy text, meeting text and public speakers each are conveying *different* newsworthiness information. We point these out because we will show in the next section, despite clear differences observed in the features that we gathered, not all are semantically useful.

Policy Text, Meeting Speech and Public Comment all cover different newsworthy topics. We see a clear pattern in the kinds of words and topics used in newsworthy policies, meeting speech and public commenters. Table 2.3 shows the top most likely words in each aforementioned text category, calculated as $\Delta p(w) = p(w|a = 1) - p(w|a = 0)$. In the written policy text, we observe topic-specific words like "housing", "covid" and "cannabis" more in newsworthy policies. Topics that were more likely to receive coverage, shown in Table 2.4, include "Hearings" and "Environment". However, meeting speech for newsworthy policies (which is primarily speech of the SFBOS Supervisors and staff) is directed at deliberation, like "think" and "know". Finally, during public comment, we see topic-specific speech, but related to a different set of concerns, like "solar", "caltrain", "hotels". We hypothesize that these are each different aspects of newsworthiness that are being conveyed.

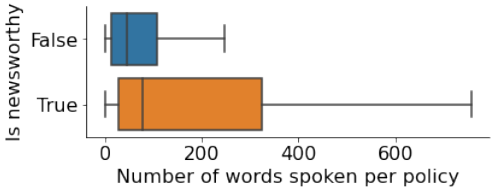


Figure 2.5: Number of words spoken per meeting for newsworthy policies versus non-newsworthy policies.

Newsworthy Policies are addressed for longer at meetings, by more people. Policies that end up getting covered in SFChron are also discussed at greater length than policies that are not: this includes (1) more words spoken (Figure 2.5), (2) more minutes spent discussing (7.7 minutes vs. 2.1), and (3) more speakers spent addressing it (4 speakers vs. 2.2. This

Full Prompt Example
(1) Policy description: "Priority for Veterans with an Affordable Housing Preference under Administrative..." Presented in 2 prior meetings, 0 news articles
(2) Introduced by 4 speakers in the meeting for 0.7 minutes: "...Without objection, this ordinance is finally passed unanimously. Madam Clerk..."
(3) 1 members of the public spoke for 1 minutes. "<SPEAKER 1> spoke for 1 minutes and said: "Hello, this is [redacted]. I would like to oppose the motions affirming..." Is this newsworthy? Answer "yes" or "no".

Table 2.5: Example prompt that shows 3 primary components: (1) **Policy text**, (2) **Meeting text** and (3) **Public commentary text** (name censored). Text is truncated at first 50 words. Further truncated in this example for brevity. Section lines/numbers shown for clarity.

number includes members of the public and council members.)[17]

The number of public commenters we are able to associate with specific policies, on the other hand, is a relatively small number. We are only able to establish an expected $n = .06$ speaker per newsworthy policy and $n = .04$ speaker per non-newsworthy policy. This amounts to 768 speakers associated, overall, with 13,089 policies. Thus, we hypothesize that public comment will not impact our modeling performance, despite observations in Figure 2.3 that public commenters tend to speak to different topics. We acknowledge this as yet another limitation of our work and dataset. We hope that future work can either (1) establish better methodologies to associate more public commenters with policies (2) collect larger public meeting datasets or (3) incorporate other channels (e.g. social media).

2.2.4.2 Newsworthiness Predictive Insights

In order to jointly model numerical and textual features, we choose to format our features jointly as a prompt. The structure of our full prompt is shown in Table 2.5, and it includes all features listed in Table 2.2. We limit the size of the prompt by providing only the first

[17]Journalists gave us initial feedback, saying that city councils sometimes shove important policies into sections of the meeting like "Consent Calendar" and "Roll Call", which are typically *not* addressed for a long period of time. This implies either that these cases are truly a minority, or that not enough attention is being paid to these sections of the meeting.

Train	F1	ROC	MRR	R@10	n
'13-'21	25.4	75.9	.26	64.4	1,595
'13-'20	18.9	68.8	.22	52.8	1,289
'13-'19	21.8	69.9	.22	53.9	1,084
'13-'18	19.5	67.8	.23	55.0	867
'13-'17	17.9	66.1	.22	52.2	693

Table 2.6: We alter the training split date cutoffs to be prior to Jan 1st on each of those years to test whether GPT is learning to fit to specific newsworthy events (e.g. "COVID-19") too well, or whether it is picking up broader newsworthy trends.

50 words of the text fields (besides "proposal text"). We do not notice any impact of this truncation in early experimentation. We use this prompt to fine-tune the GPT3-Babbage model, shown to be a robust classifier [1], outperforming architectures designed for text classification [145]. The length time spoken might be a more important variable than the time spoken itself.

Policy text is the most predictive newsworthiness attribute, followed by meeting discussion and then public comment.

In our first set of experiments, we ablate the prompt to explore which components of the policy are the most important for assessing newsworthiness. We adopt a temporal hold-out with cut date $t_0 =$ Jan 1, 2021, defining the splits $D_{\text{train}} = \{(x_i, a_i) : t_i < t_0\}$ and $D_{\text{test}} = \{(x_i, a_i) : t_i \geq t_0\}$, where $a \in \{0, 1\}$ is the label. The training set is class-balanced with counts $\mathbf{n}_{\text{train}} = (n^{(1)}_{\text{train}}, n^{(0)}_{\text{train}}) = (641, 627)$, giving empirical priors $\hat{\pi}_{\text{train}}(a = 1|x) \approx 0.506$ and $\hat{\pi}_{\text{train}}(a = 0|x) \approx 0.494$. The test set preserves the natural prevalence with $\mathbf{n}_{\text{test}} = (n^{(1)}_{\text{test}}, n^{(0)}_{\text{test}}) = (180, 2310)$, i.e., $\hat{\pi}_{\text{test}}(a = 1|x) \approx 0.072$ and $\hat{\pi}_{\text{test}}(a = 0|x) \approx 0.928$. We perform a time-based split rather than a randomized split because our goal is to test how well $\pi(a|x)$ extends into the future.

We find that the full prompt performs the best across all metrics we considered, but only marginally. Ablating "Public Comment" from the prompt barely impacts performance, while ablating all "meeting info." impacts more. Removing "policy text" information, thus forcing the model to just rely on meeting text alone impacts performance dramatically.

Model	F1	ROC	R@10	MRR
Fine-tuned GPT3-Babbage				
full	**25.1**	**75.9**	**64.1**	**29.2**
(1), (2)	24.2	71.2	63.1	27.2
(1)	16.2	64.5	52.2	23.1
(2), (3)	14.4	57.6	37.2	15.9
LR, full	19.7	67.3	51.1	22.8
GPT4, full	18.4	62.6	40.6	16.2
GPT3.5, full	13.4	63.2	46.7	21.3

Table 2.7: Results of policy model training, $\pi(a|x)$ from fine-tuning GPT3 on full and ablated versions of the prompt. Bottom sections show our baselines, Logistic Regression (LR) and vanilla GPT4/GPT3.5. All rows with (full) show models that were trained on full input prompt (Table 2.5). Rows with numbers, e.g. (1), etc. are ablation models trained with those parts of the prompt. Metrics are: F1, ROC-score over logits for "yes" tokens, Recall@10 (R@10) of each meeting (i.e. we surface the 10 most likely newsworthy items, count recall) and Mean Reciprocal Rank (MRR) of newsworthy policies, per meeting.

GPT3, unsurprisingly, outperforms a very simple classifier, TFIDF+Logistic Regression (LR in Table 2.7), but not by much, indicating there are simple textual cues we are learning.

GPT4 might be capturing national newsworthiness trends. Vanilla GPT4 outperformed our expectation. We had hypothesized that many of SFChron's newsworthiness judgements on SFBOS were local. GPT4 underperforms most other classifiers, but not by much. Manual analysis we perform finds that many errors were GPT4 failing to identify *locally newsworthy* items (e.g. "local scooter ban", local street renaming) and that many correct predictions were made on *nationally* newsworthy trends (i.e. "COVID-19 responses"). There are two likely conclusions: (1) SFChron has major overlaps for newsworthiness judgements with national newspapers, and (2) general newsworthy language and framing is *also* used for local newsworthiness.

Newsworthiness judgements are surprisingly consistent across time, with *one major exception*. Table 2.3 shows that words related to specific events (e.g. those related to "COVID-19") are reflected in the perceived newsworthiness of policy: is the model fitting

Human Validation on Different Tasks:	Metric	Score
Linking: How well can we identify prior x that was covered in news articles g (i.e. $M_\psi(x, g)$)?	Human F1	63.2
	(Model F1)	*(58.9)*
	Cohen's κ	36.3
Recommending: How useful is a recommendation system recommending the top $k = 10$ policies by estimated $\pi(a{=}1 \mid x)$ score	Preference	84%
	ID Accuracy	74.2%
	Cohen's κ	60.0

Table 2.8: Results from human evaluation. Top row: journalists identify real newsworthy policies, by meeting, given a balanced dataset of $n^{(1)} \approx 33\%$ (or $x|a = 1$) and $n^{(0)} \approx 66\%$ (or $x|a = 0$). Model f1-score is much higher than Table 2.7 because this is a balanced sample. Bottom row: preference test for lists of newsworthy minutes (generated viaour models vs. random) and identification (ID) accuracy for list-origin.

to a specific event (e.g. "COVID-19") that happens to be newsworthy in our training and test data, or is it learning either (1) larger event-types (e.g. pandemics more generally, like "ebola", are recurrent and newsworthy) or (2) newsworthy language patterns and other non-semantic attributes (e.g. framing)?

To test this question, we retrain our model and increasingly restrict the date cutoffs of our training set to ask whether a model would correctly predict the newsworthiness of policies pertaining to specific events (e.g. "COVID-19") if the likelihood of them being in the dataset were to decrease. We show in Table 2.6 that, except for a dropoff after excluding data from 2021, our performance does not significantly change. We are additionally able to replicate these findings with baseline Logistic Regression models, demonstrating that this is not simply the result of GPT3's pretraining. An error analysis shows that "COVID-19"-related news was the least likely to be predicted correctly, and is the main contributor to this performance decrease; our models correctly predicted numerous other specific events (e.g. environmental, transportation-related, fire-arms related events). We take this as evidence that *major* anomalous events, like COVID-19 specifically, do become newsworthy and are unpredictable given our current approach. This highlights an important limitation and needs to be taken into account if these tools are deployed: they must be used *along with* models tuned to these blind spots.

Human journalists find our newsworthiness judgments predictable and helpful.

We recruit two expert journalists[18] and conduct human experiments with two aims: (1) is our "newsworthiness" definition repeatable and (2) are our models helpful? For the first, we test how well *humans* able to identify newsworthy SFBOS policies. We construct a dataset by taking newsworthy policies from SFBOS meetings in our test set and a sampling non-newsworthy policies in a 1-to-2 ratio of $n^{(1)}$ vs. $n^{(0)}$. As shown in Table 2.8, our best models achieve 58.9 F1-score on this dataset, and humans score almost equivalently. It's tempting to think our models have reached a ceiling; however, the journalists are not San Francisco-based, and are thus untrained, compared to our models. Finally, to test how useful our learned policy model $\pi(a|x)$ can practically be, we use π as a recommendation model. We surface the top $k = 10$ policies where $\pi(a = 1|x)$ is the highest *from each meeting* and ask journalists to (a) indicate which policies they might write about and (b) guess whether the list was a newsworthiness list or a random sample (they were told that it was a secondary method, not random). Journalists preferred our lists to random 84% of trials.

2.2.4.3 Summary

In summary, this experiment shows the challenges of observability, even in seemingly simple horizon-1 decision-making settings where we take an *Emulation Learning* (EL) approach. We demonstrated that not only could a PRM-based linking function $M_\psi(x, g)$ help us develop a more nuanced inverse function $q_\theta(\tau|x)$ but it could help us approximate policy functions $\pi(a|x)$ that were practically useful for journalists. We will be expanding this work for other localities and seeking to gain greater insight into the specific decision-making processes by applying concept bottleneck models in the future [229].

[18] Combined have > 40 years of newsroom experience

2.3 Which stories are *more* newsworthy than others?

Now, let us revisit some of the simplifying assumptions we acknowledged when we first simplified *newsworthiness prediction* to learning a binary prediction policy model $\pi(a \in \{0, 1\}|x)$, in Section 2.2. To review, we faced two limitations. (1) First, the goal of the previous task was to predict *whether* or *not* an event would get covered *at all* in the newspaper. However, many events get covered published every day – *The New York Times* publishes 300 articles a day [187]: many are more newsworthy, many are less. (2) Secondly, we simplified the input to our policy model $s_0 = (x, c)$. x was the event and c represented external factors (e.g. newsroom coverage loads, events in the world), yet we assumed c was constant. Can we learn a better and more nuanced policy model, $\pi(a|x)$ covering a wider range of newsworthiness while also incorporating external factors c?

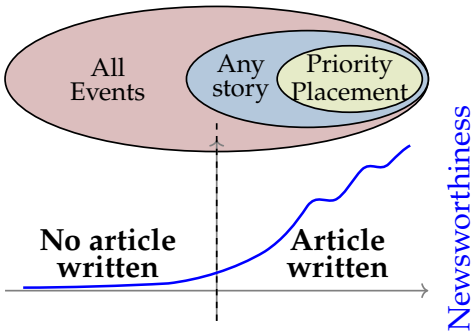


Figure 2.6: A more granular spectrum of newsworthiness, and how it is communicated to readers, goes beyond whether or not an event is covered in a news article (i.e. *Events* vs. *Stories*). Some stories are prioritized more highly within the newspaper.

I will first outline my new approach, which incorporates additional data about human decision-making, and then define it more formally as an EL task. As shown in Figure 2.6, even once an event *is* covered in a news article, it can be promoted *further* by editors based on how it is positioned relative to other articles. Visual cues for editorial preferences on homepages have a deep history in the design principles of physical newspapers [230]. At *The New York Times*, for example, top editors and designers convened daily in a "Page One" meeting [231] to determine the most important articles to put on the front page of the print newspaper the next day. Typically, the most important decision was which stories get featured on the front page, or page A1, of the newspaper; terms like "above the fold" also emerged to signal story-importance (i.e. the story is above the point at which the newspaper folds, so it is

seen on newsstands). In prior work [232], I found that these positioning decisions can help us train a useful policy model $\pi(a|x)$ for the newsworthiness prediction task. A simple policy model trained to predict whether or not a piece of text would appear on the *front page* of *The New York Times*, $\pi(a = \text{front page}|x)$, could be learned and could generalize to different textual domains.

In the digital era, Page One meetings evolved into *Homepage Meetings* [234], influencing the design and content placement on the website's homepage for the upcoming day. As such, homepages continue to be distillations of professional judgment and priorities. One visual cue editors use on homepages is **positional placement**, with articles positioned towards the top and left of a page considered more important [235]. This stems from observations that readers naturally begin scanning from the top-left corner [236]. Secondly, the **space** articles occupy is considered: larger articles or headlines are perceived as more important [237]. In print media, prominence is conveyed through more column space; in digital media, longer headlines, featured images, and extended summaries are used. Finally, **graphics and design** also play a pivotal role in signaling the importance of news stories. Articles accompanied by photographs, videos, or other multimedia elements are often deemed more significant [238]. The use of design elements (e.g. capital letters, bold fonts, and color) further enhances a story's prominence. The way humans spatially organize information reflects a key signal of preference [239]: the homepages of news organizations are one such artifact where spatial

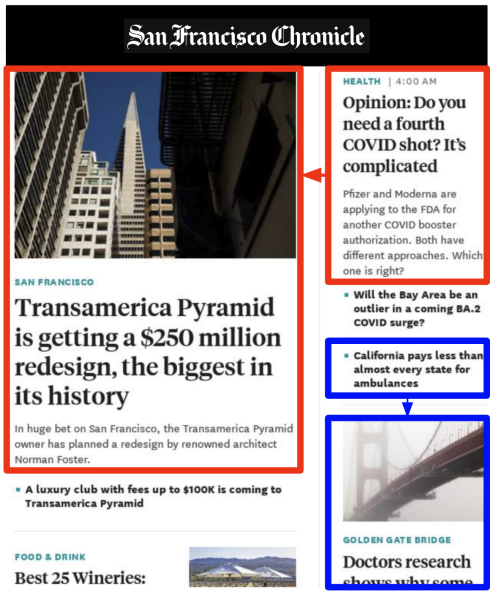


Figure 2.7: Two "newsworthiness" signals that editors make to guide reader attention are shown above. (1) **Position** (i.e. articles that are placed above, ↑, and left, ← relative to other articles are more important [233]). (2) **Size** (i.e. articles that are larger than other articles are more important) (3) **Graphics and Design** (i.e. articles with graphics and images are more important).

organization can be studied at scale. Meticulously crafted by professional human editors, their layouts reflect the informational preferences of newspapers [240] and shape public perception. Inspired by my early work modeling the page placement of articles in the physical newspaper, I will now introduce an experiment that considers spatial positioning of articles *relative to each other* on an outlet's *homepage*.

My goal here will be to push the bounds of *newsworthiness prediction* using additional information about spatial layouts. I ask two primary research questions: (1) First, *how well can spatial layout signals be used to model editorial preferences?* Can we capture these layout signals by considering pairwise comparisons between articles, as shown in Figure 2.7? (2) Secondly, *do models for editorial preferences generalize across different corpora and are they be useful in different contexts?* In other words, can they serve a newsworthiness prediction role, $\pi(a|x)$ for x that is not news (i.e. recall, in the last section, $x =$ city council policies)?

2.3.1 A Pairwise Comparison Model

Now, let us conceptualize how to approach spatial positioning with an *emulation learning* approach. Recall that, previously, our inverse function $q_\theta(\tau|g) = q_\theta(a \mid x, g_{\text{obs}})$ used the linking model $M_\psi(x, g)$ as an observation channel — with recall $R(x)$ — to convert detections/non-detections into soft posteriors over the latent action a. These posteriors supervised reward modeling to fit a policy model $\hat{\pi}(a \mid x)$ matching $q_\theta(a|x, g)$, yielding a newsworthiness predictor for new events. Now, we extend the EL framing in two ways. First, each episode now starts at $s_0 = (x, C)$, where x is a candidate article and C is the contemporaneous set of competitor articles $x' \neq x$ on the homepage. Then, to incorporate these comparisons, we extend beyond binary actions: from $a \in \{0, 1\}$ ($1 = cover$, $0 = ignore$) to an action set that is continuous, $a \in \mathbb{R}^d$, encoding a score that compares x relative to C (e.g. based on placement/visibility: size, position, font, etc.). The trajectory remains horizon-1, $\tau = (x, a, C)$, and the goal g is the observed placement of x, within the realized layout. The inverse function $q_\theta(\tau \mid g)$ recovers the

latent placement action given the observed layout: $q_\theta(a \mid x, C, g)$; the policy $\pi^*(a \mid x, C)$ maps an article and its context to a distribution over placements[19]. In practice, we do not observe a directly; instead we observe *pairwise preferences*, implied by the layout. We define an observation channel to encode these preferences, $p_o(x > x')$, that compares two articles, x and x'. $p_o(x > x') = 1$, for example, if x appears larger/more prominent than x'[20]. Pairwise preference modeling provides a natural observation channel for EL because any latent-utility model of homepage placement can be reduced to a product of pairwise comparisons. This guarantees that, under the assumption of *transitive utilities*, pairwise comparisons suffice to recover the inverse distribution q_θ and to supervise the policy π_θ [241, 243]. Aggregating $p_o(x > x')$ against all $x' \neq x \in C$ yields a posterior over a latent utility $u_\theta(x, C)$ (or over a); external factors c enter *through* C. The inverse function $q_\theta(\tau \mid g, x, C)$ depends on which $x' \neq x$ are on the page and treats newsworthiness as *relative*.

Conceptually, the preference model $p_o(x > x')$ plays a dual role. For the *inverse* step, it supplies the observation likelihoods that turn layouts into soft responsibilities over actions/utilities. For the *policy/reward* step, the same constraints inform $\hat{\pi}$ (via pairwise or listwise losses), ensuring that the learned policy reproduces observed prioritization under varying C. Moving beyond the binary formulation, $a \in \{0, 1\}$, this preference-based observation model directly encodes competition through C, and identifies a *continuous* actions/utility that governs ranking and spatial allocation. It is more data-efficient under partial observability (we need not recover exact coordinates to learn a consistent order), naturally generalizes to listwise ranking and layout optimization, and *subsumes* the binary case as a special limit. When C collapses to a single "null" competitor and observations reduce to detection/non-detection, the framework reduces to the earlier publish/ignore model, with the binary decision recovered by thresholding the learned utility/reward.

[19] In discrete-slot layouts, a Plackett–Luce policy ranks the slate by $\{u_\theta(x, C)\}$; in a continuous view, a softmax over slot utilities (with slot weights) yields a distribution over placements. We take a simpler approach in this work; as described in Section 2.3.3, we use our observation model $p(x > x')$ to perform pairwise comparisons to rank x relative to all other $x' \in C \setminus \{x\}$

[20] We can implement this in different ways (e.g. via a Bradley–Terry/Thurstone likelihood $p_o(x > x' \mid \cdot) = \sigma\big(u_\theta(x, C) - u_\theta(x', C)\big)$ [241, 242]). We train a logistic regression model, described in Section 2.3.3.

2.3.2 News Homepages Across the World: Our Dataset

To implement *Emulation Learning (EL)* in this setting, we need to be able to generate comparisons between articles, $p_o(x > x')$; thus, we need to collect observational data to support these comparisons. We compiled a large and continuing dataset of homepage snapshots, then, we bootstrapped a layout parsing model to detect the positioning of articles on homepages. I will describe each step in this process now.

We start by compiling a list of 3,489 news homepages, as of the time of this writing, which we scrape twice daily[21] on an ongoing basis over a period of five years. From 2019-2024, we have collected a total of 363,340 total snapshots. Our dataset collection is actively maintained and facilitated by a large contributing community of over 35 activists, developers and journalists. We collect homepages from national news outlets (e.g., *The New York Times*, *The Wall Street Journal*), state-level news outlets (e.g., *San Francisco Chronicle*, *Miami Herald*), as well as local and subject-matter-specific news sources. Table 2.8a provides a sample of the different categories of news homepages included in our dataset, and a full list can be found in [212]. Additionally, we collect homepages from news websites of over 32 countries in 17 languages (please see Tables 2.8c and 2.8b for a more detailed breakdown). This is an ongoing and expanding effort: we have actively encouraged contributors to add their own news homepages of interest using for our suite of tools to scrape.[22] This community helps us diversify the news sources in the dataset that we collect and helps us avoid blind spots; it also helps us to test our EL approach beyond dominant cultures.

2.3.2.1 Data Collection Pipeline

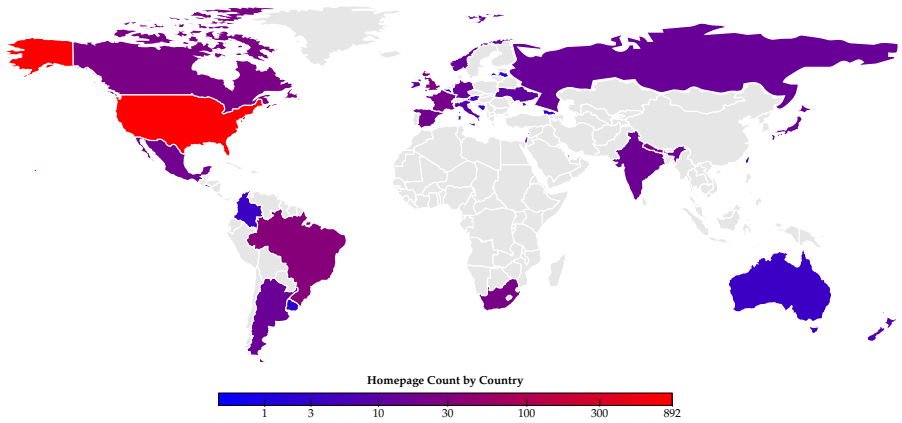
Our dataset collection runs in a `chron` job twice a day, and uploads data to Internet Archive. For each snapshot, we store the following information: 1. **All links on the page:** We store

[21] We chose a twice-daily capture, every 12 hours, to capture morning and evening publishing cycles. This is historically when many news outlets will publish new articles and update homepages [244].

[22] For more information on how to contribute, please see: `https://github.com/palewire/news-homepages`. For all code and data associated with this project, see `https://github.com/alex2awesome/homepage-newsworthiness-with-internet-archive`.

Category	Example Outlets
National	The New York Times, The Wall Street Journal, NPR, Bloomberg
State-level	San Francisco Chronicle, Miami Herald, Chicago Tribune
Local	Sturgis-Journal, The Daily Jeffersonian, LAist, The Desert Sun
Subject-specific	The Weather Channel, Chessbase, ESPN
International	India Today, Ukrinform, BBC, Prensa Grafica, Japan Times

(a) Sample of Homepages by the *type* of news outlet.



(b) Homepages being collected in each country.

Language	Count
English	975
Spanish, Castilian	44
Portuguese	36
Nepali	24
French	21
German	10
Japanese	9
Norwegian	8
Hindi	7
Hebrew	7
Russian	7
Italian	5
Ukrainian	5
Chinese	3
Other	6

(c) Homepages being collected in each language.

Figure 2.8: The *NewsHomepages* dataset is an actively maintained, twice-daily scrape of over 3,489 news homepages. It is updated and collated by a community of over 35 activists, developers and journalists. The breadth of different homepages allows us to study patterns across location and language; to generalize beyond a single set of norms.

a flat-list of hyperlinks on every homepage and associated text. 2. **Full-page screenshots:** We store JPGs of each complete homepage as we render it. 3. **Complete HTML snapshots (subset of pages):** For a subset of homepages, we save a compressed version of the webpage, including all CSS files and images, using SingleFile.[23] In addition to our Internet Archive storage,[24] we also synchronize with Wayback Machine to store these homepages, providing a secondary backup and ensuring long-term preservation.

[23] `https://github.com/gildas-lormeau/SingleFile`, incidentally the same software that Zotero uses. In initial experimentation, we observed that capturing complete, compressed HTML snapshots was far more robust than capturing assets

[24] `https://archive.org/details/news-homepages`

2.3.2.2 Parsing Homepages

To robustly extract visual attributes for each article on a homepage (i.e. size, position, presence of graphics), we need to perform a *layout parse*: we need to determine bounding boxes for all articles on a homepage. Examples of bounding boxes are shown in Figure 2.7 — each bounding box, also referred to as *article card*, covers all information directly associated with that article. Layout parsing is a well-researched field [245, 246]. However, homepages present unique challenges due to their diverse structures – text of varying size, fonts, colors and images – and lack of training data[247][25]. Although homepage layouts are easily perceived by humans, we find that existing resources fail for parsing homepages. Now I will describe how we bootstrap a state-of-the-art layout parser for homepages.

2.3.2.3 Bootstrapping a Bounding Box Detector

On a high-level, our process is: (1) we use a simple deterministic algorithm to generate candidate layout parses, (2) apply a filtering step to exclude low-quality parses, (3) use our high-precision dataset to train a more robust classifier, following other bootstrapping approaches [248]. We describe each step in turn, now.

Step 1: Find Bounding Boxes Deterministically We design a deterministic algorithm, called the DOM-Tree algorithm, to start our bootstrapping process. At a high level, the algorithm traces each <a> tag in the Document Object Model (DOM) and extracts the largest subtree in the DOM that contains *only a single* <a> *tag* (see [212] for illustration). This method can extract the maximal bounding box for each article, however it faces robustness challenges, for example, if a link exists *within* an article card (e.g. a link to an authors page). We apply this algorithm to approximately 15,000 homepages across 15 outlets in the NewsHomepages dataset. Since each outlet typically maintains a consistent layout on their homepages across samples, we include more outlets for generalizability.

[25] Existing work typically focus on parsing text around line-breaks (e.g. paragraph breaks). As can be seen in Figure 2.7, the same article box encompasses many line-breaks.

		FP#1	FP #2	FN #1	FN #2	Total Errors	% Correct
Challenge	DOM-Tree Alg.	117	137	127	265	646	61.3%
	Detectron2	25	23	27	87	162	90.3%
Clean	DOM-Tree Alg.	12	20	0	13	45	97.1%
	Detectron2	15	24	0	18	57	96.3%

Table 2.9: Error analysis of bounding box detection methods comparing the DOM-Tree algorithm and a Detectron2 model across two datasets: the challenge dataset and the clean dataset. The challenge dataset is formed by selecting the bottom 10% of articles based on the match between OCR-extracted text and retrieved link text, while the clean dataset contains well-matched articles. Error types are divided into false positives (FP #1: multiple articles in one box, FP #2: no articles in a box) and false negatives (FN #1: partially captured articles, FN #2: articles not captured). As can be seen, our trained model performs at par on the DOM-Tree algorithm in the clean settings and is far more robust in noisy settings.

Step 2: Filter Low-Quality Bounding Box Extractions We take several filtering steps to prevent dataset impurities, or "drift" [248]. First, we train a simple, reliable text classifier to identify and exclude non-news article links (e.g. log-in pages)[26]. Then, we exclude bounding boxes that did not contain enough text[27] (3) Finally, we exclude bounding boxes with improperly rendered images[28] This filtering process significantly reduced the number of bounding boxes that did not correspond to articles, were broken or corrupt, enhancing the training data quality.

Step 3: Train a Robust Classifier Now, with our dataset in hand, we trained a Detectron2 model [250] to draw bounding boxes around article cards on pictures of homepages. Detection uses ResNet-101 as a backbone with a Feature Pyramid Network (FPN) for extracting multi-scale features and Smooth L1 loss for bounding box regression. During training, we used a base learning rate of 0.02 with a linear warmup over the first 1000 steps. We trained for 10,000 steps with learning rate reductions after 5000 steps, a weight decay of 0.0001 and momentum of 0.9, on 4×A40 GPUs for 24 hours.

[26]We manually labeling 2,000 URLs as "news article" or "not" and train a Logistic Regression classifier based off a bag-of-3-gram representation of each URL. The model achieves an accuracy of 96%.

[27]We determine this by first rendering the HTML pages as images and overlaying bounding boxes, then running OCR to extract the bounding-box text.

[28]Likely due to errors in HTML extraction or dead links. To address this, we rendered HTML pages as an image and used the YOLO object detection model [249] to compare these images to the JPEGs in our archive. If a screenshot was not within 80% of the archived snapshot, we discarded the snapshot.

2.3.3 Newsworthiness Preference Modeling

With precise layout information for 363k homepages in hand, we arrive again at a core question of this *newsworthiness* formulation: can we model the editorial preferences in homepage layouts?

2.3.3.1 Preference Modeling Approach

Performing a full comparison of (x, C) presents a number of challenges. Firstly, publishing volumes are non-uniform: some days have lots of news (and many newsworthy stories) while others have less. Secondly, a homepage is intended to present a collection of articles as a cohesive bundle: individual articles do not exist in isolation [251]. Predicting the placement of a single article without considering surrounding context would limit information [252]; conversely, attempting to predict the placement of all articles simultaneously poses a combinatorial challenge. Finally, certain areas of homepages (e.g. "Latest News" feeds, which are ordered based on chronology) lack editorial decision-making altogether [253].

As stated in Section 2.3.1, we attempt to address these challenges by reducing our inverse function $q_\theta(a|x, g, C)$ and our policy function $\pi(a|x, C)$ into a pairwise preference comparison, invoking the *transitive utilities* assumption [241, 243]. Specifically, we consider pairs of articles (x, x') and train models to predict a binary preference variable p_o, where

$$p_o(x > x') = \begin{cases} 1, & \text{if outlet } o \text{ prefers } a_1 \text{ over } a_2, \\ 0, & \text{otherwise.} \end{cases}$$

The pairwise preference model $p_o(x > x')$ allows us to recover the inverse distribution q_θ and to supervise the policy π_θ by converting each homepage layout into likelihood factors: $q_\theta(a \mid x, C, g) \propto \pi_0(a \mid x, C) \prod_{x' \in C} \Pr\big[p_o(x > x') \mid a\big]$. Of course, the *transitive utilities* assumption may not hold for real newsworthiness judgments: pair (x, C) may involve higher-order interactions (e.g., thematic bundling of articles, or article diversity

constraints) that violate transitivity; in such cases, pairwise models still offer a tractable approximation that captures the dominant utility signal while leaving space for richer listwise or set-based extensions [254, 255]. In addition, the pairwise formulation learns from the set of articles that *actually* appear on a homepage. As such, it estimates relative prominence among events x that *might* be covered. We have no guarantees that it will extend to distinguishing *covered* from *uncovered* events, discussed previously.

In this work, we limit the layout variables we consider to: *size* and *position*. We explore three combinations of these variables to create weak labels for the preference variable, p:

1. **Size-based Preference**: We define $p_o(x > x') = 1$ if article x occupies more surface area on the homepage than article x': prominent articles are given more space [256].

2. **Position-based Preference**: We set $p_o(x > x') = 1$ if article x is placed in a more favorable location on the homepage than article x', such as higher up or more to the left, based on common reading patterns [257].

3. **Combined Size and Position Preference**: Here, $p_o(x > x') = 1$ if article x either occupies more surface area or is in a more favorable position than article x', particularly focusing on articles that are in the top 10% by size on the page.

While there are other design variables that could give an even finer-grained preference (e.g. font, color, images), we seek here to establish that even a coarse weak labeling can still provide valuable insights. To model our weak preference labels, p, we train a simple Transformer-based binary classifier, `distilbert-base(X)`, which classifies a text sequence X. Our model concatenates the input articles: `X=a`$_1$`<sep>a`$_2$ as input; the model learns to recognize the `<sep>` token as a boundary between the first and the second articles.

	Size	**Position × Size**	**Position**
Model Name	F1 (Weak/Human)	F1 (Weak/Human)	F1 (Weak/Human)
Flan-t5-base	91.9/28.4	70.7/65.5	64.5/56.1
Flan-t5-Large	66.6/20.2	54.9/61.0	34.5/58.2
Roberta Base	91.0/26.6	64.9/62.9	37.3/53.9
Roberta Large	85.4/25.1	47.2/65.1	49.3/56.1
Distilbert-Base-Uncased	93.1/31.1	75.2/**70.4**	70.1/61.2

Table 2.10: F1 scores for predicting pairwise preference $p_o(x > x')$ for different features, across different models (on NYTimes data). On the left, we show results in predicting the weak label — coarser variables (e.g. size) tend to have greater consistency. On the right, we show human analysis of models' decisions: finer-grained variables (position x size) have the highest performance.

2.3.3.2 Preference Modeling Variations

We explored modeling variations first on the *New York Times*[29]. We test 5 different models: {distilbert-base-uncased, flan-t5-base, flan-t5-large, roberta-base, roberta-large} and constructed a training dataset of 74,857 article-pairs and a test dataset consisting of 18,715 datapoints consisting of pairs of NYTimes articles from same homepages.

We observed exploding gradients in the flan-t5-large and RoBERTa-large models, motivating us to use a learning rate limit of 5e-5 for all the models and gradient clipping, for the sake of equal comparison. We applied Parameter-Efficient-Fine-Tuning [258] on flan-t5-base, flan-t5-large, roberta-base, roberta-large models to minimize overfitting, as we had limited of datapoints. We used 4xA40 GPUs and 16xA100 GPUs. The distilbert-base-uncased model outperforms other models (Table 2.10) for our weak labels. We run a human validation experiment, enlisting a former New York Times journalist to rank-order 100 pairs of articles in our dataset. Considering these as ground truth, we find that models trained on position and size score an $F1 = .7$. From our list of 3,000 outlets, we select 31 outlets for detailed analysis. We selected well-known outlets in various categories, including different political leanings (left-leaning vs. right-leaning[30]), local and national

[29]We start with the *New York Times* as [187] that meticulous rules, with full-time homepage editors hired, to that homepage layouts reflect preferences.

[30]As classified by MediaBiasFactCheck.com

Outlet	Accuracy	F1	Recall	Prec.
phoenixluc	57.1	70.3	57.4	90.7
newsobserver	75.0	72.5	74.3	70.7
slate	72.4	61.6	66.2	57.7
jaxdotcom	75.2	63.4	65.5	61.4
arstechnica	64.7	17.5	41.4	11.1
airwaysmagazine	72.5	73.7	78.9	69.1
denverpost	73.7	67.8	70.5	65.4
thedailyclimate	82.0	80.9	81.3	80.6
breitbartnews	68.9	22.8	54.7	14.4
foxnews	67.3	38.6	55.6	29.5
motherjones	71.4	63.0	68.7	58.2
thehill	68.8	55.5	59.8	51.7
wsj	70.0	48.0	52.0	44.6

Table 2.11: Pairwise newsworthiness preference judgments, $p_o(x > x')$ across a sampling of different outlets, made by Distilbert-Base-Uncased model trained on (position, size) cues.

levels, and varied subject matters such as science, chess and aviation. For each outlet, we collected between 200 and 300 homepage snapshots, resulting in 1,000 to 50,000 pairs of articles. We created an 80/20 train/test split and trained distilbert-base-uncased models for each outlet. We trained each model with 5e-5 learning rate limit, 3 epochs, 0.01 weight decay. Each article in our dataset includes the textual representation as it appeared on the homepage. To enhance the reliability of our models, we undertake several data processing steps informed by preliminary experiments: (1) we only sample pairs of articles that are adjacent on the homepage, to curate preference pairs that are more likely to be challenging and topically similar. Secondly, we clean the textual representations by stripping out any times, dates, and formatting elements. We also remove author names to prevent the models from learning biases based on authors who might be favored by the organization. Please refer to [212]'s Appendix for a detailed list of the outlets used and the specific number of data points associated with each.

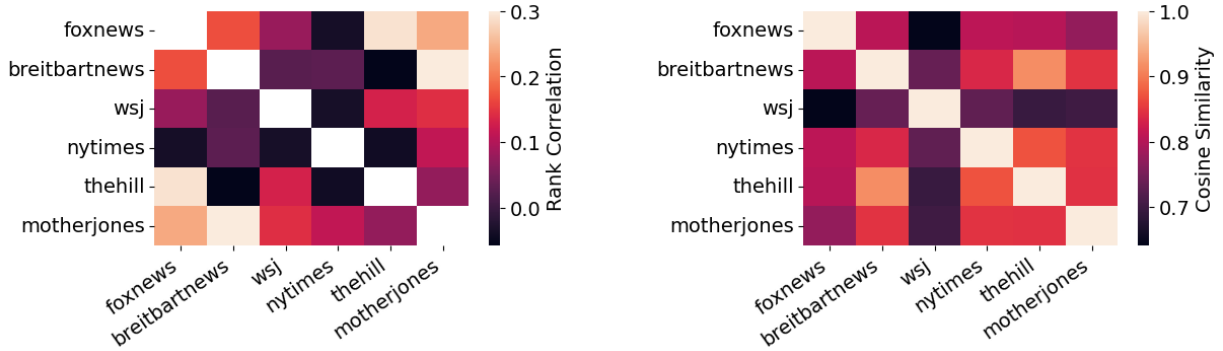
2.3.3.3 Preference Model Results

We show our results in Table 2.11. While some models (e.g. `Breitbart`) perform noticeably poorly, we note that the majority of our models score above $f_1 > .6$. We do not find a significant correlation between model performance and training set size. We were surprised to observe the tractability of this task; this indicates that many concerns we had about noise were either handled by preprocessing, or not as important as we believed.

2.3.3.4 Preference Model Comparisons between Outlets

To demonstrate the analytical insights we can obtain through policy modeling π, we interpret each outlet's trained preference model as an outlet-specific *policy* $\pi_o(a \mid x, C)$ that calculates the newsworthiness *that outlet* would assign to an event x given context C. To quantify *policy agreement*, we apply each π_o to the same held-out article pools A_k drawn from multiple outlets and compute Kendall's τ between the resulting orderings $\{\pi_o(A_k)\}$, thus comparing decisions rather than content. As a control, we contrast these policy-level correlations with topical similarity (e.g., SBERT averages), isolating convergence in editorial *policies* even when content distributions differ.

Now, we aim to rank-order lists of news items drawn from a larger pool of articles to calculate the agreement rates for newsworthiness decisions between different news outlets. Previous research has observed surprising overlaps in sentiment and preferences between right-leaning and left-leaning outlets [259], and we wish to quantitatively test this phenomenon using our preference models. We selected 9 of the 31 outlets for which we trained preference models in the previous section. From each outlet, we sampled 1,000 articles, matching on variables such as topic, length, publication date, and other potential confounders. These 9 outlets were chosen because they represent a range of political viewpoints. For each model n_{o_i} (corresponding to outlet o_i), we used it to sort lists of 1,000 articles $\{a_1, a_2, \ldots, a_{1000}\}_{j=1}^{9}$ from outlets $\{o\}_{j=1}^{9}$. In other words, the output of applying model n_{o_i} to the article list from outlet o_j is a fully sorted list $n_{o_i}(A_j)$. We used the size $\times$



(a) Kendall's τ correlation between the newsworthiness preferences expressed by preference models trained on different news outlets.

(b) Cosine distance of average SBERT similarity between articles sampled from each outlet.

Figure 2.9: Comparison of Kendall's τ rank correlation (on newsworthiness judgements) and SBERT cosine similarity (on articles) across news outlets.

position model for this experiment, as performance was similar to the size-only model, and we believed that the multivariable models capture more newsworthiness information than the single-variable models.

We calculated Kendall's τ, a correlation measure for ordinal data, between each pair of sorted lists $(n_{o_i}(A_k), n_{o_j}(A_k))$ for all i, j, k, and averaged the correlations across j. Figure 2.9a shows the resulting correlation matrix. Some surprising insights emerge: notably, *Breitbart*, a right-leaning outlet, and *Mother Jones*, a left-leaning outlet, have one of the highest rates of agreement, indicating that $\pi_{brietbart}(a|x, C)$ is similar to $\pi_{mother\ jones}(a|x, C)$. This is *despite* them not having high *topical* similarity[31] As can be seen in Figure 2.9b, topical similarity between outlets aligns more closely with political differences: distinct right-wing clusters (e.g. *Fox News*, *Brietbart* and *Mother Jones*) segment from left-wing clusters (*New York Times*, *The Hill*, and *Mother Jones*). Taken together, these results suggest that newsworthiness preference is a novel and orthogonal variable to topical similarity.

[31]To perform this comparison, we compared outlet-level embedding vectors. To derive these vectors, we sampled 100 articles per outlet and generated embeddings for each article using SBERT [221]. Then, we averaged these embeddings to create aggregated outlet-level embeddings [260].

Outlet	Top Policies LLM Summaries	Examples of Policies
Weather Channel	Environmental Policies, Public Health and Emergency Response, Infrastructure and Development	Reducing nutrient pollution from wastewater; Accepting grants for forensic science improvements
Daily Climate	Environmental and Energy Policies, Urban Planning and Development	Agreement with North Star Solar; Building code enforcement
Fox News	Community and Public Safety Policy, Education and Social Policy, Fiscal and Economic Policy	Appointment of individuals to advisory committees; Appropriating funds for San Francisco Unified School District; Developing materials on domestic violence
Mother Jones	Social Policies, Environmental and Health Policies	Sanctuary City Protection; Urging Pardons; Edible Food Recovery and Organic Waste Collection
Ars Technica	Infrastructure Policies	System Impact Mitigation Agreement; 6th St. Substation
NYTimes	Social & Cultural Awareness Policies, Labor & Employment, Economic, Housing policies	Commemorative and Awareness Events; Labor Dispute Hearings; Affordable Housing Loans
WSJ	Economic and Infrastructure Policies, Governance and Legislative Policies	Contract modifications; Bond sales; Ground lease agreements; Charter amendments concerning commissions and departments related to aging and adult services

Table 2.12: **Newsworthiness Prediction using Homepage Models applied to city council policies:** Using our pairwise preference models $p_o(x > x')$ as a policy model, we rank-order city council minutes from Section 2.2.2. Summaries of the top 10 most newsworthy policies published by the San Francisco Board of Supervisors, as ranked by models trained on 7 different homepages.

2.3.4 Newsworthiness Prediction with Homepage Preference Models

We now return to a key question for *newsworthiness prediction*: can the policy model $\pi(a \mid x, C)$ be reliably applied in real-world settings? There are reasons for skepticism. As discussed in Section 2.3.3.1, a key concern is whether a preference model trained exclusively on *published* articles (x, x') will generalize to texts outside the news domain — particularly those that may lie below the threshold of newsworthiness. To test this, we use the list of city council policies gathered in Section 2.2. We hypothesize that editorial preference rankings learned from on news homepages can help us further identify newsworthy content, by training our policy model $\pi(a|x, C)$ to detect more nuanced ranking of the *most* and *least* preferred stories of a news outlet. To test this hypothesis, we applied the preference models learned for each outlet to sort the list of the San Francisco Board of Supervisors' policies (compiled by [17]). Then, we selected the top 10 items from the ordered lists n_{o_i} and used a large language model (LLM) to summarize the key points raised in each policy.[32]

The LLM's summarization results and examples are shown in Table 2.12. We observe various themes emerge, with subject-specific outlets like *The Weather Channel* highlighting policies related to environmental issues and *Fox News* highlighting policies related to public safety. We presented these results to a group of journalists, and 81% of respondents indicated they were impressed and would consider using such a system in their workflow. These findings demonstrate the potential of our models to assist journalists in identifying newsworthy leads from large corpora of documents, thereby supporting investigative journalism and timely reporting.

Our novel dataset and experiments show that homepage editorial cues provide a wealth of resources for (1) novel news analysis and (2) newsworthiness detection [17, 225]. First, as we show in Section 2.3.3.4, editorial decision-making is distinct from simple topic preferences. In fact, information *prioritization* commonalities can be observed between outlets from vastly different political, social and topical backgrounds. Secondly, as we

[32]We used GPT-4 for this experiment.

show in Section 2.3.4, preference models trained on news homepages can be transferred to related corpora (e.g. city council meeting minutes) and can surface relevant minutes to journalists searching for stories.

2.3.5 Summary

In this chapter, we developed and evaluated models for the task of *newsworthiness prediction*, framed through the lens of *emulation learning*. Starting from a binary classification of whether an event should be covered or not, we showed how limited observability necessitated the construction of a linking function $M\_\psi(x, g)$ to infer both positive and negative examples of news coverage. By applying probabilistic relational models, we were able to decompose the linking task into tractable subproblems, yielding robust alignments between city council policies and their associated articles. This linking step enabled us to build richer training data, from which we learned predictive models $\pi(a \mid x)$ that approximate editorial judgments. Importantly, we demonstrated that features drawn from different sources—policy text, meeting transcripts, and public comment—carry complementary newsworthiness signals, though their predictive contributions differ. Our experiments highlighted both the potential and the limitations of this approach, with strong results tempered by challenges around blind spots, sparse signals, and anomalous events such as COVID-19.

Building on this, we extended the scope of *newsworthiness prediction* beyond binary coverage decisions to *relative prioritization* of stories on news homepages. By modeling homepage layouts as collections of pairwise preferences, we captured editorial judgments about prominence, size, and positioning—key signals that translate well into ordinal utility models. Our large-scale *NewsHomepages* dataset enabled us to train preference models across a wide variety of outlets, and to test their transferability to non-news corpora, such as city council proposals. The positive reception from journalists, who found the surfaced leads both credible and useful, underscores the practical promise of

this approach. Altogether, this chapter shows how EL provides a unifying framework for learning from partial, noisy, and norm-driven editorial decisions. It also demonstrates that newsworthiness, while subjective, can be approximated computationally in ways that both deepen our understanding of editorial norms and support new tools for journalists.

2.4 Chapter Conclusion

In this Chapter, we have explored how *observability* challenges arise in *emulation learning*. Specifically when actions a that are observed are distant from goal states g, we must be careful to model a that actually represent to the actions we wish to study (Sections 2.3.3); infer $\tilde{a} \sim q_\theta(a|g)$ in a way that is robust to noise; and ensure that $\tilde{a}$ covers a useful support to learn $\pi(a|x)$ (Section 2.2). I showed how, with the right observation channels, we can address these challenges and recover more robust and nuanced approximations of a. First, in the horizon-1 "publish or not" setting (Section 2.2), where $a = 1$ if an event was covered and $a = 0$ otherwise, and we noticed that relying on g, alone, would only give us information about articles where $a = 1$ and would not cover a wide support ($\text{supp}(x)$ s.t. $\exists\ g \subsetneq \text{supp}(x)$.) We introduced a *linking function* $M\_\psi(x, g)$ and treated it as an observation channel: it helps us recover $a = 1$ if $\exists\ g$ s.t. $M(x, g) = 1$ and $a = 0$ if $M(x, g) = 0\ \ \forall\ \ g$. With this constructed inverse model, $q_\theta(a|g)$, we trained a policy $\pi(a|x)$, we demonstrated that aspects of x — textual descriptions, meeting deliberation, and public comment — all convey *different* facets of newsworthiness. Human studies showed that expert journalists both *replicate* our operational definition of newsworthiness and *prefer* recommendation lists induced by $\hat{\pi}$, suggesting practical value for newsrooms. At the same time, shocks such as COVID–19 exposed a blind spot: when the world changes regime, observation channels learned on past data under-represent emerging salience. Second, to move beyond a binary notion of newsworthiness, we modeled homepages as *sets of pairwise preferences* (Section 2.3), learning $p_o(x\ >\ x')$ as an observation model over

relative prominence and using it to recover utilities (or continuous actions) that rank items within their contemporaneous context C. We showed that weak spatial cues (size, position) can be converted into dense pairwise supervision. Comparing learned policies across outlets revealed an orthogonal dimension to topical similarity: organizations that disagree on content can still *agree* on prioritization. Moreover, those learned preferences transfer: applying $\pi_o(a \mid x)$ to city–council proposals surfaces leads journalists judge as credible. Common among both approaches to inverse modeling, $q_\theta(a|g)$ is the following philosophy: generally, we consider emissions that *can* be observed in our artifacts, g, and construct observation channels that effectively generalize these emissions to latent actions a.

The inverse function $q_\theta(a|g)$ is one of the most important and distinguishing aspects of *emulation learning*, and while this Chapter focuses on some of the crucial challenges that can emerge when trying to learn it, we have barely scratched the surface. The next chapters will move on from *inverse function* modeling and will explore diverse challenges. In Chapter 3, we introduce tasks that go beyond horizon-1 to *sequential* settings to learn more complex policy models $\pi(\tau|x)$. In Chapter 4, we address the *execution* or *realization* of τ into state-space $\mathbf{s} = s_1, s_2 \ldots; s_n = g$. In Chapter 5, we explore datasets that give us richer observability into intermediate state spaces. Although we will not discuss *observability* challenges in the same degree of detail as we did in this Chapter, the challenges of constructing robust inverse models $q_\theta(a|g)$ continue to hover over all *emulation learning* tasks. Indeed, I hope in future work to continue to explore *observability*, and to do so in a more theoretical way. We need research focused on developing a *theory* about which tasks *are* observable and which tasks are not. *Explainability*, I believe, offers one theoretical path: if inferred actions $\tilde{a}$ cannot *explain* observed outputs g, then the inverse function or the action vocabulary $\mathcal{A}$ are lacking. I am excited about continuing to adapt classes of methods in latent variable analysis to *emulation learning* to improve inverse modeling. Bayesian Wake-Sleep Cycle [193, 198], for instance, is one such method that, like many probabilistic models, seeks to infer latent variables z. It bootstraps a Recognizer, $R(g) \to a$

(i.e. our *inverse* model) and a Generator, $G(a) \rightarrow g$ (i.e. our *state-transition model*), starting from synthetically constructed goal-states, g' with *known* structure a, and slowly mixing in human-generated goal-states g. Taking ever-more performant pretrained LLMs as initial Generators, I believe we can follow approaches like Wake-Sleep to extend *inverse modeling* in new and interesting ways. For practitioners of *emulation learning*, this Chapter serves as a reminder to not take inverse-modeling for granted!

Chapter 3

Learning Action Trajectories via Emulation Learning

3.1 *Source-Finding*: A Study in how Information Complements

After journalists select *newsworthy* events to report, described in Chapter 2, they must then find sources to support, confirm and expand their story. This process, *source-finding*, is the creative process we will focus on in this Chapter. As shown in Table 3.1, a typical news article uses a combination of different kinds of sources; these sources can be people, documents, or even databases.

Figure 3.1: In the *journalism pipeline* outlined in Section 1.3, we focus now on the second step: *source-finding*, or finding informational sources to confirm, contextualize and broaden the events being written about. Here the published article is the goal-state g, the (latent) sequence of sourcing actions forms a trajectory $\tau = (a_1, \ldots, a_T)$, and our inverse model $q_\theta(\tau \mid g)$ reconstructs $\hat{\tau}$ from g. We then learn a policy $\pi(\tau \mid x)$ to emulate journalists' trajectories conditioned on context x. Source-finding requires us to learn to complex relationships between information and to reason about a story's narrative needs.

Sources used to inform a sample news article	
Prime Minister **Laurent Lamothe** announced his resignation.	← *from* Statement
The announcement followed a corruption **commission**'s report.	← *from* Report
"There was no partisan intereference" said the **commission**.	← *from* Quote
However, curfews were imposed in cities in anticipation of protests.	← *from* Order
It remains to be seen whether the opposition will coalesce around a new candidate.	

Table 3.1: Different informational sources used to compose a single news article. Source attributions shown in **bold**. Some sources may be implicit (e.g. 4th sent.) or too ambiguous (last sent.). Information types used by journalists are shown on the right. Our central question: *does this article need another source?*

Some sources are used to provide factual details to establish the main event (e.g. the "Statement"); or providing background (e.g. the "Report") — a role we might be familiar with from related NLP tasks (e.g. multi-document retrieval [261, 262]). Other sources play a narrative role: they anticipate reactions (e.g. the "Order), provide anecdotes or give alternate perspectives.

Finding these sources is a crucial part of the reporting process: news articles are *driven* by the informational sources journalists use and retrieving sources takes considerable time. Research has estimated that 30% of journalists' time spent looking for sources and this is the biggest factor separating novice and expert journalists [263, 264]. The practical task we will center around in this section is as follows: imagine a retrieval system that can find different sources for the journalist, as they are reporting. This system will understand both the narrative and factual needs of the story (e.g. "contrasting voice") as well as how to find this source. Articles use 5–7 sources on average [202], and these sources are interdependent. Thus deciding which mixtures of sources to use requires us to consider *sequences* of actions. Additionally, the rewards governing source selection are complex and poorly understood [24]. Reconsider the news story example given in the Preface, the Snow Leopard story.[1] It used the following sources: Brandie Smith (i.e. director of the Smithsonian zoo), Robert Stone (i.e. former presidential advisor), the Holy Bible, and

[1]As a recap, the title of the story was: Leopards on the Potomac! Trump Is Delighted by Deal With Saudis for Rare Cats. published June 4, 2025.

Joseph Maldonado (i.e. subject of the *Tiger King* documentary). As we consider how a *source-finding* tool might begin to recover this sequence of sources to aid a journalist, we are confronted by the difficulty in even specifying *why* they were included. Is it, as has been proposed for other multi-document retrieval systems: for *diversity* and *coverage* [14, 15, 13]? *Factuality* [11, 12]? *Interestingness*, *novelty* or *fun* [265, 266, 267]? These explanations might cover some of the sources in this trajectory, but not all. Clearly, we need to understand complex, contextual, and variant rewards, making an *emulation learning* approach is *essential* for this task. Let us formalize this approach now.

Source-Finding as Emulation Learning: Source-finding requires us to push *emulation* further in this section to consider longer action trajectories, τ; no longer can we simplify the creative task we consider to a horizon-1 trajectory, as we did in Chapter 1. As shown in Figure 3.2, we consider each action in *source-finding*, $a_1, a_2 ... a_n$, to be a *Get Source* action. This is a *composite* action — each a_t includes the following sub-steps: (1) identify the informational needs of the story (2) find the source that meets those needs (3) obtain that information from the source. Once information from the source is obtained, all information from that source is added to the state-space, $s_1, s_2, \ldots$. We can only observe the final news article, g, which contains a representation of the all the information gathered so far. In general, in this Chapter, we will assume that an inverse function operating on *just* the document, i.e. $q_\theta(\tau|g)$, inferring *only* actions $\tilde{a_1}, \tilde{a_2} \ldots$ that *did* occur is enough to train robust policy models $\hat{\pi}(\tau|x)$. This raises *observability* questions: in Chapter 2, observing only actions that *did* occur was not enough to train generalizable $\hat{\pi}(\tau|x)$ functions: we also needed to make inferences about actions that did *not* occur. We assume

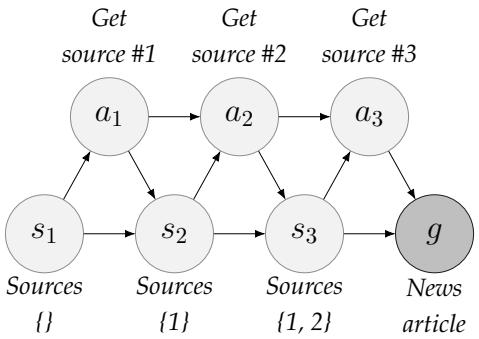


Figure 3.2: *Observability of the source-finding task:* We assume that each action, a_t = *Get source*, successfully retrieves and obtains information from a sources. The state-space, s_t, contains all information gathered so far. Only the news article, g, is observable, and contains a representation of accumulated information.

Cheat-Sheet: Emulation Learning for *Source-Finding*

From the finished article g we infer the latent sequence of "*get-source*" actions to *reproduce* human-like sourcing trajectories, given a story lead.

- **a** a_t **(action)** — composite "*Get Source*" action ("identify need" → "find source" → "obtain info").
- **s** s_t **(state)** — accumulated state (all facts/sources gathered so far) observed only through the final article's content structure (§3, Fig. 3.2).
- x x **(starting context)** — The initial query that starts the reporting process (e.g., first question, event description, press release) (§3.3, §4.3.1.2).
- τ τ **(trajectory)** — Sequence of sources chosen and added to a growing evidence base, used to write the article (§3.3, §4.3.1.2).
- g g **(goal state)** — The published news article whose content contains inferrable source details (§3).
- q $q_\theta(\tau \mid g)$ **(inverse model)** — recovers specific sources ($q_1, q_2, \ldots$ used during reporting process.
- π $\pi(\tau \mid x)$ **(policy model)** — drives "get source" actions a_t. Sub-policies: planner, π_p, *identify information needs*; executor π_e *retrieve sources*; interviewer π_i *obtain information*. Will compare to $\pi^{(\text{llm})}$, implicit policy from pretraining. (§3.3).

we do not need as complicated an approach for multiple reasons. Mostly, for convenience – modeling counterfactuals in *sequences* is harder and the pool of potential sources is infinite, compared with the closed sets of *events* x in 1-horizon trajectories we considered in Chapter 2. Secondly, we assume a larger *equivalence space* among *source-finding* actions compared with *news-finding*: source A and B might have been chosen equally if they are similar (i.e. across many factors), allowing observed actions to teach us more about unobserved than in Chapter 2. Indeed, information-retrieval research treating unjudged items as *unobserved* rather than *negative* yields stable models for such reasons [268]. Finally, state-of-the-art offline RL avoids imputing outcomes for counterfactual actions, recognizing that such imputations accumulate high variance over large horizons [269, 270, 271, 272].

[1] **Other minor notation used throughout:**

- q_i: The source itself. $a_t = q_i$ typically used interchangeably.
- $d(a)$: Discourse role of a source/action, or the *narrative role* fulfilled by the source. $d(a) \in \mathcal{D}$ (e.g., Main Actor, Background, etc). (§3.4.1.2, §3.4.3.2).
- $\nu(s_i)$ narrative needs of the story, or $\nu(g)$ latent requirements a good story should satisfy (§3.4).
- $\psi(\tau)$ — Schema-level signature (e.g., histogram over discourse roles/centrality)(§3.4).
- $L_{\text{emul}} = D(\psi(\tau), \psi(Q))$ — Emulation loss: distance between model vs. human discourse signatures (§3.4).

Chapter 3 Overview

In Chapter 3, *Learning Action Trajectories via Emulation Learning*, we will study how longer action trajectories can be inferred and predicted; how *action spaces* can be compared; and how policies $\hat{\pi}(\theta|x)$ based on *latent* actions can be evaluated. This section will unfold as follows. In Section 3.2, I describe how we train an inverse model, $q_\theta(\tau|x, g)$ to reconstruct trajectories τ from articles g. I will prove these trajectories can be *learned* – i.e. that they trajectories are composite and predictable. Then, in Section 3.3, we will use inferred trajectories, $\hat{\tau}$, to test how well pretrained language models can approximate policy functions $\pi(\tau|x)$. We conclude that policies learned during pretraining models do *not* approximate human policies, and that they specifically are less creative. Next, we begin to break apart the compositeness of the $a_t =$"*Get Source*" action, which is composed of sub-steps: (1) *identify* the story's sourcing needs (2) *find* the right source (3) *obtain* information from the source. In Sections 3.4 I will describe how we can learn *better* policies, $\hat{\pi}(\tau|x)$, including by using higher-order planners that *first* identify the story's needs, *then* find the source. In Section 3.5 we will provide metrics to justify some of the decisions we made in Section 3.4. And, finally, as a bonus, I will show in Section 3.6 how, once we *find* sources, we can use an emulation approach to train models that help us talk to and *obtain information from* sources.

Works Discussed:

- Spangher et al. (2023)". Identifying Informational Sources in News Articles". *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.
- Spangher et al. (2024)". Do llms plan like human writers? comparing journalist coverage of press releases with llms". *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.
- Spangher et al. (2025)". A Novel Multi-Document Retrieval Benchmark: Journalist Source-Selection in Newswriting". *Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing*
- Spangher et al. (2024)". Explaining Mixtures of Sources in News Articles"
- Spangher et al. (2025)". NewsInterview: a Dataset and a Playground to Evaluate LLMs' Grounding Gap via Informational Interviews". *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*

3.2 Identifying Sources in News Articles and Testing Compositionality

Emulation Learning consists of two phases: Inverse Inference, or learning $q_\theta(\tau|g)$; and Policy Learning, or learning $\pi(\tau|x)$. In this section, we will focus solely on the first phase, learning $q_\theta(\tau|g)$. As shown in Figure 3.2, we assume the following observability in our system: g is the published article, or goal-state. g contains partial information about the sources used in the reporting process ($\tau = a_1, a_2, ...a_n$). Previously, in Chapter 2, we learned $q_\theta(\tau|g)$ indirectly using *observation channels* M_ψ and $p_o(x > x')$. In this Chapter we choose to directly train a single inverse-action function, $q_\theta(\tau|g)$. Through direct supervision, we can identify sources used in news articles with high recall $Recall(g)$ (even those expressed implicitly in the final article g). We will describe this learning process now.

We approach this by representing a news article g as a set of sentences, $g = \{X_1, ...X_n\}$ and a set of informational sources $Q = \{q_1, ...q_k\}$. We define an attribution function α that maps each sentence to a subset of sources:[2]

$$\alpha(X_i) \subseteq Q \text{ for } X_i \in g$$

A sentence is *attributable* to a source if there is an *explicit* or *implicit* indication that the facts in it came from that source. A sentence *is not* attributable to any source if the sentence does not convey concrete facts (i.e. it conveys journalist-provided analysis, speculation, or context), or if it cannot be determined where the facts originated.

Computing $\alpha(X_i)$ for each sentence yields a noisy proxy for the latent source-acquisition trajectory τ, and thus informs our estimate of $q_\theta(\tau \mid g)$. Because α captures which sources support which sentences but not *when* those sources were obtained, it does not recover how $\tau = a_1, a_2, ...a_t$ is ordered directly. To induce a weak partial order $\prec$ over sources,

[2]Most sentences are attributed to only one source in the article, but some are attributed to several.

we adopt simple, empirically motivated priors (e.g., earlier mentions are more likely to have been obtained earlier; sources with more attributed sentences are more likely to have been obtained earlier). We revisit and validate these priors in Section 3.2.2 and again in Chapter 5 when introducing the *NewsEdits* dataset. For the remainder of this section, we treat $\{\alpha(X_i)\}_{i=1}^{n}$ as a sufficient surrogate of g for estimating $q_\theta(\tau \mid g)$ and leave explicit ordering to later sections.

3.2.1 Source Attribution Modeling

Sources are people or organizations and are usually explicitly mentioned. They may be named entities (e.g. "Laurent Lamothe," in Table 3.1), or canonical indicators (e..g "commission," "authorities") and they are *not* pronouns. In some cases, a sentence's source is not mentioned in the article but can still be determined if (1) the information can only have come from a small number of commonly-used sources[3] or (2) the information is based on an eye-witness account by the journalist. See Table 3.2 for examples of these latter two categories. In the first two rows, we give examples of sourced information that a knowledgeable journalist could look up quickly. The third row shows a scene that could only have been either directly observed, either in-person or via recording, and thus must be sourced directly to the journalist.

Attributing information to sources is challenging: as shown in Tables 3.1 and 3.2, while some attributions are identified via lexical cues (e.g. "said"), others are deeply implicit (e.g. one would have to know that ordering a "curfew" creates a public record that can be retrieved/verified)[4]. Previous modeling work, we show, has focused on the "easy" cases: identifying attributions via quotes,[5] resulting in high-precision, low recall techniques [276,

[3]Examples in this category include "the stock market," "legislative/executive records," "court filings." Trained journalists can tell with relative accuracy where this information came from.

[4]In one humorous example, a former Governor of New York was well known to call reporters after 5pm, offer useful information (and colorful quotes), and then request to be off-the-record. New York media outlets started referring to information from this governor as information from "an official in Albany". Experienced readers and fellow journalists were usually able to intuit who it was.

[5]By *quote*, we mean information derived from a person or a document – verbatim or paraphrased. *Sourced*

Tourist visits have declined, and the **Hong Kong stock market** has been falling for the past few weeks, but **protesters** called for more action.	Published Work, Price Signal, Statement
Mr. Trump was handed defeats in Pennsylvania, Arizona and Michigan, where a **state judge** in Detroit rejected an unusual Republican attempt to. . .	Lawsuit
Mr. Bannon, former chief strategist for President Trump, was warmly applauded when he addressed the party congress of the anti-immigrant National Front. . .	Direct Observation

Table 3.2: **Example sentences from different articles where sources are implicit**. Attribution is non-obvious and based on lexical cues: in the first two rows, we show sentences where sourcing is implicit but where a trained journalist can deduce the source. In the last row, we show a sourced sentence where the descriptive information could only have come from a direct observation by the journalist. **Bold names** are the source attribution, when it exists. In cases, not shown, where it does not exist, we label "passive voice". Underline indicates the specific information that was sourced. Colored annotations on the right are high-level information channels.

277]. Identifying sources of information in a news article is relevant to many tasks in NLP: misinformation detection [278], argumentation [279] and news discourse [130].

We split Source Attribution into two steps: *detection* (*is* the sentence attributable?) and *identification* (what is that attribution?) because, in early trials, we find that using different models for each step is more effective than modeling both jointly. Prior work in Source Attribution primarily used hand-crafted rules [280], bootstrapping [281] and distance-supervision [277] approaches to attribute sentences. Although such work has shown impressive performance on curated datasets, they typically define a source's informational contribution rather narrowly (i.e. only direct or indirect quotes). So, we test several variations of methods introduced in prior work on our dataset to confirm that these categories are not implicitly attributed. For *detection*, a binary classification task, F1-score is used. For *identification*, we use accuracy, or precision@1.

information is broader and includes actions by the journalist to uncover information: first-person observations, analyses or experiments.

		Direct Quote	Indirect Quote	State-ment	Email/ Social	Pub. Work	Other	Micro Avg.
Detection f1 score	Rules 1	64.7	69.3	81.2	76.2	72.7	37.4	59.1
	Rules 2	71.3	79.8	89.8	82.1	79.2	32.5	68.8
	Quootstrap	85.0	81.3	51.3	58.6	33.1	3.0	33.4
	Sentence	91.0	98.7	94.1	**92.7**	85.4	61.4	87.1
	Full-Doc	**92.0**	**98.7**	**96.4**	**89.8**	**86.4**	**65.1**	**88.2**
Identification Accuracy on gold-labeled sourced sents	Rules 1	47.8	48.4	43.0	51.7	37.8	30.2	46.4
	+*coref*	57.3	54.5	49.8	49.4	38.3	34.9	52.8
	Rules 2	20.7	22.5	30.3	21.3	27.4	30.2	22.5
	+*coref*	31.6	42.0	56.1	30.3	32.3	30.2	36.6
	QuoteBank	9.9	16.0	16.4	17.7	4.3	0.5	5.5
	SeqLabel	37.2	43.4	40.0	31.2	32.3	17.7	38.5
	SpanDetect	61.1	59.5	67.6	44.4	51.6	36.5	59.5
	+*coref*	51.2	56.8	60.6	79.0	54.6	42.6	53.6
	GPT3 ft, Babbage	80.9	86.9	85.0	71.9	57.9	38.3	78.9
	+*coref*	78.7	82.5	76.3	56.1	54.4	31.2	73.2
	GPT3 ft, Curie	**94.0**	**95.5**	**91.1**	**91.0**	**81.6**	57.3	**91.4**
	GPT3 ZS, DaVinci	70.9	58.8	72.5	43.1	54.6	47.6	58.5
	+*coref*	66.9	57.6	61.9	20.2	42.6	51.4	55.4
	GPT3 FS, DaVinci	74.9	56.5	70.1	52.3	49.4	**82.8**	61.6
	+*coref*	70.0	55.6	72.7	50.5	48.8	60.7	58.6
Both Acc all sents	GPT3 ft, Babbage	79.5	82.9	82.9	73.4	60.5	53.0	70.9
	+*Nones*	82.4	84.8	85.9	73.4	61.0	64.5	73.1
	GPT3 ft, Curie	90.4	90.7	89.9	91.1	78.0	**68.9**	80.0
	+*Nones*	**92.3**	**92.9**	**92.9**	**91.0**	**78.2**	68.3	**83.0**

Table 3.3: Modeling results for two steps in *Source Attribution*: *Detection* (i.e. correctly identifying source sentences) and *Identification* (i.e. correctly attributing sentences to sources). *Both* refers to the end-to-end process: first identifying that a sentence is a informed by a source *and then* identifying that source. ZS and FS refer to "Zero Shot" and "Few Shot", respectively. +*coref* refers to performing coreference resolution beforehand, and universally hurts the model. +*None* refers to Identification models trained to assign "None" to sentences without sources, possibly eliminating false positives introduced by Detection. We can attribute sources with accuracy > 80.

Baseline Methods

Rules 1 (R1): Co-Occurrence: We identify sentences where a source entity candidate co-occurs with a speaking verb. For *detection*, any sentence that contains such a co-occurence is considered a detected sentence. For *attribution*, we consider the identity of the source entity. We use a list of 538 speaking verbs from Peperkamp and Berendt [280] along with ones identified during annotation. We extract PERSON Named Entities and noun-phrase signifiers using a lexicon (n=300) (e.g. "authorities", "white house official") extracted from Newell, Margolin, and Ruths [282]'s dataset.

Rules 2 (R2): Governance: Expanding on R1, we parse syntactic dependencies in sentences [283] to introduce additional heuristics. Specifically, we identify sentences where the name is an $nsubj$ dependency to a speaking verb governor. $nsubj$ is a grammatical part-of-speech, and a governor is a higher node in a syntactic parse tree.

Quootstrap: Pavllo, Piccardi, and West [281] created a bootstrapping algorithm to discover lexical patterns indicative of sourcing. Contrasting with previous baselines, which hand-crafted lexical rules, bootstrapping allowed researchers to learn large numbers of highly specific patterns. Although the small size of our dataset compared with theirs prevents us from extracting novel lexical patterns tailored to us, we use a set of 1,000 lexical patterns provided by the authors[6]. Similary to R1 and R2, for *detection*, we consider all sentences that match these 1,000 lexical rules to be "detected" sentences. For *attribution*, we examine the entities these rules extract.

QuoteBank: In Vaucher et al. [277], authors train a BERT-based entity-extraction model on distantly-supervised data [281]. This method is less lexically focused, and thus more generalizable. They use their model to score and release a large corpus of documents. We examine this corpus and select articles that are both in their corpus and in our annotation set, finding 139 articles, and limit our evaluation to these.[7] For *detection*, we examine all

[6] `https://github.com/epfl-dlab/Quotebank/blob/main/quootstrap/resources/seedPatterns.txt`

[7] We also discard articles where *QuoteBank* reported quotations or context that are not found in our articles, because our corpus was created from *NewsEdits*, so it's possible that the version of the articles that we examined were different from theirs.

sentences with attribution, and for *identification*, we match the source name with gold-labels.

Detection Methods

Sentence: We adapt a binary sentence classifier where each token in each sentence is embedded using the `BigBird-base` transformer architecture [284]. Tokens are combined via self attention to yield a sentence embedding and again to yield a document embedding. Thus, each sentence is independent of the others.

Full-Doc: We use a similar architecture to the Sentence approach, but instead of embedding tokens in each sentence separately, we embed tokens in the whole document, then split into sentences and combine using self-attention. Thus, the sentences are not embedded independently and are allowed to share information.

Identification Methods

Sequence Labeling: predicts whether each token in a document is a source-token or not. We pass each document through `BigBird-base` to obtain token embeddings and then use a token-level classifier. We experiment with inducing a curriculum by training on shorter-documents first, and freezing layers 0-4 of the architecture.

Span Detection: predicts start and stop tokens of the sentence's source. We use `BigBird-base`, and separate start/stop-token classifiers [285]. We experiment with inducing decaying reward around start/stop positions to reward near-misses, and expand the objective to induce source salience as in Kirstain, Ram, and Levy [286], but find no improvement.

Generation: We formulate identification as open-ended generation and fine-tune GPT3 models to generate source-names. We use with the following prompt: "`<article>To which source can we attribute the sentence <sentence>?`". We need to include the whole article in order to capture cases where a source is mentioned in another sentence. We experiment with fine-tuning Babbage and Curie models, and testing zero- and few-shot for DaVinci models. Because our prompt-query as it contains an entire article/source pair, we have limited additional token-budget; so, for our few-shot setting, we give examples of sentence/source pairs where the source is mentioned in the sentence. For *+coref* variations,

	Gold (Train)	Gold (Test)	Silver
# docs	1032	272	9051
# sent / doc	30	67.5	27
doc len (chars)	3952	7885	3984
# sources / doc	6.8	12.1	8.2
% sents sourced	47.7%	46.9%	57.4%
% sents, most-used source / doc	37.5%	28.1%	31.8%
% sents, least-used source / doc	5.9%	2.4%	6.7%
source entropy	1.6	2.1	1.8
# sources added per version	n/a	n/a	+2
document sent. ↑ likely to be sourced	96th p	92th p	0th p

Table 3.4: Corpus-level statistics for our training, test, and silver-standard datasets. Shown are averages across the entire corpus. Documents in the test set are longer than the training, but the model seems to generalize well to the silver-standard corpus, as statistics match. "% sents, top source" and "% sents, bot source" refer to the % of sourced sentences attributed to the most- and least-used sources in a story. "# sources added / version" shows the number of sources added to articles each news update; it is calculated using the NewsEdits corpus which, as we will see Section 5.2, collects all *versions* of an article and can give us a finer-grained sense of temporality. "sentence most likely to be sourced" refers to the percentile sentence with the highest likelihood of being a sourced sentence.

we evaluate approaches on articles after resolving all coreferences using LingMess [287]. For +*Nones* variations, we additionally train our models to detect when sentences do *not* contain sources. We use this as a further corrective to eliminate false positives introduced during detection.

3.2.1.1 Source Attribution Results

As shown in Table 3.3, we find that the GPT3 Curie source-identification model paired with the Full-Doc detection module in a pipeline performed best, achieving an attribution accuracy of 83%. In the +*None* setting, both GPT3 Babbage and Curie can identify false positives introduced by the detection stage and outperform their counterparts. Overall, we find that resolving coreference does not improve performance, despite similarities between the tasks. The poor performance of both rules-based approaches and QuoteBank,

which also uses heuristics,[8] indicates that simple lexical cues are insufficient. Although QuoteBank authors reported it outperformed similar baselines as we tested [277], we observe low performance from Quotebank [277], even in categories it is trained to detect. GPT3 DaVinci zero-shot and few-shot greatly underperform fine-tuned models in almost all categories (except "Other"). Further, we see very little improvement in the use of a few-shot setup vs. zero-shot. This might be because the examples we give GPT3 are sentence/source pairs, which do not correctly mimic our document-level source-attribution task. We face shortcomings due to the document-level nature of our task: the token-budget required to ask a document-level question severely limits our ability to do effective few-shot document-level prompting. Approaches that condense prompts [288] might be helpful to explore in future work. Further work is necessary to show that our models can transfer well to different newspapers with different sourcing standards.

3.2.2 Insights from Source Analysis

Having built an attribution pipeline that performs reasonably well, we run our best-performing attribution model across 9051 unlabeled documents from *NewsEdits* and extract all sources. In this section, we explain derive insights into how sources are used in news articles. For statistics guiding these insights, see in Table 3.4, which shows statistics calculated on both our annotated dataset ("Gold Train" and "Gold Test" columns) and the 9051 documents we just described ("Silver" column). We ask two primary questions: *how much an article is sourced?* and *when are sources used in the reporting and writing process?*

Insight #1: $\sim 50\%$ of sentences are sourced, and sources are used unevenly. Most articles, we find, attribute roughly half the information in their sentences to sources. This indicates that the percentage of sources used is fairly consistent between longer and shorter documents. So, as a document grows, it adds roughly an equal amount of

[8] Quotebank's algorithm condenses input data to a BERT span-classifier by (1) looking for double-quotes (2) identifying candidate speakers through a lookup table.

sourced and unsourced content (e.g. explanations, analysis, predictions).[9] We also find that sources are used unevenly. The most-used source in each article contributes $\sim 35\%$ of sourced sentences, whereas the least-used source contributes $\sim 5\%$. This shows a hierarchy between major and minor sources used in reporting and suggests future work analyzing the differences between these sources.

Insight #2: Sources begin and end documents, and are added while reporting Next we examine when sources are used in the reporting process. We use the *NewsEdits* dataset, which collects all revisions made to news articles [289] (we will introduce *NewsEdits* more formally in Chapter 5, Section 5.2). We find that articles early in their publication cycle tend to have fewer sources, and add on average two sources per subsequent version, shown in Figure 3.3. This indicates an avenue of future work: understanding which kinds of sources get added in later versions can help us recommend sources as the journalist is writing. Finally, we also find, in terms of narrative structure, that journalists tend to lead their stories with sourced information: the most likely position for a source is the first sentence, the least likely position is the second. The second-most likely position is the end of the document.[10][11]

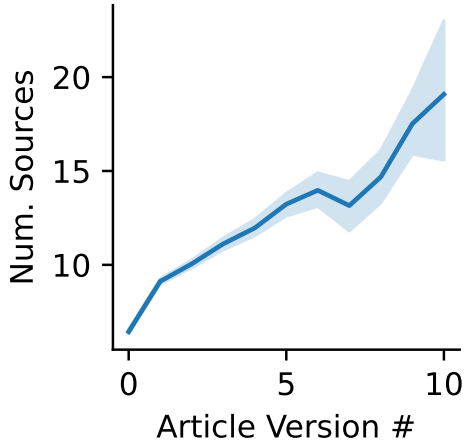


Figure 3.3: Do more sources get added to an article over time? We show the number of sources in an article as it gets republished, based on *NewsEdits* (Section 5.2) and find that as news unfolds, sources get added.

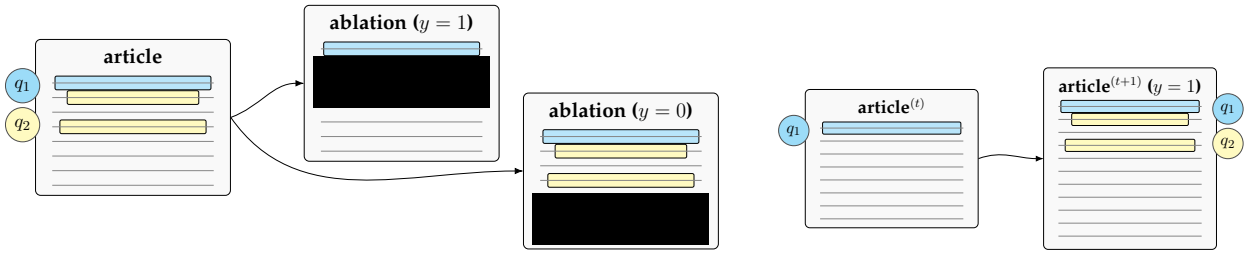
[9] The only exception, we find, is very short documents (<200 words). Manual inspection of these documents shows that they are usually breaking news alerts and take all their information from a single source.

[10] The sources might be used for different purposes: Spangher et al. [26] performed an analysis on news articles' narrative structure, and found that sentences conveying the *Main Idea* lead the article while sentences conveying *Evaluations* or *Predictions*.

[11] *A caveat to Table 3.4*: many gold-labeled documents were parsed so the first sentence got split over several sentences, which is why we observe the last sentences having highest sourcing, for example: `sents=['BAGHDAD', '–', 'Yesterday, the American military said']`. See [202].

		Other News	Disaster	Elections	Labor	Safety
Top Ablated	FastText	66.1	65.8	69.8	68.8	68.0
	+*Source-Attribution*	66.0	64.5	69.8	68.2	68.0
	BigBird	74.2	68.4	78.3	**74.0**	78.1
	+*Source-Attribution*	73.9	69.7	74.9	73.4	73.4
	GPT3 ft, Babbage	**78.3**	**75.5**	**81.5**	72.7	**80.0**
	+*Source-Attribution*	74.9	69.5	78.0	70.9	65.1
Second Source	FastText	57.6	63.2	60.8	61.0	63.3
	+*Source-Attribution*	57.8	63.2	61.1	62.3	64.1
	BigBird	63.8	61.8	63.1	64.3	61.7
	+*Source-Attribution*	65.1	**69.7**	65.7	64.9	62.5
	GPT3 ft, Babbage	**67.1**	67.9	**72.9**	58.8	65.6
	+*Source-Attribution*	65.4	65.1	68.0	**65.9**	**66.7**
Any Source	FastText	54.5	60.5	57.1	57.8	56.2
	+*Source-Attribution*	54.8	**59.2**	57.6	56.5	56.2
	BigBird	57.5	53.9	55.5	55.8	**57.8**
	+*Source-Attribution*	**59.4**	55.3	60.6	60.4	56.2
	GPT3 ft, Babbage	55.0	53.9	**63.6**	**63.4**	49.0
	+*Source-Attribution*	59.0	56.1	61.3	39.3	51.7
News Edits	FastText	58.1	48.9	62.1	58.6	48.8
	+*Source-Attribution*	56.8	55.8	61.9	61.2	49.6
	BigBird	63.5	63.9	64.5	**64.8**	64.8
	+*Source-Attribution*	**69.4**	**65.3**	62.6	60.4	**64.2**
	GPT3 ft, Babbage	65.0	63.9	**64.6**	62.4	51.0
	+*Source-Attribution*	64.0	56.1	61.3	39.3	51.7

Table 3.5: Results for *Source Prediction*, broken into four canonical news topics and 'other.' "Top Ablated" is our prediction task run on articles ablated by removing the source that has the most sentences, "Second Source" is where a source contributing more than 10% of sentences is removed, and "Any Source" is where any source is randomly removed. The *NewsEdits* task is to predict whether the article at time t will be added sources at time $t+1$. In the +*Source-Attribution* experiments, we add sourcing information, derived in Section 3.2.1, to the input (see Section 3.2.3.2). For all of these tasks, our models were able to significantly outperform random (50% acc.). In general, our expectations are confirmed that: (a) harder tasks yield lower-accuracy results and (b) more powerful models improve performance. This indicates that there is a pattern how sources are used in news writing.



(a) **Ablation probe for source predictability.** Given an original article X with sources q_1, q_2, we construct ablated documents X' by sampling with equal probability: a no-op ablation $X' = X \setminus \{\varnothing\}$, $y = 0$; and a source-removal ablation $X' = X \setminus \{X_i\}$ s.t. $\alpha(X_i)=q_1$, $y = 1$.

(b) ***NewsEdits* probe** Given an article $X^{(t)}$ with source q_1 and an updated version $X^{(t+1)}$, we check whether another source q_2 was added. If so, $y = 1$ (shown); otherwise $y = 0$.

Figure 3.4: ***Source-Predictability Probes***: We construct two supervised probes to test for compositeness in τ. The goal in both probes is to train a binary classifier f to detect either whether (1) a source is missing from X' or (2) a source will be added to $X^{(t+1)}$. Above, circles q_1, q_2 denote attributed sources. We evaluate using $\mathrm{F1}(f)$: under null-hypothesis H_0 (no coupling), $\mathrm{F1}(f) = 0.5$; evidence of predictability corresponds to $\mathrm{F1}(f) > 0.5$ under H_1.

3.2.3 Source Compositionality

Having established that we can learn a well-performing attribution function $\alpha(X_i)$ by annotating a large dataset and training state-of-the-art models, we have used $\alpha(X_i)$ to identify a broad range of sources used in news articles. Before we move to learning policy functions $\pi(\tau|g)$, in *emulation* in Sections 3.3 and 3.4, we wish to test that this is even possible; that source finding is *compositional* and *predictable*. If sources are used together predictably, then we have a hope, during *policy learning* later, of learning how to model them sequentially.

3.2.3.1 *Source Prediction*: Problem Definition

To test compositeness, we introduce a probing task, *source-prediction*. *Source-prediction* helps us probe whether the likelihood of predicting the correct source *increases* when we gain knowledge of the other sources used in the article. In other words:

$$p(q_i|q_j) > p(q_i)$$

We define two probes to frame the *source-prediction* task, both shown in Figure 3.4b:

1. *Ablation*: Given an article (X, Q) with attribution $\alpha(X_i)\ \forall\ X_i \in X$, choose one source $q_i \in Q$. To generate positive examples ($y = 1$), we remove all sentences $X \setminus X_i$ where $q_i \in \alpha(X_i)$. Then, to generate negative examples ($y = 0$), we remove an equal number of sentences where $\alpha(s) = \{\}$ (i.e. no source). This is shown in Figure 3.4a.

2. *NewsEdits*: We sample article-versions from *NewsEdits*, a corpus of news articles, with all of their updates across time (introduced in Section 5.2). We identify articles where: at time t, article$^{(t)}$, has sources $q_1, ...q_t$ and the update article$^{(t+1)}$ adds a source, q_{t+1}. article$^{(t)}$ is labeled $y = 1$. If no source is added to article$^{(t+1)}$, then article$^{(t)}$ is labeled $y = 0$.

The goal, for each probe, is to then train a binary classifier f to predict the assigned labels. The strength of the classifier tells us, then, how predictable this task is. Our null hypothesis H_0 holds that there is no predictability between sources, so the performance of the classifier f under H_0 is $\mathrm{F1}(f) = 0.5$. If we observe $\mathrm{F1}(f) > 0.5$, then we reject H_0 and accept H_1, that there is predictability among the sources. Each probe tests source usage in different ways. *Ablation* assumes that the composition of sources in an article is cohesively balanced, and induces reasoning about this balance. *NewsEdits* relaxes this assumption and probes if this composition might change, either due to the article's completeness, changing world events that necessitate new sources, or some other factor.[12]

3.2.3.2 Dataset Construction and Modeling

We use our *Source Attribution* methods discussed in Section 3.2.1 to create large silver-standard datasets in the following manner for our two primary experimental variants: *Ablation* and *NewsEdits*. To interpret results in each variant better, we train a classifier to

[12]Spangher et al. [289] found that many news updates were factual and tied to event changes, indicating a breaking news cycle.

categorize articles into four topics plus one "other" topic[13], based on articles in the *New York Times Annotated Corpus* [291] with keyword sets corresponding to each topic.

Ablation We take 9051 silver-standard documents and design three variations of this task. As shown in Table 3.4, articles tend to use sources lopsidedly: one source is usually primary. Thus, we design Easy (Top Ablated, in Table 3.5), Medium (Second) and Hard (Any Source) variations of our task. For Easy, we ablate from articles the source with the *most* sentences attributed to it (i.e. $n(q) := |\{ X_i \in X : \alpha(X_i) = q \}|, \;\; q^{\text{top}} := \arg\max_{q \in Q} n(q)$). For Medium, we randomly choose among the top three sources with the most sentences attributed to them (i.e. $(q^{(1)}, q^{(2)}, q^{(3)}) := \left[\text{argsort}_{q \in Q}(-n(q)) \right]_1^3$). And for Hard, we randomly choose any of the sources to perform ablations. Again, once we choose the source q, we generate *two* ablated documents per article by: (1) ($y = 1$): removing all sentences attributed to q: $X^{(1)}(q) := X \setminus \{ X_i \in X : \alpha(X_i) = q \}$. And (2) ($y = 0$) removing an *equal* number of sentences from the document that are not attributed to any sources: $X^{(0)}(q) := X \setminus S$, where $S \subseteq \{ X_i \in X : \alpha(X_i) = \varnothing \}$ with $|S| = |\{ X_i : \alpha(X_i) = q \}|$.

NewsEdits We sample an additional $40,000$ articles from the *NewsEdits* corpora and perform *attribution* on them. We sample versions pairs that have roughly the same number of added, deleted and edited sentences in between versions in order to reduce possible confounders — as we will see in Section 5.2, these edit-operations were predictable. We identify article-version pairs where 1 or more sources were added between version article$^{(t)}$ and article$^{(t+1)}$ and label article$^{(t)}$ these with $y = 1$. If 0 or 1 sources added to article$^{(t+1)}$, then we label article$^{(t)}$ with $y = 0$.

Modeling We use three models: (1) FastText [292] for sentence classification, (2) A BigBird-based model: we use BigBird with self-attention for document classification, similar to Spangher et al. [289].[14] Finally, (3) we fine-tune GPT3 Babbage to perform prompt-

[13]These four have been identified as especially socially valuable topics, or "beats," due to their impact on government responsiveness [290]

[14]Concretely, we obtain token embeddings of the entire document, which we combine for each sentence using self-attention. We contextualize each sentence embedding using a shallow transformer architecture.

completion for binary classification. For each model, we test two setups. First, we train on the vanilla text of the document. Then, in the *+Source-Attribution* variants, we train by appending each sentence's source *attribution* to the end of it.[15] The source annotations are obtained from our attribution pipeline.

3.2.3.3 Results and Discussion

The results in Table 3.5 show that we are broadly able to predict when major sources (Top, Secondary) are removed from articles, indicating that there is indeed *compositionality*, or intention, in the way sources are chosen to appear together in news articles. The primary source (Top)'s absence is the easiest to detect, indicating that many stories revolve around a single source that adds crucial information. Secondary sources (Second) are still predictable, showing that they serve an important role. Minor sources (Any)'s absence are the hardest to predict and the least crucial to a story. Finally, source-addition across article versions (see Section 5.2 for more details about this dataset) is the hardest to detect, indicating that versions contain balanced compositions.

Overall, we find that our experiments are statistically significant from random (50% accuracy) with t-test $p < .01$, potentially allowing us to reject the null hypothesis that positive documents are indistinguishable from negative in both settings. Evidence of structure is directly actionable for policy learning over *sets*: when item utilities exhibit complementarities and diversity pressures, set-aware objectives (e.g., submodular maximization or DPP-based selection) provide faithful inductive biases and even greedy-approximation guarantees [293, 294, 295]. Moreover, if source use unfolds in stereotyped routines across article versions (e.g., "establish claim" $\rightarrow$ "countervoice" $\rightarrow$ "context"), that is the hallmark of reusable *options* or temporally abstract skills, for which hierarchical policies are well-motivated [296].

We finally combine these sentence embeddings using another self-attention layer to obtain a document embedding for classification. We utilize curriculum learning based on document length, a linear loss-decay schedule.

[15] Like so: `<sent 1>. SOURCE: <source 1>. <sent 2> SOURCE: <source 2>... <sent n> SOURCE: <source n>.`

In short, demonstrating predictability and compositeness supports the choice of structured policy classes for selecting sources jointly.

Statistical significance does not preclude confounding, and both the *Ablation* and the *NewsEdits* setups contain possible confounders. In the *Ablation* set up, we might be inadvertently learning stylistic differences rather than source-based differences. To reduce this risk, we investigate several factors. First, we consider whether lexical confounders, such as speaking verbs, might be artificially removed in the ablated documents. We use lexicons defined in our rules-based methods to measure the number of speaking verbs in our dataset. We find a mean of $n = [34, 32]$ speaking verbs per document in $y = [0, 1]$ classes in the Top case, $n = [35, 34]$ in the Medium, and $n = [35, 37]$ in Hard. None of these differences are statistically significant. We also do not find statistically significant differences between counts of named entities or source signifiers (defined in Section 4). Finally, we create secondary test sets where $y = 0$ is *non-ablated* documents. This changes the nature of the stylistic differences between $y = 1$ and $y = 0$ while not affecting sourcing differences[16]. We rerun trials in the *Top* grouping, as this would show us the greatest confounding effect, and find that the accuracy of our classifiers differs by within -/+3 points. In the *NewsEdits* setup, we take care to balance our dataset along axes where prior work have found predictability. As we will show in Section 5.2, edit-operations[17] could be predicted. So, we balance for length, version number and edit operations.

Having attempted to address confounding in various ways in both experiments, we take them together to indicate that, despite each probing different questions around sourcing, there are patterns to the way sources are during the journalistic reporting process. To illustrate, we find in Table 3.5 that Election coverage is the most easily predictable across all tasks. This might be because of efforts to include both left-wing and right-wing voices. It also might be because the cast of characters (e.g. campaign strategists, volunteers, voters)

[16]We do not want to *train* on such datasets, because there are statistically significant length differences and other stylistic concerns ablated and non-ablated articles.

[17]E.g. Whether a sentence would be added in a subsequent version.

stays relatively consistent across stories. Two additional findings are that (1) the tasks we expect are harder do yield lower accuracies and, (2) larger GPT3-based language models generally perform better. Although not especially surprising, it further confirms our intuitions about what these tasks are probing. We were surprised to find that, in general, adding additional information in both stages of this project, whether coreference in the *attribution* stage or source information in the *prediction* stage, did not improve the models' performance. (In contrast, adding source information to smaller language model, `BigBird`, helped with harder tasks like the Medium, Hard and *NewsEdits*). We had hypothesized that the signal introduced by this labeling would not harm the GPT3-based models, but this was not the case. It could be that the larger models are already incorporating a notion of coreference and attribution, and adding this information changed English grammar in a way that harmed performance.

Why this matters for emulation and offline learning from human data Learning policies from observational human data is famously sensitive to ambiguity and support mismatch. Inverse RL highlights that many reward/process explanations can match the same artifacts, making the inverse problem non-unique [162]. Offline RL further warns that distributional shift and unobserved confounding can render policy evaluation/learning ill-posed without additional structure or assumptions [297, 298, 299]. Our probes act as *pre-tests for recoverability*: if we can reliably tell when a major source is missing or predict a soon-to-be-added source, then the observational record carries signal strong enough to constrain the hypothesis space in practice. Conversely, if predictability were at chance, that would be a red flag to augment the state with richer observables or to add interactive data collection (e.g., DAgger-style interventions) before training policies [300]. Finally, the "what source comes next?" framing parallels mature citation-recommendation settings where future or missing references are predictably inferred from context and existing citations [301, 302, 303], providing additional external evidence that this supervision signal is learnable from text and partial source sets.

3.3 Does Pretraining Implicitly Learn $\pi(\tau|x)$ for *Source-Finding*?

In the previous section, we trained a sentence-level attribution function, $\alpha(X_i)$ using our labeled dataset, which we then use to create an inverse function, $q_\theta(\tau|g)$. We then showed that inferred human trajectories τ had compositeness and predictability. Before we *learn* a policy function $\hat{\pi}(\tau|g)$, we wish to *test* whether existing, emergent policy functions are *good enough*, which we call $\pi^{(llm)}(\tau|x)$. Indeed, pretrained LLMs are being used already for these tasks; Petridis et al. [304], for instance, explored how well LLMs could suggest sources and unique angles to cover *press releases*. As discussed in Section 1.1.1, important questions remain about whether pretraining is *enough* to learn implicit human policies. How often do these implicity-learned policies align with human values? Furthermore, if we are to learn policy functions $\pi(\tau|x)$ to support complex, creative tasks in journalism, how will we assess their performance? How can we adjust such decision-making to ensure better alignment?

We seek, in this section, to build upon the previous section and demonstrate how a benchmark can be made for more broadly developing AI approaches for aiding creative tasks, ensuring they align with human values. In this section, we will use the terms *policy*, $\pi(\tau|x)$ and *planning* relatively interchangeably; indeed we will sometimes refer to the task of learning policies for *complex, creative tasks* as *creative planning*. Classically, *policy learning* learns a mapping $\pi(a \mid x, g)$ (or $\pi(\tau \mid x, g)$) that selects actions without explicit test-time search, whereas *planning* uses a model to reason over future consequences (e.g., via lookahead or search) before acting. In our setting, a story is developed as an *open-loop* sequence of source choices τ that is evaluated primarily via the final artifact g. Under this evaluation, any planner over trajectories induces a distribution $\pi(\tau \mid x, g)$, and sampling a trajectory from a learned policy constitutes a plan. Because (i) we compare distributions over τ conditioned on (x, g), (ii) the environment is effectively static within a single story

cycle, and (iii) our metrics depend on the end-state g rather than mid-trajectory feedback, we treat 'planning" and "policy learning" interchangeably without loss of specificity.

To build our benchmark, we introduce a novel, broad dataset and compare the planning decisions LLMs *would make* to the decisions humans *have made* in the past. Our work represents a generalizable[18] benchmark in creative planning tasks and can serve as a template for creative planning evaluation going forward. We start by assembling a corpus of press releases and news articles covering them, and identify articles that have *effectively covered* these releases. Like city council meetings, explored earlier, press releases are an ideal domain to explore, as they form a routine set of coverage goals pursued regularly by journalists – and, as companies often lie, exaggerate and mislead in their press releases, journalists are tasked with holding them to account with effective coverage. According to Maat and Jong [305], effective coverage substantially challenges and contextualizes press releases. We seek to focus on this subset as a basis for our benchmark, as this is likely a set of sources that are utilized well to contextualize narratives. We begin by describing our dataset collection first.

3.3.1 Press Release Dataset

Press releases offer an ideal window into the journalistic process. Press releases contain potentially valuable information, but are often "spun" by their authors to portray events positively [306]. "De-spinning" them involves challenging and contextualizing claims [305] and often requires substantial work prior to writingHere, I describe how we construct *PressRelease*, a large corpus of 650k news articles hyperlinking to 250k press releases. *PressRelease* contains data collected via two approaches in order to avoid biases with either.

3.3.1.0.1 Press Releases ← News Outlets, Hyperlinks: The first way we discover news articles linking to press releases is to collect HTML of news articles, and find hyperlinks

[18]Most prior work in this vein has limited generalizability due to small sample sizes – e.g., Petridis et al. [304] tested two articles with 12 participants.

to known press release domains in these articles. We query Common Crawl for all URLs from 9 major financial newspapersin all scrapes since 2021, resulting in 114 million URLs.

From these URLs, we discover 940,000 URLs of news articles, specifically, using a supervised model by Welsh [307] to differentiate news article URLs from other pages on news websites (e.g. login pages). Then, we find hyperlinks to press releases in these news articles by finding all links to known press release websites.[19] This yields 247,372 articles covering 117,531 press releases. We retrieve the most recent version of the press release page published before the news article from the Wayback Machine.[20] We note that this approach is biased in several ways. Firstly, we only capture the coverage decisions of the 9 major financial newspapers. Secondly, our technique to find hyperlinks to press releases, via keyword filters, introduces noise. Thirdly, we are more likely to discover popular press releases and less likely to discover ones that received less coverage. To address these biases, we retrieve data in the opposite direction as well.

3.3.1.0.2 Press releases → News Articles, Backlinks: Another way to find news articles linking to press releases is to collect press releases and discover pages *hyperlinking to them* using a backlinking service.[21] First, we compile the subdomains of press release offices for all 500 companies in the S&P 500, other organizations of interest (e.g. OpenAI, SpaceX and Theranos) and specific, notable press releases.[22] We query our backlinking service for webpages linking to each of these subdomains. We again use Welsh [307]'s model to identify backlinks to news articles. We retrieve 587,464 news articles and 176,777 press releases from the Wayback Machine. This approach, like the last, is also biased. Despite now discovering news articles from a far wider array of news outlets, we now

[19] URLs containing the following phrases: `'prnewswire'`, `'businesswire'`, `'press'`, `'release'`, `'globenewswire'`, `'news'`, `'earnings'`, `'call-transcript'` OR those with the following anchor text: `'press release'`, `'news release'`, `'announce'`, `'earnings call'`.

[20] The Wayback Machine, `https://archive.org/web/` [308], is a service that collects timestamped snapshots of webpages, allowing users to retrieve past webpages.

[21] We use Moz, `https://moz.com/`.

[22] Including: Apple IPhone releases, OpenAI's GPT2 and ChatGPT release notes, Facebook's response to the Cambridge Analytica Scandal, Equifax's response to their 2016 data breach and other major corporate events, including corporate scandals listed here: `https://www.business.com/public-relations/business-lies/`

Press Release Text	**Article Text**
(*Theranos*) Theranos will close our clinical labs, impacting approximately 340 employees. We are profoundly grateful to these teammates...	(*Mashable*) Few tears shed for E. Holmes as Theranos bleeds jobs. Theranos shot to fame in 2014. Then came an investigation from WSJ...
(*Tesla*) There is a false allegation that Tesla terminated employees in response to a new union campaign. These are the facts behind the event: Tesla conducts performance review cycles every six months... Underperforming employees are let go.	(*WKWB*) Employees said [they're] tracked down to the key stroke. "If you even go to the bathroom, you won't hit your time goal..." (*CNBC*) ...After hours on Thursday, Tesla called [retaliation] allegations false, saying [workers] had been terminated due to poor performance.
(Goldman Sachs) We found reducing the earnings gap for Black women will create 1.2-1.7M U.S. jobs and increase GDP by $300-450B.	(*BE*) Studies have found Black women's contributions to the U.S. economy as consumers, entrepreneurs, and employees play a key factor...

Table 3.6: Examples of press releases (left) and news articles that cover them in our corpus, *PressReleases*. Our corpus contains 656,000 news articles covering 250,000 press releases. Each news article introduces an angle (i.e., specific focus) and uses sources (i.e., a person or document contributing information) to support this angle. Approximately 70,000 press releases, or 28% of our corpus, are covered more than once (as the *Tesla* example shows). This indicates a rich corpus for ongoing research in narrative approaches.

overrepresent press releases from the top companies; we also miss press releases that are not directly posted on their company websites. The combination of these two methods of data collection is intended to reduce popularity biases any one direction imposes. To further clean our dataset, we exclude press release/article pairs where the press release link is in the bottom 50% of the article, and we exclude pairs that are published far apart chronologically (>1 month difference.)[23] These heuristics seek to exclude news articles where the press release is not the main topic.[24].

[23]We query the Wayback Machine to find the earliest collection timestamps of documents.
[24]We discuss additional processing steps in [273]

3.3.1.1 Dataset Details

We are left with a total of 656,523 news articles and 250,224 press releases from both directions. Examples of press releases and news articles matched in our dataset are shown in Table 3.6. As can be seen, news articles directly comment on the press releases they cover, often offering neutral or critical angles (i.e., specific areas of focus) and drawing information from sources (i.e., people or documents contributing information). 70,062 press releases, or 28% of our dataset, are covered by more than one news article (a total of 509,820 articles). This presents a rich corpus of multiply-covered stories: while in the present section, we do not utilize this direction, it opens the door for future work analyzing different coverage decisions.

3.3.2 Press Release Coverage as Contrastive Summarization

In order to narrow our benchmark to a targeted set of articles that require careful planning, we seek to identify when a news article *effectively covers* a press release [305]. These are articles, we reason, where decision-making was the most thoughtful: journalists are more careful and thoughtful with their actions, we assume, when they are criticizing a press release than simply paraphrasing or summarizing. Identifying effective coverage is not trivial: many articles uncritically summarize press releases or use them peripherally in larger narratives. We examine pairs of news articles and press releases, answering the following two questions: (1) Is this news article *substantially about* this press release? (2) Does this news article challenge the information in the press release? While many articles discuss press releases, most of them simply repeat information from the release without offering insights. After examining hundreds of examples, we devise novel framework, *contrastive summarization*, to describe "effective coverage". A piece of text is a *contrastive summary* if it not only conveys the information in a source document, but contextualizes and challenges it.

Can we automatically detect when a piece of text is a contrastive summary? To do so, we represent each press release and news article as sequences of sentences, $\vec{P} = p_1, ...p_n$, $\vec{N} = n_1, ...n_m$, respectively. We establish the following two criteria:

1. **Criteria # 1**: $\vec{N}$ *contextualizes* $\vec{P}$ if:

   $\sum_{j=1,...n} P(\text{references}|\vec{N}, p_j) > \lambda_1$.

2. **Criteria # 2**: $\vec{N}$ *challenges* $\vec{P}$ if:

   $\sum_{j=1,...,n} P(\text{contradicts}|\vec{N}, p_j) > \lambda_2$.

We define "references" (or "contradicts") as 1 if *any* sentence in $\vec{N}$ references (or contradicts) p_j, 0 otherwise. Viewed in an NLI framework [309], "contradicts" is as defined in NLI, and "references" = ["entails" $\vee$ "contradicts"]. We expect this approach can get us close to our goal of discovering press releases that are substantially *covered and challenged* by news articles. A press release is substantially *covered* if enough of its information is factually consistent or contradicted by the news article. It's substantially *challenged* if enough of its sentences are contradicted by the news article. Laban et al. [310] found that aggregating sentence-level NLI relations to the document-level improved factual consistency estimation. We take a nearly identical approach to the one shown in their work.[25] First, we calculate sentence-level NLI relations, $p(y|p_i, n_j)$, between all $\vec{P} \times \vec{N}$ sentence pairs. Then, we average the top-k_{inner} relations for each p_i, generating a p_i-level score. Finally, we average the top-k_{outer} p_i-level scores. k_{inner} is the number of times each press release sentence should be referenced before it is "covered", and k_{outer} is the number of sentences that need to by "covered" to consider the entire press release to be substantially covered. Using NLI to identify press release/news article coverage pairs provides a computationally cheap and scalable method.

[25]The only difference being that we also consider the contradiction relation, whereas they only consider entailment.

Q1: Does article *cover* press release?	
LogReg/MLP/Hist	72.1 / 72.9 / 79.0
+*coref*	74.6 / 75.2 / **80.5**
Q2: Does article *challenge* press release?	
LogReg/MLP/Hist.	60.3 / 62.9 / 69.4
+*coref*	61.2 / 62.4 / **73.0**

Table 3.7: F1-scores for our classifiers, based on document-level NLI scores, to *capture critical coverage* in news articles covering press releases. We label press releases and news articles for whether they cover and challenge the press release. $+coref$ resolution is found to increase performance in both categories.

3.3.2.1 Detecting Contrastive Summaries

To train a model to detect when a news article *contrastively summarizes* a press release, we annotate 1,100 pairs of articles and press releases with the two questions posed at the beginning of this section. Our annotations are done by two PhD students, where the first annotated all documents and The second doubly-annotated 50 articles, from which an agreement $\kappa > 0.8$ is calculated. We divide these documents into a 80/10/10% train/val/test split. We test the variations: We test resolving coreferences in each document, (+*coref*).[26] Coreference resolution can generate sharper predictions by incorporating more context into a sentence [1]. We also try three different classifiers: Logistic Regression (**LogReg**), a multilevel perceptron with l levels (**MLP**), and a binned-MLP (**Hist**), introduced in Laban et al. [310].

Table 3.7 shows how well we can detect *contrastive summarization* in press release-article pairs. We find that **Hist+*coref*** performed best, with 73.0 F1. Laban et al. [310] noted that the histogram approach likely reduces the effect of outlier NLI scores. See [273] for more experiments. Following this, we apply **Hist+*coref*** to our entire *PressRelease* corpus, obtaining Doc-Level NLI scores for all pairs of articles and press releases in *PressRelease*. In the next section, we describe three primary insights we gain from analyzing these scores.

[26] Using `LingMess` [311]

	Corr. w # Sources / Doc
Contradiction	0.50
Entailment	0.29
Neutral	-0.50

Table 3.8: Correlation between doc-level NLI labels and the # sources in the article. Sources extracted via Spangher et al. [1]'s source-attribution pipeline.

	Corr with Creativity	
	Angle	Source
Contradiction	0.29	0.10
Entailment	0.27	0.03
Neutral	-0.07	-0.11

Table 3.9: Correlation between doc-level NLI labels and the creativity of planning steps journalists took (see Section 3.3.3.2 for more information about creativity measurement).

Each insight sheds more light into how journalists cover press releases.

3.3.2.2 Analysis of Press Releases and News Articles

We frame three insights to explain more about what *effective coverage* entails. These insights lay the groundwork for our benchmark to assess implicit policy functions $\pi^{(llm)}(\tau|x)$ learned during pretraining, discussed in the next section.

	Corr. w Contra.
Person-derived Quotes	0.38
Published Work/Press Report	0.30
Email/Social Media Post	0.25
Statement/Public Speech	0.25
Proposal/Order/Law	0.25
Court Proceeding	0.18

Table 3.10: Correlation between the level of contradiction between a news article and press release and the types of sources used in the news article. Types defined by [1].

Insight #1: Effective news coverage incorporates both contextualization and challenging statements. Our first insight is that NLI-based classifiers can be useful for the task of *identifying effective coverage.* This is not entirely obvious: NLI classification is noisy [312] and contradiction relations might exist not only in directly opposing statements, but in ones that are orthogonal or slightly off-topic [313]. However, our strong results on a large annotated dataset – our annotators were instructed to determine whether a news article effectively covers a press release – indicate that this method is effective. Our performance results, between 70-80 F1-score, are within range of Laban et al. [310] (66.4-89.5 F1 across 6 benchmarks), who first used NLI to evaluate *vanilla summaries*. That a similar methodology can work for both tasks emphasizes the relatedness of the two: identifying effective coverage *is a version of* identifying a summary. Thus, we call our task *contrastive summarization*, to describe the task of condensing and challenging information in a document.

Insight #2: Articles that contradict and entail press releases (1) take more creative angles and (2) use more sources. We first noticed that articles with more creative angles[27] contradict and entail press releases more, as shown in Table 3.9. In order to further explore these kinds of articles, we analyze the sources they used. Spangher et al. [1] developed methods to identify informational sources mentioned in news articles. We utilize this work to identify sources in our corpus: as shown in Table 3.6, examples of sources we identify include a "union", an "employee" or a "study". We find that most news articles in our corpus use between 2 to 7 different sources, corresponding to Spangher et al. [1]'s findings. Next, we correlate the number of sources in an article to the degree to which it contradicts or entails a press release. Interestingly, news articles that contradict press releases *more* also use *more* sources.[28] Table 3.8 shows a strong correlation of $r = .5$ between document-level contradiction and # sources. Articles in the top quartile of contradiction scores (i.e., $> .78$) using a median of 9 sources, while articles in the bottom quartile use 3.

[27] Our methods for measuring creativity is defined further in Section 3.3.3.2.

[28] Doc-Level scores are calculated using *+coref* articles according to k_{inner} and k_{outer} thresholds from the last line in Table 3.7. See [273].

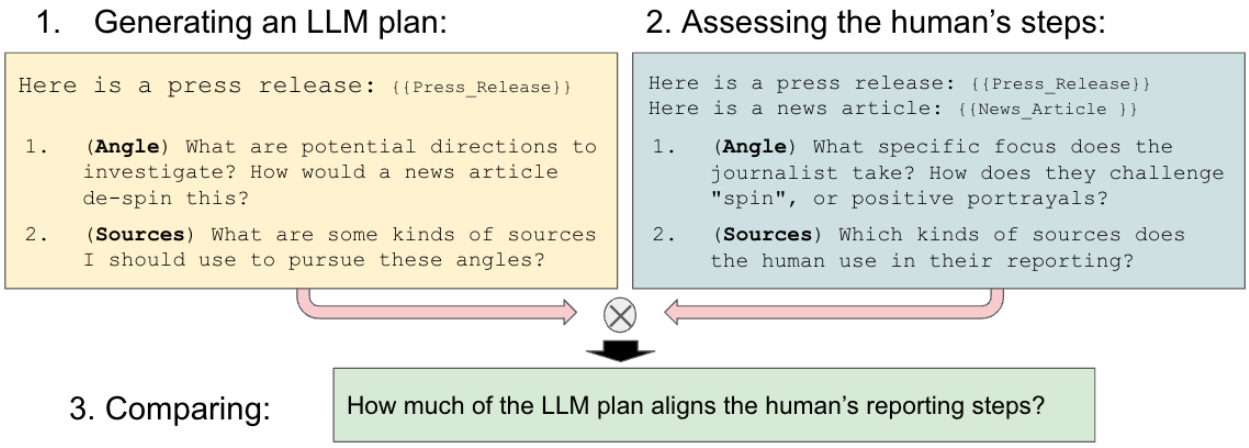


Figure 3.5: **Probing LLM's Planning Abilities:** To assess how well LLMs might assist in the planning stages of article-writing, we attempt to compare the plans suggested by an LLM with the steps human journalists *actually* took during reporting. We infer these steps from the final article. In (1) "Generating an LLM plan", the LLM is asked to suggest angles and sources to pursue. In (2) "Assessing the human's steps", we infer the steps the human took while writing the article by analyzing completed articles using LLMs. Finally, in (3) "Comparing", we compare how much of the LLM's plan aligns with τ taken by humans.

Insight #3: News articles that contradict press releases more use more resource-intensive sources. Of the kinds of sources used in news articles, the majority are either Quotes, 40%, (i.e., information derived directly from people the reporter spoke to), or Press Reports, 23% (i.e., information from other news articles). We obtain these labels by scoring our documents using models trained and described by Spangher et al. [314]. As shown in Table 3.10, the use of Quotes, or person-derived information, is correlated more with Contradictory articles. Quotes are typically more resource-intensive to obtain than information derived from other news articles. A reporter usually obtains quotes through personal conversations with sources [315]; this is a longer process than simply deriving information from other news articles [316]. Additionally, in terms of the *distribution* of sources used in each article, Court Proceedings and Proposal/Order/Laws are overrepresented in Contradictory articles: they are 124% and 112% more likely to be used than in the average article. In general, these kinds of sources require journalistic expertise to assess and integrate [317], and might offer more interesting angles.

Take-away: Taken together, our three insights suggest that any approach to assisting journalists in covering press releases must have an emphasis on (1) suggesting directions for contrastive summaries and (2) incorporating numerous sources. We take these insights forward into the next section, where we assess the abilities of LLMs to assist journalists.

3.3.3 LLM-Based Creative Planning

Based on the insights in the previous section, we now study how LLMs might assist journalists. Specifically, we ask: *How well can an LLM (1) provide a starting-point, or an "angle", for a contrastive summary and (2) How well can an LLM suggest useful kinds of sources to utilize?* Petridis et al. [304] explored how LLMs can aid press release coverage. The authors used GPT-3.5 to identify potential controversies, identify areas to investigate, and ideate potential negative outcomes. They showed that LLMs serve as useful creative tools for journalists, reducing the cognitive load of consuming press releases. While promising, their sample was small: they tested 2 press releases and collected feedback from 12 journalists.

With our dataset, *PressReleases*, we are able to conduct a more comprehensive experiment to benchmark LLMs planning abilities. In this section, we identify 300 critical news articles and the press releases they cover. We compare plans generated by LLMs with the plans pursued by human journalists: such an approach, along with recent work [318], is part of an emerging template for comparing LLM creativity with human creativity and studying how LLMs might be used in human-in-the-loop creative pipelines.

3.3.3.1 Experimental Design

We sample 300 press releases and articles scoring in the top 10% of contrastive summarization scores (identified by **Hist.+*coref*** in the previous section). We manually verify each to be true example of *effective coverage*. By implication, these are press releases that contained ample material for human journalists to criticize. We use these to explore the critical directions LLMs will take. Figure 3.5 shows our overall process. In the first step, **(1)**

		Angle			Source		
		Prec	Recall	F1	Prec	Recall	F1
zero-shot	mixtral-8x7b	35.1	24.5	28.1	15.7	16.3	14.7
	command-r-35b	57.2	61.4	57.0	28.5	26.2	25.1
	gpt3.5	56.3	54.0	52.7	23.8	15.5	17.8
	gpt4	53.6	63.4	56.3	23.2	21.5	21.2
few-shot	mixtral-8x7b	40.8	28.9	31.8	17.3	13.3	13.7
	command-r-35b	55.7	60.0	56.1	21.2	21.7	20.1
	gpt3.5	53.3	51.0	48.7	20.8	15.1	14.8
	gpt4	51.6	59.3	53.4	19.5	17.9	17.8
fine-tuned	gpt3.5	**67.6**	**62.7**	**63.6**	**31.9**	**27.5**	**27.9**

Table 3.11: The plans and suggestions made by LLMs for covering press releases do not align with human journalists. **Precision** is the number of items from the plan that the journalist actually pursued (averaged per press release). Average **Recall** is the number of items from the human-written article also suggested by the plan (averaged across news article). **Angle** is suggestions for directions to pursue, [304]. **Source** is suggestions for sources to speak with, in general terms (e.g. "a manager at the plant", "an industry expert".)

LLM as a planner, we give an LLM the press release, mimicking an environment where the LLM is a creative aide. We prompt an LLM to "de-spin" the press release, or identify where it portrays the described events in an overly positive light, and suggest potential directions and sources to pursue. [29] Our angle prompt builds off Petridis et al. [304], however, our source prompt is novel, given the importance attributed to sources in Section 3.3.2. Next, **(2) Human as a planner**, we use another LLM to assess what the human *actually* did in their reporting. Finally, **(3) Comparing**, we assess how the LLM plans are similar or different from the human plans.

[29] We keep these sources as generic sources, e.g. "a federal administrator with knowledge of the FDA approval process", not a specific person.

	Description	More Detail
1	Directly related the press release and supporting it's contents.	Can be derived just by summarizing a point in the press release.
2	Related to the press release but questioning it's points.	Little more than a simple pattern-based contradiction to a point in the press release.
3	Takes an angle outside of the press release, but relatively limited.	Can be a generic, larger-trend kind of contradiction.
4	Adds substantial and less obvious context or history.	Substantial knowledge of prior coverage and company awareness involved in making this choice.
5	Entirely new direction.	Substantial investigatory work was involved even to make this suggestion.

Table 3.12: Description of the 5-point creativity scale that we used to evaluate decisions made while covering press releases. Based on [321], our scale captures different levels of creative ideation: direct engagement with the press release (1-2), contextual/trend-level rebuttals (3-4) substantial and novel investigatory directions.

3.3.3.2 Models and Evaluations

We consider two pre-trained closed models (GPT3.5 and GPT4[30]) and two high-performing open-source models (Mixtral [319] and Command-R [320]). We conduct experiments in 3 different settings: **Zero-shot**, where the LLM is given the press release and definitions for "angle" and "source", and asked to generate plans. **Few-shot**, where the LLM is given 6 examples of press release *summaries*[31] and the human-written plans.[32] Finally, we fine-tune GPT3.5[33] on a training set composed of press releases paired with human plans. We give full prompts for all LLM queries run in this paper in [273].

3.3.3.2.1 Evaluation 1: Precision/Recall of LLM Plans We first analyze plans made by humans: we extract sources used in human-written news articles with models trained by Spangher et al. [1]. Then, we give GPT4, our strongest LLM, the press release and

[30] `gpt-4-0125-preview` and `gpt-3.5-turbo-0125`, as of February 9th, 2024.
[31] We use summaries to inform our few-shot examples because full press releases are too long for context.
[32] We manually write the summaries and the plans.
[33] Using OpenAI's fine-tuning API: `https://platform.openai.com/docs/guides/fine-tuning`

human-written news article and ask GPT4 to infer the angle that the author took. We manually validate a sample of 50 such angles and do not find any examples we disagree with. Finally, we use GPT4 to check how the sources and the angle proposed by the LLMs match the steps taken by the journalist. From this, we calculate Precision/Recall per document, which we average across the corpus.

3.3.3.2.2 Evaluation 2: Creativity of the Plans We recruit two journalists as annotators to measure the creativity of the plans pursued both by the LLMs and the article authors. We develop a 5-point scale, inspired by Nylund [321], who studied the journalistic ideation processes. They found that journalists engaged in processes of new-material ingestion, brainstorming in meetings to assess coverage trends, and individual ideation/investigation. In our scale, scores of 1-2 capture "ingestion", or a simplistic engagement and surface-level rebuttals of the press release; scores of 3-4 capture "trend analysis", or bigger-picture rebuttals; scores of 5 capture novel directions.[34]

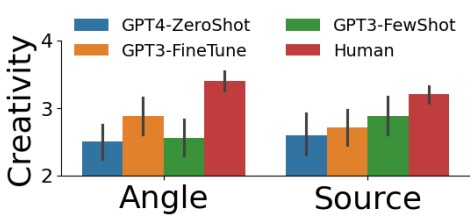
3.3.3.3 Results comparing $\pi^{(llm)}(\tau|x)$ with $\pi^{*}(\tau|x)$

Table 3.11 shows the results of our matching experiment. We find that LLMs struggle to match the approaches taken by human journalists, but LLMs are better at suggesting angles than source ideas. Few-shot demonstrations do not seem to improve performance, in fact, we observe either neutral or declining performance. Fine-tuning, on the other hand, substantially improves the performance of GPT3.5, improving to 63.6 average recall for Angle suggestions and 27.9 average recall for Source suggestions, a 10-point increase in both categories. We manually annotate 60 samples from the LLM matching to see if we concur with its annotations. We find an accuracy rate of 77%, or a $\kappa = 0.54$[35].

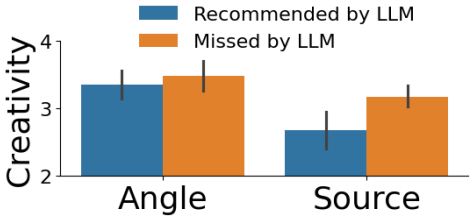
We observe slight different results for creativity. As shown in Figure 3.6a, creativity is overall lower for all categories of LLM: zero-shot, few-shot, and fine-tuning. However,

[34]We report our 5-point scale in Table 3.12.

[35]The cases of disagreement we found were either when the LLMs plans were too vague, or contained multiple different suggestions: we usually marked these "no" while the LLM marked them "yes".



(a) Average creativity of suggestions given by a sample of LLMs (1–5). Human creativity is evaluated on steps taken by actual journalists during reporting.



(b) Average creativity of human ideas *matched* to GPT-3.5 fine-tuned suggestions ("Recommended by LLM") vs. *unmatched* ("Missed by LLM"). No significant difference for Angles; significant difference for Sources.

Figure 3.6: Creativity evaluation results across models and match status.

in contrast to the prior experiment, we find that the differences between human/LLM creativity are relatively similar for source plans and angles. Further, when we observe the creativity of *just* the human plans that were retrieved by GPT3.5-fine-tuned, shown in Figure 3.6b, we observe a similar pattern: the human plans matched to GPT3.5's plans are, overall, less creative than those that were not matched.

3.3.3.4 Discussion: LLMs do *not* plan like Human Writers

We assessed how LLMs can help journalists plan and write news articles. We constructed a large corpus of news articles covering press releases to identify existing journalistic practices and evaluate how LLMs could support those processes. We found that LLM suggestions performed quite poorly compared with the reporting steps actually taken by humans, both in terms of alignment as well as creativity. Does this suggest that LLMs are poor planners in practice? Our benchmark provides a useful check for this question, but we do not believe our experiments here are conclusive. Instead, we view our approach as a first step: we compare basic prompt engineering with human actions that are observed from *final-draft writing*. Clearly, the final drafts written by humans result from multistep, iterative reporting, accumulated experience, and real-world knowledge.

Using human-decision making as a basis of comparison for LLMs is standard, even in creative, open-ended tasks: e.g. story-planning [322], computational journalism [17, 1,

289] and others [323]. If this problem were unlearnable (e.g. there were simply too many angles to take, or so much prior knowledge needed to form any kind of plan), then we would not see any improvement after fine-tuning. Crucially, the 10-point improvement we observe from fine-tuning is evidence that there are learnable patterns. Existing research into journalism pedagogy, which implies that observation of other journalists' standard practice is as important as gaining subject-matter expertise and conducting on-the-ground work [324], should further support the hypothesis that planning is learnable.

However, the low scores after fine-tuning imply the need for more fundamental work. Our current approach is naive: we expect LLMs to produce human-level plans with simple prompting and no references, besides the press release. There are two major directions for advancement in this task: **(1) creativity-enhancing techniques:** The creativity gap we observed between humans and LLMs reflect similar findings in other recent research related to creativity in AI [325, 326, 327, 328]. Chain-of-thought style prompts that explicitly include creative planning steps [318, 219], or multi-LLM approaches [328] could improve creativity. **(2) identification-oriented grounding**: we observe that many of failures in LLM plans are rooted in LLMs lack of awareness of prior events, even high-profile events that were within its training window (e.g. it interpreted many Theranos press releases without any awareness of the company's travails [329]). Retrieval-augmented generation [330] and tool-based approaches [331] might yield improvement.

As LLMs are increasingly used for planning-oriented creative tasks [318], careful analysis is required. Our goal in this work was to outline a novel task requiring planning and affirm a basic to perform this analysis. We believe that our use of LLMs in article planning represents an emerging and as-yet-underexplored application of LLMs to tasks *upstream* of the final writing output. In these cases, the decisions made by the LLM might one day have the ability to impact even more fundamental steps: which sources to talk to, which angles to take, and which details to highlight. Professional journalists ground their approach to these decisions in institutional values: fairness, reducing sourcing bias, and

confirming details. Without carefully comparing pretrained $\pi^{(llm)}(\tau|x)$ with human expert $\pi^*(\tau|x)$, we risk disregarding these values.

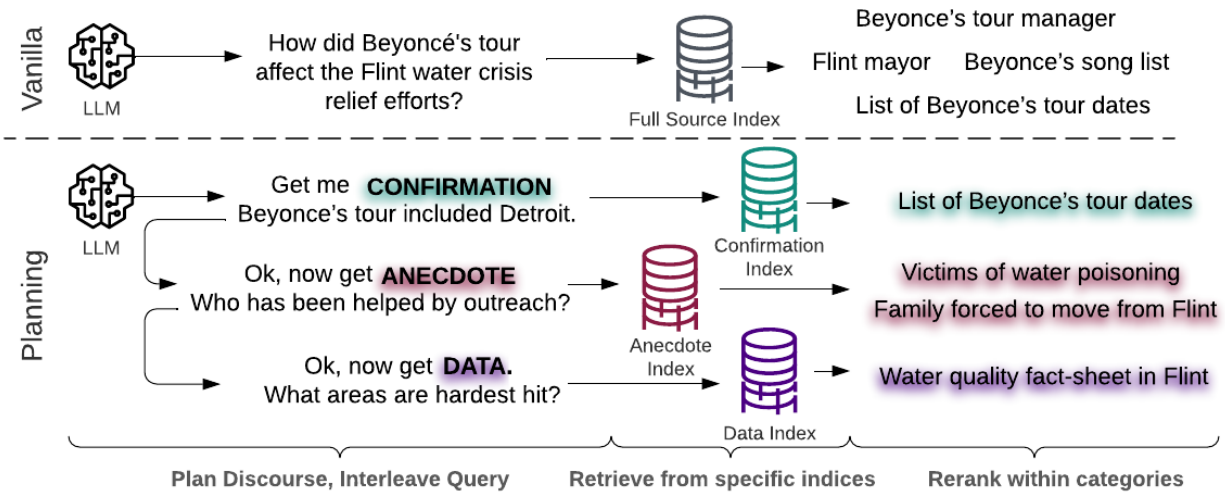


Figure 3.7: An overview of our planning-executor process for retrieving sources, demonstrated on a story idea about the Flint Michigan crisis. The *planner* decides what is needed for the story while the *executor* issues queries to retrieve sources.

3.4 Hierarchical Planning for Emulating *Source-Finding*

So far, we have demonstrated that we can make inferences about actions performed by human journalists while reporting, $q_\theta(\tau|g)$, and we have shown that pretrained LLMs struggle to replicate these actions, $\pi^{(llm)}(\tau|x) \neq \pi^*(\tau|x)$. We are now ready to explore an approach to learning a policy $\hat{\pi}(\tau|x)$ for *source-finding* that introduces a novel view on *emulation* that we have not yet explored: **aligning distributional similarities between trajectories from human expert trajectories, τ^*, inferred via $q_\theta(\tau|g)$, and our model's policies $\hat{\tau}$.**

We will explore what specific *distributions* we will use to enforce similarities, but first, let us defend it's applicability in *emulation learning (EL)*. In many human tasks, the target is not a single "optimal" trajectory, but a policy that reproduces *regularities of human end states and their schema-level structure*. Because many distinct trajectories can yield the same article g (*equifinality*), our aim is often to recover policy behaviors that match the *distributional* signatures of human work, rather than exact stepwise actions. This perspective complements reward specification: instead of hand-crafting R, we aim to *emulate* the human

policy that produced g by matching sufficient statistics derived from g. We conceptualize our policy model $\hat{\pi}(\tau|x)$ as driving a two-step process: a *planner* that determines what kinds of sources are needed to complete the narrative, and an *executor* that actually *finds* the needed source, by issuing queries to a retriever, as shown in Figure 3.7. The goal is to *emulate* the human journalist by retrieving the same kinds of sources as the journalist would of, thus demonstrating the ability to understand journalistic reasoning. We interpret the planner's choices as *options* (skills) and the executor's queries as *primitive actions* within those options. Let $\mathcal{D}$ denote the set of narrative functions sources take. The planner selects $o_t \in \mathcal{D}$ with policy $\mu(o_t \mid s_t, x)$; the executor issues a retrieval action $a_{r,t} \sim \pi(\cdot \mid s_t, o_t, x)$; each option terminates with $\beta(o_t, s_{t+1})$. This semi-MDP view provides a clean bridge between $q_\theta(\tau \mid g)$ (inferred human plans) and $\hat{\pi}(\tau|g)$ (emulated behavior).

More formally, we expand our action space a, beyond that defined at the start of this Chapter (i.e. a_1 =*Get source #1*...). We define *actions* a_r as queries issued to the retriever (the retriever and database are described in Section 4.3.1.2); *thoughts* a_t as any actions that do directly interact with the retriever but help us determine what actions a_r to take (these could be reasoning tokens generated by an LLM, or predictions made by a secondary model). This implies a hierarchical plan-and-execute process, where we realize our policy model $\pi(a|x) = \pi([a_t, a_r]\,|x) = \pi_{planner}(a_t|x)\pi_{query}(a_r|a_t, x)$. As before, the state space S to be the sources retrieved so far during the trajectory. The goal state g is defined as the published news article and, as before, we can extract a set of sources Q from g using the $\alpha(X_i)$ model trained in in Section 3.2. Note that we are not trying to explicitly define a reward function, or make any conditions on how the sources interact with each other. We simply assume, based on predictability insights learned in Section 3.2.3, that our model will learn what these interaction patterns are, yielding one kind of *emulation loss*, or goal-guided loss, *without explicit rewards*. To understand this, let $\psi(\hat{\tau})$ summarize a learned trajectory into schema-level signatures (i.e. a histogram over characteristics of each a_i like, for example, the narrative or discourse role of the source, see Section 1.2.2) and let $\psi(Q)$ be the signature

for an observed human trajectory, extracted from the human article g. With this approach, we train our policy, $\hat{\pi}$, to minimize a divergence between these summaries:

$$L_{\text{emul}}(\hat{\pi}) \;=\; \mathbb{E}_x \, \mathbb{E}_{\tau \sim \hat{\pi}(\cdot|x)} \Big[D\big(\psi(\tau),\, \psi(Q)\big) \Big],$$

where D is a distributional distance. This encourages emulation of human discourse structure without specifying explicit rewards.

3.4.1 Task and Dataset Creation

To set up our multi-document retrieval task, we wish to create *a large retrieval database where multiple "documents" are labeled as ground-truth for answering each query.* To construct our task, we apply the inverse $q_\theta(\tau|g)$ function described in Section 3.2 to extract sources from news articles. We also generate queries from press releases, and finally a latent *discourse* structure, described next. These steps follow *EL*'s backward lens: we start from end-states g and infer latent structure (queries, sources, discourse roles) that plausibly produced g. Practically, $q_\theta(\tau \mid g)$ is multi-modal; there are many valid τ for the same g. Thus, our schema learns *ensembles* or *summaries* of τ (e.g., discourse mixtures) to avoid over-committing to a single inferred path. (For a reminder on discourse and its role in *emulation*, see Section 1.4).

3.4.1.1 Dataset Creation

For each news article, we extract two items: (1) a query describing the initial question answered by the journalist and (2) the set of informational sources used by the journalist. The queries serve as the input to our retrieval problem, while the text of each source serves as the ground truth matching "document" for each query. Following the definitions in Spangher et al. [1], sources can be people (e.g., individuals interviewed or issuing statements), documents (e.g., studies, legal documents), or datasets. We use a dataset of articles released by Spangher et al. [273], described in Section 3.3, which includes 380,000

news articles covering business press releases. From this dataset, we sample 50,000 articles and their corresponding press releases.

Query Generation We provide an LLM with both the press release and the corresponding news article, asking it to generate a query that might describe an initial question the journalist had upon reading the press release, which led them to write the article.

Source Extraction First, we identify all informational sources in each news article using models trained by Spangher et al. [1]. Then, we use Llama-3.1-70B[36] to extract, for each source, a stand-alone packet of information provided by that source[37] "Standalone" means that we can accurately identify the source later in the retrieval database. In total, we extract 400,000 sources, averaging approximately 8.3 sources per document.

3.4.1.2 Discourse Schema Generation

We seek to create a low-dimensional schema to describe our sources (in order to ground our planner). We describe that process now. Inspired by Pham et al. [332], we first ask an LLM to generate descriptive labels for the discourse role of each source, based on its source extraction. This allows for a broad superset of labels (examples are given in [18].). Then, we cluster these labels by (1) annotating pairs of labels with similarity judgments using an LLM[38], (2) using these annotations to train an SBERT embedding model [333], and (3) clustering these embeddings using k-means. We identify eight distinct clusters that represent different narrative roles (e.g., "Main Actor," "Expert" "Background Info"). Definitions for each discourse role are given in [18]. Additionally, we ask the LLM to label the centrality of the source: "High" (the source is crucial to the narrative), "Medium" (the source plays a significant role but is not necessary) and "Low" (the source could be easily replaced with another source). We show the breakdown of Discourse Roles by Centrality in Figure 3.8, and give additional analysis [18]. The discourse schema

[36] `https://huggingface.co/meta-llama/Meta-Llama-3.1-70B-Instruct/`
[37] This includes: resolving all coreferences and stating the full names of places, people, and events.
[38] Specifically, whether two different narrative roles generations are substantially the same or not.

Label	%	Label	%
Main Actor	19.0%	Data	10.2%
Background	18.9%	Confirm.	9.2%
Counter.	11.3%	Analysis	7.8%
Anecdotes	10.8%	Broaden.	1.6%
Expert	10.5%	Subject	0.7%

Table 3.13: Distribution of Discourse Types in News Articles. 'Main Actor' and 'Background Info.' are the most common, and 'Subject' the least common.

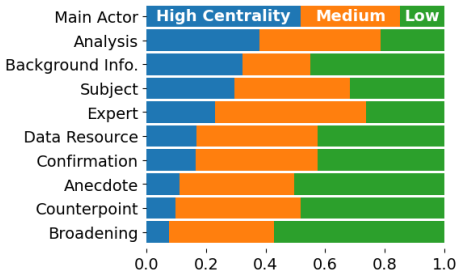


Figure 3.8: Proportion of sources within each discourse role that occupy High, Medium or Low Centrality in their stories.

serves as a low-dimensional *macro-plan* space that improves identifiability: rather than reproducing token-level actions, $\hat{\pi}$ matches stable invariants (role mixtures, centrality) that q_θ consistently attributes to g. This reduces variance from equifinality while preserving the human-meaningful structure we seek to emulate.

3.4.2 Analysis

In order to better understand our dataset, we conduct a series of analyses to show how sources are used in news writing by journalists. We make three insights.

Insight #1: Diversity and perspective alone do not characterize source inclusion Prior research typically assumes that increasing diversity, in multi-document retrieval makes retrieval more comprehensive [334, 335, 336]. However, we observe that, in news writing, diversity is not always emphasized. While many sources are chosen for diverse information, others are chosen specifically to confirm facts. For example, ∼10% of sources play a Confirmation role, as in Table 3.13. *What other theories exist to explain source-selection criteria in journalism*? Gans [337] suggests that supporting and opposing viewpoints are selected to give a balanced narrative, suggesting that *stance* is a primary driver for source selection. We conduct an analysis of sources' stances in the narrative, using Ma et al. [338]'s stance-detection method[39]. We find that while some sources do fit into the "for"

[39]Ma et al. [338] used Llama 3.1 with chain-of-thought prompts to detect stance; this scored highly on

and "against" categories, this is not universally the case. Over 30% of sources take an informational perspective *without explicitly supporting or opposing any viewpoint*[40]. This suggests that source selection is more nuanced than the binary "for and against" model implies. Journalists often include sources to provide context, background information, or expert analysis, which may not directly relate to a polarized viewpoint [340].

Insight #2: Certain Kinds of Stories Use Different Kinds of Sources We examine whether different types of news stories use sources differently. We manually identify different kinds of coverage: investigative reports, breaking news, etc. We find that different kinds coverage tend to be dominated by different source discourse roles. For instance, investigative reports tend to include more "Expert Analysis" and "Background Information" sources, while event coverage focuses on "Main Actors" and "Eyewitnesses." This analysis highlights that source selection is context-dependent and varies across different types of journalism. Understanding these patterns can inform the development of more sophisticated information retrieval systems that tailor source recommendations based on the story type.

Insight #3: Sources used in multiple documents tend to have the same discourse roles. We expected that sources would often be used in different roles in differet articles: for instance, in Story #1, a police officer might be a "Main Actor", in Story #2 the same police officer might used for "Background info." and in Story #3, for an "Anecdote". We conduct an analysis on all named sources that we name-match across two or more articles and find that, on average, sources tend to be classified in the same role (sources have .43 gini impurity[41], .33 label inconsistency[42], .95 entropy and .55 diversity[43] across discourse roles).

popular stance benchmarks. Specifically, we prompt the model to classify the stance of each source as "supporting," "opposing," or "neutral" with respect to the main event or topic of the article (see [339] for the full prompt).

[40]Shown in [339]

[41]Gini impurity is measured as $1 - \sum_i \left(\frac{l_i}{l_{total}}\right)^2$, where l_i is the count of label i and l_{total} is the sum of all label counts

[42]Inconsistency is defined as $1 - l_{max}/l_{total}$ where l_{max} is the label with the maximum count.

[43]Where diversity is defined as $l_{numunique}/l_{total}$

One possible explanation is that journalists observe how other journalists use sources, and use them similarly. This is a crucial insight: for simplicity, in the rest of the paper, we assume that sources' discourse role is only based on their original source-text.[44]

3.4.3 Discourse in Multi-Document Information Retrieval

Given our source and query dataset, described in Section 4.3.1.2, we now present our methodology for discourse-aware multi-document retrieval. Motivated by our findings in Section 3.4.2, we posit that incorporating discourse structures can significantly enhance the retrieval process. In Section 3.4.3.2, we discuss how discourse information can inform the retrieval process and in Section 3.4.3.3 we discuss ways to infer a story's discourse requirements.

3.4.3.1 A hierarchical approach to retrieving sources

We start by testing an *interleaving retrieval approach* to address this task [341]. In this approach, an LLM is used to iteratively: (1) issue queries to a retriever (2) reason about the sources returned (3) issue follow-up queries. Note that this is *also* relying on $\pi^{(llm)}(\tau|x)$ which is biased, as we explored in the previous section. Human validation, additionally, shows that these interleaved queries frequently repeat, meander, or degenerate, ultimately failing to capture the diversity of sources present in human writing (Section 3.4.5). We hypothesize that a higher-level planner can guide the interleaving process towards diversity while staying focused on the query. For example, we would like a higher-level planner to predict: *"this query is likely to answered by anecdotes, data, experts and actors"* – we can then use this plan to guide interleaving steps. Beyond instance-level relevance, our retrieval policy *should* emulate human discourse composition. To make training such a planner tractable, we first constrain the space of possible plans: we do this by developing a novel discourse

[44]We hold this constant to simplify computation. We acknowledge this is a limiting assumption, and in follow-up work we will remove that assumption. Allowing sources to adapt their discourse roles dynamically in response to novel, unseen queries is a crucial area for future research.

schema (described in Section 3.4.1.2). With this lower-dimensional planning space in hand, we train a high accuracy autoregressive planner and we evaluate both (i) set-level overlap with ground-truth sources and (ii) *schema alignment*. We will describe this now.

3.4.3.2 Overview of *Planned Interleaved Retrieval*

Our retrieval framework consists of three main stages, illustrated in Figure 3.7: (1) Query Planning, (2) Discourse-Specific Indexing and Retrieval, and (3) Re-ranking. We describe each of these steps, focusing on how discourse roles can be involved.

Stage 1: Interleaved Querying In the first stage, we employ an LLM to generate queries $q_1, ...q_n$ sequentially in order to retrieve sources, as in Trivedi et al. [341]. Discourse-awareness in this stage means the LLM can reference the discourse role of the source it desires to obtain in query round q_t while generating it's query (we will discuss in Section 3.4.3.3 how we infer these discourse roles).

Stage 2: Indexing and Retrieval Given a query, q_t, we then retrieve sources $s_1, ...s_k$ relevant to this query. Discourse-awareness in this stage means that the retrieval indices themselves are filtered to discourse roles of sources in our corpus. Traditional multi-document retrieval systems treat all documents equally **[voorhees1999trec]**, but our approach organizes the index into hierarchical, discourse-driven sub-indices. This stratification allows for more targeted retrieval. When the LLM generates a query for a particular discourse role, it is directed to the corresponding sub-index.

Stage 3: Re-ranking Finally, given a large set of sources $s_1, ...s_m$ retrieved in the prior steps, we re-rank them to surface the sources that are most relevant together. In this stage, discourse awareness means that we take the most relevant documents *within* each discourse category. This additional layer of categorization prioritizes documents that best fulfill the intended narrative role. We use a re-ranking model that incorporates both relevance and discourse compatibility, similar to the approach in Nogueira and Cho [342].

Retriever	Strategy		Overall Results			Results by Cent. (F1)		
	Seq.	A-priori	Recall	Prec.	F1	High	Med	Low
BM25 [344]			0.00	0.00	0.00	0.00	0.00	0.00
DPR [223]			13.98	9.12	11.04	14.42	6.82	5.68
Interleaving [341]			**25.81**	27.04	26.34	37.66	22.60	14.37
PIR	✓	–	24.07	25.27	24.60	33.88	21.28	14.05
	–	✓	25.49	31.61	28.04	**40.43**	22.17	14.32
	✓	✓	24.84	**33.15****	**28.12****	40.16	22.55	**14.77**
Oracle PIR	–	–	42.77	42.98	42.86	54.02	37.73	26.78

Table 3.14: We show results of running different retrieval strategies, in terms of Recall, Precision, F1 score. Each strategy uses multiple retrievers. with the Oracle strategy demonstrating the highest performance metrics. ** indicates significant increases at $p < .01$, obtained via bootstrap resampling ($b = 1,000$).

3.4.3.3 Two Different Planning Approaches

As outlined in the previous section, we can incorporate discourse information at each stage in our retrieval process. However, left unexplained was how *we would infer* these discourse roles. Now we discuss the two approaches we take.

Approach #1: Sequential Planning Here, the query-generator is informed of the possible discourse categories, and is asked to pick the next discourse role that a story requires. In other words, at turn t, the LLM views prior $q_{1,...t-1}$ and discourse roles $d_{1,...t-1}$ of retrievals, and is asked to generate the next discourse role, d_t that the story requires. By allowing an LLM to sequentially generate roles, we hypothesize that we can introduce a human-like planning ability – i.e. often humans do not know the exact discourse roles a story needs until they get deeper in [343]. However, this approach relies the LLM's inherent ability to reason independently about discourse roles without explicit guidance. Prior studies have shown that LLMs struggle with structural reasoning in complex tasks [26], suggesting that this method may be less effective.

Approach #2: A-priori Planning In this approach, we train an auxiliary planner to predict

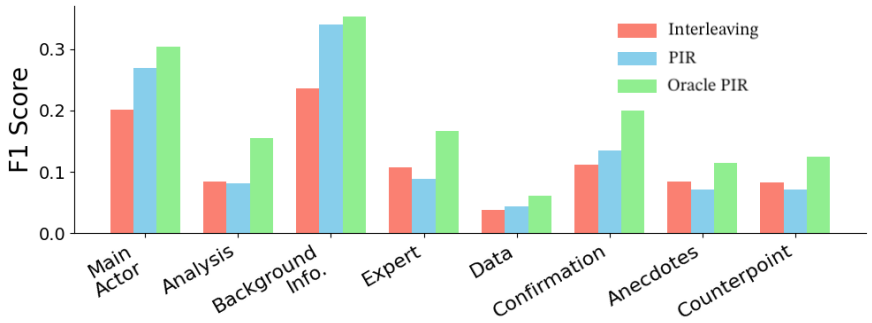


Figure 3.9: Retrieval accuracy scores, broken down by different discourse types. As can be seen, introducing my discourse planning has a greater impact on certain kinds of discourse categories (e.g. Main Actor and Background Info.) compared with other discourse types (e.g. "Experts", "Anecdotes" and "Counterpoint").

the entire distribution of discourse roles the document will take, a-priori, based on the initial query. To do this, we cluster articles based on the distribution of source narrative roles, using K-means clustering with $k = 8$ clusters and train a `DistilBERT-base` classifier [345] to *infer* which story cluster a query belongs to. In other words, the a-prior planner predicts the proportion of each discourse role expected in the final document, based on the initial query. The predicted distribution is then provided to the LLM during the query planning phase[45] We train the auxiliary model on our dataset, achieving a macro F1 score of 0.72 in classifying queries into the correct discourse clusters. The average KL divergence between the predicted and true discourse distributions is 0.7, indicating a close approximation.

3.4.4 Experiment Setup

Retriever We use SFR[46]: a 7B text-embedding model developed by Salesforce AI Research that has demonstrated superior performance across multiple benchmarks. We choose SFR as a powerful, large instruction-tuned model in order to understand richer and more nuanced queries that we anticipate our task will require.

[45] Prompt example: "`We expect this document will contain 50% Background, 30% Expert Analysis, and 20% Main Actor information. Please choose the next discourse role you want to use.`"

[46] `https://huggingface.co/Salesforce/SFR-Embedding-2_R`

LLM As in Trivedi et al. [341], an LLM is used to plan and reason about the next query to issue. As in the rest of the paper, we use Llama-3.1-70B.

Dataset We perform an 80/20 split for training and test sets. To construct the retrieval index, we aggregate all sources from both sets and organize them according to discourse role, such that each role is indexed separately. That is, for every query, a distinct retrieval index is created for each type.

Baselines (1) *BM25:* a widely-used probabilistic retrieval framework, calculating the relevance of documents to a query based on the frequency of query terms in each document. (2) *Dense Passage Retrieval (DPR)* [223]: *we fine-tune* a transformer-based model[47] to to effectively capture semantic similarities beyond keyword matching. Fine-tuned DPR allows us to test whether learned knowledge is more important than planning or reasoning. To finetune DPR, we build a training dataset that including negative samples for in-batch training [223]. For each positive pair of query q_j and its relevant sources s_j^+, we include n negative tools as negative samples. (3) *Interleaving:* we employ SFR with an identical setup to Trivedi et al. [341] in order to test the ability of LLMs to reason about the needs of the query in the absence of discourse labels.

Oracle Finally, to differentiate the role of discourse from these two noisy discourse inference techniques, we test an oracle approach. In this approach, we provide the LLM with ground-truth discourse labels extracted during our analysis. By supplying the actual distribution of discourse roles present in the target documents, we assess how well the system can perform when it has perfect knowledge of the sources' discourse structure. Also, this highlights potential improvements in retrieval planning and reasoning mechanisms.

3.4.4.1 Results

Our main finding is that incorporating discourse labels helps us retrieve sources with significantly higher accuracy than baseline approaches (we find that these improvements

[47] `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`

are significant at $p < .01$ by running bootstrapped resamples with $b = 1,000$). As evidenced in Table 3.14, including discourse labels (with both **a-priori** and **sequential** strategies) elevates the F1 score from 26.34% to 28.12% compared with the baseline *Interleave*. Further, when incorporating oracle discourse information, the F1 score boosts up to 42.86%. This indicates that discourse awareness and planning can provide insights into query needs. The gains from discourse-aware planning indicate that matching human schema-level invariants (role mixtures, centrality) provides a stronger training signal than token-level query reasoning alone, aligning with the EL claim that emulating *structure of end-states* is a powerful surrogate for explicit reward design.

Secondly, and intriguingly, our results suggest that an a-prior planning-based approach has a more pronounced impact than sequential planning. According to the results in Table 3.14, employing a-priori planning *without* sequential planning[48] yields an F1 score of 28.04%. In contrast, combining both sequential *and* a-prior planning results in a slightly higher F1 score of 28.12%. The small difference between these two trials suggests that a-priori planning alone can substantially enhance retrieval effectiveness, potentially diminishing the incremental benefits introduced by sequential planning. This contrasts with recent results on more conventional QA-based IR tasks, where prompt-based planning strategies were shown to significantly enhance retrieval performances [341, 346]. These results suggest that our task possesses inherent differences. We do caveat our results with awareness that our a-priori planner was trained while our sequential planner relied on LLM pretraining (as did [341]). This suggests both that (1) a narrative-focused query objective is distinct from purely informational query tasks like those studied previously, and (2) an a-prior plan is useful in this task, indicating that templates exists that journalists follow.

[48] In other words, we simply retrieve $k \times n$-rounds of candidates in the first round, without interleaving, and then re-rank according to the a-priori predicted discourse distribution

3.4.5 Discussion

We investigate why incorporating the discourse aspects into the systems enhances machine's source retrieval ability above the *Interleaving* approach.

Vanilla Interleaving Tends to Meander To explain the subpar performance of *Interleaving*, which has shown state-of-the-art results on QA benchmarks, we examine multiple query threads. Vanilla interleaving exhibits three notable failure modes. (1) Many queries generated by the planner tend to restate the same objectives or focus on overly narrow aspects of the broader topic without expanding into complementary dimensions (see [18]). This restricts the planner's ability to explore the full range of sources that a humans typically consider (e.g. expert opinions, counterpoints, or data analysis), thus producing a less well-rounded article. (2) Paradoxically, while interleaving often remains closely aligned with the initial query's intent, it also suffers from a tendency to drift when progressing through subsequent queries. For instance, an initial focus on the societal consequences of an issue may eventually lead to highly specific and less generalizable topics that deviate from the core inquiry. (3) Finally, even when the planner maintains alignment with the initial query, it often fails to explicitly request critical discourse roles, such as expert analyses or contrasting viewpoints. Consequently, the output of vanilla interleaving lacks the depth and balance.

Varied Centrality Improvements As shown in Table 3.14, the retrieval system shows marked improvement in handling sources of varying centrality when informed by discourse roles, particularly with the oracle setup. For high centrality sources, the Micro-F1 score leaps from 37.66 to 54.02, indicating enhanced effectiveness in identifying and retrieving crucial sources. Similarly, for low centrality sources, the Micro-F1 score rises from 14.37 to 26.78, demonstrating the system's expanded capability to incorporate less central, yet informative perspectives into the narrative, thereby enriching the overall information retrieval process. The improvement from our planning strategies, we observe, originates from the enhanced

retrieval of more central sources; this indicates that our planning strategies effectively identifies and prioritizes sources crucial for constructing detailed narratives. However, while the system excels at retrieving high centrality sources, there is room for improvement in capturing more medium and low centrality sources. Enhancing our planning to better include these sources could further enrich the comprehensiveness of the IR process.

Discourse Role F1 Analysis As shown in Figure 3.9, incorporating discourse role information significantly enhances retrieval performance across discourse roles. Since $q_\theta(\tau \mid g)$ is multi-modal, any single decoded plan may be arbitrary. Training with distributional targets mitigates overfitting to one inferred trajectory. By accounting for the specific functions that sources play in constructing a narrative, the retrieval system is more adept at identifying and selecting *comprehensive* information. The consistent enhancements across diverse categories highlight the effectiveness of a discourse-aware approach, suggesting that a nuanced understanding of narrative structures is essential for optimizing retrieval outcomes in complex tasks such as multi-document source retrieval. However, the selective improvements observed with our planning strategies indicate that while these strategies are beneficial, their effectiveness varies across different source categories. Significant gains are achieved in categories central to the narrative—such as Main Actor and Background Information—where the discourse roles are closely aligned with the main query and can be explicitly planned for. This suggests that planning strategies are most effective when the narrative role is straightforward and directly related to the primary focus of the query. In contrast, categories requiring nuanced understanding—such as Analysis, Expert, Anecdotes, and Counterpoint—exhibit less improvement, implying that current planning strategies may not fully capture the complexities inherent in these discourse roles. Consequently, further refinement of these strategies is necessary to enhance retrieval performance in categories that demand deeper contextual and interpretive analysis. EL intentionally *matches* journalistic norms; this can also replicate undesirable sourcing habits (e.g., under-represented voices). Auditing $\hat{h}_\phi(x)$ against fairness constraints (role

coverage, actor diversity) is therefore integral to safe deployment.

Retrieval Hyperparameters Our preliminary experiments reveal that the effectiveness of discourse-aware retrieval is sensitive to the choice of k, the number of documents retrieved per query. The benefits of incorporating discourse information, we find, become more pronounced with larger k values. This is consistent with findings from Craswell et al. [347], who note that re-ranking models have more impact when the initial retrieval set is large. We attempt different methods for learning the ideal k per query: we train a Poisson regression model using a simple Multilayer Perceptron (MLP) on SBERT embeddings [221]. However, the model achieves a low Pearson correlation of $r = 0.35$ between the predicted and actual optimal k values. Overall, this additional planning step fails to measurably impact performance. We leave this to future work.

While our current approach is specialized for journalistic source selection, we see the potential applicability to other domains like scientific literature and legal document retrieval. Adapting our method to these areas would involve redefining discourse categories relevant to the target domain, retraining discourse-role classifiers on domain-specific corpora, and validating with subject matter experts. Journalists often face time-constraints on the number of sources they can talk to, making news article analysis a particularly tractable domain to start in, but we anticipate that structured discursive frameworks common in these domains would particularly benefit from our planned retrieval methodology.

Additionally, we recognize the computational overhead introduced by large models such as Llama-3.1-70B and SFR-7B. In the future, we plan to explore smaller, distilled models and computationally efficient techniques, including knowledge distillation and quantization. Additionally, we look forward to testing additional baselines to validate our approach, such as token-level dense retrievers [348, 349] or in-context learning approaches [350, 351].

3.5 Examining Discourse Schemas for *Source-Finding*

The introduction of discourse roles for sources in the previous section explores how a schema can help us to tractably learn more human-like policies, $\pi^*(\tau|x)$, by introducing a kind of *emulation loss*. As a recap, we introduced in Section 3.4 a 'schema signature" $\sigma_Z(\cdot)$ (i.e. the distribution over schema elements) to describe trajectories, τ and distribution-minimizing loss between $\sigma_Z(\hat{\tau})$ of the learned policy's trajectory $\hat{\tau}$ and $\sigma_Z(\tau^*)$ the inferred human trajectory τ^*, called $D(\sigma_Z(\hat{\tau}), \sigma_Z(\tau^*))$. As our experiments showed, this approach can deliver promising results in tasks, like *source-finding*, where goal-states have equifinality (Section 3.4) and require compositional reasoning[49] (Section 3.2.3).

Although we justified a schema-driven approach in Section 3.4 primarily on the basis of computational tractability, let us now explore deeper theoretical and practical roles of schemas in *emulation learning*. A primary goal in EL is to uncover new insights about human behavior. When studying human actions, inferred from $q_\theta(\tau|x)$, we wonder: what drove these actions? What role does higher-level decision-making and cognitive organization have in behavioral trajectories? Classic theories in cognitive psychology and cognitive control view *schemas* as being *essential* [354, 355, 356, 357]; they both constrain and enable — they provide meta-structures to guide action and reasoning, while supporting flexible recombination in novel contexts [358, 359]. Within narrative storytelling, discourse theories show how higher-level structures (e.g., roles, relations, rhetorical functions) give structure to utterances [360, 361, 362, 363, 364]; event-segmentation work further indicates that humans perceive the world at boundaries that often coincide with schema transitions [365]. Hierarchical RL integrates these by treating abstract "options", that organize exploration and credit assignment, as schema-like macro-actions [366].

[49] *I want to note that schema distribution-matching is not the only way to perform the policy-learning goals of emulation learning*. I have personally been very convinced by recent results in reasoning [352, 353] which represents a way to incorporate latent. While schema-driven and, importantly, hierarchical policy learning, has an important role in *emulation learning*, other approaches for policy learning explored in Chapters 1 and 2, like direct supervision or reward learning, especially if combined with steering reasoning, to me seem more promising in their power and flexibly.

Headline: NJ Schools Teach Climate Change at all Grade Levels
Michelle Liwacz asked her first graders: what can penguins do to adapt to a warming Earth? ← *potential labels:* Academic, Neutral
Gabi, 7, said a few could live inside her fridge. ← *potential labels:* Unaffiliated, Neutral
Tammy Murphy, wife Governor Murphy, said climate change education was vital to help students. ← *potential labels:* Government, Agree
Critics said young kids shouldn't learn disputed science. ← *labels:* Unaffiliated, Refute
A **poll** found that 70 percent of state residents supported climate change being taught at schools. ← *potential labels:* Media, Agree

Table 3.15: Informational sources synthesized in a single news article. *How would we choose sources to tell this story?* We show two different explanations, given by two competing schemata: affiliation and stance. Our central questions: (1) *Which schema best explains the sources used in this story?* (2) *Can we predict, given a topic sentence, which schema to use?*

Schemas can also act as competing hypotheses to explain observed human actions (e.g., in Section 3.4, "balance opposing viewpoints" vs. "establish background"). In *source-finding* for example: why did the writer select sources $q_1, q_2, q_3...$ for document X? Let's suppose we observe an article on a controversial topic: some sources in the article "agree" with the main topic and others "disagree". Did the writer chose these sources on the basis of their *stance* [367] (or their opinion-based support)? Or is there another another explanation, like their *discourse* role (which describes their narrative function)? Each of these explanations can (and, as we will see, *will*) be operationalized with different schemas. Now we arrive at a fundamental problem: if we with to use schemas as both explanatory variables for human behavior and constructive biases for learning policies, how can we *know which schema is the "right" schema?* Schemas are typically latent: Rarely can we directly *observe* the schema categories a datum belongs to. Intuitively, if we are trying to use schema signatures using a schema that does not describe our data well, we might not be learning a *useful* $\hat{\pi}(\tau|x)$.

In this section, I will directly address these questions. I will introduce methods to compare schemas based on how well they *explain observed data*, inspired by on classical approaches to validating topic models [368].

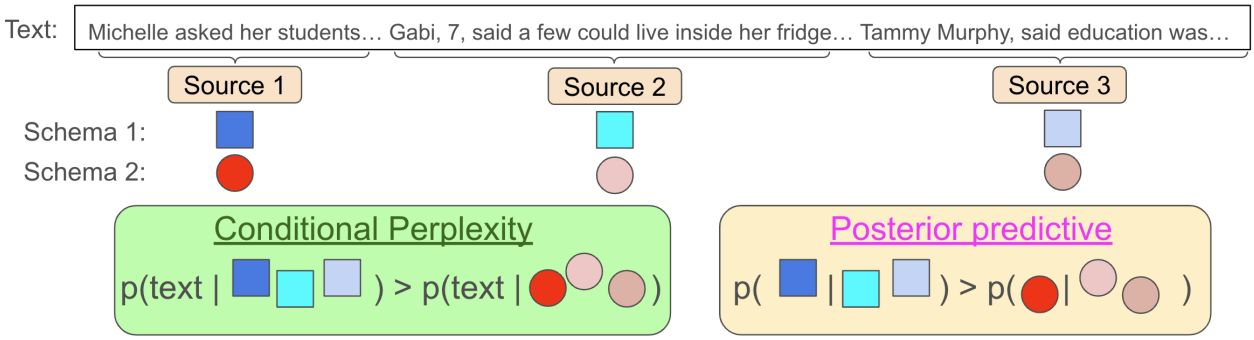


Figure 3.10: We seek to infer unobserved *plans*, or schemata, in natural data, focusing on one scenario: source-selection made by human journalists during news writing. Although the *reasons* why sources are chosen are unobservable, we show that one explanation (in the diagram, represented by *squares*: { , , }), is preferred over another (represented by *circles*: { , , }) if it better predicts the observed text (*conditional perplexity*) and the explanation is more internally consistent (*posterior predictive*). Our paper is divided into two parts: in the first part (i.e. Section 3.5.1.2 and Section 3.5.2.2), we introduce the different schemata we will compare – i.e. the top half of this diagram. In the first part (i.e. Section 3.5.3 and Section 3.5.4) we determine the right schema for a datum among competing schemata – i.e. the bottom half of this diagram – and, given minimal information about a document, we show that we can predict what schema *should* be used.

3.5.1 *Schema Criticism* as Latent-Plan Selection

To frame our methodology more directly in probabilistic graphical modeling terms [369], we describe *human source-finding* as a generative process. (The second step of this story might feel familiar; it is directly inspired by the *Planned Interleaved Retrieval* algorithm discussed in Section 3.4.3.2.)

1. First, journalists *plans* how they will choose sources, Q, for their story. They do so by selecting a *schema*, Z that describes which 1-of-k categories each sources will fall into.

2. *Then*, for each source to retrieve for the story, q_i, they sample 1-of-k categories, z_i, from their schema. They use this selection to drive what source they find.

First, to clarify our terminology: in this section, *we again use a specific and idiosyncratic definition for plan*. A *plan*, here, is a macro-level decision that governs a sequence of actions. The *plan* functions, as per our generative story, by specifying *how* actions are

categorized: this categorization drives how the journalist *selects* actions to perform. For example, in Figure 3.15, there are two possible plans shown that the journalist made before reporting (or, equivalently, there are two *schemas* that can categorize the sources they used[50]). An affiliation-based schema categorizes sources into institutional affiliation: "Academic", "Government", etc.; a stance-based schema categorizes sources as "Neutral", "Agree", etc. Each different plan is *possible*; each plan is specified by a different schema. To apply a schema to a document, we use our attribution function, α, from Section 3.2 (i.e. $\alpha(X_i) = q \in Q_X$ for $X_i \in X$) which, to recap, maps each sentence X_i in document X to a source $Q_X = \{q_1^{(X)}, ...q_k^{(X)}\}$[51] We also train classifiers, c, to assign a type $z \in Z$ from schema Z to each source:

$$c_Z(X_1^{(q)} \oplus ... \oplus X_n^{(q)}) = z \in Z \tag{3.1}$$

taking as input a sequence of sentences attributed to source $q^{(X)}$ (the full set of schemas we will consider are shown in Figure 3.11 and described in Section 3.5.1.2).

Typically, we note, when generative stories are told, as we have done, it's in the service of developing probabilistic graphical models (PGMs) to frame latent variable analysis [369]; PGMs are not usually learned with supervised classifiers. The standard *unsupervised* treatment of latent-variable PGMs learns the *assignments* z and the *semantics* of the latent space Z jointly[52]. These latent spaces often do not correspond well to theoretical schemata [371], on the other hand, supervised models trained on different schemata are challenging to compare. A latent-variable framework here is ideal: comparing different graphical

[50]Indeed, *plan* in this section is used interchangeably with *schema*, the only syntactic difference is that planning refers specifically to a-priori decision-making, while "applying the schema" refers to a-posteriori categorization on the part of researchers.

[51]These are all sources are referenced explicitly or implicitly in X. There is no consideration of sources *not* referenced in X (e.g. historical knowledge the journalist knew or background knowledge that the journalist obtained through other channels).

[52]For instance, for a model $p_\theta(x, z) = p_\theta(z)\, p_\theta(x \mid z)$, the EM algorithm [370] alternates between inferring posteriors z over latent states, $q(z \mid x, \theta) \approx p_\theta(z \mid x)$, and updating parameters, θ. The parameters θ determine $p_\theta(z)$ and the conditionals $p_\theta(x \mid z)$, thereby endowing each latent state with its "meaning". The data–wise assignments z then "choose" a particular latent state for each x by maximizing (or soft–weighting by) the posterior, e.g., $z^\star(x) = \arg\max_z\ p_\theta(z \mid x)$.

models [372, 373] *necessitates* comparing different schemata, as each run of a latent variable model produces a different schema. Methods [374, 368], to evaluate *which latent variable assignment describes the observed data the best*, give us an apples-to-apples approach for determining *which schema is better*.

3.5.1.1 Comparing Plans, or Schemata

We can compare plans in two ways: (1) how well do they explain each observed document? and (2) how structurally consistent are they?

Explainability A primary criterion for a *plan* is for it to explain the observed data well. To measure this, we use *conditional perplexity*[53]

$$p(x|z) \tag{3.2}$$

which measures the uncertainty of observed data, x, given a latent structure, z. Measuring $p(x|z)$ for different z (fixing x) allows us to compare z. Conditional perplexity is a novel metric we introduce, inspired by metrics to evaluate latent unsupervised models, like the "left-to-right" algorithm introduced by [368]. [54]

Structural Likelihood: A second basic criterion for a latent structure to be useful is for it be consistent, which is a predicate for learnability. We assess the consistency of a set of assignments, z, by calculating the *posterior predictive*:

$$p(z|z_{-}, x) \tag{3.3}$$

[376] exploring using full joint distribution, $p(z)$, *latent perplexity*, to evaluate the structure text x produced by generative language models ("*model criticism*"). We simplify using

[53] We abuse notation here, using p as both probability and perplexity: $p(x) = \exp\{-\mathbb{E} \log p(x_i|x_{<i})\}$.

[54] We note that the term, *conditional perplexity*, was originally introduced by [375] to compare machine-translation pairs. In their case, both x and z are observable; as such, they do not evaluate latent structures, and their usage is not comparable to ours.

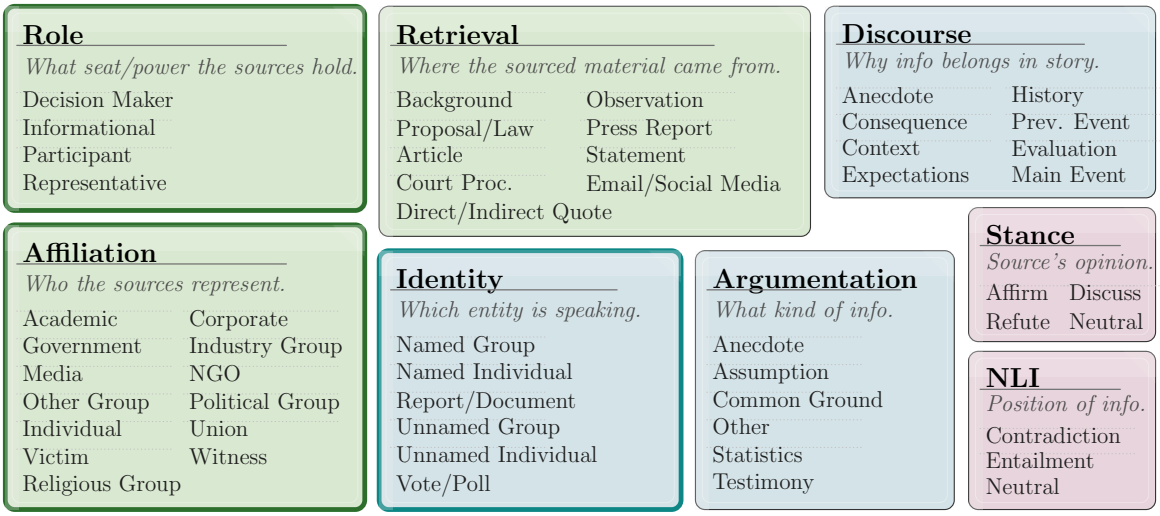


Figure 3.11: Label-sets for source-planning schemata. **Extrinsic Source Schemata** Affiliation, Role and Retrieval-method [1] capture characteristics of sources *extrinsic* to their usage in the document. **Functional Source Schemata:** Argumentation [377], Discourse [130] and Identity capture functional narrative role of sources. **Debate-Oriented Schemata**: Natural Language Inference (NLI) [309] and Stance [367] capture the role of sources in encompassing multiple sides. The three novel schemata we introduce are shown with borders: Affiliation, Identity and Role. For definitions, see [314].

the full distribution and instead evaluate the conditional predictive to study document structure. This, we find in early experiments, is easier to learn and thus helps us differentiate different Z better ("*schema criticism*").[55] Now, we describe our schemata.

For an illustration of each metric, please refer to Figure 3.10. The overall goal of the metrics is to determine *which schema, or labeling of sources, best explains the observed news article*. As the figure shows, if schema A describes an article better than schema B, then labels assigned to each source under schema A (e.g. in Figure 3.10: squares, ■, ■, ■) will outperform labels assigned under Schema B (e.g. circles, ●, ●, ●).

3.5.1.2 Source Schemata

Our schemata, or plans, are shown in Figure 3.11. We collect 8 schemata to compare, including three we introduce: *Identity, Affiliation and Role*. Each schema provides a set of

[55] Our work is inspired by [1]'s work, where z was the choice of specific source, rather than a general source-type. However, they had no concept of a "schema" to group sources.

labels, which each describe sources used in a news article. Again, our hypothesis is that the schema which *best predicts the observed text of the article* is the one the journalist most likely adhered to while planning the article (Section 3.5.3). See [314] for more details and definitions for each schema. We note that ***none*** of these schemata are complete and that real-world plans likely have elements outside of any one schema (or are combinations of multiple schema). However, this demonstration is important, we argue, to prove that we *can* differentiate between purely latent plans in long-form text. We now introduce each schema:

Debate-Oriented Schemata Both the *Stance* and *NLI* schemata are debate-oriented schemata. They each capture the relation between the information a source provides and the main idea of the article. *NLI* [309] captures factual relations between text, while *Stance* [367] captures opinion-based relations . A text pair may be factually consistent and thus be classified as "Entailment" under a *NLI* schema, but express different opinions and be classified as "Refute" under *Stance*. In our setting, we relate article's headline with the source's attributable information. These schemata say a writer uses sources for the purpose of expanding or rebutting information in the narrative, offering different perspectives and broadening the main idea.

Functional Source Schemata The following schemata: *Argumentation*, *Discourse* and *Identity* all capture the role a source plays in the overall narrative construction of the article. For instance, a source might provide a "Statistic" for a well-formed argument (*Argumentation* [377]), or "Background" for a reader to help contextualize (*Discourse* [130]). *Identity*, a novel schema, captures how the reader identifies the source. For example, a "Named Individual" is identifiable to a reader, whereas an "Unnamed Individual" is not. As identified in [378] and our journalist collaborators, this can be a strategic planning choice: some articles are about sensitive topics and need unnamed sources.

Extrinsic Source Schemata ***Affiliation***, ***Role*** **and** ***Retrieval*** schemata serve to characterize

Schema	Macro-F1	Schema	Macro-F1
Argumentation	68.3	Retrieval	61.3
NLI	55.2	Identity	67.2
Stance	57.1	Affiliation	53.3
Discourse	56.1	Role	58.1

Table 3.16: Classification F1, macro-averaged, for the 8 schemata. We achieve moderate classification scores for each of schema. When we compare schemata, we account for these differences by introducing noise to higher-performing classifiers.

attributes of sources external to the news article. They either capture aspect about how sources exist as entities in society (*Affiliation*, *Role*), or the informational channel through which is was retrieved (*Retrieval*). Stories often implicate social groups [379], such as "academia" or "government." Those group identities are extrinsic to the story's architecture but important for the selection of sources. Sources may be selected because they represent a group (i.e. *Affiliation*) or because their group position is important within the story's narrative (e.g. "participants" in the events, i.e. *Role*). *Retrieval*, introduced by [1], captures the channel through which the information was found. Although these schemata are news-focused, we challenge the reader to imagine ones that might exist in other fields. For instance, a machine learning article might compare models selected via, say, a *Community* schema: each from *open-source*, *academic* and *industry research* communities.

3.5.2 Building a Silver-Standard Dataset of Different Possible Plans

The schemata described in the previous section give us theoretical frameworks for identifying writers' plans. To *compare* schemata and select the schema that best describes a document, we must first create a dataset where informational sources are labeled *according to each schema*. We describe that process in this section.

3.5.2.1 Dataset Construction and Annotation

We use the *NewsEdits* dataset, discussed in Section 5.2, which consists of 4 million news articles, and extract sources using a methodology developed by [1], which authors established was state-of-the-art for this task. This dataset spans 12 different news sources (e.g. BBC, NYTimes, etc.) over a period of 15 years (2006-2021). For our experiments, we sample $90,000$ news articles that are long and contain more than 3 sources (on average, the articles contain ~ 7.5 sources). Then, we annotate to collect training data and build classifiers to categorize these sources. We recruited two annotators, one an undergraduate and the other a former journalist. The former journalist trained the undergraduate for 1 month to identify and label sources, then, they independently labeled 425 sources in 50 articles with each schema to calculate agreement, scoring $\kappa = .63, .76, .84$ on *Affiliation*, *Role* and *Identity* labels. They then labeled 4,922 sources in 600 articles with each schema, labeling roughly equal amounts. Finally, they jointly labeled 100 sources in 25 documents with the other schemata for evaluation data over 1 month, with $\kappa \geq .54$, *all in the range of moderate to substantial agreement* [380].

3.5.2.2 Training Classifiers to Label Sources

We train classifiers to label sources under each schema. Unless specified, we use a sequence classifier using RoBERTa-base with self-attention pooling, as in [145]. We deliberately chose smaller models to scale to large amounts of articles. We will open-source all of the classifiers trained in this paper.

Affiliation, Role, Identity We use our annotations to train classifiers which take as input all sentences attributable to source q and output a label in each schema, or $p(t|s_1^{(q)} \oplus ... \oplus s_n^{(q)})$.

Argumentation, Retrieval, Discourse We use datasets, without modification, that were directly released by the authors. Each is labeled on a sentence-level, on news and opinion datasets. We train classifiers to label each sentence of the news article, s. Then, for each

		Conditional Perplexity $p(x\|z)$			Posterior Predictive $p(\hat{z}\|z_{-}, x)$		
Schema	n	PPL	Δ base-k (↓)	Δ base-r (↓)	F1	÷ base-k (↑)	÷ base-r (↑)
NLI	3	22.8	0.62	-0.08	58.0	1.02**	1.01 **
Stance	4	21.5	-1.71	-3.21**	39.1	0.88**	0.83 **
Role	4	22.3	-0.06	-0.33**	38.7	1.11**	1.10 **
Identity	6	21.8	-0.42	-0.94	25.0	1.00	1.15 **
Argument.	6	21.7	-0.52	-1.04	30.7	1.10 **	1.12 **
Discourse	8	22.3	0.54	-0.75	19.2	1.06 **	1.08 **
Retrieval	10	23.7	1.47	0.36	15.8	1.10 **	1.12 **
Affiliation	14	20.5	-2.11**	-3.04**	10.5	1.26 **	1.16 **

Table 3.17: Results of comparing our schemata against each other. In the left results column, we show *conditional perplexity*, which shows how well each schema explains the document text. Shown is PPL (the mean perplexity per schema), $\Delta kmeans$ (PPL - avg. perplexity of kmeans) and $\Delta random$ (PPL - avg. perplexity of the random trial). Higher perplexities mean worse predictive power, so the more negative the Δ, the better. In the right results column, we show *posterior predictive*, measured via micro F1-score. We show F1 (f1-score per schema), ÷ kmeans (F1 / f1-score of kmeans), ÷ random (F1 / f1-score of random trial). Statistical significance ($p < .05$) via a t-test calculated over 500-sample bootstrapped f1-scores shown via **.

source q, we assign a single label, y, with the most mutual information[56] across sentences attributed to that source, $s_1^{(q)}, ...s_n^{(q)}$.

NLI, Stance We use an NLI classifier trained by [381] to label each sentence attributed to source q as a separate hypothesis, and the article's headline as the premise. We use mutual information to assign a single label. We create a stance training dataset by aggregating several news-focused stance datasets[57]. We then fine-tune GPT3.5-turbo[58] to label news data and label 60,000 news articles. We distill a $T5$ model with this data (Table 3.16 shows T5's performance).

[56] $\arg\max_y p(y|q)/p(y))$

[57] FNC-1 [382], Perspectrum [383], ARC [384], Emergent [385] and NewsClaims [386]. We filter these sets to include premises and hypothesis ≥ 10 words and ≤ 2 sentences.

[58] We use OpenAI's GPT3.5-turbo fine-tuning endpoint, as of November 16, 2023.

3.5.2.3 Classification Results

As shown in Table 3.16, we model schemata within a range of f1-scores $\in (53.3, 67.2)$, showing moderate success in learning each schema[59]. These scores are middle-range and likely not useful on their own; we would certainly have achieved higher scores with more state-of-the-art methods. However, we note *these classifiers are being used for comparative, explanatory purposes, so their efficacy lies in how well they help us compare plans*, as we will explore in the next section.

3.5.3 Comparing Schemata

We are now ready to explore how well these schemata explain source selection in documents. We start by describing our experiments, then baselines, and finally results. All experiments in this section are based on the $90,000$ news articles filtered from NewsEdits, labeled as described in the previous section. We split $80,000/10,000$ train/eval.

3.5.3.1 Implementing Planning Metrics

We now describe how we implement the metrics introduced in Section 3.5.1: (1) *conditional perplexity* and (2) *posterior predictive*.

Conditional Perplexity To measure *conditional perplexity*, $p(x|z)$, we fine-tune GPT2-base models [147] to take in it's prompt a sequence of latent variables, each for a different source, and *then assess likelihood of the observed article text*.[60] This is similar to measuring *vanilla perplexity* on observed text, except: (1) we provide latent variables as conditioning (2) by fixing the model used and varying the labels, *we are measuring the signal given by each set of different labels*. Our template for GPT2 is:

$\langle \texttt{h} \rangle\ h\ \langle \texttt{l} \rangle\ \texttt{(1)}\ l_1\ \texttt{(2)}\ l_2 \ldots \langle \texttt{t} \rangle$ $\texttt{(1)}\ s_1^{(q_1)} \ldots s_n^{(q_1)}\ \texttt{(2)} \ldots$

[59]When using these classifier outputs for evaluating plans, in the next section, we introduce noise (i.e. random label-swapping), so that all have the same accuracy.

[60]We note that this formulation has overlaps with recent work seeking to learn latent plans [376, 387, 219].

Red is the prompt, or conditioning, and green is the text over which we calculate perplexity. <tokens> (e.g. "(1)", "⟨text⟩") are structural markers while variables l, h, s are article-specific. h is the headline, l_i is the label for source i and $s_1^{(q_1)}...s_n^{(q_1)}$ are the sentences attributable to source i. *We do not use GPT2 for generation, but for comparative purposes, to compare the likelihood of observed article text under each schema.* We note that this implements Eq. 3.2 only if we assume green preserves the meaning of x, the article text. Our data processing (Section 3.5.2.1), based on our learned $\alpha(X_i)$ model (Section 3.2) gives us confidence in this.[61]

Posterior Predictive To learn the *posterior predictive* (Equation 3.3), we train a BERT-based classification model [285] to take the article's headline and a sequence of source-types *with a one randomly held out*. We then seek to predict *that* source-type, and evaluate using F1-score. Additionally, we follow [1]'s observation that some sources are *more important* (i.e. have more information attributed). We model the posterior predictive among the 4 sources per article with the most sentences attributed to them.

3.5.3.2 Baselines

Vanilla perplexity does not always provide accurate model comparisons [389, 390] because it can be affected by irrelevant factors, like tokenization scheme. We hypothesized that the dimensionality of each schema's latent space might also have an effect [391]; larger latent spaces tend to assign lower probabilities to each point. Thus, we benchmark each schema against baselines with similar latent dimensions.

Base-r, or Random baseline. We generate k unique identifiers[62], and randomly assign one to each source in each document. k is set to match the number of labels in each schema.

[61] Initial experiments show that text markers are essential for the model to learn structural cues. However, they also provide their own signal (e.g. on the number of sources). To reduce the effects of these artifacts, we use a technique called *negative prompting* [388]. Specifically, we calculate perplexity on the *altered* logits, $P_\gamma = \gamma \log p(x|z) - (1-\gamma) \log p(x|\hat{z})$, where $\hat{z}$ is a shuffled version of the latent variables. Since textual markers remain the same in the prompt for z and $\hat{z}$, this removes markers' predictive power.

[62] Using MD5 hashes, from python's uuid library.

Base-k, or Kmeans baseline. We first embed sources as paragraph-embeddings using Sentence BERT [221] [63] Then, we cluster all sources across documents into k clusters using the kmeans algorithm [392], where k is set to match the number of labels in the schema being compared to. We assign each source it's cluster number.

3.5.3.3 Results and Discussion

As shown in Table 3.17, the supervised schemata mostly have have lower conditional perplexity than their random and unsupervised kmeans baselines. However, only the *Stance*, *Affiliation* and *Role* schemata improve significantly (at $p < .001$), and the *Role* schema's performance increase is minor. *Retrieval* has less explainability relative to it's baselines. There is a simple reason for why some schemata have either the same or more conditional perplexity compared to their baselines: they lack explainability over the text of the document, but are not random and thus might lead to overfitting. We examine examples and find that *Retrieval* does not impact wording as expected: writers make efforts to convey information similarly whether it was obtained via a quote, document or a statement. We face a dilemma: in generating these schemata, we chose *Retrieval* because we assumed it was an important planning criterion. However, our results indicate that it holds little explanatory power. *Is it possible that some plans do not get reflected in the text of the document?* To address this question, we assign $\hat{Z} = \arg\min_Z p(x|z)$, the schema for each datapoint with the lowest perplexity, using scores calculated in the prior section[64], we calculate the lowest-perplexity schema. Table 3.20 shows the distribution of such articles. We then task 2 expert journalists with assigning their *own* guess about which schema best describes the planning for the particular article, for 120 articles. **We observe an F1-score of 74, indicating a high degree of agreement.**

Interestingly, we also observe statistically significant improvements of kmeans over random baselines in all cases (except $k = 3$). In general, our baselines have lower variance

[63] Specifically, `microsoft/mpnet-base`'s model `https://www.sbert.net/docs/pretrained_models.html`.
[64] across the dataset used for validation, or 5,000 articles

in perplexity values than experimental schemata. This is not unexpected: as we will explore in the next section, we expect that some schemata will best explain only some articles, resulting in a greater range in performance. (For more detailed comparisons, see [274]). Posterior predictive results generally show improvement across trials, with the *Affiliation* trial showing the highest improvement over both baselines. This indicates that most tagsets are, to some degree, internally consistent and predictable. *Stance* is the only exception, showing significantly lower f1 than even random baselines. This indicates that, although Stance is able to explain observed documents well (as observed by it's impact on conditional perplexity), it's not always predictable how it will applied. Perhaps this is indicative that writers do not know a-priori what sources will agree or disagree on any given topic before talking to them, and writers do not always actively seek out opposing sides. Finally, as another baseline, we implemented latent variable model. In initial experiments, it does not perform well. We show in [314] that the latent space learned by the model is sensible. Bayesian models are attractive for their ability to encode prior belief, and ideally they would make good baselines for a task like this, which interrogates latent structure. However, more work is needed to better align them to modern deep-learning baselines.

Summary In EL, we infer human trajectories with $q_\theta(\tau \mid g)$ and train $\pi(\tau \mid x)$ to emulate *structured* regularities of those trajectories. Discourse schemata give us explicit abstractions Z at which to compare humans and policies: they define signatures $\sigma_Z(\tau)$ and sequence regularities that we can penalize when $\pi(\tau \mid x)$ deviates. However, until this work there were no good ways to choose the *best* discourse schema. With *conditional perplexity* and *posterior predictive*, introduced in this work, we now have estimators to choose among competing schemata, allowing us to effectively learn how to choose planners to trust in a given context. This ties the theoretical role of schemas (as cognitive scaffolds) to concrete levers in planning and executing.

Newspaper Sections	Proportion of Sources with each Label		
Arts	Individual: 0.29	Media: 0.19	Witness: 0.17
Automobiles	Corporate: 0.41	Witness: 0.17	Media: 0.11
Books	Individual: 0.26	Media: 0.19	Witness: 0.18
Business	Corporate: 0.51	Government: 0.2	Industry Group: 0.06
Dining and Wine	Witness: 0.28	Individual: 0.18	Media: 0.17
Education	Government: 0.36	Academic: 0.19	Witness: 0.1
Front Page	Government: 0.5	Political Group: 0.09	Corporate: 0.08
Health	Government: 0.33	Academic: 0.19	Corporate: 0.12
Home and Garden	Individual: 0.21	Witness: 0.19	Corporate: 0.17
Job Market	Corporate: 0.26	Individual: 0.15	Witness: 0.14
Magazine	Witness: 0.23	Media: 0.2	Individual: 0.18
Movies	Individual: 0.28	Media: 0.18	Witness: 0.18
New York and Region	Government: 0.36	Witness: 0.13	Individual: 0.12
Obituaries	Government: 0.18	Individual: 0.18	Media: 0.16
Opinion	Government: 0.43	Media: 0.14	Witness: 0.12
Real Estate	Corporate: 0.33	Government: 0.21	Individual: 0.12
Science	Academic: 0.4	Government: 0.19	Corporate: 0.1
Sports	Other Group: 0.38	Individual: 0.15	Witness: 0.14
Style	Individual: 0.23	Witness: 0.2	Corporate: 0.17
Technology	Corporate: 0.41	Government: 0.17	Academic: 0.09
The Public Editor	Media: 0.44	Individual: 0.16	Government: 0.16
Theater	Individual: 0.34	Witness: 0.18	Media: 0.14
Travel	Witness: 0.25	Corporate: 0.21	Government: 0.15
U.S.	Government: 0.44	Political Group: 0.12	Academic: 0.08
Washington	Government: 0.6	Political Group: 0.1	Media: 0.08
Week in Review	Government: 0.37	Academic: 0.11	Media: 0.1
World	Government: 0.54	Media: 0.09	Witness: 0.09

Table 3.18: Distribution over source-types with different *Affiliation* tags, by newspaper section. Evidence that a distributional and composite view on *source-finding* has validity.

3.5.4 Using Schemata Prediction for Explanations

Taken together, our observations from (1) Section 3.5.2.3) indicate that schemata are largely unrelated and (2) Section 3.5.3.3 indicate that *Stance* and *Affiliation* both have similar explanatory power (although *Stance* is less predictable). We next ask: which kinds of articles are better explained by one schema, and which are better explained by the other? If we can answer this question, we take steps towards being able to *plan* source-selection via different schemata. Such a step could lead us towards better *multi-document* retrieval techniques, by giving us axes to combine different documents into a retriever.

In Table 3.19, we show topics that have low perplexity under the *Stance* schema, compared with the *Affiliation* schema (we calculate these by aggregating document-level perplexity across keywords assigned to each document in our dataset). As we can see, topics requiring greater degrees of debate, like "Artificial Intelligence", and "Taylor Swift" are favored under the *Stance* Topic, while broader topics requiring many different social perspectives, like "Culture" and "Freedom of Speech" are favored under *Affiliation*. We set up an experiment where we try to predict $\hat{Z} = \arg\min_Z p(x|z)$, the schema for each datapoint with the lowest perplexity. We downsample until assigned schemata, per articles, are balanced and train a simple linear classifier[65] to predict $\hat{Z}$. We get .67 ROC-AUC (or .23 f1-score). These results are tantalizing and offer the prospect of being able to *better plan source retrieval* in computational journalism tools, by helping decide an axis on which to seek different sources. More work is needed to validate these results.

Summary In conclusion, we explore ways of thinking about sourcing in human writing. We compare 8 schemata of source categorization, and adapt novel ways of comparing them. We find, overall, that *affiliation* and *stance* schemata help explain sourcing the best, and we can predict which is most useful with moderate accuracy. Our work lays the ground work for a larger discussion of discovering plans made by humans in naturally

[65] Bag-of-words with logistic regression

Stance	*Affiliation*
Bush, George W	Freedom of Speech
Swift, Taylor	2020 Pres. Election
Data-Mining	Jazz
Artificial Intelligence	Ships and Shipping
Rumors/Misinfo.	United States Military
Illegal Immigration	Culture (Arts)
Social Media	Mississippi

Table 3.19: Top keywords associated with articles favored by stance or affiliation. Keywords are manually assigned by news editors.

Affiliation	41.7%
Identity	22.7%
Stance	17.7%
Role	13.4%
Argument.	1.2%
Discourse	1.1%
NLI	1.1%
Retrieval	1.1%

Table 3.20: Proportion of our validation dataset favored by one schema, i.e. $\hat{Z} = \arg\max_Z p(x \mid z)$.

generated documents. It also takes us steps towards tools that might be useful to journalists. Naturally, our work is a simplification of the real human processes guiding source selection; these categories are non-exclusive and inexhaustive. We hope by framing these problems we can spur further research in this area.

3.6 After Source-*Finding*: A System to *Obtain Information* from Sources

So far, we have formalized *source-finding* as a creative retrieval problem within Emulation Learning. We started, in Section 3.2, infer latent sourcing trajectories (τ) from observable articles (g) with $q_\theta(\tau \mid g)$. We showed that sourcing is compositional and predictable, in Section 3.2.3; evidence we used in Section 3.3 to critique policy models learned implicitly during pretraining, $\pi^{(llm)}(\tau|x)$. We proposed our own policy models, in Section 3.4, based on enforcing distributional similarities, or *schema signatures*, and finally, we introduced methods to critique schemas, in Section 3.5.

We close this Chapter on *source-finding* with a more light interlude. We again return to the notion, outlined in Section 3.2, of what an *action*, a, is in the *source-finding* task. The most naive version is: a encompasses all that is needed to *identify*, *find*, and *obtain information from* a source. While, in Section 3.4, we split apart *identifying* from *finding* processes[66], we still assumed that *obtaining* information from sources was trivial. What if that is not the case? What if an LLM had to actually interact with sources, in a dialogue setting to extract information from them? In this section, we introduce a *dialogic subtask after* the source is found with the goal of *obtaining* quotes and facts to satisfy each *source-finding* macro-action a_t. Concretely, in this section, we view interviews as sub-trajectories $\tau_{\text{interview}} \subset \tau$ with actions $a_{t1}, a_{t2}...$, as conversational dialogue and states $s_{t1}, s_{t2}...$ as usable information that is obtained from the source and is later published in the article, g. We introduce a *NewsInterview* game, shown in Figure 3.12, to incorporate *emulation learning* in order to learn $\pi(\tau_{interview}|x)$, a proper conversational policy.

[66]Recall, by splitting the *planner* from the *query-executor*, in a hierarchical setting.
[67]The high-level objectives the LLM agent starts with are similar to a journalist's pre-interview notes

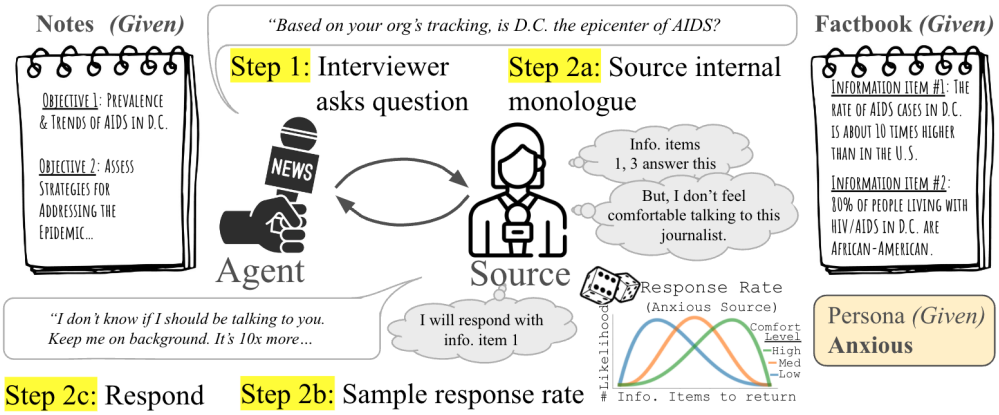


Figure 3.12: **Walkthrough of the the NewsInterview game**. An interviewer-LLM converses with a source-LLM: the interviewer-LLM is rewarded based on how many information items (shown at the right) are extracted from the source. The interviewer agent is given a set of high-level objectives[67] while the source is given a persona and a set of information items. *For k turns*: **Interviewer:** asks a question based on their goals and information obtained (Step 1). **Source:** responds with a multi-step process. First, they determine how many information items in their fact-book are relevant to the question (Step 2a). Then, they assess their comfort level. Depending on this, we randomly sample a subset of relevant information to respond with (Step 2b). The source is then prompted to craft a reply aligned with their persona (Step 2c). *After k turns:* we track on the back-end which items the source responded with and give this number as a reward to the interviewer.

3.6.1 Grounding Challenges in Human-LLM Dialogues

Before we discuss the *NewsInterview* game in more detail, we discuss the challenges that prevent pretrained LLMs from implicitly learning interviewing policies, $\pi^{(llm)}(\tau_{interview}|x)$. Large Language Models (LLMs) have demonstrated impressive capabilities in generating coherent text but often struggle with *strategic* [393] or *emotional* dialogue [394]. For example, [394] examined LLM-generated responses to dialogues and found fewer occurrences of "grounding language" [395, 396], like acknowledgments or affirmations, that humans typically use to foster comfort and trust. From an Emulation Learning perspective, these observations indicate underfitting of $\pi^{(llm)}$ to the human behavioral prior π^* in long-horizon interaction, rather than a simple modeling deficit at the token level. This can impede an LLM's ability to serve in a variety of situations: e.g., education [397], mental health [398] or conflict resolution [399]. However, prior efforts to ameliorate such gaps face

limitations: existing large datasets (1k–10k transcripts) are generated via crowdsourcing and are inherently unnatural [400, 401, 402]. More natural datasets, of educational [403] or therapeutic environments [404], are difficult to collect due to privacy concerns [405] and are small-scale (100–1k transcripts). Journalist interviews, typically conducted between an "interviewer" and a "source," with the goal is to obtain information, *also* have extensive need for grounding. Sources are often anxious or unclear [406], and human interviewers are constantly evaluating: (1) Are my questions getting fully addressed? (2) Do I need to more effectively engage or persuade a source [343]?

To study how to develop optimal policies $\pi^*(\tau|x)$ in journalistic contexts, we start by collecting interview transcripts from two major US news sources: National Public Radio (NPR) and Cable News Network (CNN), filtering to over 40,000 dyadic informational interviews.[68] As in prior sections, we frame can this in an *emulation learning* lens. Taking human interviews as the goal-state, g, we study the strategies of the human interviewer and find that pretrained LLMs suffer from the same lack of grounding as in other dialogue settings [394]. We find that significant discourse differences exist in the kinds of questions asked by LLMs: for example, LLMs are 50% less likely to make acknowledgments, and 30% less likely to pivot to higher-level questions.

Motivated by these observations, we develop a realistic game environment to serve as a playground: in this simulation, LLMs play the role of the interviewer and the source. The goal for the interviewer is to *obtain the maximal amount of information from the source in a limited number of questions*. In order to induce the need for grounding communication, we design different personas for sources (e.g., anxious, clueless, dominating), each with different communication patterns. We also add a responsiveness to strategic dialogue: sources will only return information if they are persuaded in a manner befitting their personas[69] [406, 343]. We find that our environment is realistic: source-LLMs correlate

[68] As opposed to games, questionnaires and other formats these news outlets release.

[69] We understand that "being persuaded," "being made comfortable," and "being acknowledged" are all separate forms of grounding, some more active than others. However, we use "persuasion" as a short-hand encompassing all categories.

significantly with humans in their ability to identify persuasion ($r = .43$, $p < .0001$). However, interviewer-LLMs struggle to both recognize when questions are answered and actively persuade the source, resulting in suboptimal information extraction.

3.6.2 Dataset Processing

3.6.2.1 Data Collection

We aggregate, clean and condense multiple publicly available datasets of interview transcripts from NPR and CNN in order to build a high-quality interview dataset of 45k source-to-interview transcripts. These transcripts are published records of live interviews conducted between a journalist and sources invited on the program. They provide a rich resource for analyzing natural language interactions.

3.6.2.2 Data Filtering for Interview Analysis

We want to focus on one-on-one informational interviews between a journalist and a single source. We start with 487,310 transcripts collected by Majumder et al. [407] and Zhu et al. [408]. However, initial examination of the transcripts reveals many of them to be low-quality: they include multiple sources, are formatted as panel discussions, or are not informational in nature (e.g., they include game shows). To filter the transcripts and retain only those that fit our criteria, we prompt Llama-3.1-70b[70] to classify each transcript based on the number of participants and the nature of the content. The prompts used for filtering are provided in [275]. After filtering, 45,848 interviews remain. Finally, the original transcripts do not distinguish which participant was the interviewer vs. the interviewee. So, we count each participant's use of question marks: the participant with more is labeled the interviewer.[71] We treat each validated interview as a trajectory τ^* composed of *grounding* decisions.

[70] `https://huggingface.co/meta-llama/Meta-Llama-3.1-70B-Instruct` [409] using the vLLM framework [410]

[71] Manual validation on 50 interviews showed this method correctly identified roles in $> 98\%$ of cases.

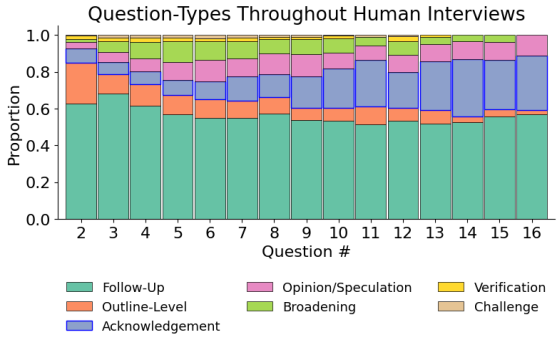
	EM	Info.	Motivation	Style	Discourse	Context
Baseline-LLM	3.9%	4.4%	4.7%	11.9%	36.2%	53.0%
Chain-of-Thought (CoT)	4.5%	3.6%	5.2%	12.8%	37.0%	56.9%
LLM w. Outline	3.7%	3.8%	4.1%	9.6%	36.2%	46.6%
Outline-CoT	3.6%	3.9%	4.3%	8.3%	29.9%	43.1%
Human	8.2%	17.5%	35.4%	40.2%	54.5%	60.3%

Table 3.21: **Discourse-Level Alignment of LLM-Generated Questions with Human Interview questions**. We give an LLM, `Llama-3.1-70b`, the prior $t-1$ turns in an interview and prompt it to ask the next question. We measure the percentage of times this question aligns to a question asked by a human at the same point in the interview across six dimensions: Exact (nearly exactly the same as the original utterance), Information (relevant factual content), Motivation (same motivation as the original question), Style (alignment with tone and phrasing), Discourse (structural role within the interview), and Context (incorporation of contextual knowledge). The prompting strategies compared are Baseline-LLM, Chain-of-Thought (CoT), LLM with an Outline, and Outline-CoT; and, we conduct a human baseline trial with a former professional journalist.

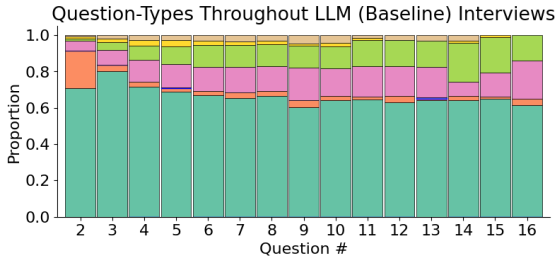
Conversations have, on average, 7.5 turns between the interviewer and source. The source speaks for longer, with an average of 551 words per conversation compared with the interviewer's 270 words (or 27 words per source utterance, 16 per interviewer). Interviewers tend to ask "what" and "how" questions the most, and conversations occur at Flesch-Kincaid Grade of 6.9 [411]. Interviews cover a range of topics, from literature, politics, academics, and international affairs (see [275]).

3.6.3 Analysis

In this section, we analyze how humans conduct informational interviews and compare this behavior to that of pretrained LLMs, to explore whether LLMs face similar grounding problems as observed in other settings [395, 412]. We keep the conversation history C and take the per-turn state as $x_t \equiv C_{t-1}$. Our approach is a combination of approaches taken in Sections 3.3, 3.4 and 3.5 in that (1) we compare human-expert policies $\pi^*(\tau|x)$ with implicitly learned policies $\pi^{(llm)}(\tau|x)$ (2) we take a distributional analysis approach,



(a) **Proportion of Discourse types throughout human interviews**. Human journalists use different discourse roles across the interview, including gradually more Acknowledging statements, increasing from 5% at the start to over 20% by the end.



(b) **Proportion of Discourse types of LLM responses in interviews**. LLMs display an increasing likelihood of asking opinion or broadening questions over the course of an interview and a lower likelihood of returning to outline-level questions.

Figure 3.13: Comparison of discourse types *throughout* an interview (the first turn, usually a greeting, is excluded). The LLM is shown the first $t-1$ turns of a human interview and asked to generate the next question.

generating discourse schemata and comparing schematic signatures. I describe these now.

3.6.3.1 Generating Counterfactual Utterances

One way to assess how an LLM would behave in an interview setting offline is to perform a counterfactual simulation [412]. Specifically, given a human interview consisting of at least t interviewer-source conversational turns $(u_1^{(I)}, u_1^{(S)})...(u_t^{(I)}, u_t^{(S)})...$, we feed $t-1$ turns into the LLM along with a prompt instructing the LLM to generate the next question. This generates a counterfactual, $\widehat{u_t^{(I)}}$ to what the human would have said, $u_t^{(I)}$; and yields a one-step, offline probe of $\pi^{(llm)}(\tau|x)$ against human reference moves, providing per-step emulation errors. We experiment with different variations: (1) **Baseline**: The LLM is simply asked to produce the next question. (2) **Chain-of-Thought (CoT)**: The LLM is instructed to reason about the information already provided in the interview, consider what might be left to ask, and then generate the next question. (3) **Outline**: the LLM is provided with an outline of the interview goals (described in Section 3.6.4.2) to incorporate

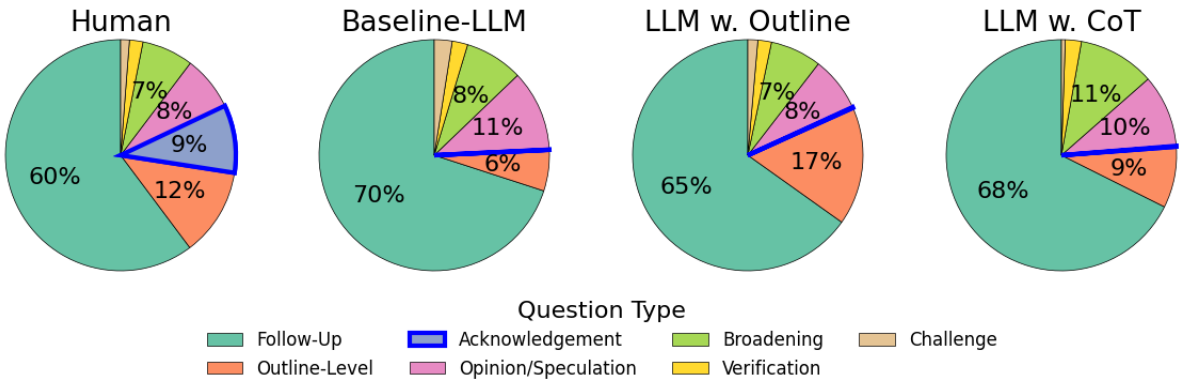


Figure 3.14: **Distribution of Discourse Roles in Questions, Across Different Prompting Strategies**. We compare the proportions of discourse roles of questions (e.g., FOLLOW-UP, ACKNOWLEDGMENT) generated by (a) human journalists, (b) Baseline-LLM (`Llama-3.1-70b`) (c) LLM prompted with an Outline and (d) with Chain-of-Thought (CoT). Acknowledgment statements, which often build empathy, are significantly underrepresented in LLM prompting approaches, compared to human-generated questions (see [275] for Outline-CoT).

into CoT reasoning.[72]

3.6.3.2 Evaluating LLM Counterfactuals

To analyze how similar LLM questions are to human questions, we perform two analyses:

Consistency Analysis: We aim to assess how similar g_t is to q_t across different comparison categories [413], specifically: *Informational* consistency (i.e., g_t and q_t seek similar informational objectives); *Motivational*, (i.e., similar outcomes); *Style*, (i.e., similar tone); *Contextual* consistency (i.e., similar appropriateness given the context); *Discourse* consistency (i.e., similar purposes in the overall conversation). Putting these together, we assess an $Exact$ match. We ask an LLM, `GPT-4o`, to perform this assessment and manually inspect its outputs and reasoning threads.

Discourse Analysis: We aim to assess whether g_t plays a similar *function* as q_t does. We develop a schema to describe the role of each question.[73] This schema includes the following

[72] We include full prompt examples for all three variations in [275]. All question-generation experiments are conducted using `Llama-3.1-70b`.

[73] To generate our discourse schema, we asked two journalists to analyze fifty interview transcripts. One had eight years of experience in newsrooms, the other was an undergraduate student studying journalism. We held three conferencing sessions to develop the schema. Then, we blindly annotated ten interviews,

elements: Follow-up Question (e.g., "Can you tell us more?"), Outline-Level Question (e.g., "Moving on, can we discuss the next event?"), Acknowledgment Statement (e.g., "I see, that sounds scary."), Opinion/Speculation (e.g., "What do you think will happen?"), Broadening Question (e.g., "How does this fit into the broader trend?"), Verification Question (e.g., "So to confirm...") and *Challenge Question* (e.g., "These dates don't line up."). See [275] in the Appendix for definitions of each role. This comparison mirrors our schema-signature approaches in Sections 3.4 and 3.5 and applies at the trajectory level.

3.6.3.3 Findings

Insight #1: Acknowledgment statements are virtually absent from all LLM variations. As shown in Figure 3.14, grounding gaps exist in journalistic interviewing similar to those observed by Shaikh et al. [394]. While human journalistic interviewers tend to make Acknowledgment statements in about 9% of their utterances, all prompting variations that we experimented with made close to zero of these statements. This lack of acknowledgment is paired with not mirroring the source's speaking style; human journalists, as shown [275], bring character and voice.

Insight #2: LLMs do not engage in strategic multi-turn questioning. Even in settings where LLMs are exposed to interview outlines, they are still undirected in their questions. As shown in Figure 3.14, LLMs are significantly more likely to ask follow-up questions than humans across all prompting variations. Introducing chain-of-thought and outline variations increases the rate at which the LLM asks outline-level questions. However, the rate remains significantly below human levels. Additionally, they are also more likely to ask either Opinion questions or Broadening questions. In fact, in Figure 3.13b, we observe that LLMs tend to ask increasing amounts of Opinion Questions and Broadening Questions

achieving a $\kappa = .6$. Given our schema, we then asked an LLM to classify discourse roles in sentences. The prompt contains the interview context, $(u_1^{(I)}, u_1^{(S)})...(u_{t-1}^{(I)}, u_{t-1}^{(S)})$, and current question $u_t^{(I)}$. To validate the LLM's labeling accuracy, we had the professional journalist label 10 additional interviews as ground-truth and scored the LLM's assignments. The LLM scored a .8 f1 score.

over time, which humans do not. These questions can be vague and open-ended. Together, these findings suggest an inability to direct an interview in a desired direction and engage in multi-turn planning.

Insight #3: LLMs are capable of understanding context, but fail in other categories of similarity to humans. Comparing the content and style of LLM interviews to human interviews in Table 3.21, we note that, overall, LLMs are broadly dissimilar to humans in style, motivation and information-seeking. One area where the LLMs succeed, relatively, is understanding the context of the interview beforehand. This is not a new observation – much recent work, e.g., in dialogue-tracking, has found LLMs to perform well [414]. The fact that LLMs can preserve context over multiple turns and do not drift away from the topic indicates that models might *one day* be able to engage in multi-turn goal-oriented dialogue, given the right reward signals and learning environment. Taken together, these findings suggest that journalistic dialogue is suitable for studying effective communication patterns, and also highlight significant gaps in current language modeling objectives. While LLMs can generate contextually relevant questions, they lack both an emotional and connective drive as well as the strategic planning exhibited by human interviewers.

3.6.4 NewsInterview: An Interview Game

As shown, LLM counterfactual questions exhibit several shortcomings: they are less likely to acknowledge the interviewee and focus excessively on follow-up questions. But do both of these shortcomings point to a lack of strategic multi-turn planning? In human dialogue, grounding exists for long-term strategic purposes [415], yet there currently exists no way to way to obtain these kinds of long-term rewards during LLM training. Motivated by this insight, our goal for the remainder of the paper is to create and validate a realistic game-environment with a delayed reward signal. We leave to future work utilization of this framework for improving strategic dialogue.

Algorithm 1 Game-play. States $x_t \equiv C_{t-1}$; terminal return $R(g) = |U_K|$.

Input Interviewer objectives $o \equiv \nu(g)$[74], Source Informational Items I, Source persona ϕ, K turns

Output Reward R

Initialize: Reward $R \leftarrow 0$, Conversation History $C \leftarrow []$, Used items $U \leftarrow \{\}$
for $i \in 1, ...K$ **do**
▷ **Step 1: Interviewer Question Generation**
    $u_i^{(I)} = \texttt{Interviewer}(C, o)$
▷ **Step 2: Source's Response Generation**
    $E_i = \texttt{getRelevantInfoItems}(I, U, u_i^{(I)})$
    $p_i = \texttt{getPersuasionLevel}(C)$
    $F_i = \texttt{getItemsToReturn}(E_i, p_i)$
    $u_i^{(S)} = \texttt{Source}(u_i^{(I)}, C, F_i, p_i, \phi)$
▷ **Update Variables**
    $U \leftarrow U \cup F_i$, $C \leftarrow C \oplus \left[u_i^{(I)}, u_i^{(S)}\right]$, $R \leftarrow R + |F_i|$
end for

3.6.4.1 Game Design Overview

We first introduce our game on a high level, illustrated in Figure 3.12, and then describe our implementation. Our game-play proceeds in a loop, shown in Algorithm 1. The "player" in our game plays the role of an interviewer and is able to ask questions to a source, based on the conversational history and the interview objectives (the `Interviewer()` step). The source is given a set of informational items and assesses whether any of these items are relevant to the question (the `getRelevantInfoItems()` step); the source then decides how persuaded or comfortable they are based on the conversational history (the `getPersuasionLevel()` step). Based on this, we determine the subset of relevant items the source returns (the `getItemsToReturn()`), and track these on the back-end as an accumulating reward. The reward, obtained at the end of the game, is the unique number of information items disclosed. We take $x_t \equiv C_{t-1}$ (state), latent factors ζ (e.g., source characteristics), and terminal return $r(g) = |U_K|$. The environment therefore supplies both trajectories and a controlled, delayed-return setting to stress-test $\hat{\pi}(\tau \mid x)$.

3.6.4.2 Game-play Design

To design our game, we draw heavily on two journalism textbooks: *Interviewing: A Guide for Journalists and Writers*, which explains how to conduct effective interviews and speak to reluctant, defensive, or poor-explaining sources [343]; and *Journalism: Principles and Practice*, which describes how to build trust [406]. We first start by describing our data processing, and then we will describe Algorithm 1 in more detail. For all game-play prompts, see [275].

Dataset Preparation for Simulation To prepare our dataset for use in the simulated game environment, we group together: (1) source responses and ask an LLM.[75] to summarize a set of *specific informational items* and (2) interviewer questions and ask an LLM to summarize them into a set of *high-level objectives*. The sources' informational items mimic the knowledge a source likely had going into the interview[76] and the interviewer's objectives represent the agendas they had prior to the conversation.[77] Both of these summaries are represented in Figure 3.12 as *Given*, and are designed to give the interviewer-LLM and the source-LLM a basis for communication. For further examples of both, see [275].

Source Design Element #1: Personas Now, we introduce the design of the source. We focus attention on this construction to build a robust game environment that accurately mimics human interactions. To make game-play varied and challenging, we draw from Sedorkin [343] to design eight different personas: *Anxious*, *Avoidant*, *Adversarial*, *Defensive*, *Straightforward*, *Poor Explainer*, *Dominating* and *Clueless*. For descriptions of each persona, as well as example responses, see [275]. These personas allow us to study how interviewers perform in a wider array of challenging scenarios.

Source Design Element #2: Persuasion The following three functions, in sequence, power our game-play: `getRelevantInfoItems` → `getPersuasionLevel` → `getItemsToReturn`. The

[75] Llama-3.1-70b

[76] Manual evaluation confirms these information items are present in initial interviews and are non-overlapping.

[77] Manual validation with professional journalists confirms that these outlines reasonably capture what a journalist might prepare before an interview and do not leak information.

first, `getRelevantInfoItems`, takes the interviewer's question and determines which of the sources' information items are most relevant; it is simply a retrieval function that we implement using an LLM. `getPersuasionLevel` is a function that determines the selected source's level of comfort or persuasion (on a five point scale) in the current conversation. `getItemsToReturn` is a stochastic engine: it randomly selects, based on the persuasion level, the number of relevant information items to return: *the more persuaded a source is, the more likely they are to return more information.* The persuadability component to our game-play increases the multi-turn strategy: because persuasion is assessed with reference to the entire interview, the interviewer gets more reward for spending words *early* in the interview persuading the source to feel comfortable. Because key drivers of disclosure are only partially observed, the setting is naturally partially observable; this supports, in the future, extending inverse inference $q_\theta(\tau|x)$ to recover auxiliary information to describe persuasiveness.

Is it sound for the source-LLM to assess its own level of persuasion? As recent research has found, LLMs are poor detectors of when they are being persuaded [416] and can even unknowingly persuade themselves [417]. Furthermore, persuadability varies from person to person [401, 418]. Luckily, source-persuasion is a well-studied field in journalism. As a starting point, we draw from Sedorkin [343], and carefully design prompts asking an LLM to rate the persuasiveness of a prior conversation. Different source personas, according to Sedorkin [343], are persuaded by different communication patterns: e.g., *Anxious* sources are distrustful of journalists; they are usually persuaded by phrases like "I will be as fair as possible." We validate this in Section 3.6.4.3.

Source and Interviewer Responses Based on the assessed persuasion level (1–5) of the conversation, we implement `getItemsToReturn`. This function takes in all relevant information items and randomly draws from a Beta distribution to determine what percentage of relevant information items to return. We choose five different parameterizations per persona, each corresponding to a different persuasion level. As can be seen in Figure 3.12,

we choose these parameterizations such that the more persuaded a source is, the more left-skewed the distribution is. Each persona has a slightly different parameterization, reflecting that some personas need less persuasion (e.g., "Dominant") while others do not drastically change how much information they return even with more persuasion (e.g., poor explainer). See [275] the Beta distributions for each source.

3.6.4.3 Game-play Validation

We conducted human trials to validate how well our game-play environment approximates real interviews, focusing on *persuasion* as a pivotal dimension. Five participants, including two professional journalists and one journalism student, each served as the "source," rating their own persuasion levels turn-by-turn on a five-point scale across 72 trials (576 turns total). The game's LLM-based source also generated persuasion estimates. We found a moderate but significant correlation of $r = 0.43$ (p < .0001). Excluding adversarial personas, correlation rose to $r = 0.68$. Bootstrapped estimates confirmed the consistency of these results, and a power analysis following guidelines from [419] showed our sample size was adequate to detect this effect.

These trials center on persuasion because the other components of our source design (i.e., retrieval of correct informational items), while crucial, leverage prior, well-studied phenomena in retrieval-augmented LLMs and prompt engineering [420, 421]. Our environment reuses standard cross-encoder reranking and chain-of-thought prompts [422, 423], meaning that the correct factual content is generally well-handled without substantial new techniques. Minimal forms of self-reflection [424, 425] were used to mitigate hallucinations, and no significant factual drift was observed. Hallucinations are well-studied in the literature [426].

Taken together, this validation suggests that modeling source *persuadability* in a turn-level simulation is reasonably accurate and stable. By capturing how LLMs adapt their strategies across different personas and persuasion thresholds, our system can potentially

Model	**Hardest** Full Game	**Medium** *sans.* Persuasion	**Easiest** *sans.* Info. withholding
gpt-4o-mini	49.3%	47.5%	84.7%
gpt-4o	50.4%	49.8%	84.2%
Llama-3.1-70b	42.6%	45.5%	80.1%
Llama-3.1-8b	42.4%	48.3%	74.9%

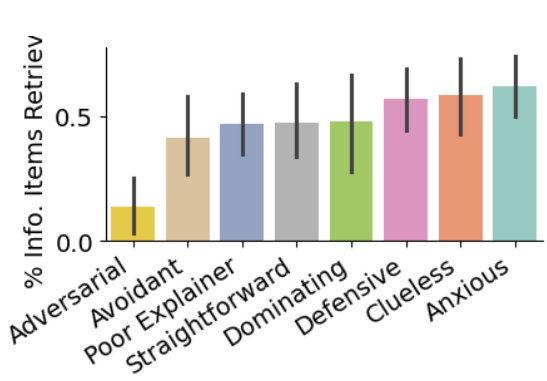
Table 3.22: **Performance of LLMs as Interviewers, with Ablations** Percentage of information items extracted (Reward percentage) in each interview by different language models (gpt-4o-mini, gpt-4o, Llama-3.1-70b, and Llama-3.1-8b) across three conditions: (1) **Hardest**: The full game, with information dependent on persuasion and persona. (2) **Medium**: an ablation removing the sources' responsiveness to persuasion. (3) **Easy**: An ablation removing the random withholding of information (i.e., a source returns all relevant information items at each turn). We observe, perhaps unsurprisingly, that removing the source's ability to withhold information (Medium → Easy) drastically increases the reward percentage at the end of the game. The removal of persuasion strategies has a smaller effect, with some models showing marginal gains (e.g., Llama-3.1-8b) and others slight losses (e.g., gpt-4o). This indicates that vanilla LLMs are poorly suited to this persuasion task.

serve as a stepping stone for training more sophisticated interview agents or supporting journalism students. Future work might expand the environment's human trials, repeat experiments at larger scale, and incorporate further realism checks to ensure robust dialogue performance and fidelity. This alignment provides face validity that the measured returns track meaningful progress signals for Emulation Learning rather than artifacts of the simulator.

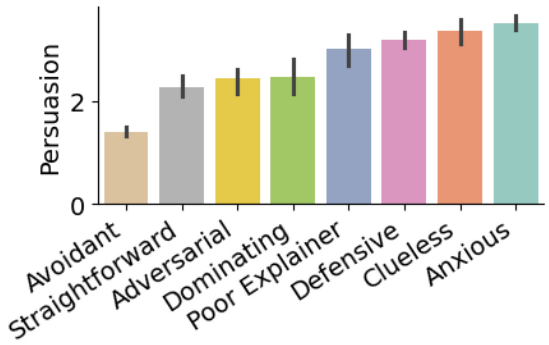
3.6.4.4 Game Simulation Results

We run our simulation for 450 interviews with four LLMs as the interviewer[78] and `gpt-4o` for the source-LLM across all personas. Table 3.22 compares the performance of LLMs across three conditions: the full game, a version without persuasion, and a version where sources do not withhold information. In the full game, where sources' responsiveness depends on persuasion and persona, the gpt-4o model performs the best, at 50.4%. However, when

[78] `gpt-4o`, `gpt-4o-mini`, `Llama-3.1-70b` and `Llama-3.1-8b`



(a) Rewards of `gpt-4o` from playing against sources of different persona types.



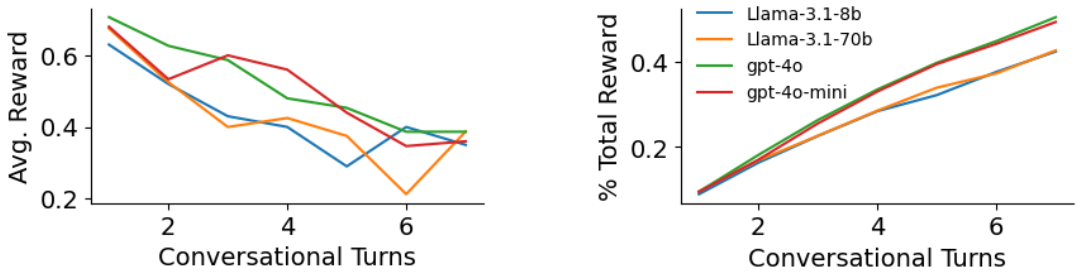
(b) Average level of persuasion, from `gpt-4o`, towards the different persona types in our evaluation.

Figure 3.15: Comparison of `gpt-4o`'s performance across different persona types. The Adversarial type is by far the hardest to extract information from, however, it is easier to persuade. LLMs might be most the thrown off by adversarial sources.

persuasion is removed, performance only marginally improves across all models (e.g., Llama-3.1-70b reaches 45.5%, while gpt-4o remains stable at 49.8%), indicating that other aspects of the game (i.e., inferring which information the source has withheld) also pose a challenge. In the easiest condition, where no information withholding occurs, all models perform significantly better, with reward percentages reaching over 80%, showing that withholding is a major obstacle.

Figure 3.15a highlights the performance of gpt-4o across different source personas. The model achieves the highest information extraction from straightforward personas, while adversarial and defensive personas are the most challenging. Despite being harder to extract information from, adversarial sources are easier to persuade (Figure 3.15b).

Figure 3.16a explores how the reward (information extraction) changes over the course of an interview. The results show a declining trend in reward per conversational turn. However, the total reward accumulated over time (Figure 3.16b) increases almost linearly, showing that the LLMs continue to extract information, albeit at a slower rate. Together, these findings highlight the limitations of current LLMs in engaging with persuasive and strategic multi-turn interviews. While larger models like gpt-4o outperform smaller ones,



(a) Average reward across conversational turns.

(b) Percentage (%) of Reward, by total reward.

Figure 3.16: Comparison of Rewards over time across language models. For all language models, the reward declines over time, shown above. However, this is not due to interviewer "maxing out" reward, as Total Reward increases nearly linearly across conversational turns.

they still exhibit significant gaps in persuasion and adaptive questioning, particularly when dealing with difficult personas. Viewed through Emulation Learning, the "Easy" ablation removes long-horizon dependencies, effectively making returns near-myopic — under which $\pi^{(llm)}(\tau|x)$ appears competent. The large performance gap to the full condition isolates deficits in temporal organization and persuasion that a better policy would limit.

3.6.5 Discussion

Our findings indicate that news interview transcripts provide a powerful, real-world resource for studying *persuasive, grounding, and multi-turn strategies* in dialogue systems. In particular, we build on prior work that highlights grounding gaps in large language models (LLMs) [394], extending insights from game-play-inspired multi-turn dialogue research [393, 427] into a domain abundant with authentic data. By examining human interviewers' behaviors, we illustrate how grounding and persuasion manifest naturally in real-world news interviews, yet remain difficult for current LLMs: counterfactual next-question experiments are necessary to evaluate $q_\theta(\tau \mid g)$ by asking whether the recovered role/content choices match human τ_{obtain}; the *game* evaluates $\pi(\tau \mid x)$ under the *true* delayed $Reward$. Both are necessary in Emulation Learning: the former ensures we

emulate human structure, the latter ensures we learn utilities that *justify* that structure. We show in Section 3.6.3.3 that humans consistently employ grounding dialogue throughout their interviews, a tactic LLMs fail to emulate effectively. In Section 4.2.7, we demonstrate how LLMs struggle to extract information from diverse source personas, particularly when those personas exhibit adversarial or avoidant traits. These findings underscore the significance of persona mismatches: while existing game-based dialogue studies often assume a single persona per environment [428, 427], our results suggest that personae with different levels of hostility or indifference pose challenges for current models.

One way to address these limitations is to incorporate *long-range reward signals* during model training [429]. Grounding dialogue and persuasion are inherently long-horizon phenomena [395, 415]. In contexts like therapy, for instance, effective grounding fosters patient openness and lasting progress [430]; in education, it encourages students' sustained engagement and deeper learning [431]. Our *NewsInterview* framework addresses this by providing an environment in which LLMs must continually strategize about which questions to ask, what information gaps need filling, and how to persuade sources to disclose details. It instantiates a training pipeline that (i) uses $q_\theta(\tau \mid g)$ on observed trajectories to extract realistic journalist goals and intentions and (ii) fine-tunes $\hat{\pi}(\tau \mid x)$ with delayed returns in controlled environments – explicitly addressing long-horizon gaps in real-world settings. This game-playing setting is less complex than fully adversarial multi-agent domains [428, 427, 393] because the source's goal is not to mislead but to selectively withhold information. Yet, even in this scenario, LLMs struggle to maintain effective information extraction over multiple turns, pointing to deeper issues in question-asking. Future directions include refining our `getPersuasionLevel` function, introducing importance-weighted or quote-centric reward signals, and further validation.

3.7 Chapter Conclusion

In this Chapter, we observed how trajectory-modeling challenges arise when we consider trajectories τ that move beyond 1-horizon tasks. Specifically, when considering longer trajectories, we must be careful to ensure that $\tilde{\tau}$ is composable and learnable (otherwise we cannot learn a useful policy) (Section 3.2.3), that our action vocabulary $\mathcal{A}$ is a *useful* vocabulary to describe the phenomena we wish to study (Section 3.5), and that the granularity of $\mathcal{A}$ is either not too granular (we risk losing long-term coherence, as modern LLMs do) and not to coarse (we cannot distinguish usefully different actions) (Sections 3.4, 3.6). In Section 3.2, we started by *directly training* an inverse model, $q_\theta(a|g)$ based on a learned $q_j = \alpha(x_i, g)$, to associate sources with sentences x_i; we then probed the *composability* of τ in order to prove a policy function $\pi(\tau|x)$ was learnable. In Section 3.3, we asked whether pretrained LLMs implicitly *learned* such policy functions, $\pi^{(llm)}(\tau|x)$ and revealed substantial gaps: LLMs were better at proposing angles than sources, but overall alignment and creativity lagged; fine-tuning helped, yet a sizable deficit remained. To close this, in Section 3.4, we introduced a hierarchical *planner–executor* model, where actions a_t are decomposed into thinking/planning actions, $a_{t,p}$ and *executing* actions, $a_{t,e}$: $a_t = [a_{t,e}, a_{t,p}]$; $\pi(a_t = [a_{t,p}, a_{t,e}]\,|x, s_t, a_{<t}) = \pi_p(a_{t,p}|x, s_t, a_{<t})\pi_e(a_{t,e}|a_{t,p})$. $a_{t,p}$ is then chosen to matches distributional signatures of human trajectories; we introduce *discourse* analysis for the first time (see Section 1.2.2 for an explanation of discourse) and introduce a low-dimensional discourse schema to align generated planning steps $\hat{a}_{t,p}$ with human $a^*_{t,p}$. Finally, in Section 3.5 we introduced methods to *compare* different *discourse schemas*, or action vocabularies $\mathcal{A}$ and showed in Section 3.6 a fun interviewing *game* that further decomposes our action space into *thinking*, *retrieving* and *obtaining* information.

Policy learning is central to *emulation learning*, and we have barely scratched the surface in this Chapter. The next chapters will move on from policy learning and explore diverse challenges (in Chapter 4, we address the *execution* or *realization* of τ into state-space

$\mathbf{s} = s_1, s_2 \ldots; s_n = g$; in Chapter 5, we explore datasets that give us richer observability into intermediate state spaces). Challenges in policy learning in the broader field of reinforcement learning continue to be an active area of research today; *emulation learning*, with it's inferred action spaces, offers yet additional challenges. Going forward, I am especially interested in exploring *inverse reinforcement learning* [2] as *deeper* approach to policy learning; only when we truly start to consider reward-learning can we (a) get closer to *emulation* as performed by humans in social learning (b) generalize *beyond* simply *replicating* goal states g and actually learn what makes goal states potent and desirable. Reward learning *also* gives a pathway towards making *active* interventions to *improve* the goal states we reach. We can interrogate rewards to discard *unwanted rewards* (e.g. bias in source-selection). I am also interested in hierarchical approaches to policy learning. Although we explored these approaches in Section 3.4, I believe we have only scratched the surface. Emerging approaches to reasoning, in domains like math and coding [432, 433, 434] including hierarchical reasoning [435] offers a tantalizing approach to latent variable modeling that generalizes *beyond* low-dimensional discourse schemata.

Chapter 4

State-Space Realization in Emulation Learning

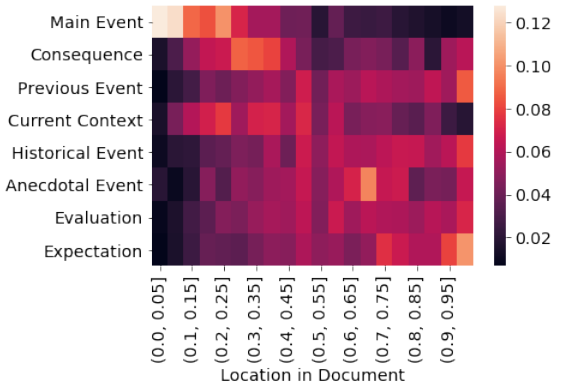
4.1 *Story-Structuring*: A Study in How Information is Organized

After the journalist has selected a *newsworthy* event, performed *source-finding*, and has compiled all the reporting material necessary to understand and narrate it, they are ready to craft a longer narrative form. This process, *story-structuring*, is the creative process we will focus on in this Chapter. We will start with the interesting observation, shown in Figure 4.2. The top part of the figure shows the distribution over discourse structures in human-written news articles. We observe a canonical, normative structure; it starts

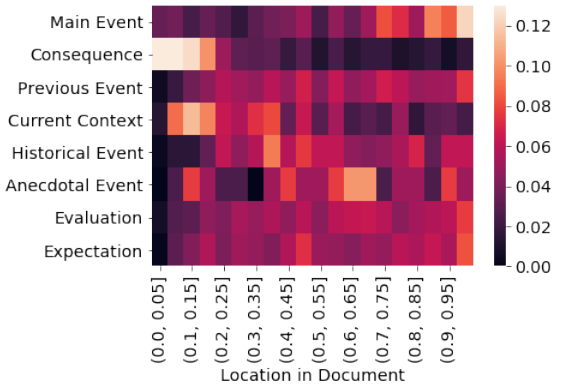
Figure 4.1: In the *journalism pipeline* outlined in Section 1.3, we focus now on the third step: *story structuring*, or taking pieces of information and organizing them together in a cohesive narrative form. Story structuring requires us to learn high-level representations of the function of text and reason about how to generate longer, coherent and human-like narratives.

with MAIN EVENTS, gives PAST CONTEXT, then finally gives FUTURE EXPECTATIONS — this is what practitioners call an *inverse pyramid* [436]. The *inverse pyramid* has developed through industry norms to deliver the most information, in the quickest form, to readers [207, 437].

Indeed, structure is an essential area of study in creative works, for two reasons. First, the structure of a work has an impact on readers: structured works allow readers to compress, navigate, and remember complex information. Research in text-comprehension work shows that readers recall top-level ideas better when discourse structure is predictable [437] and build hierarchical "macrostructures" as they read [438]. Similar effects hold for stories: when events are arranged in canonical narrative schemata, recall and perceived coherence *improve* [439, 440]. Beyond text, listeners encode hierarchical musical form and use global structure to interpret local musical events [441]. Structure functions as a cognitive technology for meaning; without it, people work harder, learn less, and forget more. Secondly, and just as importantly: structure is *deliberate* and planned by the human creator [442]. Cognitive models of *writing* treat global organization of a work as a conscious planning process that is central to creative control [443, 444] (see Section 1.2.2 for an introduction to discourse and its relation to *emulation*).



(a) Structure of human-written articles.



(b) Structure of naively generated GPT-2 articles.

Figure 4.2: Discourse structure [25] of articles generated via humans or LLMs. The likelihood of a discourse element being in the kth sentence of a news article is shown. Machine-generated structure is labeled by humans.

However, recent AI models struggle to perceive and adhere to global structures while

generating; even though surface-level generations are fluent, models can meander [445] and fail to capture deeper cohesion in long-form generations. Figure 4.2 and [446, 19] observe this in news; this observation has also been made in other domains — story-telling, dialogue and essays [318, 447, 448, 449, 450]; music [451, 452]; even images [453, 454]. In in summary, (1) structure is important for readers (2) structural cues signal human action, deliberation and thought and (3) standard self-supervised pretraining objectives fail to capture structure. Thus, *emulation learning* emerges as an appropriate tool to study structure in creative works: in this section, we will consider how our framework allows us to learn more human-like and structure-aware policies, $\pi^*(\tau|x)$, but also to better study human intentionality, $q_\theta(\tau|g)$

Story-Structuring as *Emulation Learning* Now, let us formalize *story-structuring* as an *emulation learning* problem and discuss the challenges that emerge. In previous sections, we focused on emulating the beginning (i.e. *news-finding*) and in the middle (i.e. *source-finding*) of the news creation process. Thus, previously, τ^* terminated *well* before any final *observable* goal state g existed and we performed emulation on inferences *from* goal states (i.e. in *news-finding*, we emulated *importance*, inferred from the article and it's homepage placement; in source-finding, we emulated *source mixtures*, again inferred from the article). Here, our emulation goals are closer to *observed* news articles.

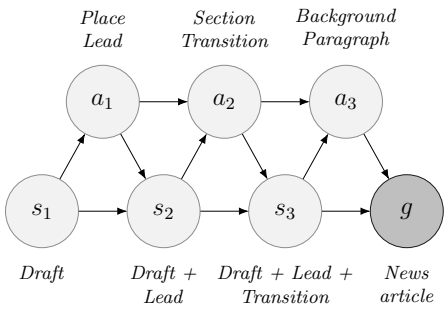


Figure 4.3: *Observability of the story-structuring task:* We assume that only the final news article, g, is observable. We assume each action, a_t corresponds to a structural decision (e.g. "place lead", "section transition") and each state-space s_t contains the draft with all realizations of structural decisions so far.

Let us formalize these goals. Let an *action* be a *structural* decision $a_t \in \mathcal{A}$ (e.g., in news, $a_t =$ "place the lead", "supply background", "introduce consequences/expectations" or "segment/transition"). A *state* $s_t \in \mathcal{S}$ is the the work with a realization of *structural decisions*

Cheat-Sheet: Emulation Learning for *Story-Structuring*

Latent *actions* are *structural decisions* are made by the writer; we focus on learning an inverse model to read structure from text and a transition model to *realize* those actions in an article.

- **a** $\boldsymbol{a_t}$ **(action)** — *structural* decisions: discourse roles, outline. Also called "control codes" $\vec{a}$.
- **s** $\boldsymbol{s_t}$ **(state)** — s_t is the draft after realizing structural decisions $a_{<t}$ (up to t). (§4.1, §4.2.1.1, Fig. 4.3).
- x x **(starting context)** — conditioning information for *state-transition* generation (e.g. headline or full article s_0, templates or prompts).(§4.1, §4.2, §4.3)..
- g $\boldsymbol{g}$ **(goal state)** — The published news article structured according to actions by the writer (§4.1, §4.2.1.1, Fig. 4.3).
- τ τ **(trajectory)** — Sequence of structural decisions and their realization in state space (§4.1, §4.2.1.1).
- P $P(s_{t+1} \mid a_t, s_t)$ **(state-space / transition model)** — *realizes* actions (structural decisions) as text. In essense, a generator trained with different methods to maintain structural coherence based on actions. (§4.2.3.2, §4.2.3.3, §4.3.2, §4.4).
- q $q_\theta(\tau \mid g)$, $q_\theta(a \mid g)$ **(inverse model)** — predicts structural decisions from text, used to *steer* structure during generation to raise label likelihood. (§4.2.3.1, Eqs. 4.2–4.5, §4.3.2.3).
- π $\boldsymbol{\pi(\tau \mid x)}$, $\pi(a_{t+1} \mid s_t)$ **(policy model)** — chooses structural actions. The *planner* in planner–executor view. (We do not train, instead provide *gold structures* $\vec{a}$.) (§4.1, §4.2, §4.4).

made at that point in the writing process (e.g. "a background paragraph", "section", "lead"). As shown in Figure 4.3, let a *trajectory* be the sequence $\tau = (a_{1:T})$, and let the *goal state* be the finalized article $g \in \mathcal{S}$. For story-structuring, our *inverse* objective $q_\theta(\tau \mid g)$ is to infer latent actions, or *structural* decisions, producing chunks of text (e.g. paragraphs) in the observed document — we will train these, in the methods that use them, using labeled datasets, as in Chapter 3. *The policy goal of emulation learning in this task* is to learn *not only* (1) a policy function, $\hat{\pi}(a|x)$ that makes structural decisions that are human-like, but *also* (2) a state-transition function, $p(s_{t+1}|a_{1\ldots t}, s_{1\ldots t})$ that generates the realization of the action a.

This formulation again suggests a *planner–realizer* (or *hierarchical emulation learning* view of narrative assembly: (1) a policy model $\pi(\tau|x)$ produces a structural sketch $a_1, \ldots a_t$ (or alternatively $\pi^*(a_{t+1}|s_t)$ selects a single structural action) and (2) a *realization* process, or *transition model* $p(s_{t+1}|a_{1\ldots t}, s_{1\ldots t})$ instantiates that sketch in natural language — producing, as a final output, a human article g. (We note the similarities with our approach in Section 3.4, which also incorporated a high-level planner and a lower-level query generator.)

Chapter 4 Overview

In Chapter 4, *State-Space Realization in Emulation Learning*, we will study how action and state-space progressions can together be modeled, to bring us closer to *observable* goal states g. This section will unfold as follows. The *first part* of this Chapter will introduce *three* techniques for *realizing actions* $a_1, ...a_t$, or *reaching* the goal-states we desire: another name for this is to learn the transition function, $P(s_{t+1}|s_t, a_{1...t})$. In Section 4.2, I formally introduce the *story-structuring* task and describe how an explicit *action-controller* can be used to guide progression in the state-space; concretely, we will learn a classifier to assess $p(s|a)$ directly; this will help us *guiding* the *state-space* transition model $\hat{P}(s_t|a, s_{t-1})$ to align with the expert $P^*(s_t|a, s_{t-1})$. I will then contrast this, in Section 4.3, a *beam-search* approach that guides states softly, via sampling. Finally, I will introduce *classifier-free guidance*, in Section 4.4, a *steering* approach for realizing actions. Then we will then concern ourselves with $q_\theta(\tau|g)$, or more generally, what *emulation* can tell us about human behavior. We will introduce a human-behavioral analysis in Section 4.5, showing how the latent structures experts developed for explaining news structure: knowledge of one, we find, can give knowledge of others. We will close, in Section 4.6 with an example *outside* of journalism, showing how more structural awareness can help interpret complex relations in legal texts.

Works Discussed:

- Spangher et al. (2022)". Sequentially Controlled Text Generation". *Findings of the Association for Computational Linguistics: EMNLP 2022*.
- Spangher et al. (2025)". DiscoSum: Discourse-aware News Summarization". *arXiv preprint arXiv:2506.06930*.
- Sanchez et al. (2024)". Stay on Topic with Classifier-Free Guidance". *International Conference on Machine Learning*
- Spangher et al. (2021)". Multitask semi-supervised learning for class-imbalanced discourse classification". *arXiv preprint arXiv:2101.00389*
- Spangher et al. (2024)". LegalDiscourse: Interpreting when laws apply and to whom". *Proceedings of the 2024 Conference of NAACL-HLT*

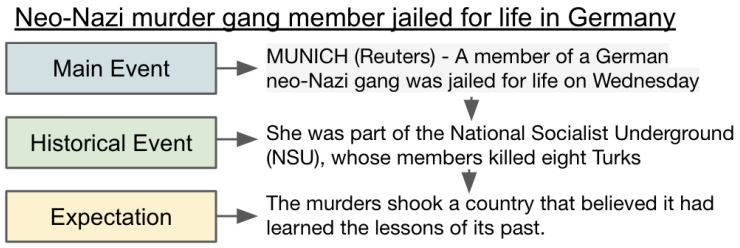


Figure 4.4: Here, we study the task of *sequentially-controlled generation*: generating documents exhibiting structure given by a sequence of local control codes. Shown is a news article with its Van Dijk structure [25] and headline. Our models take as input the headline and discourse tags and generate a sequence of sentences.

4.2 Controlling the Structure of Generated Text

The macro-structure of text (i.e. its discourse structure [207], shown in Figure 4.4) impacts both human and machine comprehension [455, 456, 457, 458]. Although naive language models generate impressively *fluent* text [147, 459, 460], the text is *structurally* dissimilar to human-written text (Figure 4.2, Section 4.2.7). Even the well-known Ovid's Unicorn generation, which *resembles* a natural news article on the surface, exhibits unnatural structure (see Table 4.1). As discussed in Section 4.1, structural decisions are actions [446]: outline-driven writing, adherence to structural form (e.g. the *inverse pyramid*) and structural critiques are all decisions that human writers make while producing their final outputs. Indeed, given our observations in Chapter 2 and 3 – that pretrained models do not always learn how to mimic these actions in creative contexts – it is unsurprising that actions governing the structure of a work should also fail to be learned. Although prior research have focused on content-planning using keywords [450], plot-design [461] and entity tracking [462], discourse/action-oriented control has been relatively understudied. We will apply *emulation* as a framework for studying structuring as a trajectory of *actions* $a_1, a_2, \ldots$ (i.e. structural decisions) and *states*, $s_1, s_2, \ldots$ (i.e. realizations of these decision). I will introduce, in this Section, our first attempt to model the *transition* function, $p(s_{t+1}|s_t, a_{1,\ldots t})$ to generate structured states. In doing so, I will introduce the basic goals and concepts in

this section.

4.2.1 Task Definition

We start with a basic view of structure, shown in Figure 4.4 and Table 4.1: structure, here, is a sequence of structural tags, or *discourse* tags (MAIN EVENT, CURRENT CONTEXT, detailed in Section 4.2.1.1). Our action space is derived from this schema a_1="Write MAIN EVENT", a_2 ="Write CURRENT CONTEXT", etc. and state space is s_1 ="Current Draft + MAIN EVENT", s_2 ="Current Draft + CURRENT CONTEXT".

Our task in this section is to learn a *transition model*, $\hat{P}(s_t|s_0, a_{1...t})$, a model that will *realize* a sequence of structural control codes[1]. As input to this model, we assume a headline sentence, s_0, and a sequence of control codes $\vec{a} = a_1, ..., a_S$ of length S (i.e., one for each sentence we wish to generate in the document. *Adjacent codes can be of the same type*.) We wish to produce, as output, a document g of length S sentences, with sentences $\mathbf{g} = g_1, ..., g_S$, each composed of a sequence of words $g_t = x_{t,1}, ..., x_{t,n_k}$ of length n_k.

We define the sequentially controlled text generation objective, our *transition model*, as:

$$p(\mathbf{g}|\vec{a}, s_0) = \prod_{k=1}^{S} \prod_{i=1}^{n_k} \underbrace{p(x_{k,i}|x_{k,<i}, g_{<k}, \vec{a})}_{t_1\text{: word likelihood}} \tag{4.1}$$

where x_i is a word in sentence k, $x_{k,<i}$ are the preceding words, $g_{<k}$ are the preceding sentences (including the headline, s_0). a_k is the control code for k. We assume that $\vec{a}$, the entire sequence of control-codes for a document, is given.

4.2.1.1 Control Codes

Now we describe more about our "control codes" or "structural actions": let us define our specific choice of structural vocabulary. Our structural vocabulary, as stated, is based on *discourse*: i.e. the functional role sentences play in a document's larger argumentative

[1]In this section, we use the terms "discourse element" and "control code" simultaneously.

0	In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.	*(Prompt)* Main Event
1	The scientist named the population, after their distinctive horn, Ovid's Unicorn.	Main Event
2	These four-horned, silver-white unicorns were previously unknown to science.	Current Context
3	Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.	Evaluation
4	Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans.	Previous Event
5	Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.	Anecdotal Event
6	Pérez and the others then ventured further into the valley.	Previous Event
7	"By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.	Anecdotal Event
8	Pérez and his friends were astonished to see the unicorn herd.	Anecdotal Event
9	These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.	Anecdotal Event
10	While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English.	Main Event
11	Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."	Evaluation
12	Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.	Historical Event
13	While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization.	Evaluation
14	According to Pérez, "In South America, such incidents seem to be quite common."	Evaluation
15	However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA.	Expectation
16	"But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization," said the scientist.	Evaluation

Table 4.1: Naïve GPT-2 output, while superficially containing familiar news elements, deviates from human news structure. To quantify structural atypicality, we train a bigram tag model $p(a_{t+1}|a_t)$ on [130]'s discourse-labeled dataset. Human (test set) log likelihood are $-1.28/-1.60/-2.01$ (5th/50th/95th perc.), whereas the Ovid's Unicorn sequence scores -2.24 — less likely than 95% of typical articles. Two notable irregularities: (i) a second Main Event appears late (row 10) after a long block of Anecdotal Event sentences (rows 5–9), and (ii) extended anecdotal runs precede key background and synthesis, patterns that are rare in human-written news.

purpose. We use a news discourse schema proposed by Van Dijk [25]. Choubey et al. [463] apply this schema and annotate a dataset, *NewsDiscourse*, consisting of 802 articles from 3 outlets[2], tagged on the sentence level. Their schema consists of 9 classes: { MAIN EVENT, CONSEQUENCE, CURRENT CONTEXT, PREVIOUS EVENT, HISTORICAL EVENT, ANECDOTAL EVENT, EVALUATION, EXPECTATION }.[3]. Although each sentence is tagged with a code, codes often repeat. For example, an entire paragraph can be tagged with MAIN EVENT sentences. We show a partial sample in Figure 4.4. We adopt this schema to describe each news article's structure. We seek frame structural control as more general and abstract than the specific kind of schema we use, though.

4.2.2 Our Approach

We use Bayes rule to factorize t_1 into:

$$t_1 = p(x_{k,i}|x_{k,<i}, g_{<k}, s_0)\frac{p(\vec{a}|x_{k,i}, x_{k,<i}, g_{<k}, s_0)}{p(\vec{a}|g_{<k}, s_0)} \propto \underbrace{p(x_{k,i}|x_{k,<i}, g_{<k}, s_0)}_{t_2\text{: naive word likelihood}}\underbrace{p(\vec{a}|x_{k,i}, x_{k,<i}, g_{<k}, s_0)}_{t_3\text{: class likelihood}} \tag{4.2}$$

t_2 is calculated using a standard pretrained language model (PTLM) and t_3 is calculated by a trained discriminator (or equivalently, inverse-action model $q_\theta(a|g)$). $q_\theta(a|g)$, here, *guides* (or equivalently, *controls*) our transition model $p(g|\vec{a}, s_0)$ to push it more in the direction of the structural tags. This factorization allows us to maximally re-use naively trained language models (i.e. t_2 stays frozen) and, as we show, is more resource efficient than fine-tuning a prompt-based model.

[2]nytimes.com, reuters.com and xinhuanet.com
[3]For a detailed class description, [26]

4.2.2.1 Past and Future Structural Awareness

Now we can use our method and task in a way that gives us real behavioral insights. Specifically: how much does awareness of the surrounding structure of the piece matter, for generating structurally sound text? In a simple example, imagine that you are tasked with: Write a "*Related Works*" section. Would it help to know the *past structure* of the article (e.g. it is coming after the "*Discussion*" section)? How about the *full structure* (e.g. after the "*Introduction*" but before the "*Conclusion*")? To answer this question, we approximate t_3 three different ways:

$$\textbf{Local-Only} \qquad t_3 \approx p(a_k|x_{k,i}, x_{k,<i}, g_{<k}, s_0) \tag{4.3}$$

In the local-only model, we assume each control code a_k is conditionally independent of other control codes given $x_{k,i}$. Thus, our generator model t_1 is made aware only of local structure: the control code a_k pertaining to the current sentence, g_k. Because of this conditional independence assumption, *local-only* control is similar to prior work that used only single-control codes, where the goal was to generate a single sentence $p(x|a) = \prod_{i=1}^{n} p(x_i|a)$ [464]. However, we show that we can remove these independence assumptions and study more complicated structural control which, as we will show, produces more coherent output.

$$\textbf{Past-Aware}: \qquad t_3 \approx \prod_{j=1}^{k} p(a_j|x_{k,i}, x_{k,<i}, g_{<j}, a_{<j}, s_0) \tag{4.4}$$

In the past-aware model, we assume autoregressive dependence between control codes, conditioned on x. Control codes for future sentences, $a_{>k}$, are conditionally independent. In Equation 4.1, this results in $x_{k,i}$ being dependent on a_k and the sequence of control codes, $a_{<k}$. To reprise our "write a *Related Works* section" anecdote, this is analogous to: "the *past* sections are: *Introduction*"; compared with "the *past* sections are: *Introduction, Problem*

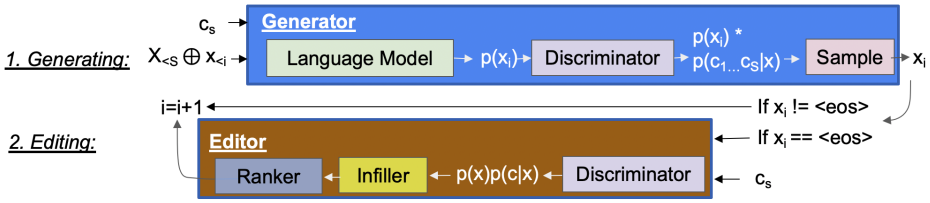


Figure 4.5: **Generation process**. First, we perturb the output of a language model using a structurally-aware classifier to approximate $p(x_i|x_{k,<i}, g_{<k})p(\vec{a}|x_{k,\leq i}, g_{<k})$ and generate word x_i by sampling from the perturbed distribution . When we generate an $< eos >$ token, we edit the sentence. We use a discriminator to identify class-salient words to mask, generating masked sentence M, and infill to boost class likelihood.

Statement, Methods, Experiments...".

$$\textbf{Full-Sequence}: \qquad t_3 = \prod_{j=1}^{S} p(a_j|x_{k,i}, x_{k,<i}, g_{<k}, a_{<j}, s_0) \tag{4.5}$$

In the full-sequence model, we make no conditional independence assumptions. Again, in the context of our "write a *Related Works* section" anecdote, this is like saying "the *past* sections are "*Introduction*", "*Methods*", ... *and* the *future* sections are: *Conclusion*". We can restrict both the past-aware and the full-sequence approximations to a sliding window around sentence k[4]. We can also add a prior on $p(\vec{a})$ to induce a discount factor[5]. This focuses the generator on control code a_k and down-weights surrounding control codes. In the next section, we show how to model these objectives. We first describe the discriminator we use as our controller, then our generation and editing techniques.

4.2.3 Additional Methodological Approaches

As described in Section 4.2.2, we can efficiently train a generative state-space transition model $P(\mathrm{g}|\vec{a}, s_0)$ by combining a *naively-trained* language model with a discriminator. *Hence, the discriminator is the main architectural component that allows us to incorporate inter-*

[4]i.e. t_3 ranges only from $j = k - w...k + w$ instead of the full sequence of sentences. In practice, we use $w = 3$.

[5]The form of our prior is: $t_3 = \prod_{j=1}^{S} m(i,j)p(a_j|x_{j,i}, x_{j,<i}, g_{<k}, a_{<j})$, where $m(i,j) = b^{|i-j|}$. We experiment with $b = [.33, .66, 1]$.

dependencies between control code sequences. We start by describing how our discriminator models different degrees of structural awareness (Equations 4.3, 4.4 and 4.5) in Section 4.2.3.1. We design a generation pipeline to balance *structural* and *local awareness*. The flow we use to accomplish this is depicted in Figure 4.5. The first step is **Generation**. Here, we sample each word, x_i using techniques described in Section 4.2.3.2 which allow us to leverage our discriminator to impose *structural control*. When we have completed a sentence, we move to **Editing**. Here, we edit the sentence to further impose *local control* on each sentence, updating x to optimize a variation of Equation 4.1: $p(x_i|x_{-i}, a_k)$, discussed in Section 4.2.3.3.

4.2.3.1 Discriminator

The discriminator we construct takes as input a sequence of sentences (g) and a sequence of local control tags ($\vec{a}$) – as such, it is literally the *inverse-action model*, $q_\theta(a|g)$ in *emulation learning*, where g is a set of generated sentences. The goal of the discriminator in this Section can be seen as a *critic* to align the structure of the generated text, $\hat{a_1}, \hat{a_2}, \ldots$ with the desired structure, $a_1^*, a_2^* \ldots$.

Our architecture combines a sentence-classification model, similar to that used in [145], with a separate label embedding architecture to incorporate knowledge of $a_{<j}$. Hence, we can make predictions for a_j based not only on x, but prior tags, $a_{<j}$, allowing us to model structural dependencies (Equation 4.2). For a full description, see [26]. We train it to model local-only, past-aware and full-sequence control variants expressed in Section 4.2.2: we train separate prediction heads to make predictions on $a_{k-w}, ...a_k, ...a_{k+w}$, i.e. labels from $-w, ..., +w$ steps away from current sentence k[6]. For local-only control (Equation 4.3) we only use predicted probabilities from the main head, k. In past-aware control (Equation 4.4), we multiply predicted probabilities from heads prior to the current sentence $< k$, and

[6]Note: we still factor label-sequences autoregressively, as in Equations 4.4 and 4.5 and learn each prediction head separately. However, keeping separate heads allows the model more flexibility in predicting how attributes of a sentence might predict future or past tags. Preliminary experiments show that this approach outperforms learning a single head for all labels.

for **full-sequence** control, we multiply predicted probabilities from all heads.[7] We now describe how we use these predictions.

4.2.3.2 Generation

We combine our discriminator's predictions with a naive PTLM to solve Equation 4.2 two ways: **Hidden-State Control**, based on [465] and **Direct Probability**, based on [466].

Hidden-State Control (HSC): Wolf et al. [467]'s GPT-2 implementation caches hidden states H to produce logits approximating $p(x_i|x_{<i})$. We perturb these hidden states H, resulting in $\hat{H}$ that produce logits approximating Equation 4.1 instead. We generate H from a naive PTLM and use this to make a prediction $\hat{c}$ using our discriminator. We then calculate the loss $L(\hat{a}, a)$ and backpropagate to H to derive $\hat{H}$.

Direct-Probability Control (DPC): We calculate $p(x_i|x_{<i}, g_{<s})$ to identify the 200 most likely x_i under the naive language model, $|x_{i,j}|_{j=0}^{200}$. Then we calculate $p(a_s|x_{i,j}, x_{<i}, g_{<s}, a_{-s})$ for each $x_{i,j}$ using our discriminator. We directly multiply these probabilities to calculate Equation 4.1[8]. Note that the HSC and DPC algorithms are extensions of previous work: the difference is that here they are used to model control code *sequences* rather than *single* tags. *The key components that allow this is our discriminator, which makes predictions based on label sequences, and our algorithm which, as shown in Figure 4.5, increments codes each time an <eos> token is generated.*

4.2.3.3 Editing

After we have finished generating a sentence, we edit it to introduce more discourse markers of the local control code. We identify words in our input sequence that have the

[7] For the editing operation, the discriminator is trained without the contexualizing layer (i.e. Transformer and a_i layers are not used) because gradients need to be computed that pertain only to the sentence being edited, not previous sentences.

[8] Note that DPC has the advantage of being simpler to implement and batch-parallelizable. However, the restriction to the top $k = 200$ words selected according to $p(x_i|x_{<i}, g_{<s})$ means that we might be limiting discriminator perturbation of word-selection.

Headline	UPDATE 1-Belarus food retailer lines up landmark overseas IPO
Main Event	SILVER SPRING, Colo. (AP) — A company in Finland is poised to list its first international grocery chain in the United States after a yearlong search, raising concerns about potential corporate conflicts of interest and legal issues around using foreign intellectual property.<|endoftext|>
Previous Event	The world's largest retailer of meat and fish products hopes to become one day the leader in convenience groceries.<|endoftext|>
Expectation	It says it could become a leader in fresh fruits and vegetables, as well.<|endoftext|>
Evaluation	About 300 Finnish companies have applied to list on the New York Stock Exchange under a new plan to bypass the U. S. government and create "licenses" that allow them in many other countries, according to a regulatory filing.<|endoftext|>
Evaluation	The European Union prohibits using foreign intellectual property for purposes outside the country, and Finland's National Stock Exchange does not require approval before selling a company'lls share in its new company.<|endoftext|>
Evaluation	"What's at stake here is transparency, a sense of fairness to all the stakeholders.<|endoftext|>
Evaluation	I don't think it's right for companies to have intellectual property rights," says Michael Vakilainen, an independent analyst who has tracked the company since 2008.<|endoftext|>
Expectation	He says there are potential conflicts of interest, because one partner is the government.<|endoftext|>
Expectation	"What if you're a government contractor?"<|endoftext|>

Table 4.2: Sample document generated. Generation Method = Direct Prob. Control. Structure = Past Aware. Edited = False. (Hyperparams = $\gamma = .75$, $b = .33$)

most impact on control-code prediction by using the gradient on our input sentence of the discriminator's loss onto tokens and masking full words, following Ross, Marasović, and Peters [468]. We use only the current sentence prediction made by our discriminator (i.e. Equation 4.3), so that we impose local control on the sequence even in settings where the generator imposes structural control.

We cull the high-gradient words based on heuristics[9] to encourage the editor to introduce explicit discourse markers. We fine-tune a label-aware infilling model [469] to generate candidate edits[10] given the masked input. We mask and infill until we have generated a

[9]Words that are *not* proper nouns, named entities (except the DATE class) or adjectives, as we find these categories are more likely to be topic words spuriously correlated with control-codes.

[10]A T5 model trained using a specific input template incorporating the label. E.g. `label: Background`.

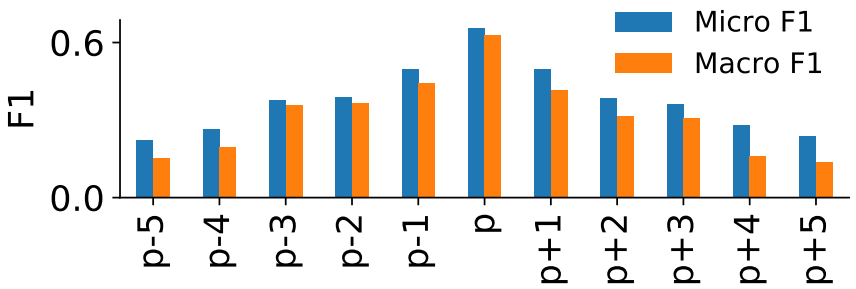


Figure 4.6: **Discriminator performance** on test data. F1 scores for $p(a_j|X_{<k}, x_{<i}, a_{<j})$ predictions. Sentence index k and word index i are fixed: we show error for using the current sentence to predict all past, current and future labels.

sentence that has an increased likelihood $p(a_k|\hat{x_k}) > p(a_k|x_k)$, and generate edit candidates ($n = 10$). We select edits on the basis of class likelihood and perplexity[11]. For more comparison and distinction from previous work for both Generation and Editing, see [26].

4.2.4 Datasets and Schema

As stated in Section 4.2.1.1, the form of sequential control we study is *discourse*: i.e. the functional role sentences play in a document's larger argumentative purpose. We adopt this schema to describe each news article's structure. We also use a dataset of unlabeled news articles[12] to fine-tune a GPT-2 model for news. We sample 30,000 documents from this dataset in a manner so that the distribution of sentence-lengths matches the distribution of sentence lengths in the Choubey et al. [463] dataset.

4.2.5 Implementation Details

We fine-tune a `GPT2-base` model on a large news corpus with a max word-piece length=2048[13]. We use this to generate naive PTLM language-modeling *as well as* sentence-embeddings

`text: The senator <MASK> to the courtroom to <MASK>.`

[11]Perplexity of the entire generated document so far is used as a selection criteria, $PPL(x_k \oplus X_{<k})$, to encourage edits preserving the logical flow of the document.

[12]`kaggle.com/snapcrack/all-the-news`. Dataset originally collected from `archive.org`. We filter to articles from nytimes.com and reuters.com.

[13]Rather than 1024 in [147]. We observe that $> 99\%$ of human-generated news articles were shorter than 2048 word pieces.

in our Discrimination model. Further implementation details are discussed in [26]. We discuss the discriminator results here briefly. As shown in Figure 4.6, the primary head, p, has a Micro F1-score of $.65$, which approaches state-of-the-art on this dataset[14]. However, performance degrades rapidly for heads farther from p. For more results on discriminator performance, including experimental variations, see [26].

4.2.6 Experiments

We sample 10 documents from the test set of our discourse dataset ($n = 200$) to test different pipeline settings. The input to our models is a headline (as a prompt) and the *full sequence of gold-truth discourse labels* of that document.

4.2.6.1 Baselines

We compare our experimental pipelines (Section 4.5.1) with the following baselines: (1) **Naive GPT-2** generation given only the headline as input (i.e. no control codes), (2) a fine-tuned **Prompting** approach and (3) the original **Human**-written articles.

For (2), we directly train a class-conditional language model to generate text by including labels in the prompt, as in [464]. Local-only prompting is achieved by only including the local control code (and prior generated sentences) in the prompt, and updating the prompt to generate a new sentence. For past-aware prompting, we include all control codes prior to our current sentence in the prompt, and update on every new sentence. Finally, for full-sequence prompting, we including the full sequence of control codes in the prompt. (See [26] for more details and examples of prompt design.) For each of these baselines, we test with and without editing (with the human-written text being edited by our algorithm in **Human** and with the generated text in all other trials being edited).

[14].71 Micro-F1 in Spangher et al. [145], which used auxiliary datasets.

4.2.6.2 Evaluation

For all pipelines, we select the best hyperparameter configurations based on perplexity and model-assigned class likelihood. Then, we manually annotate each generated document for 4 metrics: Accuracy (0-1)[15] Grammar (1-5)[16], Logical Flow (1-5)[17] and Topicality (1-5)[18]. We recruit two expert annotators with journalism experience to perform annotations blindly without awareness to which generation pipeline was used, and find moderate agreement $\kappa \in [.36, .55]$ across all categories. For more details, see [26]. We record model-dependent and non-model automatic metrics used by See et al. [470], described further in [26].

4.2.7 Results

4.2.7.1 Best Overall Trial

We show automatic and human metrics for the subset of pipelines with top-performing hyperparameters in Table 4.3. In general, the highest-performing generation pipelines are all variations of DPC with either past-aware, or full-sequence structural control. We observe that DPC with past-aware control and editing has the highest class-label accuracy, nearly approaching the human trials. The top performing pipelines for grammar and topicality are DPC with full-Sequence control and without editing. GPT-2 performed best only for Logical Flow, which was surprising but could perhaps be because the unconstrained nature of GPT-2's generation allowed it to hallucinate a flow that seemed consistent even if it was poorly structured.

[15]Accuracy: how close a generated sentence matches the discourse function of the gold-truth label for that sentence.
[16]Grammar: how grammatical *and* locally coherent a sentence is
[17]Logical Flow: how well a sentence functions in the flow of the story
[18]How well each sentence corresponds to the original headline of the article.

Generation		Human-Annotated Metrics				Automatic Metrics			
		Label Acc. ↑ (0-100)	Grammar ↑ (1-5)	Logical Flow ↑ (1-5)	On-Topic ↑ (1-5)	Perplex. ↓	Diverse Ngrams ↑ (%)	Sent. Len.**	Unseen Words ↓ (%)
GPT-2		20.0/64.4	4.2/4.5	4.7/4.3	4.6/4.2	48.2/45.4	7.1/8.3	24.9/**38.8**	4.7/3.2
Gen-Base: Prompt	L	22.2/51.1	2.8/3.9	2.4/3.0	2.3/2.8	24.4/43.4	3.7/6.5	39.7/32.4	10.6/8.7
	P	20.0/31.1	2.9/3.6	2.4/2.9	2.3/3.7	52.2/32.0	5.0/4.5	35.0/44.5	9.3/7.1
	F	46.7/64.4	4.4/4.4	3.6/3.7	3.9/3.5	42.5/49.2	7.3/7.8	35.5/42.6	4.6/4.9
Method #1: HSC	L	28.9/42.2	3.3/3.7	2.7/3.2	3.1/3.4	246/115	7.0/6.9	16.2/17.5	8.0/6.9
	P	44.4/60.0	3.4/3.8	3.0/3.0	3.2/3.3	178/147	7.5/7.5	14.8/18.8	8.1/6.7
	F	55.6/68.9	3.5/4.2	4.0/3.7	4.2/4.3	134/129	7.2/7.8	17.3/20.7	7.0/7.1
Method #2: DPC	L	44.4/64.4	4.0/4.4	3.6/4.1	3.8/3.5	42.1/39.9	5.8/8.3	24.8/42.6	4.7/3.0
	P	64.4/**88.9**	4.5/4.6	**4.4**/4.3	4.4/4.5	**37.0**/42.2	7.9/**8.4**	33.1/42.7	3.9/**3.1**
	F	66.7/68.9	**4.7**/4.5	4.3/4.3	**4.7**/4.4	42.3/45.6	8.0/8.1	28.2/40.4	4.3/3.3
Human		93.3/**95.6**	**4.9**/4.7	**4.9**/4.7	**4.9/4.9**	**34.2**/41.0	**8.7/8.7**	**37.9**/39.6	4.2/4.5

Table 4.3: Metrics on different trial runs. **L**: Local-Context only, **P**: Past only, **F**: Full sequence. Each cell shows Unedited/Edited variants. (Hyperparams = $\gamma = .75$, $b = .33$). ** Optimal sentence length is determined relative human generation, i.e. $\min |x - 37.9|$.

4.2.7.2 Effect of Different Pipeline Components

We show the distributional shifts in performance across all trials, in Figures 4.7, 4.8. Structural control has a largely positive effect on generated text. In Figure 4.7, we find that Full-Sequence models are, on average, able to generate the most label-accurate sentences with the best grammar, logical flow and topicality. Finally, editing improves accuracy, grammar and logical flow (Figure 4.8.) The original human-generated text is our gold-standard, and it is highly class-accurate, grammatical, coherent and topical. Interestingly, as seen in Table 4.3, editing can *also* be applied to human-written text to boost label accuracy, but at the expense of coherence.

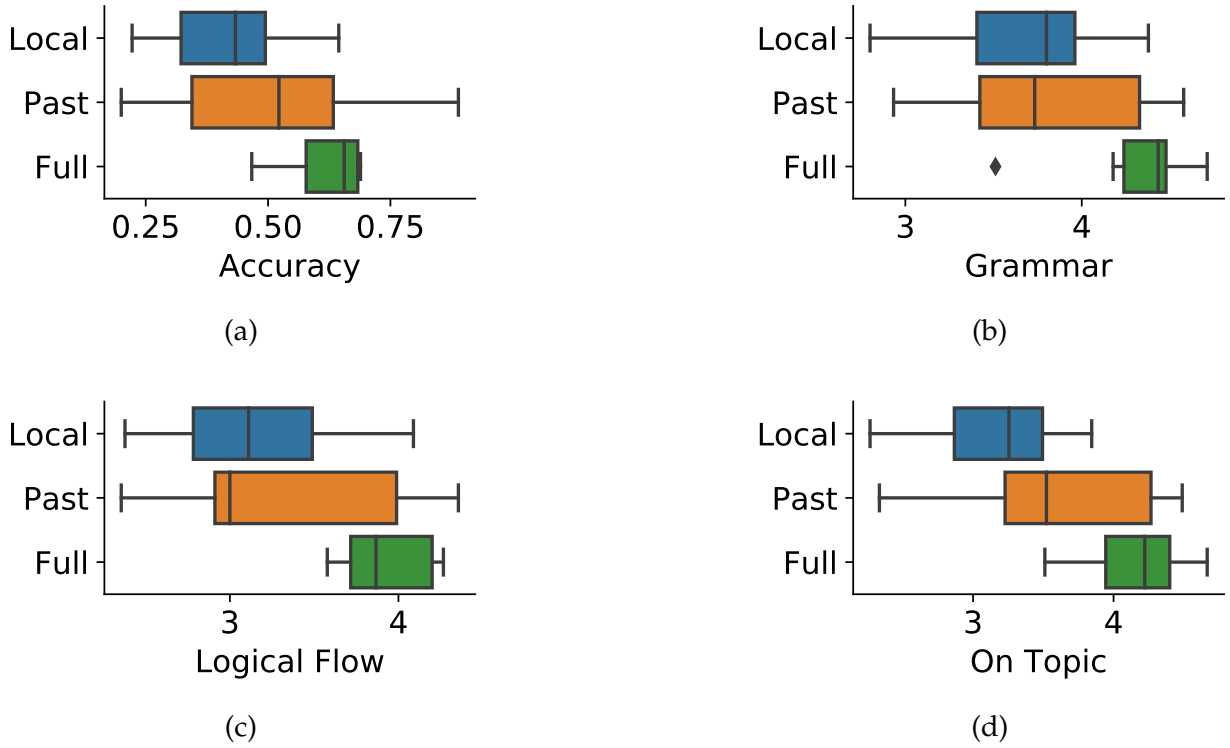


Figure 4.7: Comparison of different structural-control methods across different pipelines and hyper-parameters.

4.2.8 Discussion

We set out to answer two questions in this research: (1) whether we could impose structural control over generated documents and (2) what kinds of structural control (local-only, past-aware, or full-sequence) had the greatest effect on discourse, flow, topicality and grammaticality. Our novel pipelines, which extend various discriminator-based approaches for generation and editing, approach human-level performance. However, a gap between our model's output and human-generated text still remains across all metrics.

Insight #1: Some structural information improves all metrics of quality. Our structural exploration suggests that, for the best-performing pipelines, *past* structural information (along with editing) boosts class accuracy the most, but knowledge of the full-sequence does not. In the analogy given in the Introduction, this equates to: to write a "Related Works" section, it helps to know that it comes after the "Introduction" vs. the "Discussion",

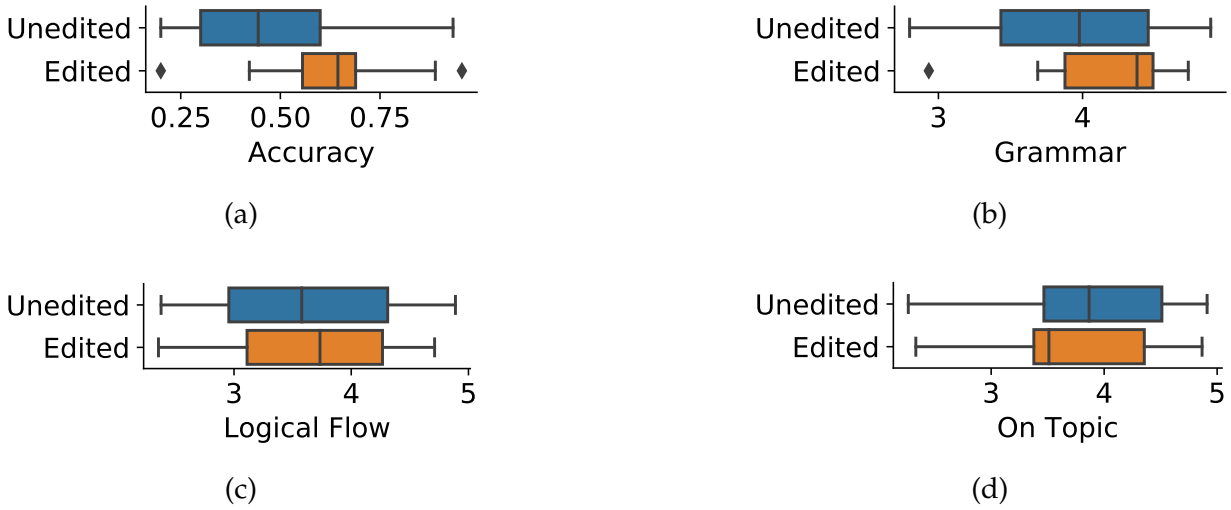


Figure 4.8: Effect of editing across different pipelines and hyper-parameters.

but not information of what sections come after. This is perhaps because enough signal is already given by the past sequence and the full sequence just adds more noise. However, full-sequence information does yield the best grammar and topicality. This might indicate a regularizing role played by the full-sequence. In general, we suspect that past-aware modeling and editing both push the model more towards the class label at the expense of topicality, flow and grammar, while full-sequence does the opposite. In practice, some combination of these pipeline components might be desired.

Insight #2: Weak discriminators can still impose accurate control. At .61 macro F1, our discriminator is a relatively weak classifier. Previous work in classifier-based controlled text generation used large training datasets and classifiers that routinely scored above .8 F1 [465, 466]. The weakness of our discriminator is one reason why HSC may have performed poorly. However, in other trials we see strong accuracy. Thus, even with a weak classifier, we can control generation. This might be because even a weak discriminator can still give relative differences between generation that does or does match the control code.

Insight #3: Evaluating text candidates using multiple model's perplexity might result in better selections. Just as surprisingly, editing also has an overall average positive effect on generation accuracy *and* generation *quality* (Figure 4.8). We had hypothesized that, because

the editor makes locally-aware infilling decisions, it would improve class-accuracy but hurt other metrics of document quality, like topicality and flow. Indeed, for the top-performing trials, like DPC and Human, Editing only improves class accuracy. However, grammar and flow improves in other trials. This could be because, as mentioned in Section 4.2.3.3, we selected candidates based on how well they make sense in the document. This also suggests that using multiple PTLMs combines different virtues of each model.

Error Analysis: We observed that sentence tokenizing remained a huge challenge. Many of the grammar errors that our annotators observed were from sentences that ended early, i.e. after decimal points. Indeed, the correlation between sentence-length and grammar is relatively high ($r = .34$). One reason for this could be that error-prone sentence tokenizing models provided faulty training data during pretrainining of LMs. This will continue to hinder document-level structural work, which often relies on a model accurately ending a sentence. Another observation, in Table 4.3, is that perplexity doesn't necessarily correlate with human judgements of quality, especially for more complex writing like *Financial* news.

Summary We have formalized a novel direction in controlled text generation: sequentially controlled text generation. We extended different techniques in controlled text generation to fit this direction, and have shown how a news discourse dataset can be used to produce news articles exhibiting human-like structure. We have explored what degrees of structural awareness yield the most human-like output: more structural control yields higher-quality output. And, we shown how to combine structural control with local editing. We have probed different parts of our pipeline to show the effects of each part.

4.3 A Beam-Search Based Approach to Generating Structural Outputs

In the prior section, we used our *inverse-action model* $q_\theta(a|g)$ to *guide* the *transition* model $p(s_{t+1}|s_t, a_{1,\ldots t})$ to perform structured story generation, introducing key concepts in how to implementing *emulation* for structural output. However, in that setup, we did not enforce *factuality* in the output, simply structure. In this task, we make two extensions. Firstly, we extend the starting state s_0 to be, not just a headline, but a *whole article*. The goal state g is *now* taken to be a *summary* of that article. As before, we also assume a set of control codes, $\vec{a}$ to drive the *structure* of that summary, but we now enforce that the outputs are factually consistent restructuring of the input s_0.

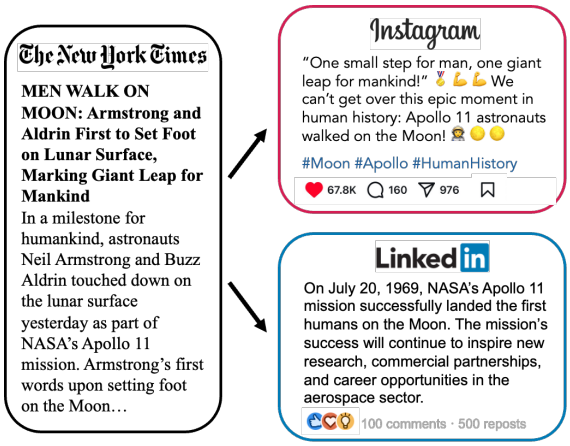


Figure 4.9: Comparative presentation of the Apollo 11 moon landing news across multiple platforms by The New York Times. This example showcases the diversity in content formatting and language adaptation for different audiences: a detailed traditional print article, a concise Instagram post, and a professionally oriented LinkedIn summary. Each platform reflects specific editorial strategies to engage its unique audience effectively.

As a practical task to frame this extension, consider Figure 4.9. Modern news organizations like *the New York Times* increasingly publish news summaries in a variety of media (e.g. print newspapers, mobile apps, podcasts, and social media) each with distinct audience expectations and content formats [471, 472]. For instance, an outlet like The New York Times may produce a child-friendly podcast edition that uses simplified language and gentler framing, a condensed Instagram version with concise, visually engaging snippets, and a longer, more detailed write-up on LinkedIn or the newspaper's own website to cater to professional or academic readers. Transforming a single piece of news into multiple styles and lengths, while preserving its core narrative and emphasis, demands **nuanced**

control over discourse structure [473, 474]. Despite the growing interest in automated news summarization [475, 476, 460, 477, 478], existing dataset approaches have overlooked this need[19]. To bridge these gaps, we propose a novel discourse-structure-aware summarization task that emphasizes the modeling of structural discourse beyond surface-level summarization coherence or factual correctness.

First, we introduce **DiscoSum**: a **Disco**urse-aware News **Sum**marization dataset. DiscoSum represents the largest and most diverse collection of professionally-written cross-platform news summaries, comprising 20k news articles from 23 different news outlets across 10 countries, multiply paired with over 100k human-written summaries from 4 distinct platforms: Facebook, Instagram, Twitter and newsletters. Next, we develop a novel discourse schema to describe structural components of news summaries, consisting of five sentence-level discourse labels. Finally, we also propose a novel discourse-driven decoding method that employs a beam search technique to evaluate and select the optimal subsequent sentences for inclusion in summaries. We evaluate our method by developing both surface-level and structural metrics to assess the effectiveness of models in producing structure-aware summaries. Our human and automated evaluations confirm that our approach effectively maintains narrative fidelity and adheres to structural demands.

4.3.1 *Structural Summarization* Task and Dataset

In this section, we describe the task formulation and evaluation metrics of structural summarization (Section 4.3.1.1). We introduce our proposed dataset including its composition and annotation process (Section 4.3.1.2).

4.3.1.1 Task Formulation

Let s_0 denote the original news document, which can consist of multiple paragraphs or sentences. We define a desired sequence of discourse labels as $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, where

[19] See [19] for a deeper comparison to Grusky, Naaman, and Artzi [479].

each a_i represents a discourse label (for instance, "contextual details," or "introductory elements," etc.) that the i-th sentence of the summary should fulfill. The objective is to generate a summary $\mathbf{g} = (g_1, g_2, \ldots, g_m)$, where each g_i is a sentence containing information in s_0, coherent, and follows. In the *structured summarization*, like structured generation before, we focus on the *transition model* – we assume that the user supplies the target label sequence $\vec{a}$ *a priori*[20]. *Predicting* an optimal structure for new input is left for future work.

We employ the same discriminator (i.e. our *inverse-action* model $\tilde{a} = q_\theta(a|g)$) that, given a sentence, predicts its discourse label. Let $\hat{\mathbf{a}} = (\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_m)$ be the sequence of labels predicted by $q_\theta(a_i|g_i) \quad \forall g_i \in \mathbf{g}$. We require $\hat{\mathbf{a}}$ to align with $\mathbf{a}$, the user-supplied labels: $\hat{a}_i = a_i$ for each position i. Although the most straightforward scenario sets $m = n$, such that the summary contains exactly n sentences, more flexible variants may allow for slight deviations while still ensuring that core positions match the targeted labels.

4.3.1.2 Dataset

We seek to construct a large, diverse dataset of news articles matched with multiple different summaries of each article, written by journalists, across different social media platforms and newsletters. We collect a list of 23 different major national and international news outlets[21] from 10 different countries (U.S., China, India, U.K., Germany, etc.), in order to capture a range of different discourse styles across different writing styles.

Social Media Collection We collect two years of social media posts on Twitter, Facebook and Instagram from each of the 23 news outlets. To do so, we build semi-automated scrolling agents that scroll down the feed of each news outlet's media page. We collect the full HTML of each post, including the text of each post as well as any linked urls. In total,

[20]This mirrors real newsroom workflows where social–media editors routinely apply pre–defined templates for different platforms. For example, commercial content–automation systems such as *Automated Insights* populate fixed headline and body layouts, and studies in discourse analysis show that canonical forms recur across news [126, 480] and even classical essay writing [481].

[21]The New York Times, The Wall Street Journal, Washington Post, AP News, BBC, Reuters, The Guardian, Bloomberg, Times of India, Le Monde, The New Zurich Times, El País, China Daily, Los Angeles Times, Chicago Tribune, The Boston Globe, USA Today, The Sydney Morning Herald, The Japan News, De Zeit

Category	Count
# of Outlets	23
# of News Articles	20,811
# of Facebook Posts	18,275
# of Instagram Posts	66,030
# of Twitter Posts	8,977
# of Newsletters	10,506

Table 4.4: Overall counts of different categories of data in our dataset.

Types	Counts
Overall	45,195
News Article → Tweet	12,516
News Article → Facebook Post	15,645
News Article → Instagram Post	7,738
News Article → Newsletter Post	9,296

Table 4.5: Statistics on the news article to summary graph, showing the number of edges between post types.

we collect 8,977 Twitter posts, 18,275 Facebook posts, and 66,030 Instagram posts (see **??** for more details). In order to identify structural summaries, we further filter these posts down to posts that contain 50 or more characters. This eliminates around 30% of our data.

Newsletter Collection We select 7 newsletter brands published by news outlets,[22] specifically searching for those that make all past newsletters within each brand available online in archives. We build scrapers to collect full HTML of each newsletter and collect 2 years worth of data, or over 20,000 newsletters. A newsletter often summarizes many news articles at the same time, yet our task is a single-document summarization task. Hence, we need to parse the text of each newsletter so that blocks of newsletter text correspond to single news article. This is *text segmentation with overlapping segments*, since links in newsletters might require larger text segments. To accomplish this, we prompted LLMs[23], building off prior work demonstrating LLM effectiveness for text segmentation tasks [482, 483, 484, 485]. We selected a prompt configuration that instructs an LLM to (1) identify all news content links, (2) extract the surrounding text context for each link, (3) exclude boilerplate content, and (4) maintain the exact original text. To mitigate potential biases or hallucinations, we implemented a verification procedure where the largest extracted blocks are cross-checked against the LLM's own outputs in multiple iterations, with any inconsistencies flagged for manual review. Manual inspection confirmed the LLM's

[22] *Axios* "The Finish Line"; *the New York Times*, "The Morning", *the LA Times*, "California Today"; *The Skimm*, "The Daily Skimm"; *The Daily Beast*, "Cheat Sheet"; *Semafor*, "Newsletters"; *CNN*, "Reliable Sources"

[23] Prompts shown in [19].

capability in this task, with segmentation quality exceeding 95% accuracy in our audits across a randomly sampled set of 100 newsletters. In total, we generate 10,506 summaries from the newsletters we collect.

News Article Collection We collect a superset of news article URLs from all the social media posts and newsletters described above. Following Spangher et al. [273], we scrape Wayback Machine for the HTML of each news article. We use an LLM (GPT-4) to clean the HTML to extract a full, complete news article (we find existing libraries[24] are insufficient). We prompt the model to filter out non-news segments (e.g., login prompts, advertisements, and extraneous content), while retaining only article content.

News Article and Summary Matching For many social media posts, we have a URL in the post that gives us an explicit match; however, for others we do not (e.g. Instagram does not allow URLs in posts). To discover as many edges as possible, we decide to match *any* news article from *any* outlet with *any* social media post or newsletter summary. To do so, we employ a two-step rank-and-check method. Specifically, we first use SBERT [221] to embed news articles and summaries; for each news article, we found the 10 closest summaries as candidates. Then, we use GPT-4 to perform a strict pairwise comparison for each candidate, returning only binary "yes" or "no" judgments on whether they describe the same news story, following the methodology validated in [211][25]. In manual audits, this matching step exceeds 95% accuracy. Not only does this approach help us recover all summaries produced by a single news outlet for each article they publish, but we can see how *other* news outlets cover the same news event.

Dataset Splits For all experiments, we use a 70%/20%/10% train/validation/test (14k/4k/2k article-summary pairs) split of the DiscoSum dataset. This split is made at the article level to prevent leakage, so all summaries of the same article are kept within the same split.

[24] https://newspaper4k.readthedocs.io/en/latest/

[25] Authors found that LLMs could be used to verify cross-document event coreference with high performance.

4.3.2 Method

In this section, we outline our methods for generating structure-aware summaries. First we describe two necessary components: (1) the discourse schema we use to drive structural summarization, and (2) a sentence-level labeler, that predicts discourse labels, which we use to guide generations (Section 4.3.2.1, Section 4.3.2.2). Then, we propose two algorithms to generate summaries conforming to a target discourse sequence a (Section 4.3.2.3): (1) an edit-based approach and (2) a beam search method.

4.3.2.1 Discourse Schema Generation

To formalize a notion of "structured" summaries, we seek to construct a low-dimensional, novel discourse schema to describe social media and newsletter summaries. First, we use an automated process to generate a schema, in contrast to prior work using manual analysis to develop schemas, typically based on O(10) examples[26]. Inspired by Pham et al. [332], we first ask an LLM to generate descriptive labels for the discourse role of each sentence in all of our summaries (O(100k) sentences). Then, we embed these labels using an SBERT embedding model [221], and cluster these embeddings using k-means.

From this embedding process, we identify five distinct clusters that represent different narrative roles: Introductory Elements, Contextual Details, Event Narration, Source Attribution and Engagement Directive). See [19] for definitions of each discourse role. We confirm the validity of this schema by asking two professional journalists to assess the quality and ideate for missing role labels. The choice of specifically five discourse labels was informed by extensive experimentation. While alternative parameter choices (e.g., k=7, 13, or 23) were feasible in our clustering approach, we selected a 5-dimensional schema based on human evaluation trials that showed high inter-annotator agreement ($\kappa = 0.615$) for assessing the validity of these labels. Though a 5-dimensional schema may appear limited for capturing the full complexity of news discourse structures—particularly across

[26]For example, Van Dijk [126] builds their schema based on an analysis of 12 news articles.

cross-cultural or niche news scenarios—it provides a strong foundation for this pilot study in discourse-aware summarization.

4.3.2.2 Discriminator

Next, in order to guide our structure-aware generation (Section 4.3.2.3), we construct a sentence-level discriminator (or, equivalently, an *inverse action model* $q_\theta(a|g)$) that assigns discourse labels to sentences, following Spangher et al. [145, 486]. Note that this is the same discriminator used in Section 4.2. The discriminator was trained on the train split of DiscoSum. To verify the quality of the validation set, we had two expert annotators independently label a subset of 500 sentences. The trained labeler achieved a high accuracy rate of over 90% on the validation set, with strong performance across all five discourse categories (the lowest per-category F1 score still exceeded 0.85, see [19] for more details). This high level of accuracy is crucial for its role in the summarization process, where it is later used as a reward guidance mechanism to ensure that generated summaries adhere to the required discourse structure.

4.3.2.3 Generation Methods

Iterative Editing Our first strategy approaches summary generation as an iterative refinement process. We begin by prompting the LLM to produce a complete initial summary, then repeatedly "edit" any sentences that do not fulfill their intended discourse labels. After the initial summary is generated, we use our discriminator $q_\theta(a|g)$ to identify which sentences carry the wrong labels. We then remove these "mismatched" sentences and generate new candidate sentences. Over several iterations, the summary gradually "evolves" to match the sequence $\mathbf{a}$. By focusing only on individual problematic sentences, this approach preserves what is already correct in the summary. It can also adapt to complex label sequences without having to restart the entire generation each time a mismatch is found.

Sentence-Level Beam Search In contrast to iteratively fixing errors, our second strategy

Algorithm 2 Sentence-Level Discourse-Driven Beam Search (beam width k)

Require: Source doc s_0; target labels $\mathbf{a} = (a_1, \ldots, a_N)$; beam width k
Ensure: Summary $\mathbf{g} = \langle g_1, \ldots, g_N \rangle$
$\mathcal{B} \leftarrow \{(\langle\,\rangle, 0)\}$ ▷ each item is (hypothesis h, score s)
for $i \leftarrow 1$ to N **do**
  $\mathcal{B}' \leftarrow \emptyset$
  for $(h, s) \in \mathcal{B}$ **do** ▷ $h = \langle g_1, \ldots, g_{i-1} \rangle$
    $candidates \leftarrow$ LLM_propose(h, s_0, k) ▷ up to k next-sentence candidates c
    for $c \in candidates$ **do**
      $h' \leftarrow h \,\|\, c$ ▷ append c to the hypothesis
      $s' \leftarrow s \;+\; \alpha \log \mathrm{LLM}(c \mid s_0, h) \;+\; \beta \log q_\theta(a_i \mid h')$ ▷ $\mathrm{LLM}(\cdot)$ = base generator likelihood; $q_\theta(\cdot)$ = inverse-action/labeler score for target label a_i on the updated hypothesis h'
      $\mathcal{B}' \leftarrow \mathcal{B}' \cup \{(h', s')\}$
    end for
  end for
  $\mathcal{B} \leftarrow \mathrm{TopK}(\mathcal{B}', k)$
end for
return $\arg\max_{(h,s)\in\mathcal{B}} s$

constructs a label-compliant summary sentence by sentence from scratch in a beam search style [487]. We begin with an empty summary and consider one position at a time (e.g., first the sentence that should have the "introductory elements" label, then the sentence that should have the "contextual details" label, and so on). At each step i, the LLM generates several candidate sentences, $candidates$ (forming a sentence-level "beam"), which are then evaluated by $q(a|\cdot)$. We choose the candidate that best matches the target label a_i. This sentence is appended to the current partial summary. By evaluating multiple options at each step and selecting the best match for the desired label, this approach ensures each summary sentence follows the intended label sequence. The detailed procedure is described as Algorithm 2.

4.3.3 Experiments

In this section, we present our experimental setup (Section 4.3.4) and evaluation framework for structured summarization with target discourse labels (Section 4.3.4.1). We introduce baseline models and methods being benchmarked (Section 4.3.4.2). Next, we present

Original News Article	
MADISON HEIGHTS, Mich. (AP) — You may have already seen them in restaurants: waist-high machines that can greet guests, lead them to their tables, deliver food and drinks and ferry dirty dishes to the kitchen. Some have cat-like faces and even purr when you scratch their heads. But are robot waiters the future? It's a question the restaurant industry is increasingly trying to answer. Many think robot waiters are the solution to the industry's labor shortages. Sales of them have been growing rapidly in recent years, with tens of thousands now gliding through dining rooms worldwide... *(continues for 823 more words...)*	
Target Labels: Event Narration, Contextual Details, Event Narration, Engagement Directive, Event Narration	
Structured Summary	
Event Narration	Robot waiters are redefining the future of the restaurant industry.
Contextual Details	The innovative technology, first introduced at a Michigan restaurant, has been gaining popularity worldwide, as they bring efficiency, convenience, and a touch of novelty to the dining experience.
Event Narration	Robot waiters are increasingly being adopted by restaurants worldwide, offering benefits such as reduced labor costs, improved accuracy, and enhanced customer engagement.
Engagement Directive	As the technology continues to evolve, we can expect to see robot waiters becoming more sophisticated, with advanced navigation systems and the ability to learn and adapt to individual customers' preferences.
Event Narration	With the potential to transform the dining experience, robot waiters are here to stay.

Table 4.6: An example news article (top), an example structural sequence inputted by the user to guide summarization (middle), and an example summary generated (bottom).

empirical results (Section 4.3.4.3), human preference evaluation (Section 4.3.4.4) and the analysis on the impact of different beam sizes (Section 4.3.4.5).

4.3.4 Implementation Details

For vanilla generation, we sample the best output among 16 trials based on automated discourse labeler. In the Sentence-Level Beam Search, we employ BeamSize $= 16$. We fine-tuned the LLaMa-3-8B model using the PEFT method on the train split of **DiscoSum**. This fine-tuning approach reduced the validation loss significantly over 20 epochs. Key hyperparameters included a learning rate of 5e-05 and a multi-GPU distributed training

setup across eight Nvidia 4090. For each generation in our experiments, we randomly generate a list of structural tags, to simulate the widest possible set of user inputs. This also prevented us from overfitting on commonly observed discourse structures.

4.3.4.1 Evaluation Protocols

Content Accuracy Evaluation. To quantify how the content accuracy of generated news summaries, we employ several metrics: *ROUGE-L* [488], originally designed for summarization, measures the longest common subsequence of tokens between the generated summary and a reference summary. *FactCC.* [489], a model-based metric that classifies whether each generated sentence is factually consistent with the source document. *AlignScore*, a consistency metric that measures the factual correspondence between texts.

Structural Evaluation. To assess the alignment between the generated summary $\mathbf{g}$ and the expected discourse structure $\mathbf{a}$, we derive a predicted label sequence $\hat{\mathbf{a}}$ from $\mathbf{g}$ via:

$$\hat{\mathbf{a}} = \text{Labeler}(g_i) \quad \forall g_i \in g$$

where Labeler is either the human annotator or our discriminator, $q_\theta(a_i|g_i)$. We employ three metrics to quantify the closeness of $\hat{\mathbf{a}}$ to the target label sequence $\mathbf{a}$: *Longest Common Subsequence (LCS)*, to measure the length of the longest subsequence common to $\hat{\mathbf{a}}$ and $\mathbf{a}$ (a higher LCS value indicates that the predicted labels closely preserve the intended label order.) *Match Score* assesses the number of exact position-wise matches between $\hat{\mathbf{a}}$ and $\mathbf{a}$. This metric reflects the precision in predicting each label at its correct position in the sequence. *Levenshtein Distance.* [490] calculates the minimum number of single-element edits (insertions, deletions, or substitutions) required to transform $\hat{\mathbf{a}}$ into $\mathbf{a}$. A lower Levenshtein Distance indicates a higher degree of sequence similarity.

Human Evaluation. Two human annotators manually assessed the discourse structure of each generated summary. Annotators evaluated 100 summaries per model.

4.3.4.2 Baselines

To evaluate the effectiveness of our proposed approach, we benchmark it against a range of baseline models that vary in architecture, training paradigms, and optimization goals. These models include both proprietary systems and open-source alternatives, providing a comprehensive overview of current state-of-the-art capabilities in text summarization.

Close-source LLMs. These models, such as DeepSeek-V3 [27], Claude-3-5-sonnet [28], and GPT-4o [29], are included primarily to help us gauge how well our approach performs with cutting-edge technology, even if these models are not the primary focus of our evaluation.

Open-Source LLMs. Models like Qwen-2.5 and various configurations of LLaMa-3-8B represent more accessible options for academic research. Each variant of LLaMa-3-8B — whether it be the vanilla version, edit-based modifications, or fine-tuned iterations — serves to illustrate different improvements and trade-offs.

4.3.4.3 Main Results

Content Accuracy Evaluation. Table 4.7 shows both surface-level and structural evaluations for a variety of models. Despite fluctuations in ROUGE-L, FactCC, and AlignScore across different systems, our approach—specifically the beam search variant of LLaMa-3-8B—maintains competitive performance in surface-level metrics. Notably, our beam search method achieves the highest AlignScore (0.3890), demonstrating superior factual consistency with source documents compared to both proprietary and other open-source models. This is particularly significant as it shows that structural improvements can be achieved without sacrificing—and in fact can enhance—factual alignment with source content. We also include the reasoning–centric model *O1*, which outperforms GPT-4o on several metrics yet still lags behind our LLaMa-3-8B beam–search variant.

[27] https://api-docs.deepseek.com/news/news1226
[28] https://www.anthropic.com/claude/sonnet
[29] https://openai.com/index/hello-gpt-4o/

	Content Accuracy			Auto Struct.			Human Struct.		
Models	**R-L (%) ↑**	**FactCC ↑**	**AlignScore ↑**	**MS ↑**	**Lev ↓**	**LCS ↑**	**MS ↑**	**Lev ↓**	**LCS ↑**
Proprietary Models									
DeepSeek-V3	<u>47.15</u>	0.47	0.3886	0.26	0.64	0.65	0.24	0.65	0.65
Claude	34.30	**0.70**	0.3882	0.25	0.68	0.64	0.20	0.49	0.75
GPT-4o	29.51	0.63	0.3884	0.11	0.80	0.62	0.15	0.58	0.68
O1	44.65	0.50	-	0.28	0.66	0.54	-	-	-
Open-sourced Models									
Qwen-2.5	40.82	0.58	0.3888	0.24	0.66	<u>0.65</u>	0.15	0.52	0.64
LLaMa-3-8B	**47.18**	0.50	0.3496	0.21	0.77	0.36	0.24	<u>0.49</u>	0.65
– Finetuned	22.01	0.61	0.3495	0.14	0.77	0.45	0.18	0.55	<u>0.72</u>
– Edit-based	15.28	0.59	-	<u>0.51</u>	<u>0.48</u>	0.56	<u>0.24</u>	0.65	0.36
– Beam Search	42.98	<u>0.64</u>	**0.3890**	**0.72**	**0.32**	**0.68**	**0.55**	**0.17**	**0.87**

Table 4.7: Comparison of models on various metrics. Metrics are categorized into content accuracy and structural assessments, both automated and human-annotated. The metrics include ROUGE-L (%), FactCC, AlignScore (for factual consistency), Match Score (MS), Levenshtein Distance (Lev), and Longest Common Subsequence (LCS). ↑ for higher is better and ↓ for lower is better. Boldfaced numbers highlight the best performance, while underscored numbers denote notable but secondary performances in each category.

Structural Evaluation. Significantly, our approach excels in both automatic and manual structural evaluations, where it demonstrates notable enhancements over both open-source baselines and the more sophisticated proprietary models. The beam search variant of LLaMa-3-8B consistently aligns more closely with the designated discourse label sequences, evidenced by its superior Match Score and reduced Levenshtein Distance. This enhancement in structural alignment underscores the model's ability to adhere rigorously to specified rhetorical structures without significant loss in surface-level accuracy. By achieving an effective balance between textual overlap and structural fidelity, our method significantly enhances the controllability and coherence of generated text.

Performances of Edit-based and Finetuned Methods. The edit-based method demonstrates a promising capability in enhancing the structural alignment of generated summaries with the desired discourse labels, as evidenced by its strong performance in structural evaluations. However, this structural fidelity comes at a cost to the content accuracy and fluency, where the ROUGE-L scores considerably lower than other methods. This

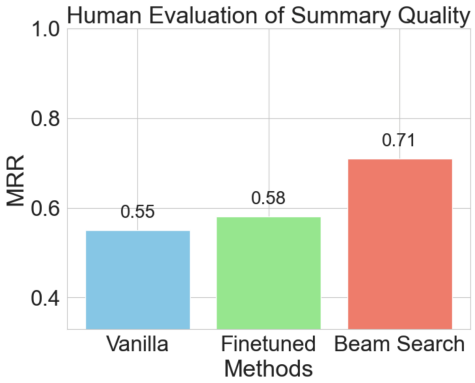


Figure 4.10: Mean Reciprocal Rank (MRR) scores from human preference evaluations of summary quality across three methods: Vanilla LLaMa-3-8B, Fine-tuned LLaMa-3-8B, and Beam Search LLaMa-3-8B.

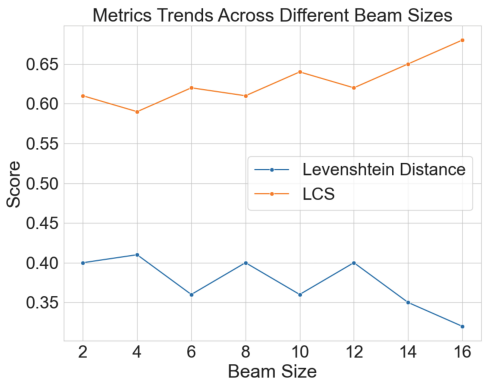


Figure 4.11: Levenshtein Distance and Longest Common Subsequence (LCS), by beam size. The graph shows a general decrease in Levenshtein Distance and a gradual increase in LCS scores, indicating improved structural alignment with larger beam sizes.

decline indicates that while the edit-based approach effectively molds the structure of the summaries, it may deviate significantly from the original text's semantic and syntactic properties. The finetuned variant of the LLaMa-3-8B model, on the other hand, shows a less impressive adaptation to the task. Despite the potential for finetuning to tailor model behavior closely to specific datasets or task requirements, the observed performance metrics suggest a failure to capture the deeper, structural nuances necessary for this specific discourse-driven summarization task. The low scores imply that mere finetuning may be insufficient for tasks that require a deep understanding and transformation of text according to complex labeling schemes. This underperformance highlights the need for more advanced approaches.

4.3.4.4 Human Evaluation of Summary Quality

We recruited two annotators to ranked the summaries based on content accuracy and structural adherence for three summary generation methods—Vanilla LLaMA-3-8B, its fine-tuned counterpart, and our beam search method. Our results, depicted in Figure 4.10, demonstrate a significant superiority of the beam search method, achieving a mean

reciprocal rank (MRR) of 0.71, compared to 0.55 and 0.58 for the Vanilla and fine-tuned.

4.3.4.5 The Impact of Beam Size

Our analysis incorporated a range of beam sizes from 2 to 16. As the beam size increases, we observe an overall improvement in the LCS scores, indicating enhanced alignment with the target discourse structure. Conversely, the Levenshtein Distance, which measures the edit distance necessary to align the predicted sequence with the target, exhibits a general decrease as the beam size increases, suggesting that larger beam sizes improve structural alignment. The observed trends open several avenues for future research. One potential area is the exploration of adaptive beam sizes that could dynamically adjust based on the complexity of the text or the specific requirements of the discourse structure at different points in a document. Additionally, while beam search techniques enhance the quality and relevance of summaries during the inference time, integrating these high-quality summaries during training could potentially elevate the model's overall performance. Future research could harness these refined outputs to boost the training process.

Summary We introduced a structural summarization approach that integrates discourse into the summarization of news articles, emphasizing factual consistency and structural alignment. Our novel dataset, DiscoSum, and evaluation metrics underscore the effectiveness of our methods, particularly the beam search technique, which ensures summaries are both contextually relevant and structurally precise. The results demonstrate significant improvements over traditional methods, suggesting that our approach enhances automated news summarization across media platforms. The shift towards a deeper understanding of discourse structures not only challenges existing models but also opens pathways for more sophisticated approaches to news narrative reconstruction. [491, 492, 493, 212].

4.4 Classifier Free Guidance

We have already observed *repeatedly*, in Chapters 2, 3 and 4 that policies learned implicitly during pretraining, $\pi^{(llm)}(a|x)$, do not seem to align with human policies $\pi^*(a|x)$. For example, in Sections 3.3 and 3.4, we observed that models lack creativity and tend to repeat queries and sources; in Sections 4.2 and 4.3, we observed that generations without structural control can meander. Simultaneously, or perhaps as a result, the *transition-model*'s generations, $P(s_{t+1}|s_t, a_{1...t})$ lack coherence over long-horizon trajectories. In the prior sections, Section 4.2 and 4.3, we addressed these by augmenting the *transition model* $p(s_{t+1}|s_t, a_t)$ with a discriminator, or the inverse-action model, $q_\theta(a|g)$. However, is this necessary? If $q_\theta(a|g)$ is noisy, is our ability to perform *story structuring* not at risk?

Similar degenerative problems have been observed in text-to-image-generation: models ignore parts of the prompt or introduce extra objects [494]. Classifier-Free Guidance (CFG) has emerged in this field as an elegant *training-free* approach to address this [495]. In this Section, we will now explore CFG as a potential alternative to using inverse-action models $q_\theta(a|g)$ for guidance (Section 4.2) or selection (Section 4.3). In CFG, the generative model *itself* is used *sans modifications* during inference to encourage guidance. While CFG might be a lightweight solution to prompt-misadherence in LLMs, it has not previously been applied in the autoregressive text-generation setting. There are many reasons to hypothesize CFG might ***not*** transfer: in text-to-image generation, the prompts are simple descriptions and outputs are fixed-size [496]. In language modeling, prompts can be highly complex and multipart, and outputs are autoregressive and unbounded. Increasing prompt adherence seems to be a promising direction for incorporating flexible, structural control; in this section, we will text whether CFG can be an effective, lightweight approach for achieving this goal.

4.4.1 Problem Statement

To understand how Classifier-Free Guidance (CFG) might be applied for *structural control* in LLMs, I will first give a broader overview and context for steering and controllability in generative models more generally. In this section, we first discuss the origins of CFG in text-to-image generation, and then discuss how autoregressive language modeling differs.

4.4.1.1 Classifier Guidance in Text-to-Image Models

Suppose $\mathrm{P}(g)$ is an unconditional model for image g and $\mathrm{P}(g|a)$ is a conditioned model with conditioning a (e.g. a label or text prompt). Generative models usually generate g by decoding from an abstract semantic space, z. In **Classifier Guidance** [497], the name in text-to-image research for the *controlled generation* methods that we covered in Sections 4.2, an auxiliary classifier $\mathrm{P}_\phi(a|g)$ guides sampling to increase the likelihood of a in g. This modification results in the following:

$$\widehat{\mathrm{P}}(g|a) \propto \mathrm{P}_\theta(g) \cdot \mathrm{P}_\phi(a|g)^\gamma \tag{4.6}$$

where γ is called the guidance strength. As Equation 4.6 show, "guidance" is a reweighting of P_θ according to the classifier likelihood P_ϕ. $\gamma = 0$ reduces 4.6 to the unconditional model $P(g)$, while $\gamma = 1$ reduces 4.6 to the conditional generation $P(g|a)$. When $\gamma > 1$, $\widehat{\mathrm{P}}$ overemphasizes the conditioning (albeit at the cost of diversity [497]). This approach has been successfully used in a variety of works [498, 499, 500]

Classifier-Free Guidance, [495] observed that by using Bayes rule, we can eliminate the external classifier. By training the same model P_θ to support both conditional and unconditional generation (via *conditioning dropout*), we can rewrite the second term in Equation 4.6 as $\mathrm{P}_\theta(g|a) \propto \frac{\mathrm{P}_\theta(g|a)}{\mathrm{P}_\theta(g)}$. Sampling is performed according to:

$$\widehat{\mathrm{P}}_\theta(g|a) \propto \frac{\mathrm{P}_\theta(g|a)^\gamma}{\mathrm{P}_\theta(g)^{\gamma-1}}. \tag{4.7}$$

Modeling $\widehat{\mathrm{P}}_\theta(x|c)$ with a diffusion process [501] reduces to predicting the distribution of the sample noise ϵ_t,

$$\log \widehat{\mathrm{P}}_\theta(\epsilon_t|g_{t+1}, a) = \gamma \log \mathrm{P}_\theta(\epsilon_t|g_{t+1}, a) - (\gamma - 1) \log \mathrm{P}_\theta(\epsilon_t|g_{t+1}). \tag{4.8}$$

We can rewrite Equation 4.8 as:

$$\log \widehat{\mathrm{P}}_\theta(\epsilon_t|g_{t+1}, a) = \log \mathrm{P}_\theta(\epsilon_t|g_{t+1}) + \gamma\big(\log \mathrm{P}_\theta(\epsilon_t|g_{t+1}, a) - \log \mathrm{P}_\theta(\epsilon_t|g_{t+1})\big) \tag{4.9}$$

Aside from its probabilistic interpretation, this equation can be seen as a vector operation in latent space: we take a step of size γ away from the unconditional vector in the direction of the conditioning. Thus, we introduce an important tool: **Negative Prompting** [502, 503, 504, 505]. Negative prompting has been proven to be effective in many situations: striking examples have been generated by interpolations latent space [506, 507, 508]. Moreover, the initial point does not have to be the unconditional latent, but any representation we want to move away from. We introduce the "negative conditioning" or "negative prompt" $\overline{a}$, as well as a generalized equation resulting in Equation 4.8 when $\overline{a} = \varnothing$:

$$\log \widehat{\mathrm{P}}_\theta(\epsilon_t|g_{t+1}, a, \overline{a}) = \log \mathrm{P}_\theta(\epsilon_t|g_{t+1}, \overline{a}) + \gamma\big(\log \mathrm{P}_\theta(\epsilon_t|g_{t+1}, a) - \log \mathrm{P}_\theta(\epsilon_t|g_{t+1}, \overline{a})\big) \tag{4.10}$$

4.4.1.2 Classifier-Free Guidance of Language Models

Unlike in image generation, where g has fixed dimensionality and *all dimensions generated dependently*, in language modeling, g is autoregressive and unbounded. Here, we apply CFG to the logits of next-token predictions. Logits, as linear transformers of word embeddings [509, 510], capture capture semantic meaning. Using the logits also avoids network editing [511] and is architecture agnostic. In modern LLMs, conditioning a is typically a *prompt* [459] which can be a context, an instruction, or the beginning of some text. Here, we assign the *prompt* the symbol a to connect it to the idea of control-codes, used in Sections 4.2 and

LLM(a_1, a_2, a_3, a_4)	$\rightarrow$	"A powerful 6.2 earthquake hit Los Angeles on Monday."
LLM(a_1, a_2, a_3, a_4)	$\rightarrow$	"No deaths or major damage have so far been reported, but rescue crews are active."
LLM(a_1, a_2, a_3, a_4)	$\rightarrow$	"One survivor said her cats freaked out more than she did."
LLM(a_1, a_2, a_3, a_4)	$\rightarrow$	"The Northridge Earthquake, on January 17, 1994, had a magnitude of 6.7."

Figure 4.12: Toy example showing how CFG with *negative prompting* might be used to guide a state-transition model, $p(s_{t+1}|s_t, a_{1,\dots t})$. a, here, is a sequence of discourse tags (e.g. a_1 =Main Event, a_2 =Current Context, a_3 =Anecdotal Event, a_4 =Historical Event) or another representation of desired structure (e.g. a_1 ="outline element #1", a_2 ="outline element #2...), along with a prompt: *"Write me a news story"*. In each line, we shift our focus by setting a_t as the *positive prompt*, a and a_{-t} as the *negative prompt*, $\overline{a}$[29].

4.3, although prompts can be more flexible and general than the discourse codes we used previously. (We will discuss at the end of this Section how to use CFG to learn a better transition model, $P(s_{t+1}|s_t, a_{1\dots t})$, realizing a *sequence* of structural codes a.)

In language modeling, in general, we wish to generate text g which has a high likelihood of starting with the prompt, a. We define the γ-reweighted distribution $\widehat{\mathrm{P}}(g|a) \propto \mathrm{P}(g) \cdot \mathrm{P}(a|g)^\gamma$, and approximate it with CFG as $\widehat{\mathrm{P}}(g|a) \propto \frac{\mathrm{P}(g|a)^\gamma}{\mathrm{P}(g)^{\gamma-1}}$. In the case of autoregressive language models, $\mathrm{P}_\theta(g) = \prod_i^T \mathrm{P}_\theta(g_i|g_{j<i})$, we can unroll the formulation and obtain Equation 4.7 again:

$$\widehat{\mathrm{P}}_\theta(g|a) \propto \prod_{i=1}^{T} \widehat{\mathrm{P}}_\theta(g_i|g_{j<i}, a) \propto \prod_{i=1}^{T} \frac{\mathrm{P}_\theta(g_i|g_{j<i}, a)^\gamma}{\mathrm{P}_\theta(g_i|g_{j<i})^{\gamma-1}} \propto \frac{\mathrm{P}_\theta(g|a)^\gamma}{\mathrm{P}_\theta(g)^{\gamma-1}} \tag{4.11}$$

An important observation we have is that, while conditioned diffusion models cannot predict unconditioned distributions without extra training, language models handle both $\mathrm{P}_\theta(g|a)$ and $\mathrm{P}_\theta(g)$ naturally due to being trained on finite context windows. In other words, dropping the prefix a is a natural feature. We thus sample the i-th token g_i in logit space:

$$\log \widehat{\mathrm{P}}_\theta(g_i|g_{j<i}, a) = \log \mathrm{P}_\theta(g_i|g_{j<i}) + \gamma\big(\log \mathrm{P}_\theta(g_i|g_{j<i}, a) - \log \mathrm{P}_\theta(g_i|g_{j<i})\big) \tag{4.12}$$

This formulation can also be extended to accommodate *Negative prompting*, as in Equation 4.10. Negative prompting is the key to how CFG can support a more robust *transition model* $P(s_{t+1}|s_t, a_{1...t})$. Instead of using the discriminator, or *inverse-model*, $q_\theta(a_t|g)$ to guide the *transition model*, $P(s_{t+1}|s_t, a_{1...t})$ towards generating with adherence to a_t, as we did in Sections 4.2 and 4.3, we can use *negative prompting*[29] by setting $a = a_t$ and $\overline{a} = a_{-t}$ in Equation 4.10 – an example of this process is shown in Figure 4.12. In other words, by setting the *current* action a_t to the positive prompt, a and the *rest* of the actions $a_{-t} = a_1, \ldots, a_{t-1}, a_{t+1}...$ to the negative prompt, $\overline{a}$, we can guide generation towards adhering to the current action over the others. We will test this concept in Section 4.4.2.5, but now, we will continue on to the next section, where we introduce our experiments exploring the effects of CFG on different variations of prompting. We note that recent works have explored variations of CFG in language models [512, 513, 514]. However, these works have been limited to specific areas of generation, like toxicity. Our work is a more general case and a broader exploration of CFG including experiments across a wide array of benchmarks, prompt variations, human-preference experiments and computing-analysis. See [388] for more details on these works.

4.4.2 Experiments

In this section we show that Classifier-Free Guidance reliably boosts performance across a variety of common prompting approaches. In Section 4.4.2.1 we show that CFG boosts zero-shot performance on a variety of standard NLP benchmarks, including achieving state-of-the-art performance on LAMBADA with LLaMA-7B. In Section 4.4.2.2 we apply CFG to *Chain-of-Thought prompts* [515, 516] an approach to allows the model to reason first before answering the question. Next, we test the performance of CFG on *text-to-text generation prompts* in Section 4.4.2.3. Finally, we show in Section 4.4.2.5 that CFG can be

[29]Note that, in all three methods introduced in Sections 4.2, 4.3 and 4.4, we do not explore *how* we will shift between *realizing* different a_t in the sequence $\mathbf{a} = a_1, ...a_n$, beyond simply generating one sentence per tag. We leave that to future work.

applied to *assistant* prompts (i.e. prompts with system-instructions).

4.4.2.1 Basic Prompting: Zero-Shot Prompts

To test *basic, zero-shot prompting*, we consider a suite of zero-shot benchmarks implemented in the Language Model Evaluation Harness [517], which includes close-book QA [518, 519], common sense reasoning tasks [520, 521, 522, 523, 524, 525, 526], and sentence completion-tasks [527]. In these settings, the desired completions are short (often 1-2 tokens), so risks of meandering [26] or degradation [528] are low. We hypothesize that the main impact of CFG in these settings will be to reduce variance in output choices, as we explore in Section 4.4.4.

We evaluate the GPT-2 model family[147], the Pythia model family [529] and the LLaMA model family[530] using different guidance strengths across a range of standard NLP benchmarks using EleutherAI's Language Model Evaluation Harness [517] and implement CFG by starting the unconditional prompt at the last token of the initial prompt. The results are shown in Table 4.8. For better visualization, the charts for the GPT2 models, the Pythia models and the LLaMA models over the standard benchmarks are shown in [388]. We observe that except ARC (challenge) and Winogrande, the boost of performances from CFG is nontrivial and consistent. The reasons for discrepancies on these tasks are still unknown. Furthermore, we note that even the smallest LLaMA 7B model achieves 81% accuracy in Lambada (OpenAI) zero-shot benchmark with $\gamma = 1.5$, outperforming the current SOTA (zero-shot) of PaLM-540B (77.9%). Despite the fact that CFG almost doubles the computation during inference, the comparison is still noteworthy given that other models with comparable performances on Lambada (OpenAI) have much more parameters and would still require more compute than LLaMA 7B with CFG. Taken together, we show that CFG increases performance in basic prompting settings significantly.

	ARC-c		ARC-e		BoolQ		HellaSwag	
	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours
G-s	22.7	**23.0**	39.5	**42.1**	48.7	**57.0**	31.1	**31.9**
G-m	**25.0**	23.9	43.6	**47.6**	58.6	**60.1**	39.4	**40.9**
G-l	**25.1**	24.7	46.6	**51.0**	60.5	**62.1**	45.3	**47.1**
G-xl	28.5	**30.0**	51.1	**56.5**	61.8	**62.6**	50.9	**52.4**
P-160M	**23.5**	23.0	39.5	**42.2**	55.0	**58.3**	30.1	**31.2**
P-410M	**24.1**	23.8	45.7	**50.3**	60.6	**61.2**	40.6	**41.6**
P-1B	27.0	**28.0**	49.0	**54.9**	60.7	**61.8**	47.1	**48.9**
P-1.4B	28.6	**29.6**	53.8	**59.6**	63.0	**63.8**	52.1	**54.3**
P-2.8B	33.1	**34.5**	58.8	**65.4**	64.7	**64.7**	59.3	**61.9**
P-6.9B	35.2	**36.1**	61.3	**67.4**	63.7	**64.6**	64.0	**66.5**
P-12B	36.9	**38.7**	64.1	**72.6**	67.6	**67.8**	67.3	**69.6**
L-7B	41.5	**43.9**	52.5	**58.9**	**73.1**	71.8	73.0	**76.9**
L-13B	47.8	**54.2**	74.8	**79.1**	**78.0**	75.8	79.1	**82.1**
L-30B	52.9	**57.4**	78.9	**83.2**	**82.7**	80.0	82.6	**85.3**
L-65B	55.6	**59.0**	79.7	**84.2**	**84.8**	83.0	84.1	**86.3**

	PiQA		SciQ		TriviaQA		WinoGrande		LAMBADA	
	Base	Ours	Base	Ours	Base	Ours	Base	Ours	Base	Ours
G-s	62.5	**63.8**	64.4	**70.8**	5.5	**6.5**	**51.6**	50.5	32.6	**44.6**
G-m	66.4	**66.9**	67.2	**76.7**	8.3	**9.3**	**53.1**	52.1	43.0	**55.8**
G-l	69.2	**70.2**	69.4	**78.8**	11.1	**12.0**	**55.4**	54.4	47.7	**60.5**
G-xl	70.5	**71.3**	76.1	**82.4**	14.7	**15.2**	**58.3**	55.6	51.2	**62.5**
P-160M	61.4	**62.1**	67.0	**75.4**	4.1	**5.3**	**52.3**	51.1	32.8	**47.4**
P-410M	67.1	**67.8**	72.1	**79.0**	7.9	**9.1**	**52.9**	50.7	51.3	**64.0**
P-1B	69.2	**70.5**	76.0	**82.9**	**12.3**	12.3	**53.9**	51.5	56.2	**69.0**
P-1.4B	71.1	**72.5**	79.4	**85.1**	15.9	**15.9**	**57.4**	56.0	61.6	**72.7**
P-2.8B	73.6	**75.8**	83.3	**88.2**	**22.1**	20.9	**60.1**	57.9	64.6	**76.5**
P-6.9B	76.3	**77.4**	84.3	**89.7**	**28.2**	27.2	**61.1**	60.3	67.1	**78.8**
P-12B	77.0	**78.4**	87.7	**91.9**	**33.4**	32.1	**65.0**	63.4	70.4	**80.6**
L-7B	77.4	**79.8**	66.3	**75.4**	**56.0**	52.7	**67.1**	65.5	73.6	**81.3**
L-13B	80.1	**80.9**	91.1	**95.1**	**62.4**	59.8	**72.8**	71.5	76.2	**82.2**
L-30B	**82.3**	82.3	94.3	**96.4**	**69.7**	67.9	**75.8**	74.1	77.5	**83.9**
L-65B	82.3	**82.6**	95.1	**96.6**	**73.3**	71.8	**77.4**	76.1	79.1	**84.0**

Table 4.8: Results of general natural language benchmarks. "G" stands for GPT2, "P" for Pythia and "L" for LLaMa. In each cell, the first value is the result for $\gamma = 1$ (baseline) and the second value is the result for $\gamma = 1.5$ (ours).

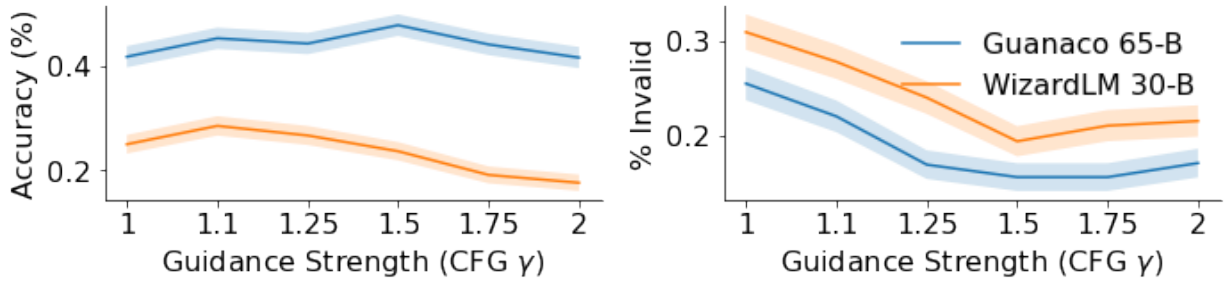


Figure 4.13: CFG's impact on chain-of-thought prompting (GSM8K dataset). Top: accuracy on task. Bottom: invalidly-formatted answers. For small γ, CFG increases the % of chains ending in a valid answer while increasing the model accuracy. For large values, the invalid % remains small but the accuracy drops.

4.4.2.2 Deliberative Prompting: Chain-of-Thought

A variation on *basic prompting* is *Chain-of-Thought (CoT) prompting* [516]. In this setting, the model is prompted to generate a series of reasoning steps before giving an answer to the task: i.e. $p(w_{cot}, w_a|c)$, where w_{cot} is a set of reasoning steps and w_a is the answer. CoT has been shown to perform well in complex reasoning tasks that cannot be fully addressed by model- or data-scaling [531]. However, as observed by [516], long reasoning chains can diverge and either not generate correct answers, or not generate parsable results. We hypothesize CFG will be able to enforce better reasoning chains with less drift. We evaluate on two arithmetic reasoning tasks: GSM8K [532] and AQuA [533]. We follow [534]'s few-shot prompt and use two open source LLM models: WizardLM-30B [535] and Guanaco-65B [536]. As can be seen in Figure 4.13, **??**, using CFG increases the percentage of CoT resulting in valid, parsable answers. For low guidance strengths, model performances increase. However, for $\gamma > 1.5$, the quality of reasoning chains degrade, and overall the performances drop[30]. We anticipate in future work being able to more fully test variations of CFG-weighting on different parts of the CoT process. For instance, instead of upweighting just w_p, we might upweight w_p, w_{cot}, or other variations.

[30] A qualitative comparison is provided in Table **??**, **??**.

4.4.2.3 Long Prompts: Generation

In contrast to *basic prompting* and *CoT-prompting* (Sections 4.4.2.1 and 4.4.2.2), where we primarily expect short answers, here we study tasks where prompts and continuations are both potentially long sequences of text. We focus on code generation here. In this setting the quality of answers is highly dependent on the model's ability to stay on target. We hypothesize that, in this setting, CFG can effectively enforce adherence to the full prompt.

4.4.2.4 Program synthesis evaluations

Program synthesis presents us with a scenario where adherence to the full prompt is essential to performance. Additionally, testing CFG on code-related tasks also demonstrates CFG's impact over formal language. Here, we prompt GPT-J [537] and CodeGen-350M-mono [538] for code generations and observe positive results (see [388]), such as an 18% improvement of the accuracy rate for GPT-J, and a 37% improvement of syntax correctness rate for CodeGen-350M-mono with positive guidance.

Next, we evaluate CFG on the HumanEval benchmark [539]. The HumanEval benchmark contains 164 coding tasks in Python, with English prompts given by a function signature and a docstring. The model generates code-based continuations of the prompt, which are tested against unit tests to evaluate the correctness of programs. We choose CodeGen-350M-mono, CodeGen-2B-mono and CodeGen-6B-mono ([538]) which are designed for Python program synthesis.[31] We test different CFG strengths[32] and different temperatures, evaluating at pass@k for $k = 1, 10, 100$ [33]. We show the results for temperature$= 0.2$ in Table 4.9[34]. The pass@1 rate, we find, increases with CFG across $1 \leq \gamma \leq 1.5$ and degrades thereafter, in accordance with findings in Section 4.4.2.2. The number of tasks where CFG outperforms is more than the one where CFG underperforms at pass@1 for $\gamma = 1, 1.25$ with

[31] *Note: CodeGen-16B-mono is omitted due to compute constraint.*

[32] $\gamma = 1.0, 1.1, 1.25, 1.5, 1.75, 2.0$

[33] The definition of pass@k according to [539]: "k code samples are generated per problem, a problem is considered solved if any sample passes the unit tests, and the total fraction of problems solved is reported."

[34] Full HumanEval results are shown in [388]

	CodeGen-350M			CodeGen-2B			CodeGen-6B		
γ	k=1	k=10	k=100	k=1	k=10	k=100	k=1	k=10	k=100
1.0	11.0%	17.0%	**22.0%**	19.5%	25.5%	**29.8%**	19.5%	25.5%	**29.8%**
1.1	**11.8%**	**18.1%**	20.1%	20.4%	25.4%	28.0%	20.4%	25.4%	28.0%
1.25	11.4%	17.3%	18.9%	19.7%	25.4%	28.0%	19.7%	25.4%	28.0%
1.5	10.9%	16.7%	18.3%	**20.9%**	**26.7%**	29.2%	**20.9%**	**26.7%**	29.2%
1.75	10.3%	16.0%	18.2%	20.4%	26.2%	28.6%	20.4%	26.2%	28.6%
2.0	8.6%	14.6%	17.6%	16.5%	22.4%	24.4%	16.5%	22.4%	24.4%

Table 4.9: CodeGen results with temperature$= 0.2$. CFG in nearly all cases increases performance, but the optimal γ value varies.

CodeGen-350M-mono.[35] We note that the improvement from CFG diminishes or harms performance at high k. Without CFG, many tasks exhibit small nonzero passing rates, while having 0% rate with CFG. This indicates that larger k significantly boosts the passing rate of difficult tasks where the rates are low but nonzero. Overall, the consistent improvement on pass@1 rates and the reduced effect on pass@100 rates support our hypothesis that CFG strengthens the adherence to the prompt at the small cost to variability/creativity.

4.4.2.5 Negative Prompting: Improving Assistants

Finally, we explore *negative prompting* in CFG, discussed in Equation 4.10 and in Section 4.4.1.2 as a method for steering our *transition model* towards action sequences. With negative prompting, traditionally, the user specifies what they do *not* want in the output (e.g. "low resolution" in text-to-image), which is then used to better meet user needs. We explore this idea, specifically, in the context of chatbots. Chatbots give us a setting where the $a =$*prompt* is expanded into a *multi-stage prompt*, a_1, a_2: as in our formulation to *structural control*.[36]. In chatbots, the language model is prompted with a two-part prompt: (1) the instruction,

[35] See the scatter plot at temperature $0.2, 0.6, 0.8$ in [388].

[36] We note that this extension to *basic-prompting* stands as a mirror to *CoT-prompting*'s extension (Section 4.4.2.2). In *CoT-prompting*, the *continuation* is expanded to a *multi-stage completion*; here, the *prompt* is expanded.

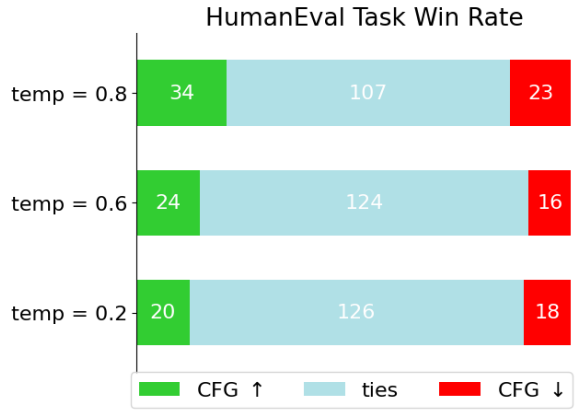


Figure 4.14: HumanEval task count comparison between $\gamma = 1, 1.25$ for CodeGen-350M-mono.

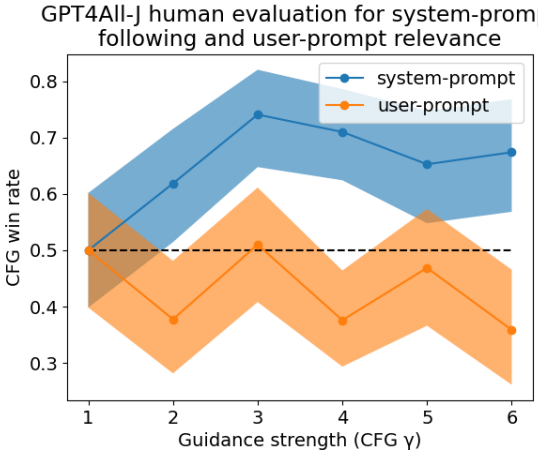


Figure 4.15: Evaluators (611 votes, 71 voters) noted that system-prompt adherence is optimal at $\gamma = 3$ while user-prompt adherence stays constant.

or "system prompt" which may give contextual information or behavioral guidelines (e.g. style, alignment, persona, etc.); and (2) the user-prompt, or the user's query. Adherence becomes an even greater concern: systems like Alpaca [540] often ignore changes to their system-prompt, and may even expose models to attacks like prompt injection [541]. We explore CFG with negative prompting to increase the success of different system prompts. We set the negative prompt $\overline{a} = a_1$ (see Equation 4.10) to be the **default system-prompt** for our models (i.e. "The prompt below is a question to answer, a task to complete, or a conversation to respond to; decide which and write an appropriate response.") and set $a = a_2$ to be **the user-prompt** (e.g. "The prompt below is a question to answer, a task to complete, or a conversation to respond to; decide which and write *a sad* response."). To test this approach with chatbots, we generate system-prompts, $n_{\overline{a} = a_1} = 25$, and user-prompts, $n_{a = a_2} = 46$, and sample 1740 random combinations of them. In [388] we include the full list of $\overline{a} = a_1$ and $a = a_2$ we use. For each (system-prompt, user-prompt) pair, we use `GPT4All-J v1.3-jazzy` to generate two completions: one without CFG and one with, with a guidance strength randomly chosen $\in$ 1,2,3,4,5,6. Our hypothesis is that CFG increases system-prompt following, ideally without hurting user-prompt adherence.

We run a human preference study on our sampled continuations, where participants are shown both, blindly, and asked to assess two things: A. which output better follows the system-prompt, $\overline{a} = a_1$ and B. which output better follows the user-prompt $a = a_2$. Our results in Figure 4.15 shows evidence that CFG emphasized the difference between $\overline{a} = a_1$ and $a = a_2$ more than sampling with $a = a_2$ alone. There is a peak at $\gamma = 3$ with 75% of system-prompt following preference over $\gamma = 1$ and user-prompt relevance (52%).

4.4.3 Cost Analysis of CFG: FLOPs and VRAM

In the previous section we showed improvements across a wide array of benchmarks and contexts. However, CFG imposes computational and memory requirements that vanilla inference does not. In this Section, we explore these requirements, which are of special interest to users with compute and memory constraints.

Compute constraints: In terms of computational requirements, CFG requires two passes through the network, effectively doubling the amount of FLOPs required for inference. Users who are compute-constrained might wonder if CFG is interesting to them at all, and if they should not run a model twice as big instead. To answer this question, we calculate the FLOP for each of the benchmark experiments that we ran in Section 4.4.2.1. We then compare across model sizes, with and without CFG. We conclude with the surprising finding that, across 5 out of 9 tasks, there there is a statistically *insignificant difference* between using CFG and using vanilla prompting with a model of twice the size at $p = .01$, according to ANCOVA regression analysis [542]. Of the significantly different tasks, 2 favor CFG and 2 favor vanilla. See [388] for more details. *In other words this indicates that, overall, a model using CFG can generally perform just as well as a model twice as large.*

Memory constraints: The impact of CFG on VRAM is nuanced. While CFG boosts the performance of smaller models, it doubles the demands of the kv cache. We conduct a memory analysis, see [388], to explore the conditions under which CFG trumps using a

	PPL $p(y\|x)$	PPL cfg	PPL instruct
PPL $p(y\|x)$			
PPL cfg	0.94		
PPL instruct	0.83	0.7	

Table 4.10: Correlation between the perplexities of CFG vs. Instruction-Tuning on the P3 dataset. We seek to identify *when* CFG is similar to instruction-tuning. Models mostly agree on the difficulty of input sentences, and in cases where they do not, CFG and Instruction-tuning have similar top-p overlaps.

larger vanilla model. We find that using CFG vs. a larger model is are highly dependent on sequence length the user wishes to generate. The doubling of the kv-cache has important implications, that qualify CFG's use, and we hope to explore these further, including memory reduction strategies, in future work.

4.4.4 Explaining the Success of Classifier-Free Guidance

In this section, we seek to explain the impact of Classifier-Free Guidance on generation. For these tests, we use the Falcon-7b-Base model [543] and, when applicable, compare against the Falcon-7b-Instruct version. We run these models on a sample dataset of $32,902$ datapoints from P3 [544]. We replicate our findings on the Open-Assistant Dataset [545] and Redpajama-3b model family[37].

Classifier-Free Guidance's Effect on Sampling Entropy We suspect that CFG, by focusing P$(y|x)$ on the prompt, will reduce the entropy of the logit distribution. CFG entropy distribution is significantly lower across generation steps than vanilla prompting, with a mean of 4.7 vs. 5.4.[38]. This restricts the number of tokens in the top-p=90% of the vocabulary distribution. We observe, in Section 4.4.4, that the top tokens re-order, showing that CFG is not simply having the same effect as temperature.

CFG's Relation to Instruction Tuning Our next question: *how* is Classifier-Free Guidance

[37] `https://www.together.xyz/blog/redpajama`
[38] See [388] for more detail)

current	top1	top2	top3	top4	top5	...	bottom5	bottom4	bottom3	bottom2	bottom1
France	flipping	destroying	waking	stopping	causing	...	guiName	ufact	Outs	kees	"}], "
,	crashing	landing	soaring	swoop	plummet	...	soDeliveryDate	POLIT	Occupations	568	publishes
landing	neigh	invis	atop	overhead	omin	...	quotas	Russo	Germans	passports	hostages
on	Buildings	skysc	rooft	Cheong	Plaza	...		MFT	ゼ	醒	DragonMagazine
Notre	Basil	Mos	Cathedral	Mosque	Eugene	...	voyage	alach	urse	arb	sb
Dame	Cathedral	monument	cathedral	Basil	Mosque	...	voyage	aila	voy	aund	wk
Cathedral	.,"	."[	slowing	blocking	ortex	...	ashore	seaf	aund	Tact	Wanted
.	Dragon	dragons	dragon	Dragon	Dragons	...	1915	1914	1944	1934	1913
It	swoop	circled	dart	hopped	bolted	...	concludes	reads	reads	culmin	marks
circled	skysc	pedestrians	architectural	hanging	skyline	...	Newfoundland	Ukrain	Zamb	Johnston	Queensland
Paris	night	amura	rum	anim	animate	...	prematurely	capit	bombed	Mé	owing
a	longer	while	long	awhile	length	...	ims	chin	chel	ille	ller
bit	longer	MORE	awhile	again	more	...	prematurely	hof	nw	arri	trop
,	startled	feathers	dragon	wings	dragons	...	inval	Junction	Palest	endas	CVE
and	dragon	dragons	golden	Winged	perched	...	CVE	inval	Ukrain	onet	Commodore
then	dragon	DRAG	dragons	neigh	DRAGON	...	CVE	onet	Kear	TPS	Tags
flew	ukong	skelet	rum	swoop	acles	...	RG	thouse	NJ	444	programmes
over	rium	Rockefeller	Plaza	Times	Symphony	...	Brittany	Newfoundland	Balt	isconsin	Yugoslavia
the	Griffith	Zeus	Hag	Science	Raphael	...	shire	Midlands	frontier	deserts	Balkans
E	BI	Rowe	ident	Methodist	allah	...	coasts	ento	bys	seys	Desire
iff	Armory	Library	restrooms	Mansion	Mahmoud	...	indo	onne	Off	itime	Norm
el	restaurant	Middle	restroom	boutique	museum	...	iband	throats	centres	detach	rift
Tower	Property	omin	Foundation	Creature	>"	...	gee	thence	pheus	hither	favourable
.	dragons	dragon	Dragons	Dragon	DRAGON	...	1944	1942	Instrument	Balt	1943
Then	dragons	dragon	dragon	Dragons	Dragon	...	Manz	Hopkins	CVE	Instrument	Squadron
it	dragon	dragons	neigh	Winged	Draco	...	CVE	udder	services	corrections	obbies
flew	upro	ukong	rum	walked	..."	...	INC	inary	lein	auxiliary	CVE
over	Chinatown	Financial	Spider	tallest	Financial	...	warr		quickShip	Newfoundland	

Table 4.11: Given the prompt **The dragon flew over Paris, France** we display, at each sampling step, the vocabulary ranked for $\mathrm{P}(w_t|w_{<t}) - \log \mathrm{P}(w_T|\hat{w})$ for the next step. We can see CFG encouraging tokens about flying dragons and Paris, and discouraging other topics or regions

affecting the vocabulary distribution? We hypothesize that CFG has similar effects to instruction-tuning, which also encourages a model to focus on the prompt [546]. Although CFG and Instruction-Tuned model variants have similar entropy across generation samples, the vocabulary distributions across our samples are largely not overlapping, indicating that CFG is *not* having a similar effect as instruction-tuning (see [388]). There are cases where the two *are* similar. As shown in Table 4.10, harder phrases for Instruction-Tuned models are typically where CFG and Instruction-Tuned models align: we observe significant spearman correlations of $r_s > .7$ between Instruction-Tuned models and CFG. As we explore more in [388], these correlations are particularly pronounced for longer prompts. We conclude that CFG is altering the model in ways that might complement instruction-tuning, opening the door to future explorations.

Visualizing Classifier-Free Guidance Finally, we provide qualitative insights into the

	FUDGE	CFG
Sentiment	.065	0.312
Toxicity	.045	0.523

Table 4.12: Percent increase in sentiment and toxicity under different guidance regimes. We compare a Classifier-Guided technique, FUDGE, [466] to CFG. (Classification likelihood judged by a secondary classifier: for sentiment we use [548]'s "positive" label; for toxicity: we use "not toxic").

reordering of the vocabulary induced by CFG. We visualize the vocabulary at each timestep ranked by the difference $\log \mathrm{P}(w_t|w_{<t}) - \log \mathrm{P}(w_T|\hat{w})$, showing which tokens are encouraged or discouraged the most. In Figure 4.11, we prompt a model with $c =$"The dragon flew over Paris, France",$\bar{c} = \emptyset$ and observe that tokens about dragons and Paris get upweighted while tokens about other locations ("Queensland"), dates ("1913"), or topics ("hostages", "voyages") are downweighted. CFG encourages tokens more related to c.

4.4.5 Discussion

Taken together, our findings indicate that CFG performs extremely well in an language-modeling setting across a wide variety of prompting techniques. This is perhaps unsurprising: recent work has demonstrated that language models can be their own reward models [547]. Indeed, CFG is to classifier-guidance for prompt adherence as Direct Preference Optimization (DPO) [65] is to Proximal Policy Optimization (PPO) [64]. From this perspective, one insights from CFG is that language models have even more expressive power than current prompting approaches are utilizing. Using the language model itself for guidance, like [65] observed, can be both more effective and efficient than using an external classifier. To prove this in our case, in Table 4.12, we show a comparison with FUDGE, an approach to Classifier Guidance in language modeling [466]. For both trials, sentiment control [548] and toxicity[39] control[40], CFG was able to steer guidance to a much greater

[39] `https://kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge`

[40] We prompt GPT2 with the prompt "That was a good movie!" for IMDB and "Don't be mean" for Toxicity. We use `bhadresh-savani/distilbert-base-uncased-emotion` and `unitary/toxic-bert` for sentiment and

degree (we tune γ as a hyperparameter for both to maximize scores while maintaining fluency). In addition, because FUDGE must be run on ever time-step, it runs 100x slower than CFG. In sum, CFG is both more effective and more efficient as a controller, without requiring any extra training. As researchers have noted, classifier guidance in language models often struggles from domain-mismatches between LMs and classifiers [549]. This perhaps can explain another key to CFG's success, with implications for RLFH and other auxiliary-model control techniques: no matter how broadly trained a classifier or agent is, it's training distribution likely not match pretraining.

However, CFG does come with it's limitations. in cases where a specific kind of control is desired, like in the two experiments shown in Table 4.12, CFG's dependency on hand-crafted prompts might be problematic. In cases where a specific generic form of control is desired (e.g. sentiment or toxicity) and a good hand-crafted prompt is *NOT* easily found, classifier-guided systems might have an advantage by being less dependent on specific system-designer prompt choices. We note that this is not the case we explore most extensively in this work, nor have we found in our extensive experiments across prompting techniques that this has observably harmed performance, but it must be acknowledged as as limitation. In future work, we hope to be able to explore prompt-optimizations to remove this barrier. Other researchers have observed that CFG is also sensitive to γ as a hyperparameter. Compared with text-to-image generation where optimal $\gamma \in 3-5$ is common, the optimal γ values for most of our prompts, except negative prompting, were small (<2). There are many reasons why text-to-image models might have higher γ values. In text-to-image generation, the pixel range is (-1, 1), whereas the range for logits in language modeling is a lot larger. In text-to-image generation, the values are independent but in text-to-text there's a softmax, and thus changing the maximum logit value dramatically alters the whole distribution. The conditional and unconditional outputs may be more different in text-to-text than in text-to-image, leading to greater

toxicity guidance, respectively, and `stevhliu/my_awesome_model` and `unitary/toxic-bert` for evaluation.

chances of text degenerating. In text-to-image diffusion models, after a very small number of iterations, the differences between the conditional and the unconditional probability should be negligible, so a stronger strength might be required.

4.5 Underlying Semantics of Structural Discourse Benefits from Multitask Learning

In the previous sections, we have introduced *three* methods for inducing *sequential* guidance and structural control in language models: two of those methods (Sections 4.2 and 4.3) depended heavily on an *inverse-model*, $q_\theta(a \mid x, \sigma)$, while the third method, CFG (Section 4.4), was more flexible to different prompting techniques. In all three methods, a key bottleneck and source of error emerges. What if we chose a *suboptimal action vocabulary*, $\mathcal{A}$, or schema σ, to specify our structural control? Either the *wrong discourse schema*, learned via $q_\theta(a \mid x, \sigma)$, or the wrong prompting approach? In Section 3.5 in Chapter 3, we faced the same question — we were comparing discourse schemas used for *source-finding* without knowing which was more optimal. We introduced methods, *conditional perplexity* and *posterior predictive*, to compare one schema against another. Here, we take the *opposite* approach. Schemata to describe textual discourse structures have been developed for a large variety of tasks: event extraction [463], sentiment analysis [550], natural language generation [551], summarization [457, 552], storyline discovery [553], and even misinformation detection [554, 458]. We ask the question: what if these schemata share *enough* similarities, and capture *enough* underlying meaning, that small variations in schemata *do not matter*? In other words, if one schema gives us enough signal about how *another* schema would label a text, we might not need to be so concerned with choosing the *right* schema.

We treat discourse tagging as learning an *inverse model* $q_\theta(a \mid g, \sigma)$ that maps an observed textual state g to a latent *discourse action* $a \in \mathcal{A}$ under a chosen schema $\sigma \in \Sigma$ (e.g., Van Dijk/NewsDiscourse, RST, PDTB; discussed in Section 4.5.2). Each dataset D_σ provides *observable labels* $y \in \mathcal{L}_\sigma$ that we view as schema-specific *emissions of the latent action*:

$$p_\theta(y \mid x, \sigma) \;=\; \sum_{a \in \mathcal{A}} C_\sigma(y \mid a)\, q_\theta(a \mid x, \sigma), \tag{4.13}$$

where $C_\sigma(y \mid a)$ is an (unknown) observation model for schema σ.

We observe that certain discourse schemata, σ, appear to offer similar and complementary information. For example, Penn Discourse and Rhetorical Structure Theory Treebanks (PDTB and RST), both tagged on news datasets, offer intrasentential, low-level discourse information [555, 556], while news discourse schemas offer intersentential, high-level, domain-specific discourse information [463, 557]. Inspired by [558]'s finding that lower-level NLP tasks (e.g. part of speech tagging) could aid higher-level tasks (e.g. semantic role labeling), our central question here becomes: can a multitask approach incorporating multiple discourse datasets can help us test the degree to which *one schema can inform another*? Specifically, by introducing *complementary information from auxiliary discourse tasks*, σ, we aim to show that we can increase performance for a primary discourse task's underrepresented classes. There is a dual purpose in this experiment. Not only do we aim to answer a scientific question — how *similar* are different discourse schemas? — we also aim to increase our ability to learn inverse models, $q_\theta(a \mid x, \sigma)$ describing any *one* schema. Indeed, even as recent advances in NLP allow us to achieve impressive results across a variety of tasks, discourse learning (often a supervised learning task — as we have framed it in Sections 3.4, 3.5, 4.2 and 4.3), faces the following challenges: (1) discourse datasets tend to be very class-imbalanced.[41] (2) Discourse learning is a complex task: human annotators require training to achieve moderate agreement [559]. (3) Discourse learning tends to be resource-poor, as annotation complexities make large-scale data collection challenging (Table 4.13). Compounding the problem, a schema often evolves across different annotation efforts, preventing the compilation of datasets.[42]

We propose a multitask neural architecture (Section 4.5.1) to address these hypotheses. We construct tasks from 6 discourse datasets, an events dataset, and an unlabeled news dataset (Section 4.5.2), including a novel discourse dataset we introduce in this Section.

[41] For example, of Penn Discourse Tree-Bank's 48 classes, the top 24 are on average 25 times more common than the bottom 24 [555].

[42] See, for instance, datasets based on variations of Van Dijk's news discourse schema [25] released in [463], [557] and the present work.

Although different datasets are developed under divergent schemas and have different goals, our framework learns correlations between schemas, and does not "waste" labeling work done by generations of NLP researchers. Our experiments show that a multitask approach can help us improve discourse classification on a primary task, *NewsDiscourse* [463], from a baseline performance of 62.8% Micro F1 to 67.7%, an increase of 4.9 points (Section 4.5.3), with the biggest improvements seen in underrepresented classes. On the contrary, two baselines — data augmentation approaches called Training Data Augmentation (TDA) and Unsupervised Data Augmentation (UDA) — fail to improve performance. We give insight into why this occurs (Section 4.5.4). In the multitask approach, the primary task's underrepresented labels *are correlated with labels in other datasets*, giving us proof into underlying similarities between these datasets. However, if we only provide more data without any correlated labels (TDA and UDA), we *overpredict* the overrepresented labels. We test many other approaches proposed to address class-imbalance and observe similar negative results [26]. Taken together, this analysis indicates that the signal from labeled datasets is essential for boosting performance in class-imbalanced settings.

4.5.1 Methodology

We formulate a multitask approach to discourse learning with the *NewsDiscourse* dataset as our primary task (Section 4.5.2). Our multitask architecture uses shared encoder layers and schema/task-specific classification heads.[43]

4.5.1.0.1 Objective. We minimize a weighted sum of schema-conditioned losses:

$$\min_{\theta} \sum_{\sigma \in \Sigma} \alpha_\sigma \sum_{(x_i, y_i) \in D_\sigma} L_\sigma\big(p_\theta(y_i \mid x_i, \sigma)\big), \tag{4.14}$$

where $D = \{D_\sigma\}_{\sigma \in \Sigma}$ is the joined dataset across schemas, $\alpha = \{\alpha_\sigma\}$ are nonnegative weights, and $p_\theta(y \mid x, \sigma)$ is the schema-specific classifier head. Conceptually, $p_\theta(y \mid x, \sigma)$

[43]Our framework can be seen as a multitask feature learning architecture [560].

factors as in Eq. 4.13, with y an *observable schema label* and a a *latent discourse action* governed by $q_\theta(a \mid x, \sigma)$. In each training step, we sample one schema σ and datum $(x_i, y_i) \in D_\sigma$.[44]

4.5.1.1 Neural Architecture

Our neural architecture (Figure 4.16) consists of a sentence-embedding layer and, in some experimental variations, embedding augmentations; a classification layer for the primary schema; and separate classification layers for auxiliary supervised schemas. The architecture we use to model our supervised schemas is inspired by previous work in sentence-level tagging and discourse learning [463, 561]. We use RoBERTa-base [562] to generate sentence embeddings (Figure 4.16). Sentences in each document are read sequentially by the same model, and the </s> token from each sentence is used as the sentence-level embedding. The sequence of sentence embeddings is passed through a Bi-LSTM layer to provide context. These layers are shared between schemas.[45]

Additionally, we experiment with concatenating different embeddings to the sentence embeddings to provide document-level and sentence-positional information. We concatenate headline embeddings and document embeddings, generated as described in [463], and sentence-positional embeddings, described in [563].[46] Each output embedding is classified using a schema-specific feed-forward layer.[47] Some of our datasets (including our primary dataset) are multiclass and others are multilabel. We discuss our datasets next.

4.5.2 Datasets

We use 8 datasets in our multitask setup, shown in Table 4.13. Four datasets contain sentence-level labels and no relational labels; two contain annotations of clausal relations; one is an events-nugget dataset where labels denote the presence of events in sentences;

[44] For UDA, which includes unlabeled data, we write $(x_i[, y_i])$ and add a consistency loss; see Section 4.5.3.3.
[45] Variations on our method for generating sentence embeddings are reported in [145]
[46] For more detail, see [145].
[47] Variations both of the classification tasks and the loss function, aimed at addressing the class-imbalance inherent in the VD2dataset, are reported in [145].

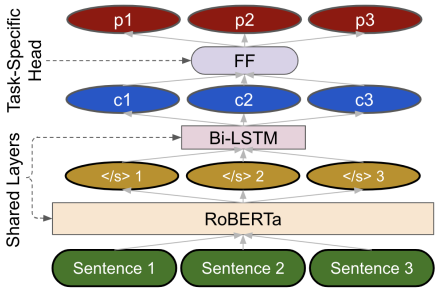


Figure 4.16: Multi-task sentence-Level classification model used for different discourse schemata. The </s> token in the RoBERTa model is used to generate sentence-level embeddings, $</s>_i$. Bi-LSTM is used to contextualize these embeddings, c_i. Finally, FF is used to make class predictions, $p_i = p_\theta(y_i \mid x_i, \sigma)$. RoBERTa and Bi-LSTM are shared between schemas. FF is the only schema-specific layer.

and one is an unlabeled news dataset. For each schema σ, we denote $D_\sigma = \{(x_i, y_i)\}$ with $y_i \in \mathcal{L}_\sigma$ as the *observable* schema label attached to sentence i (Eq. 4.13 linking y_i to a_i).

Van Dijk (VD1, VD2, VD3) and Argumentation (ARG) The Van Dijk Schema, developed by [25], was applied with no modifications [557] to 50 news articles sampled from the ACE corpus (VD1). Choubey et al. [463] expanded Van Dijk's schema to capture anecdotal discourse [564] and released a dataset, *NewsDiscourse* (VD2), consisting of 802 articles from 3 outlets[48]. ***We take VD2as our primary task due to its size.*** As shown in Table 4.13, VD2has 9 classes: Main Event (**M1**), Consequence (**M2**), Current Context (**C1**), Previous Event (**C2**), Historical Event (**D1**), Anecdotal Event (**D2**), Evaluation (**D3**), Expectation **(D4)** and Error (**E**).[49] VD2is an imbalanced dataset; its highest-support class has 1224 samples while its lowest-support has 77. We introduce a novel news discourse dataset (VD3) following the Van Dijk Schema. We expand the schema to capture discourse elements related to "Explanatory Journalism" [565]. VD3contains 67 news articles with sentence-level labels, sampled from the ACE corpus without redundancy to VD1. We additionally label 10 articles from VD1and find an interannotator agreement of $\kappa = .69$[50]. A substantial volume of news discourse is not factual assertion, but analysis, explanation,

[48]nytimes.com, reuters.com and xinhuanet.com
[49]For a detailed class description, see [463].
[50]For more information on the dataset we introduce in this paper, see [145].

Name	Label	#Docs	#Sents	#Label	Altered	Type	Cls Imb.
News discourse corpora							
NewsDiscourse	VD2	802	18,151	9	No	MC	3.01
Van Dijk [557]	VD1	50	1,341	9	No	MC	3.81
Van Dijk (present)	VD3	67	2,088	12	No	MC	6.36
Argumentation							
Argument.	ARG	300	11,715	5	No	ML	9.35
Discourse relations (filtered / altered)							
PDTB**+	PDTB-t	194	12,533	5	Yes	ML	2.28
RST**	RST	223	7,964	12	Yes	ML	2.90
KBP 14/15**	KBP	677	24,443	4	Yes	ML	4.07
Unlabeled news							
All-The-News**	U	6,000	177,530	N/A	N/A	N/A	N/A

Table 4.13: List of the datasets used, an acronym, the size, number of labels (k), whether we processed it, whether each sentence is multiclass (MC) or multilabel (ML) and the class-imbalance. ** indicates dataset was filtered. + indicates subset of tags was used. (Class Imb. $:= \frac{\sum_{j=1}^{\lfloor k/2 \rfloor} n_j}{\lfloor k/2 \rfloor} / \frac{\sum_{j=\lfloor k/2 \rfloor+1}^{k} n_j}{\lfloor k/2 \rfloor+1}$. n_j is size of class j; $n_1 > \cdots > n_k$).

and prediction [566]. We thus include the Argumentation dataset (ARG) [377], a dataset consisting of 5 labels applied to 300 news editorials.[51] The discourse tags the authors use to classify sentences are: Anecdote, Assumption, Common-Ground, Statistics, and Testimony.[52] Each of these four datasets assigns a single label to each sentence. We treat them as multiclass datasets, as shown in Table 4.13.

Penn Discourse Treebank (PDTB) and Rhetorical Structure Theory Treebank (RST) These discourse datasets each consist of spans of text in articles; labels indicate how different spans relate to each other. We process each so that sentences are annotated with the set of all relations occurring at least once in the sentence,[53] yielding multilabel $y \in \mathcal{L}_\sigma$ per sentence, and downsample documents so that the distribution of document length

[51] This dataset contains articles from 3 news outlets: aljazeera.com, foxnews.com and theguardian.com

[52] These tags share commonalities with Bales' Interactive Process Analysis categories, which delineate ways in which group members convince each other of arguments [567, 568], and have been used to analyze opinion content in news articles [566].

[53] For more details, see [145].

matches VD2.[54] Some of Van Dijk's discourse elements differ based on temporal relation: for example, some elements describe events occurring before a main event (e.g. PREVIOUS EVENT (**C2**)) while others describe events occurring after (e.g. CONSEQUENCE (**M2**)). To introduce more information about temporality, we use PDTB's tags pertaining to *Temporal* relations (we call this filtered dataset PDTB-t). When processed as described above, each of these datasets assigns multiple labels to each sentence. We treat them as multilabel datasets. This includes the labels, for PDTB: TEMPORAL, ASYNCHRONOUS, PRECEDENCE, SYNCHRONY, SUCCESSION. For RST, the final set of labels that we use: ELABORATION, JOINT, TOPIC CHANGE, ATTRIBUTION, CONTRAST, EXPLANATION, BACKGROUND, EVALUATION, SUMMARY, CAUSE, TOPIC-COMMENT, TEMPORAL.

Knowledge Base Population (KBP) 2014/2015 Some of Van Dijk's discourse elements differ based on the presence or absence of an event. For example, the elements PREVIOUS EVENT (**C2**) and CURRENT CONTEXT (**C1**) both describe the context before a main event, but the former describes events while the latter describes general circumstances. We hypothesize that a dataset identifying event occurrence can help our model differentiate these elements. We collect an additional non-discourse dataset, the KBP 2014/2015 Event Nugget dataset, which annotates trigger words for events by type: ACTUAL EVENT, GENERIC EVENT, EVENT MENTION, and OTHER. We preserve this annotation at the sentence level, similar to the PDTB and RST transformations in Section 4.5.2 and downsample documents similarly.

All-The-News (U) For semi-supervised data-ablation experiments, described in Section 4.5.3.3, we sample 6,000 documents from an unlabeled news dataset.[55] We downsample in the manner described above for PDTB and RST.

[54]Specifically, if $p_m(n)$ and $p_a(n)$ are the likelihood of a document d with n sentences in the main and auxiliary datasets respectively, we sample with weight $w_d = p_m(n)/p_a(n)$ [569]. $p_m(n)$ and $p_a(n)$ were determined empirically by N_n/N_{total}.

[55]`kaggle.com/snapcrack/all-the-news`. Dataset originally collected from `archive.org`. We filter to articles from nytimes.com and reuters.com.

	M1	M2	C2	C1	D1	D2	D3	D4	E	$F1_{Mac}$	$F1_{Mic}$
Support	460	77	1149	284	406	174	1224	540	396	4710	4710
Pretrained encoders											
ELMo	50.6	27.0	58.9	35.2	63.4	50.3	70.5	64.3	94.6	57.21	62.85
RoBERTa	52.1	9.4	65.1	27.7	68.1	51.6	72.4	65.4	96.0	56.43	64.97
+Frozen	51.2	29.3	64.3	29.8	72.2	65.8	**73.7**	**67.1**	96.5	61.08	66.54
+EmbAug	54.1	28.0	64.7	**35.9**	71.8	66.3	72.9	65.9	96.3	61.76	66.92
Data augmentation											
TDA	8.5	5.2	57.1	29.8	61.1	44.3	66.1	58.2	16.4	56.53	59.22
UDA	49.4	0.0	65.0	28.4	56.0	0.0	70.8	69.8	96.2	48.39	62.72
+TSA	51.9	34.2	63.6	33.1	70.7	66.9	72.5	66.7	96.3	61.77	66.29
Multitask											
MT-Mac	54.9	**35.5**	63.8	**35.9**	**73.7**	**70.7**	**73.7**	66.3	**96.7**	**63.46**	67.51
MT-Mic	**55.4**	25.0	**67.1**	32.8	72.5	68.9	73.6	65.8	96.0	61.89	**67.70**
Human agreement											
Hum-Pre	**58.8**	**36.1**	28.3	10.5	**75.0**	40.0	48.6	22.2	**100.0**	46.18	46.76
Hum-Post	**68.7**	**75.0**	70.3	33.3	**81.2**	**79.2**	**83.0**	**79.7**	**100.0**	**73.69**	**77.63**

Table 4.14: F1-scores of individual class labels in VD2 and Macro-averaged F1-score (Mac.) and Micro F1-score (Mic.). **ELMo** is the baseline used in [463]. **RoBERTa+Frozen+EmbAug** is our subsequent baseline. **TDA** refers to Training Data Augmentation. **UDA** is Unsupervised Data Augmentation (+TSA is for "Fine-Tuned UDA with TSA", described in Section 4.5.3.3). **MT** stands for multitask: **MT-Mac** is a trial with α chosen to maximize Macro F1-score while **MT-Mic** is a trial with α chosen to maximize Micro F1-score. **Human** is our agreement with [463]: **Hum-Pre** shows human agreement after reading VD2's annotation guidelines, conferencing and not observing labels. **Hum-Post** is after observing VD2 labels.

4.5.3 Experiments and Results

In this section, we briefly discuss experiments using VD2 as a single classification task. Then, we discuss the experiments using VD2 in a multitask setting. Finally, we discuss our experiments with data augmentation as ablations. We give a more detailed analysis of single-task experiments in [145], focusing here on multi-task experiments.

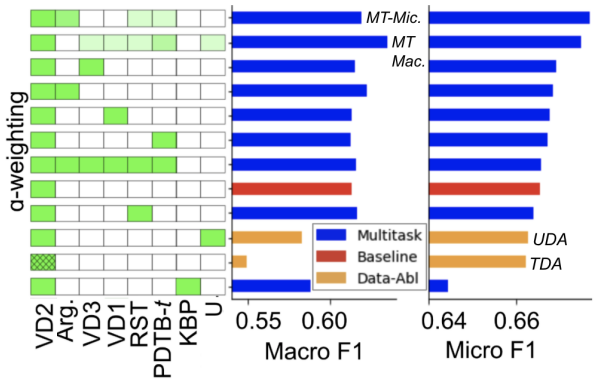


Figure 4.17: Optimal loss coefficients (α) across tasks shown for: (a) trials, (First two blue bars; **MT-Micro** and **MT-Macro** trials) (b) pairwise multitask tasks (other blue bars), (c) baseline (red bar) (d) data ablation (yellow bar; UDA and TDA). Tasks are green in strength, α value. When **U** is used, it is used with UDA head. Hashed VD2, for TDA, is data-augmented (Section 4.5.3.3).

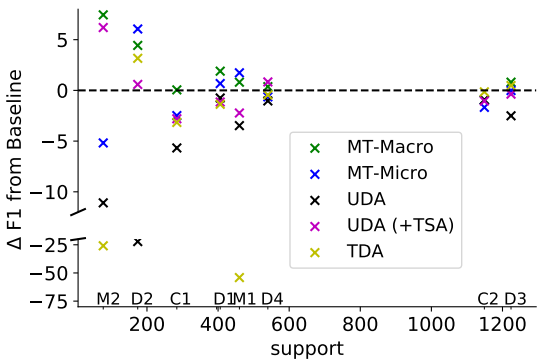


Figure 4.18: Comparison of class-level accuracy vs. label # for 3 models: MT-Micro, TDA (which underperforms baseline for lower-represented labels like M2, C1), and MT-Macro (which overperforms baseline for lower represented labels M1, M2, D1, D2). Split y-axis shown for clarity, due to TDA outliers.

4.5.3.1 Single Task Experiments

We observe, perhaps unsurprisingly, a 2-point F1-score improvement by using RoBERTa as a contextualized embedding layer rather than [463]'s baseline, ELMo [570] (**Roberta** in Table 4.14). We observe an additional 1.5 F1 score improvement by freezing layers in RoBERTa (**+Frozen** in Table 4.14). We find that freezing layers closer to the input results in greater improvement, replicating [571]. Finally, we observe a .5 F1 improvement by incorporating document, headline, and sinusoidal information (**+EmbAug** in Table 4.14).[56]

4.5.3.2 Multi-Task Experiments

As shown in Table 4.14, multitask achieves better results than any single-task experiment. We conduct our multitask experiment by performing a grid-search over loss-weighting, α_σ (defined in Equation 4.14). We select top-performing α for Micro F1-score as well as Macro F1-score based on a validation split, and report results on a test split.[57] As can be

[56]The .5 F1 improvement is observed across different sentence embeddings variations. See [145].
[57]Train, test and validation splits are specified by [463].

Dataset	$F1_{Micro}$	$F1_{Macro}$	Dataset	$F1_{Micro}$	$F1_{Macro}$
Main	.83	1.15	ARG	.05	.83
RST	.50	.73	PDTB	-.69	1.41
VD3	.49	.53	U	1.14	.68
VD1	.21	.61	KBP	2.17	2.94
β_0	66.26	61.13			

Table 4.15: We run LinReg (LR) on the α weights from multitask trials to predict Micro and Macro F1-scores (i.e., $\text{LR}(\alpha) = \text{Mic. F1, Mac. F1}$). LR coefficients ($\beta$) for each dataset show the effects of each dataset on the scores.

seen in Figure 4.17, the weighting achieving the top Micro F1-score includes datasets VD2, ARG, RST and PDTB-t, while the weighting achieving the top Macro F1-score includes datasets VD2, ARG, VD3, and RST. To understand the effect of each dataset individually, we run linear regression on the α and F1-scores found in our grid search.[58] The regression coefficients, β, displayed in Table 4.15, approximate the effect each dataset has. We conduct over 600 trials in our grid search.

4.5.3.3 Data Ablation Experiments

To test our hypothesis that labeled information in the multitask setup helps us achieve higher accuracy, we perform the following ablation: we test using additional data that does not contain new label information. We test two methods of data augmentation: Training Data Augmentation (TDA) and Unsupervised Data Augmentation (UDA). TDA enhances supervised learning [572] by increasing the size of the training dataset through data augmentations on the training data; it exploits the smoothness assumption in semi-supervised learning to help our model be more robust to local data perturbations [573]. For each datapoint (x_i, y_i) in our primary dataset, we generate $k = 10$ noisy samples $(x_{i1}, y_i), ..., (x_{ik}, y_i)$. We use a sampling-based backtranslation function to generate

[58] I.e. $y = \beta X$, where $X = \alpha$, the loss-weighting scheme for each trial, and $y =$ F1-score.

augmentations for TDA and UDA. [574].[59] UDA is a form of semisupervised learning that propagates signal from labeled to unlabeled datapoints, making use of the manifold assumption in semi-supervised learning [577, 573]. UDA seeks to promote consistency between model predictions on unlabeled datapoints $p_\theta(x_i)$ and their augmentations $\{p_\theta(\hat{x}_i)\}_{j=1}^{k}$ by minimizing their KL-divergence.[60] Both techniques were chosen as they have been shown to boost performance of low-resource NLP classifiers above other semi-supervised methods [572, 578, 576, 577, 579]. Because both techniques introduce more data without introducing more labels, they address the question: did multitask learning improve accuracy only by introducing more data?

As shown in Table 4.14 and Figure 4.17, **TDA** and **UDA** fail to improve performance above single-task experiments (**RoBERTa+EmbAug**). To interrogate further, we explored approaches introduced by [577] and [579] to improve convergence of UDA. Specifically, we use a confidence threshold, r, to mask out uncertain unlabeled data; Training Signal Annealing (TSA), to mask out uncertain labeled data; suppression coefficient β, to decrease unsupervised loss contributions for low-support classes; and other methods.[61] We test a range of values for each of these hyperparameters. In particular, we find that TSA with a *Linear* schedule has a dramatic effect on accuracy, nearly rescuing the performance of UDA. We show UDA with and without TSA (Figure 4.18, Table 4.14) to demonstrate, yet we are unable to achieve a setting whereby UDA or TDA beats multitask. Additionally, we add UDA as an unsupervised head in our multitask setup, similar to [558] introducing language modeling as an unsupervised head. We find only one setting where it contributes to our multitask accuracy (MT-Macro in Figure 4.17 and Table 4.15).

[59]To perform backtranslation, we use Fairseq's English to German and English to Russian models [575]. Inspired by [576], we generate backtranslations using random sampling with a tunable temperature parameter instead of beam search, to ensure diversity in augmented sentences.

[60]KL-divergence is minimized via consistency loss: $L_{con} = \mathbb{E}_k[D(p_\theta(x_i)||p_\theta(\hat{x}_{i,k}))]$

[61]See [145] for a detailed discussion on these approaches and our reported explorations. The top-performing hyperparameters we found were: $r = .8$, $TSA = Linear$, $\beta = 0$, $k = 5$, $p = 8$, $\alpha_{UDA} = .8$, $\tau = .8$,; [577] do not share their explorations; we find that the choice of p (the number of unlabeled data) and k (the number of augmentations per datum) have significant impact on performance.

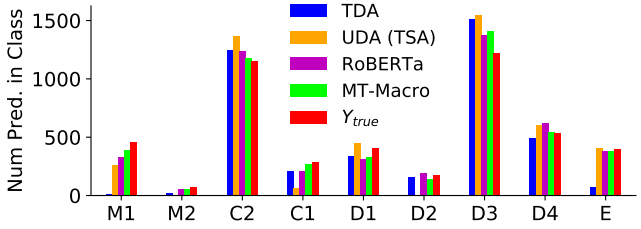


Figure 4.19: UDA and TDA over-predict better-represented classes (C2, D3) relative to Y_{true}, and under-predict lesser-represented classes (M1, M2, C1, D1). MT-Macro prediction rates are closer to Y_{true}. Specifically, $D_{\text{KL}}(Y_{\text{UDA}} \,\|\, Y_{\text{true}}) = 0.45$, $D_{\text{KL}}(Y_{\text{TDA}} \,\|\, Y_{\text{true}}) = 0.27$, $D_{\text{KL}}(Y_{\text{MT-Macro}} \,\|\, Y_{\text{true}}) = 0.01$.

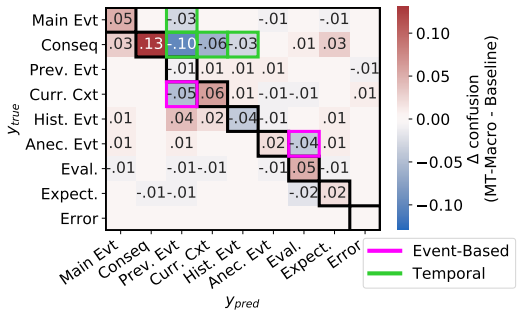


Figure 4.20: Change in Confusion between **MT-Macro** and **Baseline** (RoBERTa+EmbAug). Except for Historical Event, all classes show an improvement. Classes with Event-Based and Temporal error improvement highlighted (see Section 4.5.4 for discussion on confusion categories.)

4.5.4 Discussion

As shown in Figure 4.18, a multitask approach significantly increases performance for underrepresented classes while not hurting performance for others. This is in contrast to pure data augmentation approaches, like UDA or TDA. Improving performance in low-support classes improves overall Macro F1, as expected, and Micro F1 (Table 4.14).

Under the emulation view, auxiliary datasets provide distinct $C_\sigma(y \mid a)$ that make certain latent actions a more or less observable. Temporal relations in PDTB-t increase observability of actions that differ by temporal orientation (e.g., **C2** vs. **M2**), while argumentation tags increase observability of analytic actions (e.g., **D3**, **D4**). Thus, gains on underrepresented ND labels are expected when auxiliary C_σ reduce ambiguity about a in precisely those regions. We pause to comment on the differences in task weightings observed in Figure 4.17 for **MT-Micro** and **MT-Macro**. For example, ARG is one of the most important datasets for **MT-Micro**, but ignored in **MT-Macro**. In class imbalanced settings, Micro F1-score is weighted more towards high-support classes while Macro F1-score favors each class equally. Because different auxiliary tasks boost performance for different classes, it is

reasonable to assume that the same α will lead to different Macro F1 and Micro F1 scores[62]

One future direction is to identify criteria for including promising discourse tasks in a multitask framework. [580] performed such an analysis for multitask setups including POS-tagging and Keyphrase detection and the present work demonstrates the impact such criteria could have in aiding discourse tagging. One criteria for inclusion might be based on the label correlations between the main discourse task and a candidate task. However, obtaining correlations would require training a multitask model; at that point, directly calculating the accuracy boost would be trivial. Identifying discourse-relevant features in the input data, x, as [580] did in their work, might be more fruitful. A competing explanation to our hypothesis that multitask improves performance through label correlations is that additional datasets simply expose the model to more of the data-input space, x. Both UDA and TDA serve as ablation studies for this. [579] show that, for class-imbalanced problems, regions of the data manifold that contain the underrepresented classes generalize poorly when data augmentation is used. Indeed, we show in Figure 4.19 that TDA and UDA over-predict overrepresented classes, perhaps showing that the algorithms misjudge the extent of under-represented classes on the data manifold. One approach to improving semi-supervision would be to consider a more sophisticated annealing algorithm. As discussed in Section 4.5.3.3, TSA nearly rescued UDA's performance for all labels. Another would be to generate more augmentations for underrepresented classes [581]; on the training data for TDA [582] or using a model to identify promising unlabeled points for UDA. Upsampling underrepresented labels in sequences, which our data are, presents a challenge because we can only sample the entire sequence (i.e. the document). Thus, if we try to upsample individual underrepresented classes (i.e. sentences), we will also be upsampling overrepresented classes in the sequence.

As a final piece of analysis on our multitask setup, we show the reduction of confusion between **MT-Macro** and **Baseline** in Figure 4.20.[63] We identify reductions in two main

[62]For more information, see [145].
[63]For a more extended analysis, see [145]

classes of confusion: **Temporal** confusion, or confusion between temporal ordering of discourse elements (i.e. PREVIOUS EVENT and CONSEQUENCE); and **Event-based** confusion, or confusion between tags semantically similar except for the presence of an event (i.e. CURRENT CONTEXT and PREVIOUS EVENT). We hypothesize the reduction is due to the addition of temporal information in PDTB-t and event information in RST.

We close our discussion with an analysis of VD2's task difficulty. We ask expert annotators to relabel VD2data. Our annotators read [463]'s annotation guidelines and labeled a few trial examples. Then they sampled and annotated 30 documents from VD2without observing VD2's labels. Annotations in this **Blind** pass were significantly worse than predictions made by our best model (Table 4.14). Then, our annotators observed VD2's labels on the 30 articles, discussed, and changed where necessary. Surprisingly, even in this **Post-Reconciliation** pass, our annotators rarely scored more than 80% F1-score.

Thus, Van Dijk labeling task might face an inherent level of legitimate disagreement, which **MT-Macro** seems to be approaching. However, there are two classes, M1 and M2, where **MT-Macro** underperformed even the **Blind** annotation. For these classes, at least, we expect that there is further room for modeling improvement through: (1) annotating more data, (2) incorporating more auxiliary tasks in the multitask setup, or (3) learning from unlabeled data, by fine-tuning RoBERTa [583], using an adapter-based method [584] or another semi-supervised algorithm (one candidate besides UDA is [578]).

Summary We framed discourse tagging as schema–conditioned inverse modeling, learning $q_\theta(a \mid x, \sigma)$ over latent discourse actions $a \in \mathcal{A}$, with observable labels $y \in \mathcal{L}_\sigma$ arising via a schema–specific emission $C_\sigma(y \mid a)$. A shared encoder with schema–specific heads, trained across multiple schemas, yields a state-of-the-art improvement of $+4.9$ Micro F1 on *NewsDiscourse* ($62.8\% \rightarrow 67.7\%$) and higher Macro F1 (e.g., 63.46 for MT–Macro), with the largest gains on underrepresented labels. We show in exhaustive experiments in [145] that data-only augmentations (TDA, UDA) fail to surpass a strong single-task baseline and bias predictions toward majority labels. Multitask gains, on the other hand, are explained by

label-correlated signal: auxiliary schemas provide complementary observation models C_σ that make rare/ambiguous actions more *observable*, improving estimation of $q_\theta(a \mid x, \sigma^{\text{main}})$ in low-density regions of the state manifold. *Crucially, the fact that cross-schema training helps while data-only augmentation does not is evidence for an underlying semantic overlap across observation channels: distinct C_σ appear to be different lenses on a shared latent action space $\mathcal{A}$, so small schema variations do not erase the core semantics being emulated.* We show an additional benefit that our approach can reconcile datasets with slightly different schema, allowing NLP researchers not to "waste" valuable annotations.

Overall, in this Section, we treat schemas as observational lenses. Treating each dataset as an observation model C_σ clarifies why combining them helps: PDTB–t contributes temporal orientation; RST contributes relational structure; Argumentation contributes analytic/explanatory cues—together reducing ambiguity over a. *The observed improvements thus support our original hypothesis that these lenses share substantial semantic content over a.* Multitask improves minority classes without hurting majority ones, unlike TDA/UDA, which overpredict frequent labels; TSA can partly stabilize UDA but does not match multitask. The view is not uniform, though — which auxiliaries help depends on the metric; we observe different optimal weightings for Micro vs. Macro, consistent with their sensitivity to class frequency; linear-regression coefficients on α_σ align with the qualitative roles above. Confusions shrink along *temporal* and *event-based* axes (e.g., **C2** vs. **M2**, **C1** vs. **C2**), matching the added observability from PDTB–t and event signals. Finally, post-reconciliation human agreement remains well below 100%, suggesting inherent ambiguity; nonetheless, MT–Macro approaches human accuracy on many labels while still trailing on **M1**/**M2**, indicating room for better temporal/event modeling or expanded $\mathcal{A}$.

4.6 Structural Discourse and Computational Law

...in counties having a metropolitan form of government and in counties having a population of not less than three hundred thirty-five thousand (335,000) nor more than three hundred thirty-six thousand (336,000), according to the 1990 federal census or any subsequent federal census, the magistrate or magistrates shall be selected and appointed by and serve at the pleasure of the trial court judge...

Figure 4.21: Paragraph from a sample law, Tennessee § 36-5-402, referencing a bureaucratic process impacted by population counts determined by the upcoming federal census. The colored blocks represent the following legal discourse elements from our schema: PROBE, TEST, SUBJECT, CONSEQUENCE, OBJECT (see Section 4.6.1). We train LLMs to identify these spans and build a web application to aggregate these span tags across state-level laws.

Having both introduced new methodologies for *transition-modeling* and offered another lens to justify the use of discourse schemas, we now close with a lighter, "bonus" section, showing how emulation can be utilized for creative, interpretive tasks *outside* of journalism, specifically computational law. AI practitioners have long explored how to use automation to *interpret the law*[64] [585]. Recent advances in NLP and information retrieval have already enabled practical applications [586], such as legal question answering bots [65], contract generation[66], and automatic motion-filing [587]. The legal reasoning capabilities of large language models (LLMs) are promising [588, 589] – GPT4 has been demonstrated to pass the bar exam.

However, fundamental challenges remain. As noted by [590], GPT3 models fail when confronted with simple, yet ambiguous conditions (or "tests") present in legal rules [591], a challenge documented in other models as well [592, 593]. Additionally, the majority of legal study has been focused a few specific domains, like contracts [594, 595], privacy policy [596, 597], and corporate law [598], and the kinds of tasks heretofore studied have

[64] Specifically: legal codes, court opinions and regulations.

[65] https://www.chatbotsecommerce.com/nrf-launches-parker-first-australian-privacy-law-chatbot/

[66] As well as other documents: documents – i.e. laws, court opinions and regulations https://legal.thomsonreuters.com.au/products/contract-express/, https://turbotax.intuit.com/

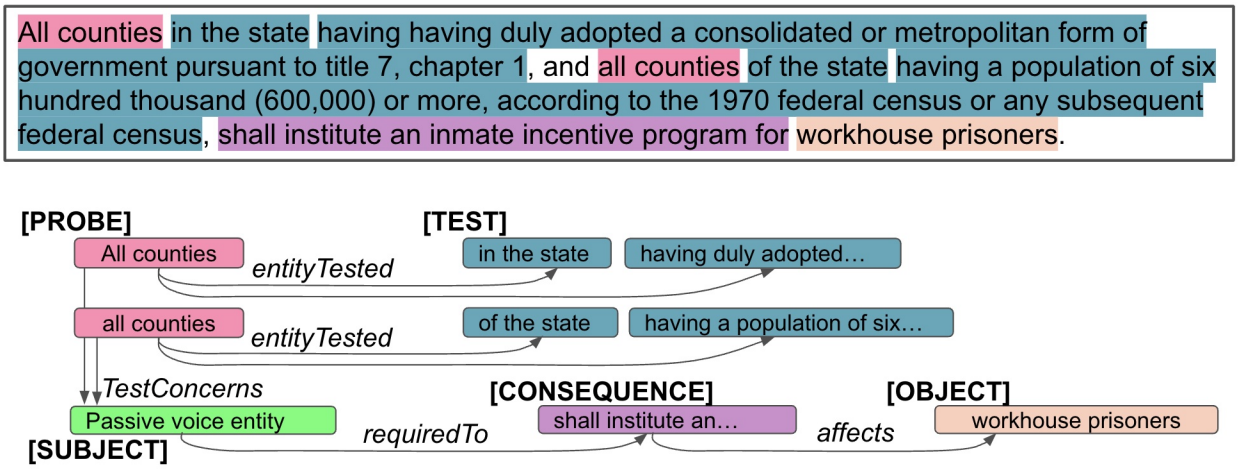


Figure 4.22: A sample span-and-relation discourse tree generated from a paragraph of legal text. Above, the highlighted text shows the original law text with discourse-spans annotated. Below, relations are drawn between discourse blocks, shown with double-black curved lines and categorically annotated. Note that the SUBJECT responsible for carrying out the CONSEQUENCE is passively implied.

been highly domain specific[67]. Benchmarks like [588] are dominated by these use-cases, limiting our ability to get a *general* assessment of a model's abilities. It also limits our confidence about models' reasoning in understudied legal domains which are important to policy makers, journalists and academics, like state-level administrative law.

We see the need to introduce a unified mode of study that can quickly incorporate new areas and applications of law. *Discourse analyses*, or the study of functional role of text and its relations within in a document [556, 555], has been successfully applied to areas like argumentation [599], dialogue [600] and journalism [446, 145]. In journalism, for instance, we used discourse schemata in Sections 3.4, 3.5, 4.2 and 4.3 to describe textual relations and drive *emulation learning* for various tasks.

In this work, we develop a *legal discourse schema* for characterizing a legal text, which we apply to state-level legal texts. At the core, our schema seeks to answer the following key questions: (1) When does a law apply? (2) What are its consequences? (3) Who is affected? We show that large language models struggle to model this schema, yet it is

[67] An example of a domain-specific task: *"Classify if the clause limits the ability of a party to transfer the license being granted to a third party"* from [595].

useful for human practitioners. In Section 4.1, we argued that discourse relations map to writer actions. Here, our discourse schema corresponds to a sequence of writer actions $\boldsymbol{\tau} = (\mathbf{a}_1, \ldots, \mathbf{a}_T)$ that produced the observed legislative text $\mathbf{g}$ and that, we show, give us deeper insights into to *intended meaning* of this text. Our inverse function is $q_\theta(\boldsymbol{\tau} \mid \mathbf{g})$, which seeks to recover these actions from the observed text.

This Section unfolds as follows. We outline our discourse schema and modeling in Section 4.6.1. Next, we discuss our dataset collection process, including the web-scrapers we release for gathering public-domain U.S. state law text (Section 4.6.2.1). In Section 4.6.2.2 we describe our lightweight and modular span and relation annotation interface which we used to collect data. Next, in Section 4.6.5, we describe our web-app, where we surface our model's output to journalists and engage volunteers to improve our annotations. Finally, we discuss an ongoing use-case to illustrate how one might use our app in Section 4.6.5.1.

4.6.1 A Legal Discourse Schema

A legal rule is a *hypothetical imperative* [601], or a conditional consequence. Reasoning about these rules requires practitioners to understand how and whether conditions of the law are met; what the consequences are [590]; and *who* is affected by these consequences.

As shown in Figure 4.22, modeling the different components of a legal doctrine as *discourse units* and how they interact as *relations* can be an effective way of discern meaning [556, 555]. Identifying these parts poses a basic test of a model's legal reasoning and can also lead to practical use-cases (as [446] showed in the journalism domain). We introduce the key parts in our schema, starting with span annotations and then relations.

4.6.1.1 Span-Level Schema

A *span-level discourse element* corresponds to a span action $\mathbf{a}_{\text{span}}(s_i, z_i)$ that specifies a micro-intention of the writer: who they intend to affect and what the effect will be. In our generative story of legal texts, the writer (1) first selects a sequence of discourse actions

Edge Case Type	Example
Passive Subject and Object:	*Taxes shall be collected at the beginning of every month.*
Subject-Consequence relation without an Object:	*The trial court judge shall begin session at or before 9am.*
Subject-Consequence-Object relation $>$ 1-hop	The magistrate shall designate to the county clerk, who shall adjudicate among taxpayers

Table 4.16: **Edge Cases and Extensions**: Our discourse schema flexibly handles different variations of legal expressions. Shown here are variations of the Subject-Object-Consequence relation. In the top variation, the Subject and Object (i.e. "Tax-collector" and "Tax-payer") are not actively expressed. In the middle relation, no Object is entailed. In the bottom relation, a multi-hop relational chain is formed.

$\boldsymbol{\tau} = (\mathbf{a}_1, \ldots, \mathbf{a}_T)$ (for example, "create Test on Probe", "link Subject to Consequence"). Multi-hop Subject→Consequence→Object chains arise by composing actions in $\boldsymbol{\tau}$. (2) These actions are realized into surface form as the observed paragraph $\mathbf{g}$. (3) Given $\mathbf{g}$, we recover $\hat{\boldsymbol{\tau}}$ using the inverse function $q_\theta(\boldsymbol{\tau} \mid \mathbf{g})$. Supervision uses gold actions $\mathbf{a}^\star$ for a subset of spans and relations. The eight discourse elements we identify are Subject, Object, Probe, Consequence, Test , Exception, Definition and Class. The first three are entities (noun phrases); the others are predicates (verb or prepositional phrases). Subject, Consequence and Object capture first-degree interactions between entities, inspired by [602]. We describe each a_{span} in turn.

- A Subject is an entity that gains powers or restrictions under a law. (e.g. *"The **trial court judge** shall adjudicate property disputes between claimants."*) Subjects aren't always explicit, and can be expressed passively (see Table 4.16 for examples of edge-cases).

- The Consequence is the specific power or restriction conferred by the law. Consequences nearly always are attributed to the Subject , either passively or explicitly. (e.g. *"The trial court judge **shall adjudicate property disputes** between claimants."*)

- An OBJECT is an entity (noun phrase) affected by the SUBJECT , under a law. Typically, when the SUBJECT gains powers, the OBJECT usually faces more restrictions; if the SUBJECT faces restrictions, the OBJECT usually faces fewer restrictions. (e.g. *"The trial court judge shall adjudicate property disputes between* ***claimants.****"*) Like SUBJECTS, OBJECTS are not always present in the text, or might be expressed passively.

Often, the SUBJECT-CONSEQUENCE-OBJECT involves a longer chain than a 1-hop relationship (for an example, see Table 4.16)[68]. In these cases, an entity is both an OBJECT and a SUBJECT . We label this entity as an OBJECT to prioritize the first CONSEQUENCE . The next three elements in our schema, TEST , PROBE and EXCEPTION , indicate when laws apply.

- A TEST is an explicit condition applied to an entity (i.e. an OBJECT, SUBJECT or PROBE) that determines *when* a SUBJECT -CONSEQUENCE -OBJECT relation holds. (e.g. *"In counties* ***with a population above 10,000****, the trial court judge shall adjudicate... unless claimants settle."*

- A PROBE is an entity to which a TEST is applied to that is *not* a SUBJECT or an OBJECT. If the TEST is applied to a SUBJECT or an OBJECT, there may not be a need for a PROBE . *"In* ***counties*** *with a population above 10,000, the trial court judge shall adjudicate... unless claimants settle."*

- An EXCEPTION is a corollary to a TEST ; it specifies when a law does NOT apply. An EXCEPTION usually modifies a TEST *"In counties with a population above 10,000, the trial court judge shall adjudicate...* ***unless claimants settle****."*.

Finally, the remaining two classes in our schema, DEFINITION and CLASS, serve to more fully characterize the entities mentioned in legal text. These terms have already been well-described in the literature [603, 590] and incorporated into tasks [588]. We give definitions in and examples of all span-level discourse types in [129].

[68] Example of a SUBJECT-CONSEQUENCE-OBJECT that is greater than 1-hop: "The magistrate shall designate to the county clerk, who shall adjudicate among taxpayers".

4.6.1.2 Relational Schema

Links *between* spans arise from a more macro-level action, a_{rel}. Here, the writer intends to specify how these spans relate and creates a greater meaning in the text. Cross-type (e.g., Test $\rightarrow$Probe) and same-type (e.g., Test $\wedge$Test) links are different families of $\mathbf{a}_{\text{rel}}$ but share the same inverse predictor $q_\theta^{\text{rel}}(\mathbf{a}_{\text{rel}} \mid \mathbf{g})$. We define 21 relational categories during our annotation process. There are two categories of relations. (1) The first category occurs between discourse units of *different types*. The type of these relations is usually singular based on the type of the discourse units (e.g. a Test -Probe relation means that the Test is being applied to the Probe entity), so we do not enumerate them here (we give full definitions in [129]). (2) The second category applies between discourse units of the *same type*. These are typically simple grammatical or logical relations. For instance, **sameEntity** indicates that two entities are instances of the same class of entity or the same instance of an entity. **Or**, **And** refers to how two predicate interact (e.g. if Test $_1$ *OR* Test $_2$ is passed...).

4.6.1.3 Parsing Level

Our framework can be conceptualized recursively, with spans being further parsed, tree-like [604]. For example, a Subject "*trial court judge*" can be also interpreted as "*trial court* *judge*". We define the parse-level in relation to the interpretation of the law. For instance, if "*trial court judges*" are being compared with other judges, e.g. "*county judges*", we need the "*trial court* *judge*" and "*county* *judge*" parses, which create conditions for comparison.

4.6.2 Dataset Creation

In this section, we describe how we operationalized the schema discussed in Section 4.6.1. We scrape a dataset of all state-level laws from 52 U.S. states and territories, which we discuss in Section 4.6.2.1. We then sample a set of paragraphs to annotate. We build an annotation framework, described in Section 4.6.2.2, and enlist four annotators, who

collectively annotate 602 law paragraphs.

4.6.2.1 Dataset Construction

Our full legal dataset comprises the more than 100,000 active state-level laws in the United States. We compile this dataset by building a scraper for a public-domain law website called Justia.[69] We then manually audit the output collected by Justia by comparing to state websites and find 19 states where either Justia is incomplete, not updated, or unparsable.[70] We build individual state-level parsers for these states. State law is public domain,[71] yet it is often inaccessible for bulk downloads and web scraping. For instance, many websites license LexisNexis, a for-profit company, as the official provider for their state codes[72]. Although these websites are publicly accessible, they employ a range of mechanisms (e.g. timeouts, dynamically-generated URLs, cookie-based access) that make them difficult to scrape.[73] To circumvent these, our scrapers are robust and mimic human web-browsing behavior. We develop a generalized scraper for LexisNexis Public Access websites using `scrapy`[74] and `selenium-webdriver`[75]. In order to scrape Justia, we launch three Google Compute Engine (GCE) instances for a total of 60 compute hours[76].

[69] `https://www.justia.com/`

[70] Some of the laws provided by Justia, such as those for Colorado, contain data in PDF files (see `https://law.justia.com/codes/colorado/2019/`), which, due to formatting, have a high OCR error rate, so in these cases we we extract directly in these cases.

[71] `https://fairuse.stanford.edu/overview/public-domain/welcome/`

[72] Ex. Colorado, Georgia and Tennessee: `http://www.lexisnexis.com/hottopics/colorado`, `http://www.lexisnexis.com/hottopics/gacode`, `http://www.lexisnexis.com/hottopics/tncode`

[73] The practical effect of mechanisms to block bulk downloads is the hindrance of law corpora collection for journalistic or academic study.

[74] `https://scrapy.org/`

[75] `https://www.selenium.dev/`.

[76] We will release our code for scraping with Docker images created to perform these scrapes. Given the difficulty in creating this dataset, we believe these routines constitute a considerable resource for academic inquiries into state-level law.

	% annots	% of docs	# / doc
TEST	28%	91%	2.4
SUBJECT	20%	95%	1.7
CONSEQUENCE	19%	83%	1.8
OBJECT	15%	69%	1.7
PROBE	9%	46%	1.5
CLASS	6%	34%	1.5
DEFINITION	2%	11%	1.6
EXCEPTION	1%	6%	1.1

Table 4.17: The prevalence of different discourse units across our annotated dataset. The left column shows the percentage of units across all annotations. Center shows the percentage of documents in our corpus that have at least one discourse unit. Right shows the average number of units per document, when present.

4.6.2.2 Annotation

We recruited 4 annotators, including one former journalist and 2 undergraduate researchers[77]. We trained all of the annotators for multiple rounds, until they were achieving above an 80% accuracy in both span and relation identification tasks, based on a gold-label set that we constructed. After reaching this agreement level, we begin accepting completed tasks from annotators. We had multiple rounds of conferencing throughout the period of annotation where we discussed edge-cases, and maintained a Slack channel throughout the annotation process that was continually monitored. Together, the annotators annotated 602 laws, with a 10% overlap, from which we calculated a $\kappa = .8$ We found that our annotators could learn to identify different span and relation levels in most contexts quite easily. However, most of the error and ambiguity of the annotation process derived from when to split spans into sub-spans (e.g. the TEST in: *"clerks of the superior court of the county* can be split further: *"clerks of the superior court of the county*). The decision to do so usually depends on many factors, e.g. if entities will be coreferenced elsewhere. Despite rounds of

[77]We compensated the undergraduate researchers fairly at a rate of $20 per hour through AMT, according to University policy

Relation	%
ENTITY ↔ PREDICATE	61
ENTITY ↔ ENTITY	20
PREDICATE ↔ PREDICATE	19

Table 4.18: Types of relations common in our corpus. ENTITY includes: Subject , Object and Probe . PREDICATE includes all others.

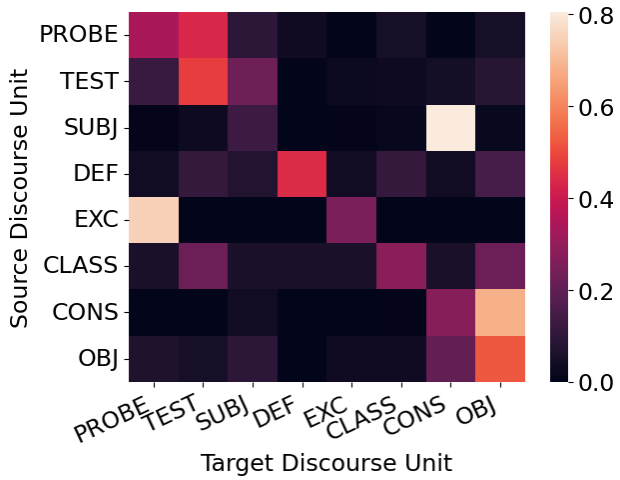


Figure 4.23: The conditional likelihood of a target discourse class, given a source discourse class. The color scale is $p(t|s)$ where s is the source node and t is the target node.

training, annotators still sometimes struggled; our directive in these circumstances was to parse to the lowest-level. See discussion in Section 4.6.1.3.

We built a Javascript-based framework to handle span and relation tagging and (1) serve as a standalone web-app for annotators (2) compile to Amazon Mechanical Turk (AMT) tasks[78] (3) integrate into a web-site built for journalists using our work (described in Section 4.6.5). Although many NLP-focused annotation tools exist[79] we found that none were flexible enough to be integrated easily into larger websites or automatically generate AMT tasks.[80] We plan to distribute our interface as a stand-alone Javascript package. For more details about the annotation interface, see [129].

[78] `https://docs.aws.amazon.com/AWSMechTurk/latest/AWSMturkAPI/ApiReference_HTMLQuestionArticle.html`.

[79] There were 87 frameworks as of [605]'s count, including BRAT [606], YEDDA [607] and WebAnnon [608]

[80] We will release the annotation code as part of this framework

4.6.2.3 Dataset Statistics

Corpus Description The length of the legal paragraphs we annotate averages 490 characters. The types of content that we focused on in our sample included topics on Government, education and environment. Certain states in our sample emphasized different topics. For example, California has a higher proportion of laws aimed at Poverty and Development compared with Tennessee, which has a higher proportion of laws focused on Administration (see [129] for more information and visualizations).

Discourse-level Analysis Discourse unit-level statistics vary widely. As can be seen in Table 4.17, TEST and SUBJECT are the most common discourse unit, accounting for 48% of all span-level annotations. TEST occurs in 91% documents. Surprisingly, EXCEPTION units were relatively rare, accounting for only 1% of annotations and occurring in only 6% of documents. There are many more TEST units per document, at 2.4 TEST units, than others.

Relation-level Analysis Next, we analyze the nature of the relations between discourse units. Two discourse spans are much more likely to directly relate if they are closer together in the law text. 62 characters, on average, separate discourse units with relations, while 195 characters, on average, separate all pairs of discourse units without relations. In Section 4.6.3.3, we describe how we balance our training datasets to remove this adjacency bias.

Figure 4.23 shows the likelihood of transitioning to a target discourse type, given a source discourse type. We order the x and y axes by the most likely starting points of discourse elements in a document (Discourse elements that are appear first in the document to be connected with discourse elements later. See [129] for more information). We see a strong diagonal bias: all discourse elements are likely to transition to elements of the same type. We also notice the strong SUBJECT $\rightarrow$ CONSEQUENCE and CONSEQUENCE $\rightarrow$ OBJECT relation, as well as the PROBE $\rightarrow$ TEST relation. This reinforces insights by [602], [601] and [590] about the key role of *hypothetical imperative* language in legal texts (discussed in Section 4.6.1).

	SUBJECT	CONS	OBJECT	PROBE	TEST	EXC	Macro	Micro
Baselines								
ASP [610]	35.7	39.4	26.3	38.9	44.6	33.3	37.7	36.6
PURE [592]	41.5	45.2	25.0	56.1	17.3	36.4	34.3	36.5
GPT3.5								
0-shot	34.4	9.7	14.8	13.4	35.4	54.7	27.1	22.7
3-shot	31.7	23.3	20.4	28.2	43.9	46.2	32.3	30.1
5-shot	30.7	24.1	15.9	30.8	49.8	45.2	32.8	30.8
8-shot	29.7	23.4	15.8	33.5	48.4	53.8	34.1	31.0
GPT FT	42.1	49.9	35.9	34.9	53.0	56.0	45.3	44.3

Table 4.19: F1 scores shown for span-identification for our 6 primary discourse elements: SUBJECT , CONSEQUENCE , OBJECT, PROBE , TEST and EXCEPTION . Average Precision, Recall and F1 across all samples are shown. Although fine-tuning improves performance across most categories, leading to +10-point increases in macro and micro f1-scores, although some, like EXCEPTION , are able to be handled relatively well even in zero-shot settings. F1 scores are still below human levels of agreement.

On the other hand, we find that several categories of relation are simply unlikely to ever occur. For instance, EXCEPTION is almost never applied to CONSEQUENCE . We hope in future work to investigate if these patterns hold up across a wider body of legal text. See [129] for more details. We TEST the implication of this in Section 4.6.3.3.

4.6.3 Legal Entity and Relational Modeling

We frame a new task using the data we collect: Legal Entity and Relational Modeling, or *extracting legally significant spans and their relations*. This task is analogous to end-to-end relation extraction (ERE) [609]. We will first describe two subtasks that traditionally compose ERE, and how legal discourse can be modeled in this framework, then we will discuss methods, with a particular focus on how we can use this setup to interrogate the reasoning capabilities of large language models.

4.6.3.1 Tasks and Datasets

Span-Level Tagging Given the observed text g and candidate spans $S = \{s_1, \ldots, s_m\}$, we train the span head $q_\theta(\mathbf{a}_{s_i}^{(\text{span})} \mid \mathbf{g}, s_i)$ to predict span actions. Let a be the set of actions we supervise — we use a subset of our discourse tags: SUBJECT, CONSEQUENCE, OBJECT, TEST, PROBE and EXCEPTION. We focus on these types because they have more within-text impact compared with DEFINITION and CLASS, which are primarily about adding context and helping to reason across texts [611]. For each candidate span s_i, we predict a type $a_{s_i}^{(span)} \in \mathbf{a} \cup \{\epsilon\}$, where ϵ is the null class. Supervision uses gold span actions $a_{s_i}^{\star}$. We optimize the inverse by minimizing the negative log-likelihood

$$\mathcal{L}_{\text{span}}(\theta) \;=\; -\sum_{a_{s_i} \in \{a_{s_1}, a_{s_2}, \ldots\}} \log q_\theta\big(\mathbf{a}_{s_i}^{(\text{span})} \,\big|\, \mathbf{g}, s_i\big)\,.$$

In legal reasoning, this subtask can help test a model's awareness of the function of — or *action generating* — each span of text. We filter our task dataset so that each document has at minimum two of the primary 6 spans, and we additionally remove spans that are at most one word, as these were the most ambiguous for our annotators to agree on. The ambiguity, we observed, was primarily due to annotator disagreement around how far each span should be parsed, discussed in Section 4.6.1.3 and 4.22. This filtering leaves us with 3,559 spans across 413 documents. We measure classification accuracy using F1 per class, and we consider a span to be valid if it contains 80% of more of the same words as the gold-annotated span, after removing stop words and punctuation, and is no longer than twice in length.

Relation Extraction Let R be a set of pre-defined relation types. For every pair of spans $s_i, s_j \in S \times S$, we seek to predict a relation-type, $q_\theta(a_{s_i,s_j}^{(rel)} | s_i, s_j, g) \in \{R, \epsilon\}$, where ϵ is the null class. We consider two versions of this task: *detection* and *classification*. Detection asks whether some rel exists, that is, whether $\mathbf{a}_{s_i,s_j}^{(rel)} \neq \epsilon$; classification asks which r instantiated

the action $r = \mathbf{a}^{(rel)}_{s_i,s_j}$. This can test how well model identifies which other spans are modified by a given span. We train the relation head $q_\theta(\mathbf{a}^{(\text{rel})}_{s_i,s_j} \mid \mathbf{g}, s_i, s_j)$ and evaluate F1 for detection and for classification. To construct a challenging legal relation classification dataset, we take a subset of relations $\hat{R} \in R$ that are observed occurring between span pairs of different span-types. This allows us to focus less on modeling the semantics of each span's type and more on the relation between them. We sample negatives, i.e. $a^{(rel)}_{s_i,s_j} = \epsilon$, and notice that discourse units that are more proximal in the text are more likely to be related, as noted in Section 4.6.2.3. We find in early trials that our models were overfitting to proximity in text and not generalizing well to cases where relations are more distant. So, to make the task more challenging, we sample negative examples that the same distribution of offsets our labeled examples: in other words, so that the character-offset distribution $|\text{pos}(s_i) - \text{pos}(s_j)|$ of negative pairs matches that of positives. We are left with 1,482 datapoints. We measure model accuracy using F1, focusing on three main groupings: relations between entities and entities (ENT↔ENT), relations between entities and predicates (ENT ↔ PRED) and relations between predicates and predicates (PRED ↔ PRED).

4.6.3.2 Baselines

Relation extraction is a widely studied field, with classical and current work focusing on modeling each subtask separately [612, 613], as well as end-to-end modeling [614]. As such, we build upon two recent methods focused on each approach:

PURE [615]: separately models two embedding spaces, one focused on span identification and the other focused on relation extraction, using masked language modeling [285].

ASP [610]: trains a generative T5 model [616] to create structured predictions.

4.6.3.3 Generative Modeling

Recent work has shown that large language models can also be effective relation predictors [617]. To test this hypothesis, and to add to a growing body of work focused on bench-

marking LLMs for legal tasks [588], we parameterize the inverse function $q_\theta(\boldsymbol{\tau} \mid \mathbf{g})$ with GPT-style models and instantiate task-specific heads $q_\theta(\mathbf{a}_{s_i}^{(\text{span})} \mid \mathbf{g}, s_i)$ and $q_\theta(\mathbf{a}_{s_i,s_j}^{(\text{rel})} \mid \mathbf{g}, s_i, s_j)$ in order to fine-tune GPT3.5 models[81]. We format each action prediction as constrained generation. For span prediction, the model lists all spans for a given discourse type. For example, for $a_{span}(s) =$ SUBJECT , we prompt with the question: `You are a legal assistant. I will show you a paragraph of law. Which entities gain powers, restrictions or responsibilities under this law? <Legal Text>`). Additionally, as each law may contain several discourse elements of the same type, we ask the LLM to generate *all* elements of a certain discourse type in mentioned in the given law. For prompts for all relation-types, filled in with examples, see [129]. For relations, the model answers a yes/no detection query (detection) or selects a relation type from a closed set (classification). We evaluate zero-shot, few-shot and fine-tuned settings with identical train/test splits to the baselines. In other words, for relation *detection* we generate a "Yes"/"No" indicator, $I \sim llm(s_1, s_2, g)$ if a relation is present between two spans. We construct a prompt where the LLM is given the legal text and two discourse elements, and ask if they are related. Our prompt is: "`Are span A and B related in Law X?`". For *classification* we generate the relation-type, $r \sim llm(s_1, s_2, g)$. In other words, our prompt is: "`What is the relation between span A and B in Law X? Answer from the following set: {..., 'no relation'}.`"). We include $\epsilon \in R$ so that our experiments with GPT are comparable to the baseline models. We test two different prompt settings. In the first setting, we simply give the two spans of text and the law, and ask the LLM to determine if they are related. In the second setting, we give the LLM the class labels of the discourse units, as well as definitions for what each label means (**w. def**, in Table 4.20). See [129] for all relational prompts, with examples. We test both tasks in zero-shot, few-shot, and fine-tuned settings[82] and for each test sample, we repeatedly query the LLM for 3 trials, randomizing the few-shot examples it receives.

[81] Specifically, we use `GPT3.5-turbo` as of October 11, 2023.

[82] For fine-tuning experiments, we use GPT3.5's finetuning endpoint, which prompts OpenAI to fine-tunes GPT3.5 under the hood. This requires us to upload a file of {"`prompt`":<>, "`completion`"} pairs. We generate this file using the prompting structure described above, with the same train splits used in baseline trials.

	ENT ↔ P		ENT ↔ ENT		P ↔ P		All (Macro)		All (Micro)	
	Det.	Cls.	Det.	Cls.	Det.	Cls.	Det.	Cls.	Det.	Cls.
Baselines										
ASP	26.5	14.2	4.5	3.8	4.0	2.2	13.6	6.7	19.5	11.1
PURE	73.9	64.5	15.4	5.3	45.7	38.2	49.5	40.5	63.1	53.9
GPT3.5										
0-Shot	54.9	0.0	42.5	27.1	25.2	23.2	40.8	16.8	48.5	7.2
0-S +def	69.4	0.0	54.2	39.5	60.8	48.2	61.5	29.2	65.1	12.8
10-Shot	50.6	55.3	56.8	53.9	40.2	34.2	49.2	47.8	50.5	51.7
10S +def	72.6	60.1	68.5	65.9	65.1	35.2	68.7	53.7	70.8	56.7
GPT-FT	82.6	85.9	76.5	88.7	81.0	65.9	80.0	80.2	81.1	82.9

Table 4.20: **Relation Detection and Classification** F1 score. We examine scores between three categories of relations: ENTITIES ↔ ENTITIES, ENTITIES ↔ PREDICATES, and PREDICATES ↔ PREDICATES. ENTITIES are SUBJECT, OBJECT and PROBE, and PREDICATES are all other discourse types. Classification is only run for discourse-type pairs where more than one relation can exist (see Section 4.6.1).

4.6.4 Results and Discussion

Span-Level Tagging: Table 4.19 shows F1 scores from our span-tagging experiments. Interestingly, our inverse model $q_\theta(\boldsymbol{\tau} \mid \mathbf{g})$, via its span head $q_\theta(\mathbf{a}_{s_i}^{(\text{span})} \mid \mathbf{g}, s_i)$ underperforms trained annotators on identifying span actions even after fine-tuning. Distinguishing entity roles (SUBJECT, OBJECT, PROBE) is notably harder than predicate types (TEST , CONSEQUENCE, EXCEPTION): GPT was especially challenged by distinguishing between different entities' roles: SUBJECT , OBJECT and PROBE (GPT Fine-tuned scores 35-42 F1 on entities, compared with 50-59 F1 for predicates. EXCEPTION stands out as a particular category where even 0-shot GPT performs well.) SUBJECT and OBJECT roles can be particularly ambiguous, consistent with the edge-cases in Section 4.6.1, as there are cases when an entity can be in both a SUBJECT and OBJECT role (we annotated OBJECT, in those cases). Interestingly, too, the gap between GPT and the baseline models is not as large in this task than it is in relational modeling. Perhaps our generative setup for this step, $p(s|\zeta, X)$, with 6 different

prompts, allowed GPT to generate the same entity for different categories. We might see improvements with a post-hoc disambiguation step that predicts a_i given $(s_i, \mathbf{g})$, when a single span is generated in multiple categories. Our broader finding, though, is that this remains a challenging task. Although our task dataset, at 400 documents, is small relative to other language resources, the spans in our schema are syntactically low-level. The spans divide relatively well into different parts of speech, like noun phrases and verb phrases; identifying such chunks in text has long been within the capability of even classical language models [618]. Future work either fine-tuning on other resources, or using law-specific models, might show improvements in these areas.

Relation Identification and Classification Table 4.20 show F1 scores from relation detection (Detect) and classification (Class). For relation actions a_{rel}, $q_\theta(\boldsymbol{\tau} \mid \mathbf{g})$, via its relation head $q_\theta(\mathbf{a}^{(\text{rel})}_{s_i,s_j} \mid \mathbf{g}, s_i, s_j)$ approaches (and sometimes matches) human annotators. In other words relation extraction is a category where fine-tuned GPT performs just as well as our annotators. We notice, too that in some cases GPT does even better on the classification task than it does on the identification task (e.g. ENT↔PRED and ENT↔ENT). It's possible that the semantics of classification task enforce greater reasoning and justification than the identification task [219]. The relation identification task also shows a clear different between the baseline models, which we do not observe in the span-level tagging task. One explanation for the especially poor performance of ASP [610] is that the jointly learned model requires the model to make use of more data to fully learn the embedding layers. In fact, tasks that ASP performs well on, like ACE2005 [612], have $\tilde{1}$0x more documents and annotation than our dataset. We show more details in [129].

4.6.5 Practical Use Case: Census 2020

To get feedback on our work from a preliminary group of users, we apply our models to a domain of state-level law pertinent to journalists. In 2020, the U.S. Census count faced multiple challenges, notably the Trump administration's attempt to add the question: *"Are*

you a legal citizen?". Many researchers hypothesized that populations, especially minorities, might be inaccurately counted [619, 620, 621]. Scant insight existed, especially on the state-level, into how population counts were being used in law[83]: the corpus of state-level laws was too large and varied for journalists to parse. On the other hand, this provided an interesting case for discourse-based reasoning. Population counts typically get used as a relatively unambiguous TEST . For example, see Figure 4.21, e.g. *"In counties with less than 20,000, adjudicators shall.."*. Our discourse models help us identify this occurring, and then we can develop ways to parse out the specific ways population is in TEST discourse. We describe the website we built to facilitate different explorations, and then we describe two such explorations that we received permission from the journalists collaborating with us to write about. We will focus on our own contributions in these collaborations.

4.6.5.1 Website Design

We design a website[84], shown in Figure 4.24, to enable exploration of our dataset and modeling output. Users can (1) perform full-text search on all laws in our database, (2) view the spans our models have extracted, by their discourse role, across laws and (3) correct or provide new annotations. Users interact with the inferred schema extracted from (e.g., enumerate TEST thresholds; trace SUBJECT →CONSEQUENCE paths) rather than raw text alone. For more detail on the website, including flow diagrams, see [129]. The website's overall goal is to facilitate both ***deep* explorations** and ***wide* explorations**.

Going deep: Going "*deep*" here, essentially, means finding a subset of the laws to study first, via keyword filtering, and *then* analyzing the discourse relations within the laws. The web search functionality[85] helps users do this by exploring a specific term or concept in the law's plain text or in specific discourse role (e.g. laws affecting OBJECT="taxpayer"). After the user finds an interesting subset of laws they wish to study, we use our discourse

[83]Besides federal budgeting and Congressional representation, which have already been manually programatized [622, 621].

[84]To view the website, see: `http://www.statecensuslaws.org/`

[85]Powered by ElasticSearch [623]

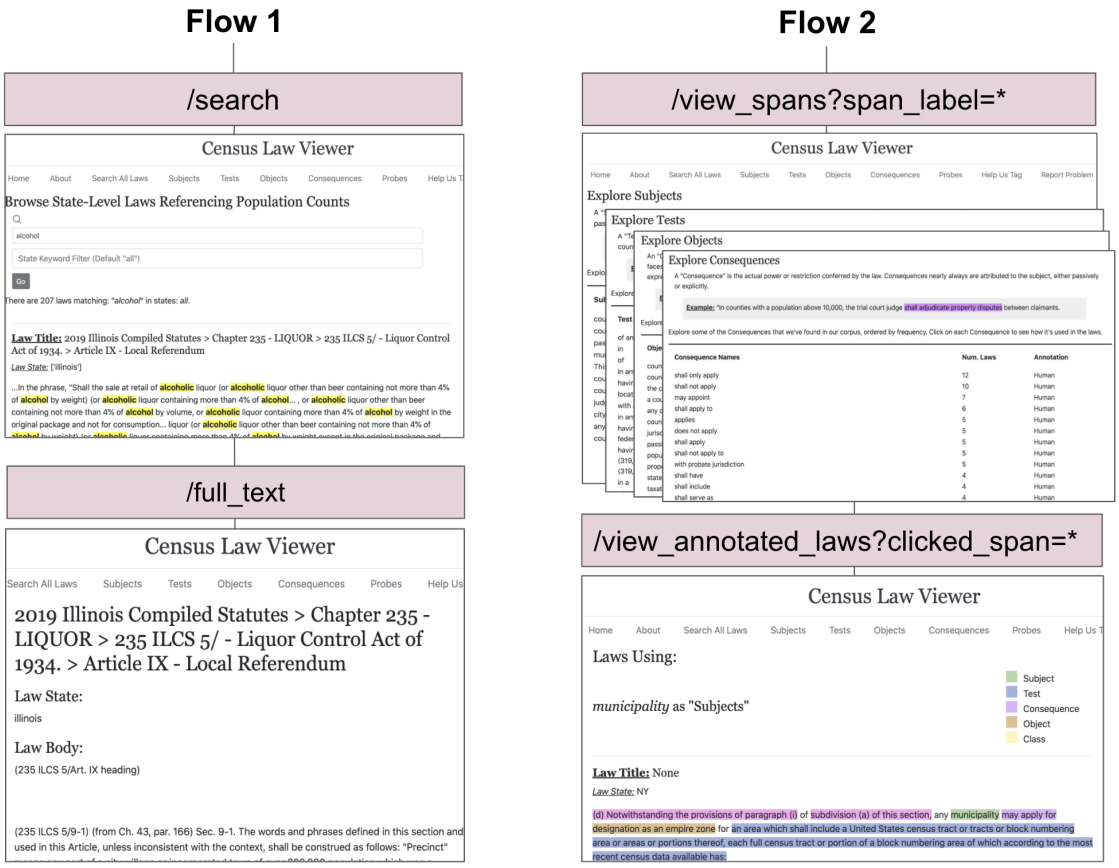


Figure 4.24: A flow-based sitemap for our website, `statecensuslaws.org`, with some details about the back-end and database setup. The left-column shows **Flow 1**, where a user can search and view full-text results. The right-column shows **Flow 2**, where a user can view top law-discourse spans. Each flow leads to the annotation framework.

models to answer: *who* is being affect, under *what conditions*, and *how*?

Going wide: Conversely, going "*wide*" means studying discourse units and relations first, then analyzing the laws. The website includes a second functionality: allowing users can view aggregate counts of different discourse units and relations. This helps users notice patterns among the ways in which discourse was being used. After a user notices a specific pattern in discourse roles (e.g. EXCEPTION units modifying TEST units about taxes), then we can analyze the laws that include, or do not include, these elements. In both flows, visitors can access our annotation framework, described in Section 4.6.2.2, which helped us gather more data. We now describe two example articles explored by users of our system.

Case Study #1: Going Deep (Liquor Store Licenses)

In the first example, journalists hypothesized that the allocation of new liquor licenses might be population-based. To explore this, they used the search interface; they searched for the term "alcohol OR liquor OR beverage" in the search interface and discovered that interface returned 270 laws. Together, we analyzed the breakdown of liquor-related law by state. We found that the states most likely to base liquor licenses off population counts were Tennessee, New York and Illinois. They then asked us to extract all TEST S from these laws. We found that mid-size cities would be the most likely to be impacted by a 5% or 10% undercount in population. The journalists identified key cities and are seeking sources in these areas.

Case Study #2: Going Wide (Slim Population Thresholds)

In another example, journalists explored the top-level discourse annotations. They noticed that some TESTS are based on explicit population thresholds (ex. Figure 4.21) and that some of these thresholds were very narrow. We identified all TEST s in our dataset, using our discourse schema. We then compiled several keyword filters and regular expressions extract specific population thresholds.

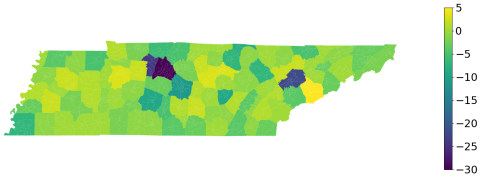


Figure 4.25: **Illustration of a Use-Case**: A heatmap of the state of Tennessee, colored by the number of laws that would no longer apply in counties, if a 5% undercount in the census were to occur. Counties with Nashville and Knoxville are particularly effected. Population-related TEST s were identified using our discourse framework.

We found that in Tennessee, in particular, over 40% of all Census-related laws imposed narrow population tests of fewer than 500 people (e.g. *"for counties with no less than 400,000 and no more than 400,500 inhabitants"*) and 10% imposed tests of fewer than 100 people. We show in Figure 4.25 a vivid illustration of the number of laws that would be affected with a 5% undercount in population, based on population projections made prior to 2020 [624]. As can be seen, major population centers like Nashville and Knoxville are the most affected centers. This raised questions:

what is the purpose of these narrowly targeted laws? Were they trying to target specific counties without mentioning them by name? The journalists are now investigating further by tracking down the authors of these laws.

4.6.6 How does a discourse approach fit within the broader computational law field?

Although the field of AI-driven legal aids is multifaceted and growing [625], free and open-source frameworks remain few [626, 586, 627]. Our discourse-driven web application, designed for legal exploratory analysis is one of the few AI-powered, free applications that exist, and the first to open source tools for legal document collection. For-profit legal inquiry systems, as mentioned above, are numerous. Bloomberg Law[86], Westlaw[87], LexisNexis[88] and Wolters Kluwar[89] are the four main services for legal research [586], which provide subscription-based, Google-style searches. CaseText[90] and Ravel[91] were two upstart case-text search engines (although both have now been aquired); CaseText offered crowdsourced annotations and Ravel linked cases together to create visual maps of important cases [628]. We similarly provide a way of collecting user-annotations, and a novel way linking together cases, although ours takes a discourse approach rather than an unsupervised clustering approach.

Various discourse schemas have been developed to understand law texts, including deontological logic-based schemas [629, 630], and subject matter-specific schemas [631]. Ours is the first discourse-based approach to take steps towards a big-data approach by setting up a framework for the ingestion of crowdsourced annotations. Finally, outside of the legal domain, other areas have experienced a growth in academically-oriented

[86] https://pro.bloomberglaw.com/
[87] https://www.westlaw.com/
[88] https://www.lexisnexis.com
[89] https://www.wolterskluwer.com
[90] https://casetext.com/
[91] https://home.ravellaw.com/

systems for human-in-the-loop inquiry. The COVID-19 pandemic has produced a burst in NLP-driven corpora-collection [632], demonstrations [633, 634, 635] and workshops [636, 637]. Such concerted effort in the NLP domain to expose resources and build open tools for subject matter experts is an inspiring guide for how NLP researchers can contribute to wider inquiries. We hope such efforts expand to other domains as well, forming a common alliance between academics, civil-minded journalists and other researchers and end-users.

Summary We have sought to take steps towards a semantic understanding of legal texts [602]. Framed as emulation, our objective is to recover the writer's discourse actions a from the observed law g via $q_\theta(a \mid g)$. While $q_\theta(a|g)$ excels at relation actions, role-level span actions remain challenging and likely benefit from stronger structural constraints or richer supervision. We show that large language models, while achieving impressive results in some parts of our task, show surprisingly weak performance compared to human annotators in others. Language models have an important role to play in interpreting law and lowering the barrier of access to legal systems. Our task is an important step towards assessing a sturdy foundation and opening the door to more intensive legal tasks [588]. In this work, we have presented three open-source components. (1) A web-app exposing a novel discourse schema and its application to state law referencing U.S. Census counts. (2) A flexible and modular annotation framework that can be seamlessly embedded into web-apps to allow visitors to contribute and update annotations. (3) A set of web-scrapers to help researchers gather public-domain legal text. Our longer-term goal is to collect feedback and data, and improve our database and machine learning systems. We hope that such efforts can continue to push legal tech [638] into a more open and accessible domain, and make it easier to understand the laws governing our society.

4.7 Chapter Conclusion

In this Chapter, we explored how to *realize* a set of actions $\mathbf{a} = a_1, a_2 \ldots$ into changes in the state space $\mathbf{s} = s_1, s_2 \ldots; s_n = g$. Specifically, in this section we defined our action space $\mathbf{a}$ to be a representation of *human writing-structuring process* (e.g. an *outline* or *discourse structure*, where a_1 ="Give Background" and a_2 ="Write transition"). We introduced *three* methods to realize these actions into a *structured piece of writing*: in Section 4.2, we introduced a method for *sequential* control using the *inverse model* $q_\theta(a_t|s_{<t}, s_t, a_{-t})$ to *steer* an LLM's generations *towards* a more desirable structure. We followed this with a similar approach in Section 4.3 that *further* enforced not just *structural* actions but also *factual* consistency (represented as s_0, the starting state). We introduced in Section 4.4 a third, more general method, called *Classifier Free Guidance (CFG)* for NLP, which flexibly extends *beyond* discourse structures to any kind of multi-part structure, expressed in a prompt. After introducing these methods, we further interrogated the "rightness" of latent action vocabularies $\mathcal{A}$ for writing structure in Section 4.5, but took a different approach than in Section 3.5; here we made the point, actually, the specific choice of vocabulary *might not matter* as much as we think and different vocabularies have an underlying correlation that appears in multitask learning setup. Finally, in Section 4.6, we showed that structural analysis can be useful *outside* of *transition* or *policy* models; interpreting intent of the writer can yield novel analytical insights.

Looking forward, for AI to continue to make strides, generative models will need to maintain coherence over long passages: (a) to reason more effectively, (b) execute longer workflows, and (c) interact with other agents in agentic systems. *Planning* and *structured generation*, I believe, are important research topics to make progress in these directions. The *state space transition* framing that *emulation learning* sets up is a beneficial framing for such advancements, as it allows us to study the interplay between *human planning* (through inferred actions) and *human generation*. I see similar approaches as we explored

in this Chapter being able to model human conversational dialogues, human interaction systems, game-playing and intent systems and other *found* data online. I am also excited for approaches to synthetic data creation, which I believe can give us more insights into unobserved state space transitions. In this vein, Bayesian Wake-Sleep Cycle, discussed in Section 2.4, is again a promising candidate for training *state-space models* — Wake Sleep's Generator is a close parallel to the state transition model $P(s_{t+1}|s_t, a_t)$; the Generator *also* takes a *structured latent variable input*, z, and is tasked with learning a function to project it into the output (for a recap, please refer back to that section). Looking forward, these methods and others, I believe, will play an important role in improving methods for performing *state-space transitions*, which will continue to play a larger and larger role in AI systems.

Chapter 5

State-Space Observability in Emulation Learning

5.1 *News-Edits*: A Study in How Information is Updated

Even after the journalist has *found*, *sourced* and *structured* their story, the work is still not done. Stylistic and factual corrections need to be made; events in the world update requiring updates. As a practical matter, throughout this chapter, imagine the following use-case: a *breaking news* event – i.e. a broadly *newsworthy* event that updates quickly – is occurring, and a journalist needs to publish *and update* (or republish) their article *quickly*. Which *sources* does the journalist need to retrieve in order to craft the first version of the article? After how many versions, after the basic contours of the event have been established, is the audience ready for background contextualization? When will the article

Figure 5.1: In the *journalism pipeline* outlined in Section 1.3, we focus now on the final step step: *editing*, or updates that are made to news articles to correct errors, add information, make stylistic changes, and update facts. Observing edit patterns gives us insights into an article *updates* through time, giving us a more temporal granularity into *emulation*.

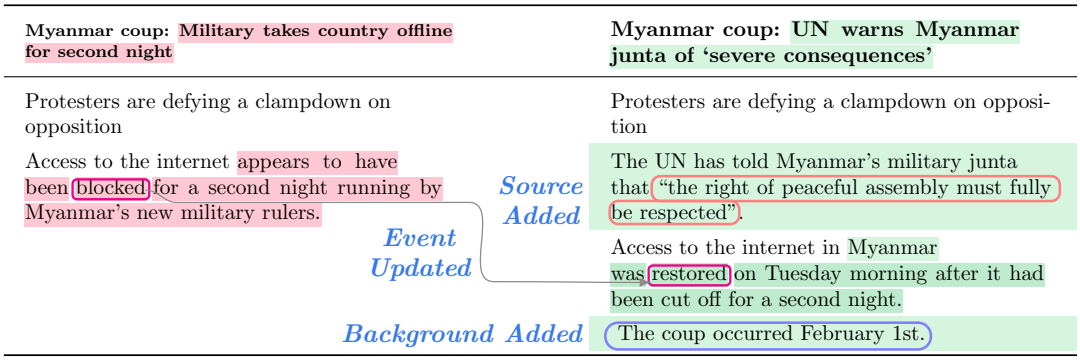


Figure 5.2: **Two versions of a news article** D^2 — t (left), $t+1$ (right) — covering a coup in Myanmar. Pink spans denote information that was removed or revised between versions, while green spans indicate information was added. The transformations shown (i.e. Event Update, Source Added, Background Added) are examples of actions taken in the article writing process *that we can assign to* t. Observing edits allows us to see how the state space of the article unfolds step by step, and to localize when actions a_t occur.

stop updating? Simply observing the final state, as we have done in previous chapters, will not give us any insights – we will not be able to understand what actions were taken *when*[1], just that they were taken *at some point*. The *news article* is seldom a static artifact [639], but is more a fluid, "liquid" narrative [640, 641, 642] that evolves over time according to formalized processes [643, 644]. Observe, for instance, two versions of an updating article (versions *t* and *t+1*) shown in Figure 5.2 covering a coup occurring in Myanmar: between *t* and *t+1*, *events update* (i.e. the internet, which was blocked, is now restored); a source is added (i.e. a quote from the UN); and background is added (i.e. more information about the coup). Observing this arc shows us how the framing, details and information provided change over time – even in the same article. We will now introduce an experimental setting that will allow us to study these updates in more detail. *Article versions* of news articles exist in online archives and are generated each time a news outlet republishes a story to the same URL [642]. Newsroom cultures have emerged that prioritize speed and efficiency, especially for "breaking news" articles [645, 646]. This means that many

[1]Recall, in prior sections, that we usually inferred temporality using heuristics: i.e. which source was a *major* source, or which structural discourse element occurred first.

[2]From the *BBC*. February 16, 2021. `https://www.bbc.com/news/world-asia-56074429`

more article versions are generated for these events throughout the coverage arc, forming rich histories. These can provide a wealth of information about evolving world states and actions driving the article forward; while article versions do not capture the exact timing of updates made *between* versions, they give us far more observability into an evolving article than we had previously.

News editing and it's role in emulation learning. We now begin our final exploration of emulation and creative works. Until now, the only states we assumed we could observe were either *starting states*, s_0 (i.e. story leads, press releases) and/or *goal states*, g, (i.e. sets of sources, completed news articles). In some of these settings, we had *synthesized* intermediate states, by say, seeking to predict missing sources. However, until now we had not *observed* intermediate states. As shown in Figure 5.3, we now assume in this section that we *can* observe *some* intermediate states, $s_{1,n_1}, s_{2,n_2}...$, each corresponding to the publication of an *article version* (we reserve $s_{\cdot,n}$ for the *published draft* of an article version, assuming within-version edits, Figure 1.7). Between each *version* $(t, n) \rightarrow (t+1, n)$, we assume actions $a_{t,1}, a_{t,2}, \ldots$ occur. These can encompass any kind action we have so far considered; for instance, in Figure 5.2 we show three actions: $a_{t,1}$ = "Source Added", $a_{t,2}$ = "Event Updated" and $a_{t,3}$ ="Background Added'. The analysis of article versions helps us perform *emulation learning* because it opens the door to understanding temporal dynamics of action sequences better. For instance, in Chapter 3 when we studied *source-finding*, we either *ignored* temporal dynamics of when sources that were added to the news article or we used rough heuristics to impose an ordering. An analysis of article versions, on the

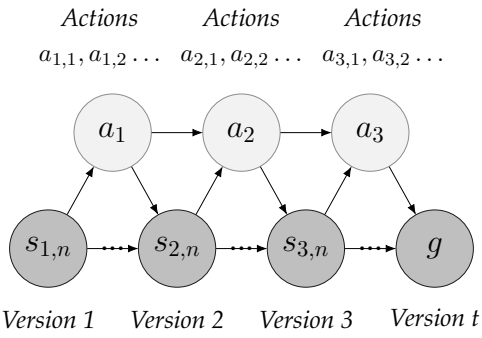


Figure 5.3: *Observability into article* writing given by edit analysis. In the state space, i refers to *version* number, and j refers to *draft* number: $s_{1,1} \ldots s_{1,n_1}; s_{2,1} \ldots s_{2,n_2}; \ldots$. We assume observability into the *final draft* of each version, $s_{\cdot,n}$. This lets us to infer actions $a_{t,1}, a_{t,2}... < a_{(t+1),1}, a_{(t+1),2}...$, between versions $(t, n_t) < (t+1, n_{t+1})$. Actions $a_{t,i}$ are any $\mathcal{A}$ considered in Chapters 2-4.

(a) *Imitation*: A demonstrator cracks a nut. *Actions* **a** and *states* **s** . . . are visible.

(b) *Ghost conditions*: A pully moves the hammer to crack a nut. Only *states*, **s** are visible.

(c) *End-state observation*: only the final state, $s_n = g$ is visible.

Figure 5.4: Three different forms of social learning, pictures from [647]

other hand, gives us much more granularity into this temporality: if we filter $a_{1,1}, a_{1,2} \ldots a_{2,1}, a_{2,2} \ldots a_{3,1}, a_{3,2} \ldots$, extracted from versions, to actions that map to source inclusion, then we can impose ordering on far more groups of sources. Thus, our primary interest in this Chapter is *not* to establish an explicit *goal state* g and work backwards to *emulate* that, but to use patial state-information to get more detailed action inferences for goal states g in other contexts we might wish to study.

The observation of intermediate states has a long history as a foundational part of the cognitive approach to studying *emulation*. In social-learning research, a *"ghost condition"* removes the visible agent and shows only an apparatus producing outcomes (e.g. a door sliding open to reveal a reward), allowing researchers to dissociate *imitation* (copying actions, $a_1, ...a_t$) from *emulation* (learning rewards r and finding goal-states, g) [648]. As shown in Figure 5.4, "ghost experiments" sit between *imitation* and *end-state observation* in terms of observability. Ghost-condition experiments have demonstrated that young children can learn from partial state–sequence information (e.g., seeing only an apparatus change state) in tasks such as opening "artificial fruit" puzzle boxes and tool-use devices [649] or where doors slide [203]: these uncovered *emulation* learning mechanisms from $intermediate$ state observations [649, 149][3].

[3]Indeed, even *watching* a teacher is not clear evidence of *imitation*: learners do not inherently replicate *every* motor movement of the teacher, but often discover new pathways to goals (i.e. *emulation*).

Cheat-Sheet: Emulation Learning for *Edit-Prediction*

We observe sentence-level edits across *article versions* (i.e. *atomic* state transitions) and use emulation learning to infer the *latent intentions*. We use these to gain more temporality into action sequences.

- **s** s_t **(states)** — this is the *published article state*. Also referred to as D_t. We also refer to $s_{t,i}$ as the ith sentence in version t[a] (§5.2, §5.3).
- **a** a **(actions)** — $a_t \in \mathcal{A}$: the *latent action/intention* between versions that drive the update (e.g., update an event, add a quote). We also use sentence-pair actions $a_{t,ij}$ to refer to the action generating sentences $(s_{t,i}, s_{t+1,j})$ (§5.2, §5.3).
- τ τ **(trajectory)** — The sequence of state–action steps temporally ordered by *article versions*, used to order other actions studied in this work (ghost-conditions) (§5.1).
- x x **(starting state)** — No starting state, x. Aim of section is to analyze *inner* states.
- g g **(goal state)** — No goal state, g. Aim of section is to analyze *inner* states.
- q $q_\theta\big(a_{t,ij} | s'_{t,i},\ s'_{t+1,j},\ D_t,\ D_{t+1}\big)$ **(inverse model)** — maps paired sentences i, j in article versions D_t, D_{t+1} to latent edit intentions; where $s' := s \cup \varnothing$. When $s_t = \varnothing$, it did not exist in that version. (§5.3, §5.3.1).
- π $\pi(a \mid s_{t,i}, D_t)$ **(policy model).** — predictor used to forecast which actions are likely to be taken. Specifically used to trigger cautious behavior (abstention) in QA when *factual updates* occur and evidence is likely stale (§5.3.4, §5.3.5).
- P $P_\phi(s_{t+1} \mid s_t, a_t)$ **(state transition model).** — how an intention changes the article, in practice approximated by "Edit Prediction" tasks. (§5.3, §5.2.3 *without* a: $P_\phi(s_{t+1} \mid s_t)$).

[a] In this introduction, we have referred to $s_{t,n}$ as the *final draft* of version t; we do not carry this notation into the main body. We reserve $s_{\cdot,i}$ as the ith sentence of version t, not ith draft.

In more creative domains, drafts and revisions are treated as observable traces of process rather than mere byproducts. Classic cognitive models characterize writing as cyclical planning–translating–reviewing, not a linear pipeline [133, 650]; empirical work shows experienced writers revise at conceptual levels (claims, structure) more than at the surface, making intermediate versions diagnostic of strategy and control [651]. In literary and creativity studies, *genetic criticism* systematizes the analysis of manuscripts, notes, and successive versions to reconstruct the making of a work and explicitly treats drafts as

[3] **Other notation used throughout**:
· $\Delta_t = \Delta(s_t, s_{t+1})$: the observable delta between article versions, or the "atomic edit action". The measurable state change between versions labeled as Addition, Deletion, Edit, Refactor via sentence alignment—used both for analysis and as supervision to learn transitions (§5.2.1.2, §5.2.1.3).
· $E(\cdot)$: the emission/observation channel. The alignment/labeling pipeline that emits Δ_t from (s_t, s_{t+1}) without committing to a specific action; provides training signal for q_θ and for transition tasks (§5.2.1.3, §5.2).

production protocols [652]; the geneplore model [653] models process of creation as an iterative process that proceeds through states. These fields further motivate, in *emulation learning*, modeling version histories as observable state–action sequences: given versions $s_{1,n_1}, s_{2,n_2}, \ldots$, we can infer trajectories $\tau = (a_{0,*}, a_{1,*}, \ldots)$ and learn policies $\pi(\tau \mid s_0)$ with temporal dynamics aligned to human creative practice.

Chapter 5 Overview

In Chapter 5, *State-Space Observability in Emulation Learning*, we will study how increased observability into intermediate state spaces $s_1, s_2, ...s_t$ can help us learn more precise sequences of actions $a_1, ...a_t$. This section will unfold as follows. In Section 5.2, I will introduce the *NewsEdits* dataset, a large collection of *article versions* we collected. I will discuss how we aligned sentences between these versions to better understand when: information was ADDED was REMOVED; UPDATED; or REPRIORITIZED. I will introduce a task, *Edit Prediction*, where the goal is to predict *observed* edits, or s_{t+1} from prior $s_{1...t}, a_{1...t}$ sequences (i.e. the *state-transition model* $p(s_{t+1}|s_t, a_t)$ is learnable); we will show that although machines struggle to do this, human journalists bring expert intuition. Our focus in this section will *primarily* be on *state-spaces* and what they can tell us. Then, in Section 5.3 I will more concretely codify the action space, A. We introduce a schema for edit actions, show that we can predict a_{t+1} from s_t and a_t, (i.e. the *policy* model $\pi^*(a_{t+1}|s_t, a_t)$ is learnable). Finally, I will conclude by showing that learning better *policy* $\hat{\pi}(a_{t+1}|s_{1...t}, a_{1,...t})$ and *state-transition* models $p(s_{t+1}|s_t, a_t)$ can help us understand informational staleness and help with *model abstention*.

Works Discussed:

- Spangher et al. (2022)". NewsEdits: A News Article Revision Dataset and a Novel Document-Level Reasoning Challenge". *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
- Spangher et al. (2024)". NewsEdits 2.0: Learning the Intentions Behind Updating News". *arXiv preprint arXiv:2411.18811*.

5.2 Measuring State-Change: The NewsEdits Dataset

As discussed in Section 5.1, *article versions* get generated every time a journalist publishes and *republishes* an article to the same URL (i.e. in "breaking news" scenarios). This gives us a unique opportunity to observe news revision-histories: in this Section, I will introduce the first *journalistic* edits dataset in the academic literature [4]. *NewsEdits* is a dataset of 1.2 million articles and 4.6 million versions. In this Section, we will study *NewsEdits from a purely state-space centered view* and will seek to prove that state-space progressions in revisions histories are atomic, informative and predictable [5] (we will address *actions* more specifically in Section 5.3). We treat each published version of an article as an *observable state* s_t. The ordered pair (s_t, s_{t+1}) induces *observed state changes* that we categorize using *edit types*. These are *not* actions; they are *observable state changes* emitted by unobserved editorial processes that transform s_t into s_{t+1}. In Section 5.2.1.1, I discuss how we gathered our dataset; in Section 5.2.1.2, I discuss how we identify different *state-space changes*; and in Section 5.2.3, I discuss how we *predict edits*, or in other words, study *transition regularities* in state space changes.

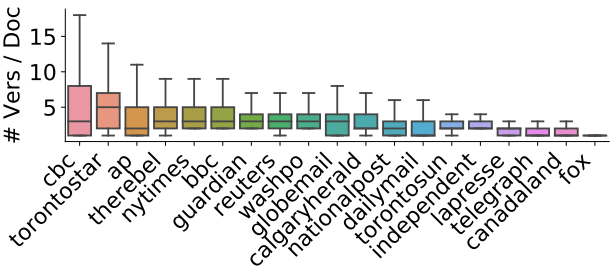


Figure 5.5: Number of versions per article, by outlet, in the *NewsEdits* dataset.

[4] To be clear, *NewsEdits* is not the *first* revision histories dataset. Revision datasets have been gathered from various natural language domains like Wikipedia [655], Wikihow [656] and student learner essays [657], and have primarily been studied to explore stylistic changes, grammatical error correction [658] and argumentation design [659]. However, as explored in the rest of this thesis, we are interested in questions: *What voices and perspectives are needed to complete a narrative? What is the process by which a story is written?* and later, more specific to edits: *which facts are uncertain and likely to be changed? Which events are likely to update?* Existing corpora do not suffice. News creation is a normative, professionalized process [105, 24] that involves performing *actions* (e.g. *source-finding*) *extrinsic* to the act of writing [103]. As such, it is both more intensive *and* regular than other more variable, writing-only processes like student essay creation.

[5] Our *predictability* experiment follows from the same logic as the *compositeness* experiment in Section 3.2.3.

Corpus	# Revisions	Language(s)	Source	Goal
WikEd Error Corpus [655]	12M changed sentences	English	Wikipedia	Grammatical Error Correction (GEC)
WikiAtomicEdits [660]	43M "atomic edits"[6]	8 languages	Wikipedia	Language modeling (edits), semantics/discourse
WiCoPaCo [661]	70,000 changed sentences	French	Wikipedia	GEC and sentence paraphrasing
wikiHowToImprove [662]	2.7M changed sentences	English	wikiHow	Version prediction, article improvement
NewsEdits [289]	36.1M changed, 21.7M added, 14.2M removed sentences; 72M atomic edits	English and French	22 media outlets	Language modeling, event sequencing, journalism

Table 5.1: A comparison of revision-history corpora, their size and composition, and the intention of their release, to situate *NewsEdits*.

5.2.1 Dataset Creation

5.2.1.1 Data Collection

We collect a dataset of news article versions. An article is defined by a unique URL, while a version is one publication (of many) to that same URL. We combine data from two online sources that monitor news article updates: NewsSniffer[7] and Twitter accounts powered by DiffEngine[8]. These sources were chosen because, together, they tracked many major U.S., British and Canadian news outlets [663]. Our corpus consists of article versions from 22 media outlets over a 15-year timescale (2006-2021), including *The New York Times*, *Washington Post* and *Associated Press*. Although the median number of updates per article is 2, as shown in Figure 5.5, this varies depending on the outlet. More dataset details in [289].

5.2.1.2 Categories of State-Space Change

Since we are interested in how an *entire* news article updates between versions, we focus on *sentence-level changes*, rather than on token-level rewrites. Identifying that sentences are added and deleted (vs. updated), can help us study the degree of change an edit introduces in the article [664, 665, 666]. Again, these labels identify *state changes*: they describe

[7] https://www.newssniffer.co.uk/
[8] https://github.com/DocNow/diffengine

$s_{t+1} \setminus s_t$ (and relocations within s_{t+1}) without committing to the underlying editorial *actions*. Thus, we define the following sentence-level state-space changes, shown in Figure 5.6: Addition, Deletion, Edit and Refactor. Additions introduce novel content; Deletions remove content; Edits preserve core meaning while revising syntax or updating specific facts (merges/splits are special cases); see Section 5.2.1.3 for more details. Refactors move sentences independent of other changes and thus reveal shifts in positional importance.[9]. Refactors are important because, based on the *inverse pyramid* [10] [207] of article structure, sentences that are higher in an article are more important [667].

5.2.1.3 State-Space Change Extraction

Our objective is to recover $\Delta(s_t, s_{t+1})$—the observed state change between article versions s_t, s_{t+1}. To extract these state-space changes, we construct a bipartite graph linking sentences in s_t and s_{t+1} (example graph shown in Figure 5.6). If an edge exists between a sentence in one version and a sentence in the other, the sentence is an Edit (or Unchanged). If no edge exists, the sentence is an Addition (if the sentence exists in the newer version only) or Deletion (if it exists in the older version only). We identify Refactors based on an algorithm we develop: in short, we identify a minimal set of edges in the graph which causes all observed edge-crossings. For details on this algorithm, see [289]. Conceptually, this pipeline estimates the *emission* $E(\Delta_t \mid s_t, s_{t+1})$; it deliberately avoids modeling any latent action space.

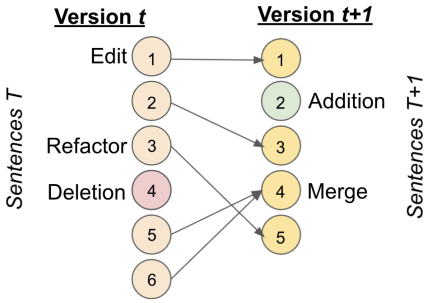


Figure 5.6: Sentence-level changes – Edit, Addition, Deletion and Refactor – between two versions of a news article (merges and splits are a special cases of Edits).

In order to construct this bipartite graph, we need a scalable, effective, sentence-

[9] As an example, in Figure 5.6, the addition of Sentence 2 in version$_{t+1}$ shifts Sentences 3, 4, 5 down. These are *not* refactors, just incidental moves caused by other changes. However, Sentences 5, 6 in version$_t$ are shifted upwards in version$_{t+1}$, which is movement that is not caused by other changes. We label this as a Refactor.

[10] An inverse pyramid narrative structure is when the most crucial information, or purpose of the story, is presented first [667].

Article Version t	**Article Version** $t+1$
The Bundesbank would only refer to an interview Mr. Weidmann gave to Der Spiegel magazine last week, in which he said, "I can do my job best by staying in office."	The Bundesbank would only refer to an interview published in Der Spiegel magazine last week, in which Mr. Weidmann said, "I can carry out my duty best if I remain in office."

(a) *Edit*: When the information conveyed in a sentence is substantially the same across versions, it should be connected, regardless of how many surface-level edits are made. Our algorithm successfully matches these sentences.

Article Version t	**Article Version** $t+1$
DALLAS—Ebola patient Thomas Eric Duncan told his fiancée the day he was diagnosed last week that he regrets exposing her to the deadly virus and had he known he was carrying Ebola, he would have "preferred to stay in Liberia and died than bring this to you," a family friend said.	DALLAS—Ebola patient Thomas Eric Duncan told his fiancée the day he was diagnosed last week that he regrets exposing her to the deadly virus . Had he known he was carrying Ebola, he would have "preferred to stay in Liberia and died than bring this to you," a family friend said.

(b) *Split:* When two sentences in version $t+1$ contain substantially the same information as a sentence in version t, they should be matched (the opposite is a *merge*).

Article Version t	**Article Version** $t+1$
"The mother, this was the first time seeing her son since he got to the States." "She has not seen him for 12 years, and the first time she saw him was through a monitor," said Lloyd.	"She has not seen him for 12 years, and the first time she saw him was through a monitor," said Lloyd. "The mother, this was the first time seeing her son since he got to the States." "She wept, and wept, and wept."

(c) *Refactor*: When the position of a sentence is moved in a document, we determine *heuristically* that the sentence moving *up* is the refactor, while the sentence moving down is incidental.

Table 5.2: Three challenging examples illustrating how our sentence-matching algorithms help us track information change across sentences, in article versions t and $t+1$. (red =removed/replaced word, green =inserted/replacement word).

similarity algorithm. There is a wide body of research in assessing sentence-similarity [668, 669, 670, 671]. However, many of these algorithms measure *symmetric* sentence-similarity. As shown in Figure 5.6, two sentences from the old version can be merged in the new version[11]. The symmetric similarity between these three sentences would be low, leading us to label the old sentences as DELETIONS and the new one an ADDITION, even if they were

[11]E.g. "ipsum. Lorem" → "ipsum; and Lorem". Conversly, one sentence can also be split.

BERT-Based			Subsequence Matching		BLEU-Based	
Method		F1-Score	Method	F1-Score	Method	F1-Score
Hungarian	TB-mini	88.5	ngram-1	86.0	BLEU-1	86.7
	TB-medium	88.7	ngram-2	88.7	BLEU-2	89.2
	RB-base	88.6	ngram-3	88.5	BLEU-3	88.8
Max	TB-mini	89.0	ngram-4	88.2	BLEU-1,2	88.8
	TB-medium	**89.5**			BLEU-1,2,3	89.1
	RB-base	89.4				

Table 5.3: F1 scores on validation data for matching algorithms. Left-hand group shows embedding-based methods (TinyBert (TB) and RoBERTa (RB)) with Maximum or Hungarian matching. Middle group shows ngram methods. Right-hand group shows BLEU for different ngram weightings (1,2 and 1,2,3 are uniform weightings over unigrams, bigrams and trigrams).

minimally edited (for concrete examples, see Table 5.2). This violates our tag definitions (Section 5.2.1.2). So, we need to measure one-way similarity between sentences, allowing us to label merged and split sentences as EDITS. Our algorithm is an asymmetrical version of the *maximum alignment* metric described by Kajiwara and Komachi [672]:

$$\text{Sim}_{asym}(x, y) = \frac{1}{|x|} \sum_{i=1}^{|x|} \max_j \phi(x_i, y_j)$$

where $\phi(x_i, y_j) :=$ similarity between words x_i in sentence x and y_j in sentence y. We test several word-similarity functions, ϕ. The first uses a simple lexical overlap, where $\phi(x_i, y_j) = 1$ if $lemma(x_i) = lemma(y_j)$ and 0 otherwise[12]. The second uses word-embeddings, where $\phi(x_i, y_j) = Emb(x_i) \cdot Emb(y_j)$, and $Emb(x_i)$ is the embedding derived from a pretrained language model [673, 562]. Each ϕ function assesses word-similarity; the next two methods use ϕ to assess sentence similarity. *Maximum alignment* counts the number of word-matches between two sentences, allowing many-to-many word-matches between sentences. Hungarian matching [674] is similar, except it only allows one-to-one matches. We compare

[12]We extend this to non-overlapping ngram matches.

	Total Num.	% of Sents.
Edits	26.6 mil.	17.6 %
Additions	10.2 mil.	6.8 %
Deletions	5.4 mil.	3.6 %
Refactors	1.6 mil.	1.1 %

Table 5.4: Summary statistics, after running sentence-matching algorithms, of state-space changes between article versions t and $t+1$.

these with BLEU variations [675], which have been used previously to assess sentence similarity [656].

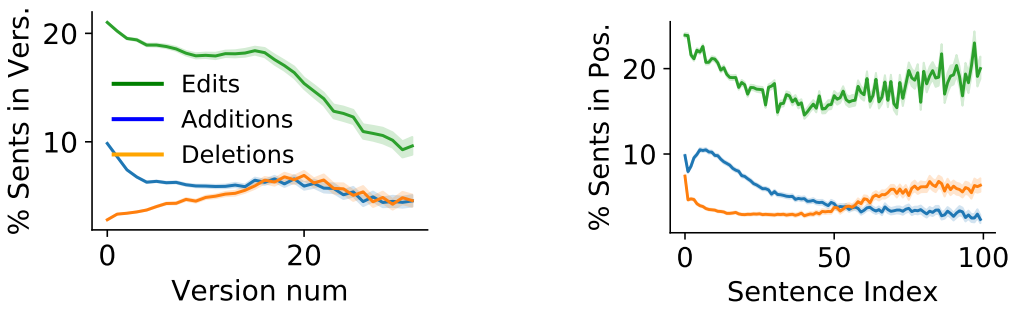
5.2.1.4 Edit Type Extraction Quality

Although our sentence-similarity algorithm is unsupervised, we need to collect ground-truth data in order to set hyperparameters (i.e. the similarity threshold above which sentences are considered a match) and evaluate different algorithms. To do this, we manually identify sentence matches in 280 documents. We asked two expert annotators to identify matches if sentences are nearly the same, they contain the same information but are stylistically different, or if they have substantial overlap in meaning and narrative function. See [289] for more details on the annotation task. We use 50% of these human-annotated labels to set hyperparameters, and 50% to evaluate match predictions, shown in Table 5.3. Maximum Alignment with TinyBERT-medium embeddings [673] (**Max-TB-medium**) performs best[13].

5.2.2 Exploratory Analysis

We extract all edit types in our dataset using methods described in the previous section. Statistics on the total number of changes are shown in Table 5.4. In this section, we analyze Additions, Deletions and Edits to explore when, how and why these states evolve during

[13]For more details and examples, see [289]



(a) State changes by version (% total in vers) (b) State changes by sentence position (%).

Figure 5.7: Dynamics of state-space changes across article version number and across the article body.

news event coverage and the clues this provides as to why articles are updated[14]. To reiterate, we interpret these statistics as *constraints on plausible latent actions*, not as actions.

Insight #1: Timing and location of state-changes reflect patterns of breaking news and inverse pyramid article structure. How do edit types evolve from earlier to later versions, and where do they occur in the news article? In Figure 5.7a, we show that state-space changes in an article's early versions are primarily adding or updating information: new articles tend to have roughly 20% of their sentences edited, 10% added and few deleted. This fits a pattern of breaking news lifecycles: an event occurs, reporters publish a short draft quickly, and then they update as new information is learned [676, 677]. We further observe that updates occur rapidly: outlets known for breaking news[15] have a median article-update time of < 2 hours [289]. An article's later lifecycle, we see, is determined by churn: $\approx 5\%$ of sentences are added and 5% are deleted every version. As seen in Figure 5.7b, additions and edits are more likely to occur in the beginning of an article, while deletions are more likely at the end, indicating newer information is prioritized in an inverse pyramid structural fashion. These regularities suggest a transition regime in early versions characterized by growth and rewriting, shifting to lower-magnitude churn later.

Insight #2: Additions and deletions are more likely to contain fact-patterns associated

[14]We leave a descriptive analysis of Refactors to future work.
[15]E.g. *Associated Press*, *The New York Times* and *Washington Post*

	Addition	Deletion	Unchanged
Contains Event	38.5	39.3	31.4
Contains Quote	48.4	50.0	39.2
Discourse: Main	4.4	4.9	3.6
Discourse: Cause	29.0	30.2	23.6
Discourse: Distant	63.5	61.4	68.1

Table 5.5: % Additions, Deletions or Unchanged sentences that contain Events or Quotes, or have news discourse role: Main (main events), Cause (immediate context) or Distant (history, analysis). $F < .01$, $n = 7,368,634$.

with breaking news (quotes, events, or main ideas) than unchanged sentences. In the previous section, we showed that the timing and position of state-space changes reflects breaking news scenarios. To provide further clues about the semantics of state-space changes, we sample Additions, Deletions and unchanged sentences and study the kinds of information contained in these sentences. We study three different fact-patterns associated with breaking news: events, quotes and main ideas [678, 679]. To measure the prevalence of these fact-patterns, we sample 200,000 documents (7 million sentences) from our corpus and run an event-extraction pipeline [680], quote-detection pipeline [681], and news discourse model [145]. As shown in Table 5.5, we find added and deleted sentences have significantly more events, quotes and Main-Idea and Cause discourse than unchanged sentences. (See [289] for more details.) Thus, $\Delta_t(s_t, s_{t+1})$ correlate with semantic payloads (events, quotes, main ideas) rather than purely stylistic variation.

Insight #3: Within-sentence edits frequently reflect event updates. The analyses in the previous sections have established that state-space changes both are positioned in the article in ways that resemble, and contain information that is described by, breaking news epistemologies [678]. A remaining question is whether the state-space changes change fact-patterns themselves, rather than simply changing the style or other attributes of sentences. One way to measure this is to explore whether state-space changes update the events in a story [682]. We focus on pairs of edited sentences. We randomly sample

Event Chains
(attack, killed), (injured, killed), (shot, dead), (shot, killed), (attack, injured), (injured, died), (election, won), (meeting, talks), (talks, meeting), (elections, election), (war, conflict)

Table 5.6: Selection of top event extracted from edited sentence pairs across article versions.

EDITS from documents in our corpus ($n = 432,329$ pairs) and extract events using Ma et al. [680]'s model. We find that edited sentence pairs are more likely to contain events (43.5%) than unchanged sentences (31.4%). Further, we find that 37.1% of edited sentences with events contain *different* across versions. We give a sample of pairs in Table 5.6. This shows that many *within*-sentence edits update events. Taken together, we have shown in this analysis that *factual* updates drive many of the edit types that we have constructed to describe *NewsEdits* revision histories. Next, we measure the predictability of update patterns.

5.2.3 Predictive Analysis on NewsEdits

As shown in Section 5.2.2, many state-space changes show breaking news patterns, which Usher [679] observed follow common update patterns. Now, we explore how *predictable* these edit types are as a transition problem. Like in Section 3.2.3, we aim prove that these transitions regular and can support learning for downstream research questions, like those outlined in Section 5.1 around narrative design (e.g. "which facts in the current version of this article are likely to change?", "what resources should a journalist access to improve this article?", "what voices should be added to this story?"). To reiterate, we explicitly frame the tasks below as predicting next states s_{t+1} and observable deltas Δ_t from s_t, deferring any commitment to an action schema to Section 5.3. In this section, we outline three tasks[16] that involve predicting the future states of articles based on the current state. These tasks, we hypothesize, outline several modeling challenges: (1) identify indicators of

[16]These tasks were inspired by Story Cloze and narrative understanding tasks [683, 684].

uncertainty used in news writing[17] [678], (2) identify informational incompleteness, like source representation [681] and (3) identify prototypical event patterns [685]. These are all strategies that expert human evaluators used when performing our tasks (Section 5.2.3.6). The tasks range from easier to harder, based on the sparsity of the data available for each task and the dimensionality of the prediction. We show that they are predictable but present a challenge for current language modeling approaches: expert humans perform these tasks much more accurately than LLM-based baselines. In addition to serving a model-probing and data-explanatory purpose, these tasks are also practical: journalists told us in interviews that being able to perform these predictive tasks could help newsrooms allocate reporting resources in a breaking news scenario[18].

5.2.3.1 Task Description and Training Data Construction

We now describe our tasks. For all three tasks, we focus on breaking news by filtering *NewsEdits* down to short articles (# sents $\in$ [5, 15]) with low version number (<20) from select outlets[19].

Task 1: Will this document update? Given the text of an article at version t, predict if $\exists v+1$. This probes whether the model can learn a high-level notion of change, irrespective of the fact that different state-space changes have different consequences for the information presented in a news article. For **Task 1**, $y^{(1)} = 1$ if a newer version of an article was published and 0 otherwise. We sample $100,000$ short article versions from *NewsEdits*, balancing across length, version number, and $y^{(1)}$.

Task 2: How much will it update? Given the text of an article at version t, predict in the next version how many Additions, Deletions, Edits, Refactors will occur. This moves beyond Task #1 and requires the model to learn more about *how* each edit-type

[17] E.g. "Police to release details of the investigation."
[18] See [289] for more details.
[19] The *New York Times*, *Associated Press*, *Washington Post*, *BBC*, *Independent*, *Guardian* and *Reuters* were used, as they are more known for breaking news [679]. See [289] for more details.

changes an article. For **Task 2**, $y^{(2)}$ = counts of sentence-level labels (Num. Additions, Num. Deletions, Num. Refactors, Num. Edits) described in the previous sections, aggregated per document. Each count is binned: $[0, 1)$, $[0, 3)$, $[3, \infty)$ and is predicted separately as a multiclass classification problem. We sample $150,000$ short article versions balancing for sources, length and version number.

Task 3: How will it update? For each sentence in version t, predict whether: (1) the sentence itself will change (i.e. it will be a Deletion or Edit) (2) a Refactor will occur (i.e. it will be moved either up or down in the document) or (3) an Addition will occur (i.e. either above or below the sentence). This task, which we hypothesize is the hardest task, requires the model to reason specifically about the informational components of each sentence *and* understand nuance about structure and form in a news article (i.e. like the inverse pyramid structure [207]). For **Task 3**, $y^{(3)}$ = individual sentence-level labels. Labels are derived for the following subtasks mentioned above: (1) *Edit Type* is a categorical label comprising: [Deletion, Edit, Unchanged], expressed as a one-hot vector. (2) Refactor is a categorical label comprising: [Up, Down, Unchanged], also expressed as a one-hot vector. (3) Addition Above and Addition Below are each binary labels expressing whether > 1 sentences was added above or below the target sentence. Because some sentences had Additions above and below, we chose to model this subtask as two separate classification tasks. We sample $100,000$ short article versions, balancing for sources, length and version number. For each task, the input X is a document represented as a sequence of sentences. For each evaluation set, we sample $4k$ documents balancing for class labels[20].

5.2.3.2 Modeling

We benchmark our tasks using a RoBERTa-based architecture shown in Figure 5.8. Spangher et al. [145] showed that a RoBERTa-based architecture [562] with a contextualization layer outperformed other LLM-based architectures like Reimers and Gurevych [221] for

[20]With the exception of some tasks, like Refactors, which are highly imbalanced and cannot be balanced.

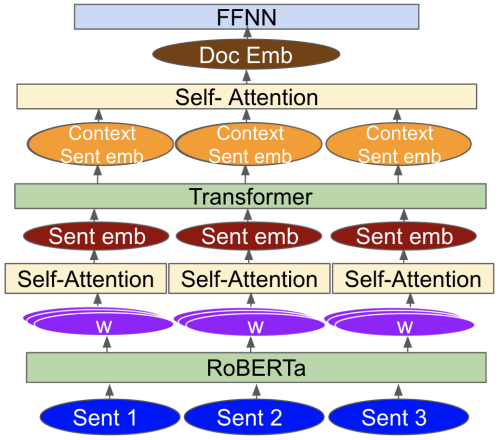


Figure 5.8: Architecture diagram for the model used for edit-prediction tasks. Word-embeddings are averaged using **Self-Attention** to form sentence-vectors. A minimal transformer layer is used to contextualize these vectors (+Contextual Layer). In Tasks 1 and 2, self-attention is used to generate a document-embedding vector.

document-level understanding tasks (further insight given in Section 5.2.3.6). In our model, each sentence from document d is fed into a pretrained `RoBERTa Base` model[21] to obtain contextualized word embeddings. The word embeddings are then averaged using self-attention, creating sentence vectors. For **Task 3**, these vectors are then used directly for sentence-level predictions. For **Tasks 1** and **2** these vectors are condensed further, using self-attention, into a single document vector which is then used for document-level predictions. The sentence vectors are optionally contextualized to incorporate knowledge of surrounding sentences, using a small Transformer layer[22] (+Contextualized in Tables 5.7, 5.8, 5.9). We experiment with the following variations. For **Task 2**, we train with less data ($n = 30,000$ version pairs) and more data ($n = 150,000$ version pairs), balanced as described in Section 5.2.3.1, to test whether a larger dataset would help the models generalize better. We also experiment, for all tasks, with freezing the bottom 6 layers of the RoBERTa architecture (+Partially Frozen) to probe whether pretrained knowledge is helpful for these tasks. Additionally, we experiment giving the version number of the older version as an additional input feature alongside the text of the document (+Version).

[21]We used Wolf et al. [467]'s version, found here `https://huggingface.co/roberta-base`.

[22]Specifically, we initialize a 2-layer, 2-headed GPT2 transformer block to perform autoregressive contextualization.

	Num. Adds		Num. Deletes		Num. Edits		Num. Refactors	
	F1 *Mac*	F1 *Mic*	F1 *Mac*	F1 *Mic*	F1 *Mac*	F1 *Mic*	F1 *Mac*	F1 *Mic*
Most Popular	19.8	25.0	25.6	47.8	21.9	32.0	39.2	64.5
Random	32.5	33.9	30.2	36.4	31.7	35.1	25.8	35.1
Baseline ($n = 30k$)	22.1	27.9	25.6	46.5	21.4	30.6	35.2	64.5
($n = 150k$)	29.7	36.3	25.7	48.1	22.4	32.8	39.2	64.6
+Partially Frozen	**52.2**	54.0	44.8	59.0	49.3	53.1	44.3	**65.6**
+Contextual	50.7	52.2	41.0	57.4	**50.8**	**54.8**	**45.0**	64.3
+Version	52.0	**54.5**	**45.3**	**59.8**	49.9	53.7	43.8	63.1
+Multitask	46.7	50.2	28.2	48.4	42.1	49.5	40.3	55.1
Human	**66.4**	**69.3**	**64.6**	**67.5**	**65.9**	**75.6**	**71.3**	**70.7**

Table 5.7: **Task 2 Benchmarks:** Baseline model performance for document-level edit-prediction. Counts of Added, Deleted, Edited and Refactored sentences are binned into roughly equal-sized "low" ($[0, 1)$ sentences), "medium" ($[0, 3)$ sentences), "high" ($[3, \infty)$ sentences) bins. Macro and Micro F1 calculated across bins. (Scores shown are median of $1,000$ bootstrap resamples of the evaluation dataset.)

Finally, for **Tasks 2** and **3**, we attempt to jointly model all subtasksusing separate prediction heads for each subtask but sharing all other layers. We use uniform loss weighting between the tasks. Spangher et al. [145] showed that various document-level understanding tasks could benefit by being modeled jointly. For our tasks, we hypothesize that decisions around one operation might affect another: i.e. if a writer deletes many sentences in one draft they might also add sentences, so we test whether jointly modeling has a positive effect. We do not consider any feature engineering on the input text, like performing event extraction [680], even though results in Section 5.2.2 show that certain types of edits are more likely to contain events. We wish to establish a strong baseline and test whether models can learn salient features on their own. For more discussion on modeling choices and hyperparameter values, see [289]. In summary, these experiments probe a transition view —approximating $p(s_{t+1} \mid s_t)$ — while remaining agnostic about latent actions.

	Additions		Edit Types		Refactors	
	Above (F1)	Below (F1)	F1 *Mac*	F1 *Mic*	F1 *Mac*	F1 *Mic*
Most Popular	0.0	0.00	18.1	20.2	34.7	53.3
Random	**11.8**	**14.4**	28.0	38.3	24.7	34.7
Baseline	8.3	0.1	**36.5**	**61.9**	35.2	54.2
+Partially Frozen	3.5	0.0	35.4	60.9	35.4	54.6
+Version	0.1	0.0	30.3	59.0	**41.6**	**57.2**
+Multitask.	0.0	0.0	27.5	57.8	39.5	54.8
Human	**38.6**	**46.7**	**63.8**	**63.5**	**45.6**	**91.5**

Table 5.8: **Task 3 Benchmarks:** Baseline model performance for sentence-level edit-prediction. Addition tasks are: "Was a sentence added *below* the target sentence?", "Was a sentence added *above* the target sentence?" Edit Types columns are three edits that occur on the target sentence: "Deletion", "Editing", "Unchanged". Refactor is binned into whether the target sentence is "Moved Up", "Moved Down" or "Unchanged". (Scores shown are median of $1,000$ bootstrap resamples of the evaluation dataset.)

	F1		F1
Most Popular	56.6	Baseline	60.8
Random	50.6	+Partially Frozen	66.0
Human	**80.1**	+Contextual	61.7
		+Version	**77.6**

Table 5.9: **Task 1 Benchmarks:** Baseline model performance for next-version edit-prediction task. Label is binary. (Scores are median of $1,000$ bootstrap resamples of the evaluation dataset.)

5.2.3.3 Human Performance

To evaluate how well human editors agree on edits, we design two human evaluation tasks and recruit 5 journalists with ≥ 1 year of editing experience at major U.S. and international media outlets.

Evaluation Task 1: We show users the text of an article and ask them whether or not there will be an update. Collectively, they annotate 100 articles. After completing each round, they are shown the true labels. This evaluates **Task 1**.

Evaluation Task 2: We show users the sentences of an article, and they are able to move sentences, mark them as deleted or edited, and add sentence-blocks above or below sentences. They are **not** asked to write any text, only mark the high-level edits: "I *would* add a sentence," etc. Collectively they annotate 350 news articles. After each annotation, they see what edits *actually* happened. The raw output evaluates **Task 3**; we aggregate their annotations for each article to evaluate **Task 2**. They are instructed to use their expert intuition and they are interviewed afterwards on the strategies used to make these predictions. (See [289] for task guidelines and interviews).

5.2.3.4 Results

As shown in Tables 5.7, 5.8, and 5.9, model-performance indicates that our tasks do range from easier (**Task 1**) to harder (**Task 3**). While our models show improvements above **Random**, and **Most Popular** in almost all subtasks, a notable exception is **Task 3**'s Addition subtasks, where the models do not clearly beat **Random**. We note that this was also the most difficult subtask for human evaluators.

We observe that +Partially Frozen increases performance on **Task 2**, boosting performance in all subtasks by ≈ 10 points. In contrast, it does not increase performance on **Task 3**, perhaps indicating that the subtasks in **Task 3** are difficult for the current LLM paradigm. Although adding version embeddings (+Version) boosts performance for **Task 1**, it does not seem to measurably increase performance for the other tasks. Finally, performing **Task 2** and **3** as multitask learning problems decreases performance for all subtasks. In contrast, human evaluators beat model performance across tasks, most consistently in **Task 2**, with on average performance 20 F1-score points above Baseline models. On **Task 3**, human performance also is high relative to model performance. We observe that, despite Additions in **Task 3** being the hardest task, as judged by human and model performance, humans showed a ≈ 40 point increase above model performance. Humans are also better at correctly identifying minority classes, with a wider performance gap seen for Macro F1

Topic (↑)	F1	Topic (↓)	F1
U.S. Pol.	38.1	Local Pol.	66.8
Business	48.4	War	61.8
U.K. Pol.	50.4	Crime	58.3

Table 5.10: Predictability of edit patterns for $y^{(2)}$ on documents grouped by topic (document topics derived from running LDA [686] and assigning the top topic to the document. Edit patterns in topics (e.g. "local politics") are easier to predict than others (e.g. "U.S. politics").

y (Add)	F1
$[0, 1)$	16.2
$[1, 5)$	59.7
$[5, 100)$	0.9

Table 5.11: Predictability of $y^{(2)}$ by growth rate: $[0, 1)$ often reflects stylistic updates; $[5, 100)$ is often breaking news. Both are harder to predict than medium.

scores (i.e. see *Edit Types*, where the majority of sentences are unchanged).

5.2.3.5 Error Analysis

We perform an error analysis on the **Task 2** task and find that there are several categories of edits that are easier to predict than others. We run Latent Dirichlet allocation on $40,000$ articles, shown in Table 5.10[23]. We assign documents to their highest topic and find that articles covering certain news topics (like War) update in a much more predictable pattern than others (like $Business$), with a spread of over 26 F1-score points. Further, we find that certain edit-patterns are easier to differentiate, like articles that grow between 1-5 sentences (Table 5.10). This show us ways to select for subsets of our dataset that are more standard in their update patterns. The class imbalance of this dataset (Table 5.4) results in the **Most Popular** scoring highly. To mitigate this, we evaluate on balanced datasets. Class imbalanced training approaches [687, 145] might be of further help.

5.2.3.6 Evaluator Interviews

To better understand the process involved with successful human annotation, we conducted evaluator interviews. We noticed that evaluators first identified whether the main news event was still occurring, or if it was in the past. If it was still occurring, they tried to predict

[23]Topic words shown in [289].

when the event would update (i.e. *state-change* (s_t, s_{t+1}) inference).[24] For the latter, they considered discourse components to determine if an article's narrative was complete (i.e. see Sections 4.1: *state-change* and *action* inference, (s_t, s_{t+1}, a_t)) and analyzed the specificity of the quotes (i.e. see Sections 3.2.3, 3.6: *source action* a_t inference.).[25] They determined where to add information in the story based on structural analysis (i.e. see Section 4.2, *state* and *action* (s_t, a_t) inference) [26], and stressed the importance of the inverse pyramid for *informational uncertainty*: information later in an article had more uncertainty; if confirmed, it would be moved up in later versions (i.e. see Section 2.3.3). [27] Finally, they considered the emotional salience of events; if a sentence described an event causing harm, it would be moved up (action inference, a_t)[28]. Clearly, these tasks demand strong world-knowledge and common sense, as well and high-level discourse, structural and narrative awareness. [29] I have also tried to point out, in parentheticals, where different kinds of reasoning (i.e. narrative, factual, news-value; state and action) tie into creative questions that we have modeled, with *emulation learning*, throughout thesis. Combining these different forms of reasoning, as we have seen repeatedly, is challenging for current language models to do.

In fact, current LLMs, for many subtasks, perform worse than guessing. **+Multitask** performance actually decreases performance for both **Task 2** and **Task 3**, indicating that these models learn features that do not generalize across subtasks. This contrasts with what our evaluators said: their decision to delete sentences often used the same reasoning as, and were dependent on, their decisions to add. However, we see potential for improvement in these tasks. Current LLMs have been shown to identify common arcs in story-telling [689], identify event-sequences [682] and reason about discourse structures [145, 127].

[24] The longer the timespan, the more information they predicted would be added between drafts.

[25] E.g. Generic quotes, say a public announcement, would be updated with specific, eye-witness quotes.

[26] They identified the paragraph that introduced the main event – i.e. the Lede and the Nut Graf – and added information right after that

[27] One evaluator called this a *"buried cause"*. For example, a story about a building collapse had a sentence near the end about a source mentioning a faulty inspection: in a later draft, this sentence was moved up as it was confirmed with a second source.

[28] See [289] full interviews.

[29] Evaluators told us they "thought like the AP." The AP, or the *Associated Press*, has a styleguide [688] that many outlets use to guide their writing.

Further, for the ROCStories challenge, which presents four sentences and tasks the model with predicting the fifth [690, 683], LLMs have been shown to perform scene reconstruction [691], story planning [450, 692], and structural common sense reasoning [693]. These are all aspects of reasoning that our evaluators told us they relied on. Narrative arcs in journalism are often standard and structured [694], so we see potential for improvement.

Summary We introduced a large-scale dataset of news version histories and operationalized *state-change types* that make state change observable at the sentence level. We showed that many changes are fact-driven and that next-state patterns are predictable by experts but remain challenging for current LM-backed classifiers. Our analysis in this Section is *state- and emission-focused*: Δ_t are observables of $s_t \to s_{t+1}$, not actions. Going forward, we will develop a schema describing the types of edits. We are inspired by the Wikipedia Intentions schema developed by [695], and will present work inspired by this in Section 5.3. We will introduce an explicit edit-intention schema and study policy learning over latent decisions, connecting our state-change observations in this Section with the *actions* behind these edits.

5.3 Mapping an *Action-Space* $\mathcal{A}$ Onto News Edits

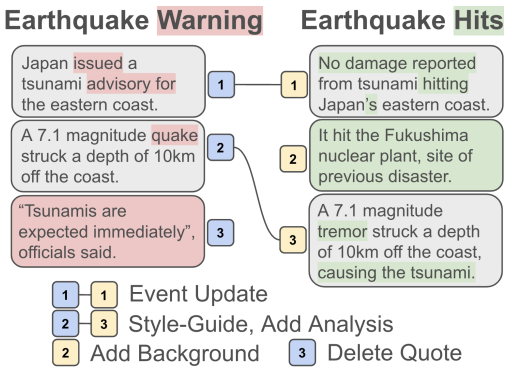


Figure 5.9: We demonstrate the insights we gain from comparing two versions of an updating article. We can identify factual updates (e.g. "Event Update" between 1-1), stylistic updates (e.g. "Style-Guide" between 2-3) and narrative updates (e.g. "Add Background" for sentence addition 2).

In Section 5.2, we introduced the *NewsEdits* dataset; we treated version pairs (s_t, s_{t+1}) as *fully observable states* and categorized different *edit types*, Addition, Deletion, Edit, and Refactor. To put this in terms of *emulation*, these edit types were estimated *emissions* over sentence-level state changes[30]. We demonstrated that these edit types were predictable, thereby providing evidence we could model *full* edit trajectories $(s_1, a_1), (s_2, a_2), \ldots$ to better emulation humans. Now, we are ready to try to parse the intentions of editors. In this Section, we adopt the following terminologies from *emulation learning*. We posit a *latent action* variable $a_t \in \mathcal{A}$ that drives state transitions, $s_{t+1} \sim P_\phi(s_{t+1} \mid s_t, a_t)$ and a (history-dependent) policy, $a_t \sim \pi_\theta(a_t \mid s_t)$. P_ϕ captures *how* edits change a document when a particular edit *intention* is taken. We abuse terminology here and allow s_t, s_{t+1} to refer to *emissions*, or observed edits, *as well as* full article version updates. Finally, we have our inverse model, $q_\theta(a_t | s_t, s_{s+t})$ which maps observable sentence-level edits we extract to latent intentions.

[30]To recap, this is not the first time we are seeing an *emissions* model. *Emissions* are signals we use to draw inferences about the latent variable of interest (a or s). We saw *emissions* or *observation* models in Sections 2.2 and 2.3.3, where we could view a only as an *emissions*, or *observation channels* $M_\psi(x, g)$ and $p(x > x')$ into the phenomena we cared about (in Chapter 2, the phenomena we cared about *emulate* was *newsworthiness*). We also saw them in Section 4.5, where we defined *emissions modes* $C_\sigma(y|a)$ to utilize different discourse schemas (in Chapter 4, the phenomena we sought to *emulate* was *story structure*.)

5.3.1 Learning Edit Intentions in Revision Histories

News articles update for different reasons, especially during breaking news cycles where facts and events update quickly [696]. In this Section, we introduce the edit-intentions discourse schema to describe actions a taken by the journalist to drive an article from $s_t \rightarrow s_{t+1}$. Next, I will introduce our annotation, and our models to label edit-pairs. This lays groundwork for Section 5.3.4, where we will predict when facts change. Our goal is to identify categories of edits, in order to enable different investigations into these different update patterns. In other words, we describe the following *inverse* model:

$$q_\theta(a_{t,ij}|s_{t,i} \cup \varnothing, s_{t+1,j} \cup \varnothing, D_t, D_{t+1}) \tag{5.1}$$

where $a_{t,ij}$ is an *action* or *intention* (e.g. a "Correction needs to be made") corresponding to sentences i and j in article versions t and $t+1$ – or $\varnothing$ if sentence i was a Deletion or sentence j was an Addition. D_t and D_{t+1} represent the full text of the article versions t and $t+1$. i, j, as stated before, are sentence indices, and range from $i \in \{1, ...n+1\}$, $j \in \{1, ...m+1\}$ (where n, m are the number of sentences in D_t, D_{t+1} and $n+1, m+1 = \varnothing$).

5.3.2 Edit Intentions Schema

Our schema (Fig. 5.10) supplies a hierarchical action ontology $\mathcal{A} = \mathcal{A}_{\text{fact}} \cup \mathcal{A}_{\text{style}} \cup \mathcal{A}_{\text{narr}}$. At the sentence pair $(s_{t,i} \cup \varnothing, s_{t+1,j} \cup \varnothing)$, we annotate an intention label $a_{t,ij} \in \mathcal{A}$ (e.g., Event Update, Quote Added, Add Background). We work with two professional journalists and one copy editor[31] to develop this ontology. Building off work by Zhang and Litman [657] and Yang et al. [695], we start by examining 50 revision-pairs sampled from *NewsEdits*. We developed our schema through 4 rounds of conferencing: tagging examples finding edge-cases and discussing whether to add or collapse schema categories. Figure 5.10

[31]Collectively, these collaborators have over 50 years of experience in major newsrooms.

Figure 5.10: Discourse schema for edit *actions* $\mathcal{A}$ across news edit versions. We organize revisions into four macro categories: Factual Edits capture changes to the state of the world — updating events, sources, etc. and making corrections. Style Edits modify form rather than substance — simplifying, updating syntax or tonality and moves that emphasize or de-emphasize importance. Narrative/Contextual edits reshape the story's framing — adding background, analysis, or anecdotes to situate facts. Other covers housekeeping cases such as unchanged pairs and sentence-linking errors (see [654] for definitions).

shows our schema, which we organize into coarse and fine-grained labels. We incorporate existing theories of news semantics into our schema. For instance, "Event Updates" incorporates definitions of "events" [697], while "Add Background" incorporates theories of news discourse [698] (discussed more in Sections 4.1, 4.5). "Add Quote" incorporates definitions from informational sourcing [1](discussed in Sections 3.2) and "Add Anecdote" incorporates definitions from editorial analysis [128]. See [654] for a deeper discussion of the theoretical schemas that inform our edit-action schema. Finally, "Incorrect Link" is an attempt to correct sentence pairs that were erroneously (un)linked by our linking algorithm in Section 5.2.1.3. As such, our edit schema brings together several different tasks — each with their own action vocabulary — that we have considered so far. It is a distillation of *emulation* in many parts the creative process of news writing.[32]

[32]A criticism could be: why did we need to annotate and learn a single *inverse* model for multiple different parts of the creative process, especially when we already performed large-scale annotation in other prior Sections for specific parts of this process? Firstly, and most importantly, we wanted to incorporate information about s_t *and* s_{t+1} in our inverse model. Secondly, we aimed to confirm that the multiple discourse schemata we introduced were converging and agreeing, which we explore in our **Experimental Variations**.

	All		Fact		Style		Narrative	
Features	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro
Baseline, *fine-grained*	45.8	73.6	32.0	47.2	58.6	39.9	52.0	39.9
+ NLI	48.6	74.1	45.7	50.4	55.2	38.7	43.6	38.7
+ Event	46.7	74.1	39.0	49.0	59.3	41.4	41.7	41.4
+ Quote	46.3	72.8	49.8	54.7	31.9	28.0	42.4	28.0
+ Collapsed Quote	**51.2**	73.9	38.7	47.6	58.3	39.4	51.4	39.4
+ Discourse	45.8	75.1	37.7	49.6	63.8	44.6	43.2	44.6
+ Argumentation	48.9	73.6	37.1	47.9	57.1	37.7	53.5	37.7
+ Discourse & Event	46.3	74.3	38.9	49.9	62.1	42.2	42.4	42.2
+ Discourse & Argumentation	47.8	74.1	56.8	50.5	31.4	32.2	41.1	32.2
+ Argumentation & Event	50.0	75.1	38.0	48.6	46.4	44.9	58.5	44.9
+ Quote & Discourse	**51.2**	72.2	40.5	45.3	62.8	43.0	48.7	43.0
+ Collapsed Quote & Discourse	49.6	73.9	45.6	49.4	58.9	39.1	47.9	39.1
+ Collapsed Quote & NLI	45.4	72.8	41.9	50.4	46.7	31.2	39.3	31.2
+ Collapsed Quote & NLI & Event	49.0	73.8	44.9	48.9	57.4	37.0	44.0	37.0
+ All	47.2	73.6	40.0	49.7	58.6	36.0	43.5	36.0
Baseline, *coarse-grained*	49.4	56.7	46.6		65.1		10.4	
+ Discourse & Arg. (Best model, Fact)	65.4	70.7	59.4		66.2		49.2	

Table 5.12: F1 scores (%) on our test set of the fine-tuned LED model with different combinations of features. Fact/Style/Narrative F1 scores are computed on instances that contain the corresponding labels, whereas All F1 scores are derived from all instances.

5.3.2.1 Schema Annotation

We build an interface for annotators to provide intention labels for news article sentence pairs (details given in [654]). Annotators are shown definitions for each fine-grained intention and the articles to tag; they are instructed to tag each sentence. We develop our interface in D3. To recruit annotators, we posted on two list-serves for journalism industry professionals[33]. We train our annotators until they are all tagging with $\kappa > .6$ agreement, compared with a gold-set of 50 article revision-pairs that we annotated, described previously (Section 5.3.2). See [654] for more details.

5.3.2.2 Edit Intentions Modeling

Now, we are ready to classify edit intentions between sentences in article revisions. As stated previously, edit intentions $a_{t,ij}$ are labeled on the sentence-pair level (including Additions and Deletions), and sentence-pairs has potentially multiple intention-labels. Document-level context is important: as shown in Figure 5.9, understanding that Sentence 2, right, adds background ("*It hit the Fukushima plant, site of previous disaster.*") is aided by the surrounding sentences contextualizing that a major event had just occurred. So, we wish to construct models that can produce flexible outputs and reason about potentially lengthy inputs. Generative models have recently been shown to outperform classification-based models in document understanding tasks [699, 700]. Inspired by this, we develop a sequence-to-sequence framework using LongFormer [34] [460] to predict the intent behind each edit. We adopt three weak assumptions to make $a_{t,ij}$ learnable: (1) Sparsity: $|a_t|$ is small (few intentions per version step). (2) Locality: each $a_{t,ij}$ primarily depends on a bounded window in D_t, D_{t+1}. (3) **Stability:** the inverse model $q_\theta(a_{t,ij}|s_{t,i} \cup \varnothing, s_{t+1,j} \cup \varnothing, D_t, D_{t+1})$ is time-stationary within an outlet/topic up to noise. The decoding target is multilabel, i.e. $a_{t,ij} = \left[a^{(1)}_{t,ij}, a^{(2)}_{t,ij}, \ldots\right]$ is a concatenation of ≥ 0 intention labels for the pair $s_{t,i}, s_{t+1,j}$.

Experimental Variants As discussed in Section 5.3.2, we developed our schema to bring together different theories of news semantics and large parts of the creative process that we have explored so far. So, we hypothesize that incorporating insights from these theories into our modeling – specifically, by utilizing labels from trained models in these domains – might improve our performance. We run models from the following papers over our dataset: *Discourse* (Section 4.5), *Quote-Type Labeling* (Section 3.2), *Event Detection* [701], *Textual Entailment* [702] and *Argumentation* [128]. Labels generated from these models, denoted as $f_{s_{t,i}}$ and $f_{s_{t+1,j}}$, are appended to the model input: $[s_{t,i} \cup \varnothing || s_{t+1,j} \cup \varnothing || D_t || D_{t+1} || f_{s_{t,i}} || f_{s_{t+1,j}}]$.

[33] The Association of Copy Editors (ACES) `https://aceseditors.org/` and National Institute for Computer-Assisted Reporting (NICAR) `https://www.ire.org/hire-ire/data-analysis/`.

[34] `https://huggingface.co/allenai/led-base-16384`

Edit-Intention Tagging Model Performance As shown in Table 5.12, our baseline tagging models that solely use article features score 45.8 Macro F1 and 73.6 Micro F1, respectively. These scores are moderate-to-low. The category we are most interested in, Factual updates, scores at 32 Macro-F1 (derived from macro-averaging the fine-grained categories). However, incorporating additional features increases overall Macro and Micro F1 by 5.5 and 1.5 points, respectively, in the *Quotes & Discourse* trial. And for Factual updates, additional features increase Macro and Micro F1 accuracy by 17.8 and 7.5 points, respectively. While low-to-moderate scores are not ideal, this likely reflects the noisy nature of our problem. For details and schema definitions, see [654].

5.3.3 Exploratory Insights

Different edit-intentions distribute differently across different edit types. We run the models trained in Section 5.2 over the entire *NewsEdits* corpus to generate silver-labels *state-change categories* on all edit pairs (i.e. Add, Deletion, Update). We present an exploratory analysis of these silver labels, with more material shown in [654]. Table 5.13 shows the correlation between syntactic edit categories (defined by [289]) and our semantic categories. As can be seen, categories like Addition have far more Narrative and Factual updates than Stylistic updates; Stylistic updates, on the other hand, are far more likely to occur between sentences. This is logical; Stylistic updates are likely smaller, local updates, while Narrative and Factual updates might include more rewriting.

Different edit-intentions distribute differently across different kinds of news (e.g. Business, Politics). Next, we explore if certain *kinds of articles* are more likely to have certain *kinds of edits*. We start by looking at broad news categories, shown in Table 5.14, obtained from classifier we train on CNN News Groups dataset[35]. "Politics" and "Sports" coverage are observed to have the highest level of Factual updates, relative to other categories, while Stylistic updates are prevalent in "Health" and "Entertainment" pieces. Although we

[35] https://www.kaggle.com/code/faressayah/20-news-groups-classification-prediction-cnns

	Narrative	**Fact**	**Style**
Addition	840329	358900	104
Deletion	330039	21671	6088
edit	411292	102499	644243

Table 5.13: Counts of coarse-grained semantic edit types, broken out by syntactic categories (for fine-grained counts, see [654]).

	Fact	Style	Narrative
Business	1.6	62.0	36.4
Entertainment	3.3	65.5	31.1
Health	2.1	61.0	36.9
News	2.8	57.0	40.2
Politics	5.9	57.8	36.3
Sport	3.5	59.3	37.2

Table 5.14: Distribution over update-types, across CNN section classifications.

focus on Factual updates for the rest of the paper, we believe that there are many fruitful directions of future work examining other categories of updates. For instance, stylistic edits made in "Health" news might reach more readers – understanding these patterns might be crucial during times of crisis. We include additional exploration in [654].

5.3.4 Predicting Factual Updates

In Section 5.3.1, we learned high-scoring models to categorize edit pairs (Equation 5.1). Now, we wish to leverage these to learn a predictive policy function:

$$\pi(a = \text{Factual-Update} | s_{t,i}, D_t) \tag{5.2}$$

Where $s_{t,i}$ and D_t are the *older* half of a revision pair. This policy function (Eq 5.2) seeks to predict how D *might* change; in other words, it asks *should there be a factual edit on sentence* $s_{t,i}$*?* The problem statement builds off of our line of inquiry introduced previously,

in Section 5.2.3. There, we introduced tasks aimed at predicting news article developments across time. We tried to predict whether a "sentence will be ADDED to, DELETED from, or UPDATED in" an older draft, to induce reasoning about article changes. However, we stopped at this *state-change* analysis. Here, we build off of this mode of inquiry and train a *true* policy function: with an *action-oriented* understanding of edits introduced in this section, we try to predict *how* information will change.

5.3.4.1 Factual Edit Prediction Dataset

To construct our task dataset, we sample revision pairs with a non-negligible amount of updates. We sample a set of 500,000 articles from *NewsEdits* that have $> 10\%$ sentences added and $> 5\%$ deleted. We acknowledge that this introduces bias into our dataset, as we focus solely on a subsection of data we *know* will update. However we build off of our broader analysis of syntactic edits patterns in Section 5.2, where we found that these kinds of articles could be predicted with reasonable accuracy. We reason that our construction makes it more likely that we are focusing on factual updates that have more significant impact on the article (as they require more substantial rewrites.) Then, we use the best-performing edit-intentions model, in Section 5.3.2.2, to produce silver labels. We assign labels $\tilde{a}_{t,ij}$ using our *inverse model* (Equation 5.1); then we discard D_{t+1}, $s_{t+1,j}$ and try to predict $a_{t,i} = \{a_{t,ij}\}_{j=1}^{m+1}$ using *just* $D_t, s_{i,t}$ (Equation 5.2).

5.3.4.2 Predicting Factual Edits

For training and development, we chronologically split our dataset into train/development sets with 80/20 ratios. The earliest 80% is our training set, the next 20% for development, etc. To keep cost reasonable, we sample 16,000 sentences for the training set and 2,000 for the development set. We test all approaches on the same gold-labeled documents D_{test}^{gold}, which were part of our gold-annotated test set (Section 5.3.2.1). In early experiments, we noticed that many fine-grained labels were too infrequent to model well, so we switched to

Model	Features	Fact F1	Not Fact F1	Macro F1	Micro F1
GPT-3.5	Sentence-Only	11.3	79.1	30.4	74.2
	Direct Context	3.4	91.8	32.2	85.2
	Full Article	7.9	91.1	49.8	85.4
GPT-4	Sentence-Only	11.1	66.3	38.9	62.4
	Direct Context	14.8	88.8	52.7	84.1
	Full Article	15.4	90.6	53.2	84.9
FT Longformer	Sentence-Only	21.2	92.3	57.4	87.0
	Direct Context	22.3	93.0	87.8	87.4
	Full Article	25.4	91.4	58.0	86.4
Human Performance	Sentence-Only	41.2	75.3	58.6	69.2

Table 5.15: How well can models predict if a sentence will have a fact update, or not? We test GPT3.5 and GPT4. Individual, macro and micro F1 scores (%) on the golden test set for various evaluated models.

predicting coarse-grained labels. We balance the training dataset to have an equal number of classes.

Factual Edit Prediction We test three different variants of Equation 5.2 to provide different degrees of article context to the policy model: (1) Sentence-Only, $\pi(a_{t,i}|s_{t,i})$; (2) Direct Context, $\pi(a_{t,i}|s_{t,i-1}, s_{t,i}, s_{t,i+1})$; and (3) Full Article, $\pi(a|s_{t,i}, D_t)$. This helps us understand how much local vs. global article features predict Factual Updates. For each variant we test zero-shot (i.e. prompted `gpt-3.5-turbo` and `gpt-4`); and fine-tuning approaches (i.e. longformer models)[36].

Results are shown in Table 5.15. Performance is moderate-to-low for detecting factual updates. However, we do observe performance increases from fine-tuning the longformer model, so to some degree this task is learnable. We recruit a former journalist, with 4 years of experience in major newsrooms, to predict labels for this task, in order to provide a human upper bound to Equation 5.2. The journalist observes the training data, and then

[36]The longformer is trained with the same approach as the silver-label prediction step from Section 5.3.2.2. In early trials, we try different variations on these experiments, like restricting the dataset to different subsets based on topic, like "Disaster" or "Safety". These topic categories, as shown in Section 5.3.3, are more fact-heavy. However, we find negligible impact on F1-score.

Sent. Contains:	Fact U.	$\overline{\text{Fact U.}}$	Δ
Recent Event	50%	8%	42%
Developing Event	30%	0%	30%
Statistic	28%	8%	19%
Info. request	12%	0%	12%
Historical Event	0%	17%	-17%
Opinion/Analysis	2%	39%	-36%
Description	10%	50%	-40%

Table 5.16: **Linguistic Cues characterizing Factual Updates**: Manual annotations of characteristics in D^{gold}_{test} sentences that either Factually Update, or not. We show the % of sentences containing these characteristics, ordered by those most salient for Factual Updates.

Sentences with $\uparrow p(l\|s_i, D)$
There are no immediate reports of casualties.
His trial has not yet started.
Officials said attackers fired as many as 30 rockets in Friday's assault.
The rebel group did not immediately comment.

Table 5.17: A small sample of sentences in the high-likelihood region of $p(l|s_i, D)$. More examples shown in Table 5.21.

scores the test set. At 41.2 F1-score, the journalist sets a moderately higher upper bound.

Linguistic Cues Characterize Factual Edits. LLMs are bad at detecting these. Interestingly, sentence-level characteristics seem to contain much of the signal for this task: as shown in Table 5.15, the performance barely increases by including the Full Article as context (a finding we did not observe in our tagging task, in Section 5.3.2). To gain a deeper intuition about these sentence-level cues, we sample 100 sentences from D^{gold}_{test} that have been labeled as either having a Factual Update or not (i.e. another kind of update, or no update at all). We show results in Table 5.16. We identify cues like the temporality of an event described in the sentence as important, and whether the sentence contains statistics, analysis or other kinds of news discourse [698]. Interestingly, sentences that Factual Update are more likely

to contain Recent Events and Developing Events, compared with Opinion, Historical Events and Description. (See [654] for definitions of these discourse patterns). This would explain in part why language models underperform human reasoning in predicting updates. We find that GPT4 generally has low agreement with human annotators on these tasks, at $\kappa = .2$. Researchers have generally found that LLMs struggle with this kind of reasoning [703, 704]. Recent modeling advancements might help us perform these tasks better [705].

This prediction task is noisy: many sentences may look similar, but may or may not have had Factual Updates, due to chance. Indeed, even expert human annotators have low prediction scores. However, we hypothesize that data that the model is most confident about (or the high-precision region), are more uniformly predictable. We show samples of these sentences in Table 5.17. These sentences contain many of the linguistic cues identified in 5.16. See Table 5.21 for more examples of high-probability sentences (and Table **??** for examples of low-probability sentences). We focus on these high-precision sentences in the next section.

5.3.5 Question Answering with Outdated Documents

We are ready to test whether the prediction models learned in the last section, to predict whether a sentence will have a Factual update, can help us in dynamic LLM Q&A tasks. We set up a RealTimeQA-style task [706], where an LLM is supplied by a retrieval system with potentially *out-of-date* information. We would like the LLM to *abstain* from answering a question if it suspects it's information might be outdated. Consider the scenario in Table 5.18. As humans, we could infer that the ongoing events in the old sentence would be of relatively short time-scale. Thus, if a retriever retrieves the old sentence for the LLM, without knowledge of the new sentence, we would like the LLM to answer the question with something like: "*I do not have the most updated information and this might change quickly*". Confidently answering without any caution as to the updating nature of events is *wrong*.

Old sentence: The White House ~~is~~ on lockdown after a vehicle struck a security barrier.
New sentence: The White House was on lockdown for about an hour after a vehicle struck ...
Question: *"Can I visit the White House right now?"*

Table 5.18: **LLM Abstention Demonstration**: In this example, the LLM only has access to the old, outdated article. We wish to probe whether LLMs can reason about the information's likelihood of being outdated and be cautious about answering this question.

	No-Conflict			**Maybe-Conflict**			**Likely-Conflict**		
	F1 Micro	F1 Macro	Avg.	F1 Micro	F1 Macro	Avg	F1 Micro	F1 Macro	Avg.
No Warn	55.9	35.8	55.9	8.8	8.1	8.8	38.8	28.0	38.8
Const. Warn.	52.9	**49.6**	52.9	90.0	47.4	90.0	64.7	54.0	64.7
w. Pred.	**59.4**	48.9	**59.4**	**90.6**	61.1	**90.6**	**67.1**	**62.4**	**67.1**
w. Oracle	57.6	47.7	57.6	90.0	**63.3**	90.0	66.5	61.1	66.5

Table 5.19: **LLM-QA Abstention Accuracy**: we measure how often GPT4 correctly abstains from answering user-questions, based on the ground truth of whether the facts in an article updated or not. Each variant shows different information that GPT4 is given. We generate questions in three categories: No-Conflict, Maybe-Conflict, Likely-Conflict, representing how likely the answer to the question will be outdated after a factual update.

5.3.5.1 LLM-QA Experiments

Experimental Design We take pairs of sentences in the gold test set of our annotated data where an update occurred, and we ask GPT4 to ask questions based on the older sentence. *No-Conflict*: 5 questions based on information in the older sentence that does *NOT* update in the newer one. *Maybe-Conflict*: 5 questions based on information in the older sentence that *might* update in the newer one. *Likely-Conflict*: 5 questions based on information from the older sentence *likely* updates with a newer one. (For all prompts, see [654]).

Experimental Variants We devise the following experimental variants. Each variant take in the *old sentence* and a *question*, generated previously. *No Warning (Baseline #1)*: We formulate a basic prompt to GPT4, without alerting it to any possibly outdated material. *Uniform Warning (Baseline #2)* We warn GPT4 that some information might be outdated. The warning is the same for all questions, so GPT has to rely on its own reasoning to detect

	No-Conflict	Maybe-Conflict	Likely-Conflict
No Warning	0.0	0.0	0.0
Uniform Warning	30.0	87.1	98.8
w. Update Pred.	10.6	74.1	95.9
w. Oracle Update	12.4	75.9	94.1

Table 5.20: **Likelihood of abstaining** in the three test cases: **No** factual conflict, **Maybe** factual conflict, **Likely** factual conflict. In general, we wish to refrain only when we need to. Over-refraining is bad.

information that could be potentially outdated. *w/ Our Update Likelihood*: We give GPT4 predictions from our Factual Update model, binned into "low", "medium", "high" update likelihood. (We use the highest-scoring LED variation).

w/ Oracle Update: We give GPT4 gold labels that a fact-update *did* or *did NOT* occur. This is designed to give us an upper bound on abstention.

Abstention Rate Evaluations We evaluate performance of each prompting strategy using a GPT4-based evaluation. We ask GPT4: (1) Is this question answerable given the information in the old sentence? (2) Is the answer consistent with the information presented in the revised sentence? We manually label a small set of 100 questions, to verify that GPT4 can perform this task, and find high agreement $\kappa > .74$ for both questions. If the answer to both questions is yes, the LLM should attempt to provide an answer. If either of the answers is "no", then we want the LLM to ABSTAIN from answering. Abstaining when it *should* is a success; any other answer is a failure. We show F1 scores in Table 5.19. Interestingly, and perhaps unexpectedly, the variant with Update Predictions does as well if not better than the variant with Oracle Updates. Perhaps the categories of the prediction score helps GPT4 better understand the task compared with the simple yes/no gold labels. The Uniform Warning (Baseline #2) variation has surprisingly strong performance as well, perhaps an indication that GPT4 does have some emergent abilities to detect the linguistics of outdated information. However, when we examine overall abstention rates, shown in Table 5.20, we find that this baseline has a far abstention rate. Meanwhile, the variant with Update

Top Predictions for Content Evolution Prediction, $p(l = \text{Fact Update} \mid s_i, D)$
The company takes this recommendation extremely seriously," it said in a statement.
KABUL, Afghanistan — An Afghan official says a powerful suicide bombing has targeted a U.S. military convoy near the main American Bagram Air Base north of the capital Kabul.
WASHINGTON — The U.S. carried out military strikes in Iraq and Syria targeting a militia blamed for an attack that killed an American contractor, a Defense Department spokesman said Sunday.
Mr. Causey, who reported his concern to authorities, was not charged in the indictment, which a grand jury returned last month, and did not immediately comment.
His trial has not yet started.
MEXICO CITY — A fiery freeway accident involving a bus and a tractor-trailer killed 21 people in the Mexican state of Veracruz on Wednesday, according to the authorities and local news outlets.
The indictment accuses Mr. Hayes, a former congressman, of helping to route $250,000 in bribes to the re-election campaign of Mike Causey, the insurance commissioner.
No Kenyans died in the attack, Kenya's military spokesman Paul Njuguna said Monday.
Mr. Manafort, 70, will most likely be arraigned on the new charges in State Supreme Court in Manhattan later this month and held at Rikers, though his lawyers could seek to have him held at a federal jail in New York, the people with knowledge said.
Officials said attackers fired as many as 30 rockets in Friday's assault.
KABUL, Afghanistan — Gunmen attacked a remembrance ceremony for a minority Shiite leader in Afghanistan's capital on Friday, wounding at least 18 people, officials said.
BEIRUT — A senior Turkish official says Turkey has captured the older sister of the slain leader of the Islamic State group in northwestern Syria, calling the arrest an intelligence "gold mine. "
Paul J. Manafort, President Trump's former campaign chairman who is serving a federal prison sentence, is expected to be transferred as early as this week to the Rikers Island jail complex in New York City, where he will most likely be held in solitary confinement while facing state fraud charges, people with knowledge of the matter said.
The watchdog, the Securities and Exchange Surveillance Commission, said Tuesday it made the recommendation to the government's Financial Services Agency on the disclosure documents from 2014 through 2017.
There are no immediate reports of casualties.
It said the U.S. hit three of the militia's sites in Iraq and two in Syria, including weapon caches and the militia's command and control bases.
The rebel group did not immediately comment.
Kep provincial authorities later announced a total of five dead and 18 injured.
QUETTA, Pakistan — Attackers used a remotely-controlled bomb and assault rifles to ambush a convoy of Pakistani troops assigned to protect an oil and gas facility in the country's restive southwest, killing six soldiers and wounding four, officials said Tuesday.
WASHINGTON — Senator Bernie Sanders of Vermont raised $18.2 million over the first six weeks of his presidential bid, his campaign announced Tuesday, a display of financial strength that cements his status as one of the top fund-raisers in the sprawling Democratic field.

Table 5.21: Sample of the most likely fact-update sentences, as judged by our top-performing model. Top predictions reflect a combination of statistics, recent or upcoming events, and waiting for quotes.

Predictions abstains at nearly the same rates as that with Oracle Updates.

Summary The ability of our prediction tags to recover near-oracle performance signals that factual edit prediction can serve a useful role in LLM Q&A. We do suspect there to be an inherent upper bound in our ability to model such revision patterns. Randomness undoubtedly exists in the editing and revision process; for many factual updates where, perhaps, the ethical stakes of outdated information are lower, journalists may choose not to go back and revise. We still see such work as promising. Indeed, it is surprising that, despite low scores on the modeling components for Part 1 (Edit-Intention Tagging) and Part 2 (Factual Edit Prediction), we still observe useful downstream applications in Part 3. The linguistic insights we are observe concord with human intuition, and identify known shortcomings of current language models.

5.4 Chapter Conclusion

In this Chapter, we observed how additional, partial observability into human workflows could yield tangible insights and improvements in *emulation learning*. Mirroring "ghost condition" experiments in cognitive science [89], we show how edit revision histories can be leveraged to provide intermediate state information: given $s_{i,j}$ where i is the version number and j is the draft number within the version, we have derived observability into: $s_{1,n_1}, s_{2,n_2} \ldots$, the sequence of *final* drafts in each version. In Section 5.2, we introduced the *NewsEdits* dataset, which provides this version-level observability; we developed *observation channels* to parse *atomic* state-changes between versions (i.e. Addition, Edit, Deletion). We also showed composability and predictability in edits; in other words, we showed that $P(s_{t+1}|s_t)$ was predictable, highlighting the role that edit histories can play in providing useful temporal orderings. In Section 5.3, we introduced an action space, $\mathcal{A}$ on top of *atomic* state-changes. We showed that not only could we learn *and predict* these actions, but policy models $\pi(a_{t+1}|a_t, s_t)$ were practically useful for us to integrate into downstream tasks like QA abstention.

This Chapter suitably closes my work, as it completes the *emulation* formulation of news edits. Edits are an especially exciting direction in *emulation*: despite the existence of many revisions datasets [707, 708], they are not commonly used, to my knowledge, for the purposes of increased state observability and better action inference. This direction and it's potential in *emulation learning* has barely been scratched. I hope it can emerge as an important part of learning complex, creative workflows. I am also personally proud of this work, as it explicitly brings together so many of the multiple different discourse schemata, covering multiple parts of the news-generating creative process into one unified action vocabulary. I hope more broadly that the work introduced here has rich directions forward. We hope in future work to revise directions around stylistic and narrative edits, both of which we believe can lead to better tools for computational journalists.

Bibliography

[1] Alexander Spangher et al. “Identifying Informational Sources in News Articles”. In: *arXiv preprint arXiv:2305.14904* (2023).

[2] Pieter Abbeel and Andrew Y Ng. “Inverse reinforcement learning”. In: *Encyclopedia of machine learning*. Springer, 2011, pp. 554–558.

[3] Pengjie Gao, Chang Lee, and Dermot Murphy. “Financing Dies in Darkness? The Impact of Newspaper Closures on Public Finance”. In: *Political Economy: Government Expenditures & Related Policies eJournal* (2019). URL: https://api.semanticscholar.org/CorpusID:85451380.

[4] Jonas Heese, Gerardo Perez Cavazos, and Caspar David Peter. “When the Local Newspaper Leaves Town: The Effects of Local Newspaper Closures on Corporate Misconduct”. In: *Political Institutions: Federalism & Sub-National Politics eJournal* (2021). URL: https://api.semanticscholar.org/CorpusID:236975019.

[5] James M. Jr. Snyder and David Strömberg. “Press Coverage and Political Accountability”. In: *Journal of Political Economy* 118 (2008), pp. 355–408. URL: https://api.semanticscholar.org/CorpusID:154635874.

[6] James T Hamilton. *Democracy’s detectives: The economics of investigative journalism*. Harvard University Press, 2016.

[7] Thomas Peele et al. “Don’t Stop the Presses! When Local News Struggles, Democracy Withers”. In: *Wired* (Nov. 30, 2017).

[8] Victor Pickard. “A new business model for journalism”. In: *Axios* (Feb. 22, 2024).

[9] K. T. Greene et al. “An evaluation of online information acquisition in US news ...” In: *Nature (Scientific Reports)* (2024).

[10] Northwestern University Medill School of Journalism. “State of Local News: 2024 Report”. In: *Reuters Institute (via Medill)* (2024).

[11] Peter Carragher, Evan M. Williams, and Kathleen M. Carley. “Misinformation Resilient Search Rankings with Webgraph-Based Interventions”. In: *ACM Transactions on Intelligent Systems and Technology* 16.1 (2025), pp. 1–27. DOI: 10.1145/3670410.

[12] Rajvardhan Oak et al. “Re-ranking Using Large Language Models for Mitigating Exposure to Harmful Content on Social Media Platforms”. In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Vienna, Austria: Association for Computational Linguistics, July 2025, pp. 894–908. DOI: 10.18653/v1/2025.acl-long.44. URL: https://aclanthology.org/2025.acl-long.44/.

[13] Bowen Jin et al. *Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning*. arXiv preprint. 2025. DOI: 10.48550/arXiv.2503.09516. arXiv: 2503.09516 [cs.CL]. URL: https://arxiv.org/abs/2503.09516.

[14] Puxuan Yu et al. "Search Result Diversification Using Query Aspects as Bottlenecks". In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23)*. Birmingham, United Kingdom: Association for Computing Machinery, 2023, pp. 3040–3051. DOI: 10.1145/3583780.3615050. URL: https://dl.acm.org/doi/10.1145/3583780.3615050.

[15] Shivanshu Gupta. "Demonstration Selection and Task Formulation for Effective In-Context Learning". Ph.D. dissertation. PhD thesis. Irvine, CA: University of California, Irvine, 2025. URL: https://escholarship.org/uc/item/3h22z31f.

[16] Vladimir Karpukhin et al. "Dense Passage Retrieval for Open-Domain Question Answering". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020, pp. 6769–6781. DOI: 10.18653/v1/2020.emnlp-main.550. URL: https://aclanthology.org/2020.emnlp-main.550/.

[17] Alexander Spangher et al. "Tracking the newsworthiness of public documents". In: *arXiv preprint arXiv:2311.09734* (Aug. 2023). Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar, pp. 14150–14168. DOI: 10.18653/v1/2024.acl-long.763. URL: https://aclanthology.org/2024.acl-long.763.

[18] Alexander Spangher et al. "A Novel Multi-Document Retrieval Benchmark: Journalist Source-Selection in Newswriting". In: *Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing*. Ed. by Weijia Shi et al. Albuquerque, New Mexico, USA: Association for Computational Linguistics, May 2025, pp. 180–204. ISBN: 979-8-89176-229-9. DOI: 10.18653/v1/2025.knowledgenlp-1.18. URL: https://aclanthology.org/2025.knowledgenlp-1.18/.

[19] Alexander Spangher et al. "DiscoSum: Discourse-aware News Summarization". In: *arXiv preprint arXiv:2506.06930* (2025).

[20] Louis Bradshaw et al. "Scaling Self-Supervised Representation Learning for Symbolic Piano Performance". In: *arXiv preprint arXiv:2506.23869* (2025).

[21] Ryan Lee, Alexander Spangher, and Xuezhe Ma. "Patentedits: Framing patent novelty as textual entailment". In: *arXiv preprint arXiv:2411.13477* (2024).

[22] Susanne K Langer. *Philosophy in a new key: A study in the symbolism of reason, rite, and art*. Harvard University Press, 1942.

[23] Johan Galtung and Mari Holmboe Ruge. "The structure of foreign news: The presentation of the Congo, Cuba and Cyprus crises in four Norwegian newspapers". In: *Journal of peace research* 2.1 (1965), pp. 64–90.

[24] Stuart Hall. *Writings on media: History of the present*. Duke University Press, 2021.

[25] Teun A Van Dijk. *News as discourse*. Routledge, 2013.

[26] Alexander Spangher et al. "Sequentially Controlled Text Generation". In: *arXiv preprint arXiv:2301.02299* (2023).

[27] Arnold Schoenberg, Gerald Strang, and Leonard Stein. "Fundamentals of musical composition". In: (1967).

[28] Shunyu Yao et al. "React: Synergizing reasoning and acting in language models". In: *International Conference on Learning Representations*. 2023.

[29] B. F. Skinner. *Walden Two*. New York: Macmillan, 1948.

[30] Burrhus Frederic Skinner. *Science and human behavior*. 92904. Simon and Schuster, 1965.

[31] Willem JM Levelt. "Producing spoken language: A blueprint of the speaker". In: *The neurocognition of language* 83 (1999), p. 122.

[32] Christiane Donahue and Theresa Lillis. "Models of writing and text production". In: *Handbook of writing and text production* (2014), pp. 55–78.

[33] Hao Sun and Mihaela van der Schaar. "Inverse Reinforcement Learning Meets Large Language Model Post-Training: Basics, Advances, and Opportunities". In: *arXiv preprint arXiv:2507.13158* (2025). arXiv: 2507.13158.

[34] Azra Ismail et al. "Public Health Calls for/with AI: An Ethnographic Perspective". In: *Proceedings of the ACM on Human-Computer Interaction* 7 (2023), pp. 1–26. URL: https://api.semanticscholar.org/CorpusID:263621116.

[35] Asbjørn Malte Pedersen and Claus Bossen. "Data Work Between the Local and the Global: An Ethnography of a Healthcare Business Intelligence Unit". In: *Proceedings of the ACM on Human-Computer Interaction* 8 (2024), pp. 1–28. URL: https://api.semanticscholar.org/CorpusID:269461412.

[36] Riyaj Isamiya Shaikh et al. "Fleeting Alliances and Frugal Collaboration in Piecework: A Video-Analysis of Food Delivery Work in India". In: *Comput. Support. Cooperative Work.* 33 (2024), pp. 1289–1342. URL: https://api.semanticscholar.org/CorpusID:270678856.

[37] Kalle Kusk. "Flexible Platforms? An Ethnographic Study of Flexible Scheduling in Platform-Mediated Delivery". In: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (2025). URL: https://api.semanticscholar.org/CorpusID:278063039.

[38] Ju-Yeon Jung et al. "How Domain Experts Work with Data: Situating Data Science in the Practices and Settings of Craftwork". In: *Proceedings of the ACM on Human-Computer Interaction* 6 (2022), pp. 1–29. URL: https://api.semanticscholar.org/CorpusID:248002739.

[39] Sachita Nishal, Jasmine Sinchai, and Nicholas Diakopoulos. "Understanding practices around computational news discovery tools in the domain of science journalism". In: *Proceedings of the ACM on Human-Computer Interaction* 8.CSCW1 (2024), pp. 1–36.

[40] S. Petridis et al. "AngleKindling: Supporting Journalistic Angle Ideation with Large Language Models". In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (2023). URL: https://api.semanticscholar.org/CorpusID:258217880.

[41] Jacob W. Getzels and Mihaly Csikszentmihalyi. *The Creative Vision: A Longitudinal Study of Problem Finding in Art*. New York: John Wiley & Sons, 1976. ISBN: 0471014869.

[42] Pranab Sahoo et al. "A Survey of Prompt Engineering Methods in Large Language Models for Different NLP Tasks". In: *arXiv preprint arXiv:2407.12994* (2024). URL: https://arxiv.org/abs/2407.12994.

[43] Pranab Sahoo et al. "A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications". In: *arXiv preprint arXiv:2402.07927* (2024). URL: https://arxiv.org/abs/2402.07927.

[44] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, et al. "WebGPT: Browser-assisted Question Answering with Human Feedback". In: *arXiv preprint arXiv:2112.09332* (2021).

[45] Yijia Shao et al. "Assisting in Writing Wikipedia-like Articles From Scratch with Large Language Models". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 2024, pp. 6252–6278.

[46] Rujun Han et al. “Deep Researcher with Test-Time Diffusion”. In: *arXiv preprint arXiv:2507.16075* (2025).

[47] Ann Yuan et al. “Wordcraft: story writing with large language models”. In: *Proceedings of the 27th International Conference on Intelligent User Interfaces*. 2022, pp. 841–852.

[48] Mina Lee, Percy Liang, and Qian Yang. “Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities”. In: *Proceedings of the 2022 CHI conference on human factors in computing systems*. 2022, pp. 1–19.

[49] Piotr Mirowski et al. “Co-Writing Screenplays and Theatre Scripts with Language Models: An Evaluation by Industry Professionals”. In: *arXiv preprint arXiv:2209.14958* (2022).

[50] Katy Ilonka Gero, Tao Long, and Lydia B Chilton. “Social dynamics of AI support in creative writing”. In: *Proceedings of the 2023 CHI conference on human factors in computing systems*. 2023, pp. 1–15.

[51] Tuhin Chakrabarty et al. “Creativity support in the age of large language models: An empirical study involving emerging writers”. In: *arXiv preprint arXiv:2309.12570* (2023).

[52] Joon Sung Park et al. “Generative agents: Interactive simulacra of human behavior”. In: *Proceedings of the 36th annual acm symposium on user interface software and technology*. 2023, pp. 1–22.

[53] Guanzhi Wang et al. “Voyager: An open-ended embodied agent with large language models”. In: *arXiv preprint arXiv:2305.16291* (2023).

[54] Liwei Jiang et al. “Investigating machine moral judgement through the Delphi experiment”. In: *Nature Machine Intelligence* 7.1 (2025), pp. 145–160.

[55] Caleb Ziems et al. “NormBank: A knowledge bank of situational social norms”. In: *arXiv preprint arXiv:2305.17008* (2023).

[56] Badr Alkhamissi et al. “Investigating Cultural Alignment of Large Language Models”. In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2024, pp. 12404–12422.

[57] Seungju Han et al. “Reading Books is Great, But Not if You Are Driving! Visually Grounded Reasoning about Defeasible Commonsense Norms”. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 2023, pp. 894–914.

[58] Sachita Nishal, Eric Lee, and Nicholas Diakopoulos. "De-jargonizing Science for Journalists with GPT-4: A Pilot Study". In: *arXiv preprint arXiv:2410.12069* (2024).

[59] Joris Veerbeek and Nicholas Diakopoulos. "Using Generative Agents to Create Tip Sheets for Investigative Data Reporting". In: ().

[60] Jiho Shin et al. "Prompt Engineering or Fine Tuning: An Empirical Assessment of Large Language Models in Automated Software Engineering Tasks". In: *CoRR* (2023).

[61] Boqi Chen, Fandi Yi, and Dániel Varró. "Prompting or fine-tuning? A comparative study of large language models for taxonomy construction". In: *2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. IEEE. 2023, pp. 588–596.

[62] Ive Botunac, Marija Brkić Bakarić, and Maja Matetić. "Comparing fine-tuning and prompt engineering for multi-class classification in hospitality review analysis". In: *Applied Sciences* 14.14 (2024), p. 6254.

[63] Zezhong Wang et al. "Fine-tuning after Prompting: an Explainable Way for Classification". In: *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*. 2024, pp. 133–142.

[64] John Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[65] Rafael Rafailov et al. "Direct preference optimization: Your language model is secretly a reward model". In: *Advances in neural information processing systems* 36 (2023), pp. 53728–53741.

[66] Andriy Mnih and Karol Gregor. "Neural variational inference and learning in belief networks". In: *International Conference on Machine Learning*. PMLR. 2014, pp. 1791–1799.

[67] Tao Lei, Regina Barzilay, and Tommi Jaakkola. "Rationalizing Neural Predictions". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Ed. by Jian Su, Kevin Duh, and Xavier Carreras. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 107–117. DOI: 10.18653/v1/D16-1011. URL: https://aclanthology.org/D16-1011/.

[68] Zeqiu Wu et al. "Fine-grained human feedback gives better rewards for language model training". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 59008–59033.

[69] Xiaobao Wu. "Sailing by the Stars: A Survey on Reward Models and Learning Strategies for Learning from Rewards". In: *arXiv preprint arXiv:2505.02686* (2025).

[70] Shuhe Wang et al. "Reinforcement learning enhanced llms: A survey". In: *arXiv preprint arXiv:2412.10400* (2024).

[71] Daya Guo et al. "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning". In: *arXiv preprint arXiv:2501.12948* (2025).

[72] Ishita Dasgupta et al. "Language models show human-like content effects on reasoning tasks". In: *arXiv preprint arXiv:2207.07051* (2022).

[73] Qing Lyu et al. "Faithful chain-of-thought reasoning". In: *The 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL 2023)*. 2023.

[74] Samuel Gershman and Noah Goodman. "Amortized inference in probabilistic reasoning". In: *Proceedings of the annual meeting of the cognitive science society*. Vol. 36. 36. 2014.

[75] Noah D Goodman and Michael C Frank. "Pragmatic language interpretation as probabilistic inference". In: *Trends in cognitive sciences* 20.11 (2016), pp. 818–829.

[76] Sang Michael Xie et al. "An Explanation of In-context Learning as Implicit Bayesian Inference". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=RdJVFCHjUMI.

[77] Miles Turpin et al. "Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting". In: *Advances in Neural Information Processing Systems*. Vol. 36. 2023, pp. 74952–74965.

[78] Tamera Lanham et al. "Measuring Faithfulness in Chain-of-Thought Reasoning". In: *CoRR* (2023).

[79] Huatong Song et al. "R1-Searcher: Incentivizing the Search Capability in LLMs via Reinforcement Learning". In: *CoRR* (2025).

[80] Mingyang Chen et al. "Learning to reason with search for llms via reinforcement learning". In: *arXiv preprint arXiv:2503.19470* (2025).

[81] Yuxiang Zheng et al. "Deepresearcher: Scaling deep research via reinforcement learning in real-world environments". In: *arXiv preprint arXiv:2504.03160* (2025).

[82] Zhepei Wei et al. "Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning". In: *arXiv preprint arXiv:2505.16421* (2025).

[83] Ethan Hsu et al. “WebDS: An End-to-End Benchmark for Web-based Data Science”. In: *arXiv preprint arXiv:2508.01222* (2025).

[84] Nisan Stiennon et al. “Learning to summarize with human feedback”. In: *Advances in neural information processing systems* 33 (2020), pp. 3008–3021.

[85] Alexander Gurung and Mirella Lapata. “Learning to reason for long-form story generation”. In: *arXiv preprint arXiv:2503.22828* (2025).

[86] Rafael Pardinas et al. “Leveraging human preferences to master poetry”. In: *The AAAI-23 Workshop on Creative AI Across Modalities*. 2023.

[87] David J. Wood. *How Children Think and Learn*. Oxford: Basil Blackwell, 1988.

[88] Christine A Caldwell et al. “End state copying by humans (Homo sapiens): implications for a comparative perspective on cumulative culture.” In: *Journal of Comparative Psychology* 126.2 (2012), p. 161.

[89] Lydia M Hopper et al. “Observational learning in chimpanzees and children studied through ‘ghost’conditions”. In: *Proceedings of the Royal Society B: Biological Sciences* 275.1636 (2008), pp. 835–840.

[90] Christine A Caldwell and Ailsa E Millen. “Studying cumulative cultural evolution in the laboratory”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 363.1509 (2008), pp. 3529–3539.

[91] Christine A Caldwell and Ailsa E Millen. “Experimental models for testing hypotheses about cumulative cultural evolution”. In: *Evolution and Human Behavior* 29.3 (2008), pp. 165–171.

[92] David G Jansson and Steven M Smith. “Design fixation”. In: *Design studies* 12.1 (1991), pp. 3–11.

[93] Ut Na Sio, Kenneth Kotovsky, and Jonathan Cagan. “Fixation or inspiration? A meta-analytic review of the role of examples on design processes”. In: *Design Studies* 39 (2015), pp. 70–99.

[94] Nicolas Fay, Simon Garrod, and Leo Roberts. “The fitness and functionality of culturally evolved communication systems”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 363.1509 (2008), pp. 3553–3561.

[95] Annette Barnes. *Languages of Art: An Approach to a Theory of Symbols*. 1971.

[96] Kendall L Walton. *Mimesis as make-believe: On the foundations of the representational arts*. Harvard University Press, 1993.

[97] Tessa Verhoef. "The origins of duality of patterning in artificial whistled languages". In: *Language and cognition* 4.4 (2012), pp. 357–380.

[98] Nori Jacoby and Josh H McDermott. "Integer ratio priors on musical rhythm revealed cross-culturally by iterated reproduction". In: *Current biology* 27.3 (2017), pp. 359–370.

[99] Bruno Latour. "Visualisation and cognition: Drawing things together". In: *AVANT. Pismo Awangardy Filozoficzno-Naukowej* 3 (2012), pp. 207–257.

[100] Steven Shapin and Simon Schaffer. *Leviathan and the air-pump: Hobbes, Boyle, and the experimental life*. Princeton University Press, 2011.

[101] Bruno Latour, Jonas Salk, and Steve Woolgar. "Laboratory life: The construction of scientific facts". In: (2013).

[102] Michael Schudson. *Discovering the news: A social history of American newspapers*. Basic books, 1981.

[103] Michael Schudson. *The power of news*. Harvard University Press, 1995.

[104] Michael Schudson. "The objectivity norm in American journalism". In: *Journalism* 2.2 (2001), pp. 149–170.

[105] Michael Schudson. "The sociology of news production". In: *Media, Culture & Society* 11.3 (1989), pp. 263–282.

[106] Oscar Gandy. "Beyond agenda-setting". In: *Agenda setting*. Routledge, 2016, pp. 263–275.

[107] Raymond A Harder, Julie Sevenans, and Peter Van Aelst. "Intermedia agenda setting in the social media age: How traditional players dominate the news agenda in election times". In: *The international journal of press/politics* 22.3 (2017), pp. 275–293.

[108] Michael Polanyi. "The tacit dimension". In: *Knowledge in organisations*. Routledge, 2009, pp. 135–146.

[109] Harry Collins. *Tacit and explicit knowledge*. University of Chicago press, 2019.

[110] Gilbert Ryle and Julia Tanney. *The concept of mind*. Routledge, 2009.

[111] Lucille Alice Suchman. *Plans and situated actions: The problem of human-machine communication*. Cambridge university press, 1987.

[112] Donald A Schön. *The reflective practitioner: How professionals think in action*. Routledge, 2017.

[113] Jean Lave and Etienne Wenger. *Situated learning: Legitimate peripheral participation*. Cambridge university press, 1991.

[114] Harry Collins and Robert Evans. *Rethinking expertise*. University of Chicago press, 2019.

[115] Hubert Dreyfus and Stuart E Dreyfus. *Mind over machine*. Simon and Schuster, 1986.

[116] R Thomas McCoy et al. "Embers of autoregression show how large language models are shaped by the problem they are trained to solve". In: *Proceedings of the National Academy of Sciences* 121.41 (2024), e2322420121.

[117] Michael A.K. Halliday and Ruqaiya Hasan. "Cohesion in English". In: 1976. URL: https://api.semanticscholar.org/CorpusID:62192469.

[118] Jerry R. Hobbs. "Coherence and Coreference". In: *Cogn. Sci.* 3 (1979), pp. 67–90. URL: https://api.semanticscholar.org/CorpusID:45706253.

[119] Rashmi Prasad, Bonnie Webber, and Aravind Joshi. "The Penn Discourse Treebank: An annotated corpus of discourse relations". In: *Handbook of linguistic annotation*. Springer, 2017, pp. 1197–1217.

[120] Bonnie Webber et al. "The penn discourse treebank 3.0 annotation manual". In: *Philadelphia, University of Pennsylvania* 35 (2019), p. 108.

[121] Lynn Carlson, Mary Ellen Okurowski, and Daniel Marcu. *RST discourse treebank*. Linguistic Data Consortium, University of Pennsylvania, 2002.

[122] William C. Mann and Sandra A. Thompson. "Rhetorical Structure Theory: Toward a functional theory of text organization". In: *Text & Talk* 8 (1988), pp. 243–281. URL: https://api.semanticscholar.org/CorpusID:60514661.

[123] Yan Huang, ed. *The Oxford Handbook of Pragmatics*. Oxford: Oxford University Press, 2017.

[124] Alex Lascarides and Nicholas Asher. "Segmented Discourse Representation Theory: Dynamic Semantics With Discourse Structure". In: 2008. URL: https://api.semanticscholar.org/CorpusID:8730950.

[125] Teun A Van Dijk. "Discourse analysis: Its development and application to the structure of news". In: *Journal of communication* 33.2 (1983), pp. 20–43.

[126] Teun A Van Dijk. *News as discourse*. Routledge, 1988.

[127] Xiangci Li, Gully Burns, and Nanyun Peng. "Scientific Discourse Tagging for Evidence Extraction". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 2550–2562. DOI: 10.18653/v1/2021.eacl-main.218. URL: https://aclanthology.org/2021.eacl-main.218.

[128] Khalid Al-Khatib et al. "A News Editorial Corpus for Mining Argumentation Strategies". In: *26th International Conference on Computational Linguistics (COLING 2016)*. Ed. by Yuji Matsumoto and Rashmi Prasad. Association for Computational Linguistics, Dec. 2016, pp. 3433–3443. URL: https://aclanthology.org/C16-1324/.

[129] Alexander Spangher et al. "LegalDiscourse: Interpreting when laws apply and to whom". In: *Proceedings of the 2024 Conference of NAACL-HLT*. 2024, pp. 8528–8551.

[130] Prafulla Kumar Choubey et al. "Discourse as a function of event: Profiling discourse structure in news articles around the main event". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 5374–5386. DOI: 10.18653/v1/2020.acl-main.478. URL: https://www.aclweb.org/anthology/2020.acl-main.478.

[131] L. S. Vygotsky. *Thinking and Speech*. Available online at Marxists Internet Archive. 1934. URL: https://www.marxists.org/archive/vygotsky/works/words/Thinking-and-Speech.pdf.

[132] Peter Carruthers. "The cognitive functions of language". In: *Behavioral and brain sciences* 25.6 (2002), pp. 657–674.

[133] Linda S. Flower and J. R. Hayes. "A Cognitive Process Theory of Writing". In: *College Composition & Communication* (1981). URL: https://api.semanticscholar.org/CorpusID:18484126.

[134] Zae Myung Kim et al. "Align to Structure: Aligning Large Language Models with Structural Information". In: *arXiv preprint arXiv:2504.03622* (2025).

[135] Barbara J Grosz and Candace L Sidner. "Attention, intentions, and the structure of discourse". In: *Computational linguistics* 12.3 (1986), pp. 175–204.

[136] Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. "Centering: A framework for modeling the local coherence of discourse". In: *Computational Linguistics* 21.2 (1995), pp. 203–225.

[137] Willem J. M. Levelt. *Speaking: From Intention to Articulation*. Cambridge, MA: MIT Press, 1989.

[138] Willem J. M. Levelt. "Producing spoken language: A blueprint of the speaker". In: *The Neurocognition of Language*. Ed. by Colin M. Brown and Peter Hagoort. Oxford: Oxford University Press, 1999, pp. 83–122.

[139] Zenzi M. Griffin and Kathryn Bock. "What the Eyes Say about Speaking". In: *Psychological Science* 11.4 (2000), pp. 274–279. DOI: 10.1111/1467-9280.00255.

[140] J. Kathryn Bock. "Syntactic persistence in language production". In: *Cognitive Psychology* 18.3 (1986), pp. 355–387. DOI: 10.1016/0010-0285(86)90004-6.

[141] J. Kathryn Bock and Richard K. Warren. "Conceptual accessibility and syntactic structure in sentence formulation". In: *Cognition* 21.1 (1985), pp. 47–67. DOI: 10.1016/0010-0277(85)90023-X.

[142] Fernanda Ferreira and Benjamin Swets. "How incremental is language production? Evidence from the production of utterances requiring the computation of arithmetic sums". In: *Journal of Memory and Language* 46.1 (2002), pp. 57–84. DOI: 10.1006/jmla.2001.2797.

[143] T Jaeger and Roger Levy. "Speakers optimize information density through syntactic reduction". In: *Advances in neural information processing systems* 19 (2006).

[144] T. Florian Jaeger. "Redundancy and reduction: Speakers manage syntactic information density". In: *Cognitive Psychology* 61.1 (2010), pp. 23–62. DOI: 10.1016/j.cogpsych.2010.02.002.

[145] Alexander Spangher et al. "Multitask semi-supervised learning for class-imbalanced discourse classification". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 498–517. DOI: 10.18653/v1/2021.emnlp-main.40. URL: https://aclanthology.org/2021.emnlp-main.40.

[146] Chris L Baker, Rebecca Saxe, and Joshua B Tenenbaum. "Action understanding as inverse planning". In: *Cognition* 113.3 (2009), pp. 329–349.

[147] Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[148] Alex Warstadt et al. "BLiMP: The benchmark of linguistic minimal pairs for English". In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 377–392.

[149] Andrew Whiten et al. "Emulation, imitation, over-imitation and the scope of culture for child and chimpanzee". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 364 (2009), pp. 2417–2428. URL: https://api.semanticscholar.org/CorpusID:15697790.

[150] Dongyeop Kang and Eduard Hovy. "Plan ahead: Self-Supervised Text Planning for Paragraph Completion Task". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 6533–6543.

[151] Jonathan Evans. "Dual-processing accounts of reasoning, judgment, and social cognition." In: *Annual review of psychology* 59 (2008), pp. 255–78. URL: https://api.semanticscholar.org/CorpusID:12246493.

[152] Stephanie O'Donohoe and Adam Ferrier. "Thinking, Fast and Slow". In: *International Journal of Advertising* 31 (2012), pp. 445–446. URL: https://api.semanticscholar.org/CorpusID:149191349.

[153] Jonathan Evans and Keith Frankish. "In two minds: Dual processes and beyond." In: 2009. URL: https://api.semanticscholar.org/CorpusID:142629976.

[154] Violet Xiang et al. "Towards system 2 reasoning in llms: Learning how to think with meta chain-of-thought". In: *arXiv preprint arXiv:2501.04682* (2025).

[155] Yoshua Bengio, Yann LeCun, and Geoffrey E. Hinton. "Deep learning for AI". In: *Communications of the ACM* 64 (2021), pp. 58–65. URL: https://api.semanticscholar.org/CorpusID:235495130.

[156] Zhong-Zhi Li et al. "From system 1 to system 2: A survey of reasoning large language models". In: *arXiv preprint arXiv:2502.17419* (2025).

[157] A. Bandura. "Social learning theory". In: *Canadian Journal of Sociology-cahiers Canadiens De Sociologie* 2 (1977), p. 321. URL: https://api.semanticscholar.org/CorpusID:227319622.

[158] Andrew N. Meltzoff. "'Like me': a foundation for social cognition." In: *Developmental science* 10 1 (2007), pp. 126–34. URL: https://api.semanticscholar.org/CorpusID:7157186.

[159] Cristine H. Legare and Mark Nielsen. "Imitation and Innovation: The Dual Engines of Cultural Learning". In: *Trends in Cognitive Sciences* 19 (2015), pp. 688–699. URL: https://api.semanticscholar.org/CorpusID:3664635.

[160] Harriet Over and Malinda Carpenter. "The social side of imitation". In: *Child Development Perspectives* 7 (2013), pp. 6–11. URL: https://api.semanticscholar.org/CorpusID:145356998.

[161] Pieter Abbeel and Andrew Y Ng. "Apprenticeship learning via inverse reinforcement learning". In: *Proceedings of the twenty-first international conference on Machine learning*. 2004, p. 1.

[162] Andrew Y Ng, Stuart Russell, et al. "Algorithms for inverse reinforcement learning." In: *Icml*. Vol. 1. 2. 2000, p. 2.

[163] Faraz Torabi, Garrett Warnell, and Peter Stone. "Behavioral Cloning from Observation". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*. 2018.

[164] Faraz Torabi, Garrett Warnell, and Peter Stone. "Generative Adversarial Imitation from Observation". In: *arXiv preprint arXiv:1807.06158* (2018).

[165] Henry A. Kautz and James F. Allen. "Generalized Plan Recognition". In: *Proceedings of the 5th National Conference on Artificial Intelligence (AAAI)*. 1986.

[166] Miquel Ramírez and Hector Geffner. "Plan Recognition as Planning". In: *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*. 2009, pp. 1778–1783.

[167] Tom Schaul et al. "Universal Value Function Approximators". In: *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. 2015.

[168] Marcin Andrychowicz et al. "Hindsight Experience Replay". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2017.

[169] Emmanuel Bengio et al. "Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2021.

[170] Nikolay Malkin et al. "Trajectory Balance: Improved Credit Assignment in GFlowNets". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2022.

[171] Allen Newell and Herbert A. Simon. *Human Problem Solving*. Englewood Cliffs, NJ: Prentice–Hall, 1972.

[172] Richard E. Fikes and Nils J. Nilsson. "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving". In: *Artificial Intelligence* 2.3–4 (1971), pp. 189–208.

[173] Gabriel Peyré and Marco Cuturi. *Computational Optimal Transport*. Now Publishers, 2019.

[174] Christian Léonard. "A Survey of the Schrödinger Problem and Some of its Connections with Optimal Transport". In: *Discrete and Continuous Dynamical Systems - Series A* 34.4 (2014), pp. 1533–1574.

[175] Diederik P Kingma and Max Welling. "Auto-encoding variational {Bayes}". In: *Int. Conf. on Learning Representations*.

[176] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. "Variational inference: A review for statisticians". In: *Journal of the American statistical Association* 112.518 (2017), pp. 859–877.

[177] Yann LeCun, Sumit Chopra, and Raia Hadsell. "A Tutorial on Energy-Based Learning". In: *Predicting Structured Data*. MIT Press, 2006.

[178] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *Proceedings of the 9th International Conference on Learning Representations (ICLR)*. 2021.

[179] Sergey Levine. "Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review". In: *arXiv preprint arXiv:1805.00909* (2018).

[180] Emanuel Todorov. "Linearly-Solvable Markov Decision Problems". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2006.

[181] Brian D. Ziebart et al. "Maximum Entropy Inverse Reinforcement Learning". In: *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI)*. 2008.

[182] Chao Yang et al. "Imitation Learning from Observations by Minimizing Inverse Dynamics Disagreement". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2019.

[183] Seonghyeon Ye et al. "Latent Action Pretraining from Videos". In: *The Thirteenth International Conference on Learning Representations*.

[184] Judea Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Reading, MA: Addison–Wesley, 1984.

[185] Richard E. Korf. "Depth-first Iterative-deepening: An Optimal Admissible Tree Search". In: *Artificial Intelligence* 27.1 (1985), pp. 97–109.

[186] Chong Wang and David M Blei. "Collaborative topic modeling for recommending scientific articles". In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2011, pp. 448–456.

[187] Alexander Spangher. "Building the next New York Times recommendation engine". In: *The New York Times* (2015), pp. 08–26.

[188] Prem Gopalan, Laurent Charlin, and David M Blei. "Content-based recommendations with Poisson factorization". In: *Advances in neural information processing systems* 27 (2014).

[189] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. "Maximum Likelihood from Incomplete Data via the EM Algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–38.

[190] Carl Doersch. "Tutorial on variational autoencoders". In: *arXiv preprint arXiv:1606.05908* (2016).

[191] Lawrence R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[192] K. Lari and S. J. Young. "The Estimation of Stochastic Context-Free Grammars Using the Inside–Outside Algorithm". In: *Computer Speech & Language* 4.1 (1990), pp. 35–56.

[193] Geoffrey E. Hinton et al. "The Wake–Sleep Algorithm for Unsupervised Neural Networks". In: *Science* 268.5214 (1995), pp. 1158–1161.

[194] Hagai Attias. "Planning by Probabilistic Inference". In: *Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics (AISTATS)*. 2003.

[195] Marc Toussaint. "Robot Trajectory Optimization Using Approximate Inference". In: *Proceedings of the 26th International Conference on Machine Learning (ICML)*. 2009.

[196] H. J. Kappen. "Linear Theory for Control of Nonlinear Stochastic Systems". In: *Physical Review Letters* 95.20 (2005), p. 200201.

[197] Will Grathwohl et al. "Your Classifier is Secretly an Energy Based Model". In: *Proceedings of the 8th International Conference on Learning Representations (ICLR)*. 2020.

[198] Kevin Ellis et al. "DreamCoder: Bootstrapping Inductive Program Synthesis with Wake–Sleep Library Learning". In: *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation (PLDI)*. 2021.

[199] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. "Human-Level Concept Learning Through Probabilistic Program Induction". In: *Science* 350.6266 (2015), pp. 1332–1338.

[200] Michael Janner et al. "Planning with Diffusion for Flexible Behavior Synthesis". In: *Proceedings of the 39th International Conference on Machine Learning (ICML)*. 2022.

[201] Anurag Ajay et al. "Is Conditional Generative Modeling all you Need for Decision Making?" In: *Proceedings of the 11th International Conference on Learning Representations (ICLR)*. 2023.

[202] Alexander Spangher et al. "Identifying Informational Sources in News Articles". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 3626–3639. DOI: 10.18653/v1/2023.emnlp-main.221. URL: https://aclanthology.org/2023.emnlp-main.221.

[203] Lydia M Hopper, Vania de la Luz, and Andrew Whiten. "'Ghost' experiments and the dissection of social learning in humans and animals". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 365.1551 (2010), pp. 3635–3640. DOI: 10.1098/rstb.2010.0140.

[204] Sarah Cohen, James T. Hamilton, and Fred Turner. "Computational journalism". In: *Communications of the ACM* 54 (2011), pp. 66–71. URL: https://api.semanticscholar.org/CorpusID:30295912.

[205] Francesco Marconi and Alex Siegman. *A day in the life of a journalist in 2027: Reporting meets AI*. Apr. 11, 2017. URL: https://www.cjr.org/innovations/artificial-intelligence-journalism.php (visited on 08/29/2025).

[206] J Gatlung and Mari Holmboe Ruge. "The structure of foreign news". In: *Journal of Peace Research* 2.1 (1965), pp. 64–91.

[207] Horst Po
ttker. "News and its communicative quality: the inverted pyramid—when and why did it appear?" In: *Journalism Studies* 4.4 (2003), pp. 501–511.

[208] Phyllis Kaniss. *Making local news*. University of Chicago Press, 1991.

[209] David L Hamilton and Roger D Fallot. "Information salience as a weighting factor in impression formation." In: *Journal of Personality and Social Psychology* 30.4 (1974), p. 444.

[210] Sarah Cohen, James T Hamilton, and Fred Turner. "Computational journalism". In: *Communications of the ACM* 54.10 (2011), pp. 66–71.

[211] Alexander Spangher et al. "Tracking the Newsworthiness of Public Documents". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 14150–14168. DOI: 10.18653/v1/2024.acl-long.763. URL: https://aclanthology.org/2024.acl-long.763/.

[212] Alexander Spangher et al. "NewsHomepages: Homepage Layouts Capture Information Prioritization Decisions". In: *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2025. URL: https://arxiv.org/abs/2501.00004.

[213] Jonathan Spencer et al. "Feedback in imitation learning: The three regimes of covariate shift". In: *arXiv preprint arXiv:2102.02872* (2021).

[214] Pim De Haan, Dinesh Jayaraman, and Sergey Levine. "Causal confusion in imitation learning". In: *Advances in neural information processing systems* 32 (2019).

[215] Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. "Data quality in imitation learning". In: *Advances in neural information processing systems* 36 (2023), pp. 80375–80395.

[216] Naoki Shibata, Yuya Kajikawa, and Ichiro Sakata. "Link prediction in citation networks". In: *Journal of the American society for information science and technology* 63.1 (2012), pp. 78–85.

[217] Amit Bagga and Breck Baldwin. "Cross-document event coreference: Annotations, experiments, and observations". In: *Coreference and Its Applications*. 1999.

[218] Lisa Getoor et al. "Learning probabilistic models of link structure". In: *Journal of Machine Learning Research* 3.Dec (2002), pp. 679–707.

[219] Jason Wei et al. "Chain of thought prompting elicits reasoning in large language models". In: *arXiv preprint arXiv:2201.11903* 35 (2022), pp. 24824–24837.

[220] Juan Ramos et al. "Using tf-idf to determine word relevance in document queries". In: *Proceedings of the first instructional conference on machine learning*. Vol. 242. 1. Citeseer. 2003, pp. 29–48.

[221] Nils Reimers and Iryna Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks". In: *arXiv preprint arXiv:1908.10084* (2019), pp. 3982–3992.

[222] OpenAI. *Introducing text and code embeddings*. https://openai.com/index/introducing-text-and-code-embeddings/. Accessed: 2025-08-19. Jan. 2022.

[223] Vladimir Karpukhin et al. "Dense passage retrieval for open-domain question answering". In: *arXiv preprint arXiv:2004.04906* (2020).

[224] Stephen Robertson, Hugo Zaragoza, et al. "The probabilistic relevance framework: BM25 and beyond". In: *Foundations and Trends® in Information Retrieval* 3.4 (2009), pp. 333–389.

[225] Nicholas Diakopoulos, Mor Naaman, and Funda Kivran-Swaine. "Diamonds in the rough: Social media visual analytics for journalistic inquiry". In: *2010 IEEE Symposium on Visual Analytics Science and Technology*. IEEE. 2010, pp. 115–122.

[226] Jian Zhao et al. "# FluxFlow: Visual analysis of anomalous information spreading on social media". In: *IEEE transactions on visualization and computer graphics* 20.12 (2014), pp. 1773–1782.

[227] Max Bain et al. "WhisperX: Time-Accurate Speech Transcription of Long-Form Audio". In: *INTERSPEECH 2023* (2023).

[228] Rodrigo Uribe and Barrie Gunter. "AreSensational'News Stories More Likely to Trigger Viewers' Emotions than Non-Sensational News Stories? A Content Analysis of British TV News". In: *European journal of communication* 22.2 (2007), pp. 207–228.

[229] Pang Wei Koh et al. "Concept bottleneck models". In: *International conference on machine learning*. PMLR. 2020, pp. 5338–5348.

[230] Kevin G. Barnhurst and John Nerone. *The Form of News: A History*. Guilford Press, 2001.

[231] Nikki Usher. *Making news at the New York times*. University of Michigan Press, 2014.

[232] Alexander Spangher, Nanyun Peng, and Emilio Ferrara. "Modeling "Newsworthiness" for Lead-Generation Across Corpora". In: *cj2020* (2020).

[233] Stephanie Hays. "An Analysis of Design Components of Award-winning Newspaper Pages". In: *Elon Journal of Undergraduate Research in Communications* 9.2 (2018), pp. 44–63.

[234] Margaret Sullivan. "The End of the Page One Meeting: Making Way for the Reader in Choosing the News". In: *The New York Times* (Mar. 2016). `https://publiceditor.blogs.nytimes.com/2016/03/16/the-end-of-the-page-one-meeting-making-way-for-the-reader-in-choosing-the-news/`.

[235] Jakob Nielsen. *F-Shaped Pattern For Reading Web Content*. `https://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/`. Accessed: 2023-10-06. 2006.

[236] Hans-Jürgen Bucher and Peter Schumacher. "The relevance of attention for selecting news content. An eye-tracking study on attention patterns in the reception of print and online media". In: (2006).

[237] Mario R. García. *Contemporary Newspaper Design: Shaping the News in the Digital Age*. Prentice Hall, 1987.

[238] Dolf Zillmann, Silvia Knobloch, and Zhao Yu. "Effects of photographs on the selective reading of news reports". In: *Media Psychology* 3.4 (2001), pp. 301–324.

[239] George A Miller. "The magical number seven, plus or minus two: Some limits on our capacity for processing information." In: *Psychological review* 63.2 (1956), p. 81.

[240] Mark Boukes, Natalie P Jones, and Rens Vliegenthart. "Newsworthiness and story prominence: How the presence of news factors relates to upfront position and length of news stories". In: *Journalism* 23.1 (2022), pp. 98–116.

[241] Ralph Allan Bradley and Milton E Terry. "Rank analysis of incomplete block designs: I. the method of paired comparisons". In: *Biometrika* 39.3/4 (1952), pp. 324–345.

[242] Louis L Thurstone. "A law of comparative judgment". In: *Scaling*. Routledge, 2017, pp. 81–92.

[243] LL Thurstone. "A law of comparative judgment". In: *Psychological Review* 34.4 (1927).

[244] Guy Bergstrom. *Understanding the Newspaper News Cycle*. Accessed: 2025-05-19. 2019. URL: https://www.liveabout.com/understanding-the-news-cycle-2295933.

[245] Zejiang Shen et al. "LayoutParser: A Unified Toolkit for Deep Learning Based Document Image Analysis". In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. ACM. 2021, pp. 4528–4538.

[246] Minghao Li et al. "DocBank: A Benchmark Dataset for Document Layout Analysis". In: *Proceedings of the 28th International Conference on Computational Linguistics*. 2020, pp. 949–960.

[247] Xiaojie Zhong, Jianbin Tang, and Antonio Jimeno Yepes. "PubLayNet: largest dataset ever for document layout analysis". In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE. 2019, pp. 1015–1022.

[248] Massih-Reza Amini et al. "Self-training: A survey". In: *arXiv preprint arXiv:2202.12040* (2022).

[249] Joseph Redmon and Ali Farhadi. "YOLOv3: An Incremental Improvement". In: *arXiv* (2018).

[250] Yuxin Wu et al. *Detectron2*. https://github.com/facebookresearch/detectron2. 2019.

[251] Edward R. Tufte. *Envisioning Information*. Graphics Press, 1990.

[252] Matthew J. Salganik, Peter S. Dodds, and Duncan J. Watts. "Experimental study of inequality and unpredictability in an artificial cultural market". In: *Science* 311.5762 (2006), pp. 854–856.

[253] Christin Angèle. "Metrics at work: Journalism and the contested meaning of algorithms". In: (2020).

[254] Robin L Plackett. "The analysis of permutations". In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 24.2 (1975), pp. 193–202.

[255] Gerard Debreu. *Individual choice behavior: A theoretical analysis*. 1960.

[256] Alan Lambert and Julie Brock. "Layout complexity and visitors' attention on web pages: An eye-tracking study". In: *Journal of Digital Information* 6.2 (2005).

[257] Jakob Nielsen and Kara Pernice. *Eyetracking Web Usability*. New Riders, 2009.

[258] Sourab Mangrulkar et al. *PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods*. `https://github.com/huggingface/peft`. 2022.

[259] Matthew Gentzkow and Jesse M Shapiro. "What drives media slant? Evidence from US daily newspapers". In: *Econometrica* 78.1 (2010), pp. 35–71.

[260] Sonal Sannigrahi, Josef Van Genabith, and Cristina España-Bonet. "Are the best multilingual document embeddings simply based on sentence embeddings?" In: *arXiv preprint arXiv:2304.14796* (2023).

[261] Zhilin Yang et al. "HotpotQA: A dataset for diverse, explainable multi-hop question answering". In: *arXiv preprint arXiv:1809.09600* (2018).

[262] Devendra Singh et al. "End-to-end training of multi-document reader and retriever for open-domain question answering". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 25968–25981.

[263] CDP Institute. *Knowledge Workers Lose 30% of Time Looking for Data: Forrester Study*. `https://www.cdpinstitute.org/news/knowledge-workers-lose-30-of-time-looking-for-data-forrester-study/`. Accessed: 2025-08-20. Jan. 2023.

[264] Signe Ivask, Heleri All, and Kairi Janson. "Time-efficient and time-consuming practices among journalists in communicating with the sources". In: *Catalan Journal of Communication & Cultural Studies* 9.1 (2017), pp. 25–41.

[265] Zhang-Wei Hong et al. "Curiosity-driven Red-teaming for Large Language Models". In: *The Twelfth International Conference on Learning Representations*.

[266] Sahithya Ravi et al. "Small But Funny: A Feedback-Driven Approach to Humor Distillation". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2024, pp. 13078–13090.

[267] Guangxuan Xu et al. "EnDex: Evaluation of Dialogue Engagingness at Scale". In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. 2022, pp. 4884–4893.

[268] Chris Buckley and Ellen M Voorhees. "Retrieval evaluation with incomplete information". In: *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. 2004, pp. 25–32.

[269] Doina Precup, Richard S. Sutton, and Satinder Singh. "Eligibility Traces for Off-Policy Policy Evaluation". In: *International Conference on Machine Learning*. 2000. URL: https://api.semanticscholar.org/CorpusID:1153355.

[270] Philip Thomas and Emma Brunskill. "Data-Efficient Off-Policy Policy Evaluation for Reinforcement Learning". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 2139–2148. URL: https://proceedings.mlr.press/v48/thomasa16.html.

[271] Scott Fujimoto, David Meger, and Doina Precup. "Off-policy deep reinforcement learning without exploration". In: *International conference on machine learning*. PMLR. 2019, pp. 2052–2062.

[272] Aviral Kumar et al. "Conservative q-learning for offline reinforcement learning". In: *Advances in neural information processing systems* 33 (2020), pp. 1179–1191.

[273] Alexander Spangher et al. "Do llms plan like human writers? comparing journalist coverage of press releases with llms". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 2024, pp. 21814–21828.

[274] Alexander Spangher et al. "Explaining Mixtures of Sources in News Articles". In: (2024).

[275] Alexander Spangher et al. "NewsInterview: a Dataset and a Playground to Evaluate LLMs' Grounding Gap via Informational Interviews". In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2025, pp. 32895–32925.

[276] Sebastian Padó et al. "Who sides with whom? towards computational construction of discourse networks for political debates". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 2841–2847.

[277] Timote Vaucher et al. “Quotebank: a corpus of quotations from a decade of news”. In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 2021, pp. 328–336.

[278] Momchil Hardalov et al. “A survey on stance detection for mis-and disinformation identification”. In: *arXiv preprint arXiv:2103.00242* (2022). URL: https://aclanthology.org/2022.findings-naacl.94.

[279] Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. “Neural end-to-end learning for computational argumentation mining”. In: *arXiv preprint arXiv:1704.06104* (2017).

[280] Jeroen Peperkamp and Bettina Berendt. “Diversity Checker: Toward recommendations for improving journalism with respect to diversity”. In: *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*. 2018, pp. 35–41.

[281] Dario Pavllo, Tiziano Piccardi, and Robert West. “Quootstrap: Scalable unsupervised extraction of quotation-speaker pairs from large news corpora via bootstrapping”. In: *Twelfth International AAAI Conference on Web and Social Media*. 2018.

[282] Edward Newell, Drew Margolin, and Derek Ruths. “An attribution relations corpus for political news”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.

[283] Joakim Nivre. “Dependency parsing”. In: *Language and Linguistics Compass* 4.3 (2010), pp. 138–152.

[284] Manzil Zaheer et al. “Big bird: Transformers for longer sequences”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 17283–17297.

[285] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018), pp. 4171–4186.

[286] Yuval Kirstain, Ori Ram, and Omer Levy. “Coreference Resolution without Span Representations”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 2021, pp. 14–19.

[287] Shon Otmazgin, Arie Cattan, and Yoav Goldberg. “LingMess: Linguistically Informed Multi Expert Scorers for Coreference Resolution”. In: *arXiv preprint arXiv:2205.12644* (2022).

[288] Jesse Mu, Xiang Lisa Li, and Noah Goodman. “Learning to Compress Prompts with Gist Tokens”. In: *arXiv preprint arXiv:2304.08467* (2023).

[289] Alexander Spangher et al. "NewsEdits: A News Article Revision Dataset and a Novel Document-Level Reasoning Challenge". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2022, pp. 127–157.

[290] James T Hamilton. "All the news that's fit to sell". In: *All the News That's Fit to Sell*. Princeton University Press, 2011.

[291] Evan Sandhaus. "The new york times annotated corpus". In: *Linguistic Data Consortium, Philadelphia* 6.12 (2008), e26752.

[292] Armand Joulin et al. "Bag of Tricks for Efficient Text Classification". In: *arXiv preprint arXiv:1607.01759* (2016).

[293] Daniel Golovin and Andreas Krause. "Adaptive submodularity: Theory and applications in active learning and stochastic optimization". In: *Journal of Artificial Intelligence Research* 42 (2011), pp. 427–486.

[294] Andreas Krause and Daniel Golovin. "Submodular function maximization." In: *Tractability* 3.71-104 (2014), p. 3.

[295] Alex Kulesza, Ben Taskar, et al. "Determinantal point processes for machine learning". In: *Foundations and Trends® in Machine Learning* 5.2–3 (2012), pp. 123–286.

[296] Richard S Sutton, Doina Precup, and Satinder Singh. "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning". In: *Artificial intelligence* 112.1-2 (1999), pp. 181–211.

[297] Sergey Levine et al. "Offline reinforcement learning: Tutorial, review, and perspectives on open problems". In: *arXiv preprint arXiv:2005.01643* (2020).

[298] Nathan Kallus and Angela Zhou. "Confounding-robust policy evaluation in infinite-horizon reinforcement learning". In: *Advances in neural information processing systems* 33 (2020), pp. 22293–22304.

[299] Alizée Pace et al. "Delphic offline reinforcement learning under nonidentifiable hidden confounding". In: *arXiv preprint arXiv:2306.01157* (2023).

[300] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. "A reduction of imitation learning and structured prediction to no-regret online learning". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 627–635.

[301] Chandra Bhagavatula et al. "Content-Based Citation Recommendation". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). 2018, pp. 238–251.

[302] Chanwoo Jeong et al. "A context-aware citation recommendation model with BERT and graph convolutional networks". In: *Scientometrics* 124.3 (2020), pp. 1907–1922.

[303] Kehan Long et al. "Recommending Missed Citations Identified by Reviewers: A New Task, Dataset and Baselines". In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 2024, pp. 13699–13711.

[304] Savvas Petridis et al. "Anglekindling: Supporting journalistic angle ideation with large language models". In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 2023, pp. 1–16.

[305] Henk Pander Maat and Caro de Jong. "How newspaper journalists reframe product press release information". In: *Journalism* 14.3 (2013), pp. 348–371.

[306] Edward Spence and Peter Simmons. "The practice and ethics of media release journalism". In: *Australian Journalism Review* 28.1 (2006), pp. 167–181.

[307] Ben Welsh. *Story Sniffer*. Tech. rep. The Reynolds Journalism Institute, University of Missouri, 2022. URL: https://palewi.re/docs/storysniffer/.

[308] Greg R Notess. "The Wayback Machine: The Web's Archive." In: *Online* 26.2 (2002), pp. 59–61.

[309] Ido Dagan, Oren Glickman, and Bernardo Magnini. "The Pascal recognising textual entailment challenge". In: *Machine Learning Challenges Workshop*. Springer. 2005, pp. 177–190.

[310] Philippe Laban et al. "SummaC: Re-visiting NLI-based models for inconsistency detection in summarization". In: *Transactions of the Association for Computational Linguistics* 10 (2022), pp. 163–177.

[311] Shon Otmazgin, Arie Cattan, and Yoav Goldberg. "F-coref: Fast, Accurate and Easy to Use Coreference Resolution". In: *Asia-Pacific Chapter of the Association for Computational Linguistics (AACL)*. 2022.

[312] Yixin Nie et al. "Adversarial NLI: A New Benchmark for Natural Language Understanding". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 4885–4901.

[313] Erik Arakelyan, Zhaoqi Liu, and Isabelle Augenstein. "Semantic Sensitivities and Inconsistent Predictions: Measuring the Fragility of NLI Models". In: *Proceedings*

of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers). 2024, pp. 432–444.

[314] Alexander Spangher et al. "Explaining Mixtures of Sources in News Articles". In: *Conference on Empirical Methods in Natural Language Processing*. 2024.

[315] Brant Houston and Mark Horvit. *Investigative Reporters Handbook*. Bedford/Saint Martin's, 2020.

[316] Elena Bruni and Anna Comacchio. "Configuring a new business model through conceptual combination: The rise of the Huffington Post". In: *Long Range Planning* 56.1 (2023), p. 102249.

[317] Marcel Machill, Markus Beiler, and Iris Hellmann. "The selection process in local court reporting: A case study of four Dresden daily newspapers". In: *Journalism Practice* 1.1 (2007), pp. 62–81.

[318] Yufei Tian et al. "Are Large Language Models Capable of Generating Human-Level Narratives?" In: *2024 Conference on Empirical Methods in Natural Language Processing*. 2024.

[319] Albert Q. Jiang et al. "Mixtral of Experts". In: *arXiv* abs/2401.04088 (2024).

[320] Aidan Gomez. "Command R: Retrieval-Augmented Generation at Production Scale". In: (2024). URL: https://txt.cohere.com/command-r/.

[321] Mats Nylund. "Toward creativity management: Idea generation and newsroom meetings". In: *International Journal on Media Management* 15.4 (2013), pp. 197–210.

[322] Nasrin Mostafazadeh et al. "A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kevin Knight, Ani Nenkova, and Owen Rambow. San Diego, California: Association for Computational Linguistics, June 2016, pp. 839–849. DOI: 10.18653/v1/N16-1098. URL: https://aclanthology.org/N16-1098.

[323] Yufei Tian et al. "Unsupervised Melody-to-Lyrics Generation". In: *arXiv* abs/2305.19228 (2023).

[324] David M Ryfe. "How journalists internalize news practices and why it matters". In: *Journalism* 24.5 (2023), pp. 921–937.

[325] Fabrice Harel-Canada et al. "Measuring Psychological Depth in Language Models". In: *2024 Conference on Empirical Methods in Natural Language Processing*.

[326] Yufei Tian et al. "MacGyver: Are Large Language Models Creative Problem Solvers?" In: *North American Chapter of the Association for Computational Linguistics*. 2023.

[327] Ken Gilhooly. "AI vs humans in the AUT: Simulations to LLMs". In: *Journal of Creativity* (2023), p. 100071.

[328] Yunpu Zhao et al. "Assessing and Understanding Creativity in Large Language Models". In: *arXiv* abs/2401.12491 (2024).

[329] Lauren Rogal. "Secrets, Lies, and Lessons from the Theranos Scandal". In: *Hastings LJ* 72 (2020), p. 1663.

[330] Patrick Lewis et al. "Retrieval-augmented generation for knowledge-intensive NLP tasks". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9459–9474.

[331] Timo Schick et al. "Toolformer: Language Models Can Teach Themselves to Use Tools". In: *arXiv* abs/2302.04761 (2023).

[332] Chau Pham et al. "TopicGPT: A Prompt-based Topic Modeling Framework". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 2024, pp. 2956–2984.

[333] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2019. URL: https://arxiv.org/abs/1908.10084.

[334] Jaime Carbonell and Jade Goldstein. "The use of MMR, diversity-based reranking for reordering documents and producing summaries". In: *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM. 1998, pp. 335–336.

[335] James Allan. "Topic detection and tracking: event-based information organization". In: *Topic Detection and Tracking*. Springer, 2003, pp. 1–16.

[336] Charles LA Clarke et al. "Novelty and diversity in information retrieval evaluation". In: *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM. 2008, pp. 659–666.

[337] Herbert J Gans. *Deciding What's News: A Study of CBS Evening News, NBC Nightly News, Newsweek, and Time*. Northwestern University Press, 1979.

[338] Junxia Ma et al. “Chain of Stance: Stance Detection with Large Language Models”. In: *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer. 2024, pp. 82–94.

[339] Alexander Spangher et al. “A Novel Multi-Document Retrieval Benchmark: Journalist Source-Selection in Newswriting”. In: *Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing*. 2025, pp. 180–204.

[340] Gaye Tuchman. “Making news: A study in the construction of reality”. In: *Free Pres* (1978).

[341] Harsh Trivedi et al. “Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2023, pp. 10014–10037.

[342] Rodrigo Nogueira and Kyunghyun Cho. “Passage Re-ranking with BERT”. In: *arXiv preprint arXiv:1901.04085*. 2019.

[343] Gail Sedorkin. *Interviewing: A Guide for Journalists and Writers*. 4th. Allen & Unwin, 2015.

[344] Stephen E Robertson and Steve Walker. “Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval”. In: *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*. Springer. 1994, pp. 232–241.

[345] Victor Sanh et al. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: *arXiv preprint arXiv:1910.01108* (2019).

[346] Tenghao Huang, Dongwon Jung, and Muhao Chen. “Planning and Editing What You Retrieve for Enhanced Tool Learning”. In: *ArXiv* abs/2404.00450 (2024). URL: https://api.semanticscholar.org/CorpusID:268819329.

[347] Nick Craswell et al. “Overview of the TREC 2019 Deep Learning Track”. In: *Proceedings of the Twenty-Eighth Text REtrieval Conference (TREC 2019)*. NIST. 2020.

[348] Omar Khattab and Matei Zaharia. “Colbert: Efficient and effective passage search via contextualized late interaction over bert”. In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 2020, pp. 39–48.

[349] Kartik A Santhanam et al. “ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction”. In: *Proceedings of the 45th International ACM SI-*

GIR Conference on Research and Development in Information Retrieval. ACM. 2022, pp. 337–347.

[350] Zihao Zhao et al. "Calibrate before use: Improving few-shot performance of language models". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 12697–12706.

[351] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. "Learning To Retrieve Prompts for In-Context Learning". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2022, pp. 2655–2671.

[352] Shunyu Yao et al. "ReAct: Synergizing Reasoning and Acting in Language Models". In: *arXiv preprint arXiv:2210.03629* (2022).

[353] Eric Zelikman et al. "Star: Bootstrapping reasoning with reasoning". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 15476–15488.

[354] Frederic Charles Bartlett. *Remembering: A study in experimental and social psychology*. Cambridge university press, 1932.

[355] David E Rumelhart. "Schemata: The building blocks of cognition". In: *Theoretical issues in reading comprehension*. Routledge, 1980, pp. 33–58.

[356] Roger C Schank and Robert P Abelson. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Psychology press, 1977.

[357] Marvin Minsky. "A framework for representing knowledge". In: (1974).

[358] Vanessa E Ghosh and Asaf Gilboa. "What is a memory schema? A historical perspective on current neuroscience literature". In: *Neuropsychologia* 53 (2014), pp. 104–114.

[359] Marlieke TR Van Kesteren et al. "How schema and novelty augment memory formation". In: *Trends in neurosciences* 35.4 (2012), pp. 211–219.

[360] Walter Kintsch and Teun A Van Dijk. "Toward a model of text comprehension and production." In: *Psychological review* 85.5 (1978), p. 363.

[361] Jean M Mandler and Nancy S Johnson. "Remembrance of things parsed: Story structure and recall". In: *Cognitive psychology* 9.1 (1977), pp. 111–151.

[362] Vladimir Propp. *Morphology of the Folktale*. University of Texas press, 1968.

[363] Barbara J Grosz and Candace L Sidner. "Attention, intentions, and the structure of discourse". In: *Computational linguistics* 12.3 (1986), pp. 175–204.

[364] William C Mann and Sandra A Thompson. "Rhetorical structure theory: Toward a functional theory of text organization". In: *Text-interdisciplinary Journal for the Study of Discourse* 8.3 (1988), pp. 243–281.

[365] Gabriel A Radvansky and Jeffrey M Zacks. "Event perception". In: *Wiley Interdisciplinary Reviews: Cognitive Science* 2.6 (2011), pp. 608–620.

[366] Matthew M Botvinick, Yael Niv, and Andew G Barto. "Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective". In: *cognition* 113.3 (2009), pp. 262–280.

[367] Momchil Hardalov et al. "Cross-Domain Label-Adaptive Stance Detection". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 9011–9028.

[368] Edoardo M Airoldi and Jonathan M Bischof. "Improving and evaluating topic models and other models of text". In: *Journal of the American Statistical Association* 111.516 (2016), pp. 1381–1403.

[369] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[370] Todd K Moon. "The expectation-maximization algorithm". In: *IEEE Signal processing magazine* 13.6 (1996), pp. 47–60.

[371] Jonathan Chang et al. "Reading tea leaves: How humans interpret topic models". In: *Advances in neural information processing systems* 22 (2009).

[372] David Bamman, Brendan O'Connor, and Noah A Smith. "Learning latent personas of film characters". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2013, pp. 352–361.

[373] David Bamman and Noah A Smith. "Unsupervised discovery of biographical structure from text". In: *Transactions of the Association for Computational Linguistics* 2 (2014), pp. 363–376.

[374] Hanna M Wallach et al. "Evaluation methods for topic models". In: *Proceedings of the 26th annual international conference on machine learning*. 2009, pp. 1105–1112.

[375] GuoDong Zhou and KimTeng Lua. "Word Association and MI-Trigger-based Language Modeling". In: *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*.

Montreal, Quebec, Canada: Association for Computational Linguistics, Aug. 1998, pp. 1465–1471. DOI: 10.3115/980691.980808. URL: https://aclanthology.org/P98-2239.

[376] Yuntian Deng, Volodymyr Kuleshov, and Alexander M Rush. "Model Criticism for Long-Form Text Generation". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 2022, pp. 11887–11912.

[377] Khalid Al Khatib et al. "A news editorial corpus for mining argumentation strategies". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 2016, pp. 3433–3443.

[378] Margaret Sullivan. "Tightening the screws on anonymous sources". In: *New York Times* (2016).

[379] Kate C McLean et al. "Narrative Identity in the Social World: The Press for Stability". In: *Handbook of Personality Psychology* (2019).

[380] J Richard Landis and Gary G Koch. "The measurement of observer agreement for categorical data". In: *biometrics* (1977), pp. 159–174.

[381] Adina Williams, Tristan Thrush, and Douwe Kiela. "ANLIzing the Adversarial Natural Language Inference Dataset". In: (2022).

[382] Dean Pomerleau and Delip Rao. "Fake news challenge stage 1 (FNC-I): Stance detection". In: *Retrieved March* 15 (2017), p. 2023.

[383] Sihao Chen et al. "Seeing Things from a Different Angle: Discovering Diverse Perspectives about Claims". In: *Proceedings of NAACL-HLT*. 2019, pp. 542–557.

[384] Ivan Habernal et al. "The argument reasoning comprehension task: Identification and reconstruction of implicit warrants". In: *arXiv preprint arXiv:1708.01425* (2017).

[385] William Ferreira and Andreas Vlachos. "Emergent: a novel data-set for stance classification". In: *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*. ACL. 2016.

[386] Revanth Gangi Reddy et al. "NewsClaims: A New Benchmark for Claim Detection from News with Background Knowledge". In: *arXiv preprint arXiv:2112.08544* (2021).

[387] Xinyi Wang et al. "Large Language Models Are Latent Variable Models: Explaining and Finding Good Demonstrations for In-Context Learning". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[388] Guillaume Sanchez et al. "Stay on Topic with Classifier-Free Guidance". In: *International Conference on Machine Learning*. 2024. URL: https://api.semanticscholar.org/CorpusID:272330615.

[389] Clara Meister and Ryan Cotterell. "Language model evaluation beyond perplexity". In: *arXiv preprint arXiv:2106.00085* (2021).

[390] Byung-Doh Oh, Christian Clark, and William Schuler. "Comparison of structural parsers and neural language models as surprisal estimators". In: *Frontiers in Artificial Intelligence* 5 (2022), p. 777963.

[391] Kun Lu et al. "Vocabulary size and its effect on topic representation". In: *Information Processing & Management* 53.3 (2017), pp. 653–665.

[392] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. "The global k-means clustering algorithm". In: *Pattern recognition* 36.2 (2003), pp. 451–461.

[393] Wichayaporn Wongkamjan et al. "More Victories, Less Cooperation: Assessing Cicero's Diplomacy Play". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 12423–12441. DOI: 10.18653/v1/2024.acl-long.672. URL: https://aclanthology.org/2024.acl-long.672/.

[394] Omar Shaikh et al. "Grounding Gaps in Language Model Generations". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 2024, pp. 6279–6296.

[395] Herbert H Clark. *Using language*. Cambridge university press, 1996.

[396] Hyundong Cho and Jonathan May. "Grounding Conversations with Improvised Dialogues". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 2398–2413.

[397] Enkelejda Kasneci et al. "ChatGPT for good? On opportunities and challenges of large language models for education". In: *Learning and individual differences* 103 (2023), p. 102274.

[398] Per Carlbring et al. "A new era in Internet interventions: The advent of Chat-GPT and AI-assisted therapist guidance". In: *Internet Interventions* 32 (2023).

[399] Lisa P Argyle et al. "Ai chat assistants can improve conversations about divisive topics". In: *arXiv preprint arXiv:2302.07268* (2023).

[400] Hannah Rashkin et al. "Towards Empathetic Open-domain Conversation Models: A New Benchmark and Dataset". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 5370–5381. DOI: 10.18653/v1/P19-1534. URL: https://aclanthology.org/P19-1534.

[401] Xuewei Wang et al. "Persuasion for Good: Towards a Personalized Persuasive Dialogue System for Social Good". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 5635–5649. DOI: 10.18653/v1/P19-1566. URL: https://aclanthology.org/P19-1566/.

[402] Siyang Liu et al. "Towards Emotional Support Dialog Systems". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2021, pp. 3469–3483.

[403] Andrew Caines et al. "The Teacher-Student Chatroom Corpus". In: *Proceedings of the 9th Workshop on NLP for Computer Assisted Language Learning*. Ed. by David Alfter et al. Gothenburg, Sweden: LiU Electronic Press, Nov. 2020, pp. 10–20. URL: https://aclanthology.org/2020.nlp4call-1.2/.

[404] Jonathan Gratch et al. "The Distress Analysis Interview Corpus of human and computer interviews". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Ed. by Nicoletta Calzolari et al. Reykjavik, Iceland: European Language Resources Association (ELRA), May 2014, pp. 3123–3128. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/508_Paper.pdf.

[405] Dympna Casey. "Challenges of collecting data in the clinical setting". In: *NT Research* 9.2 (2004), pp. 131–141.

[406] Tony Harcup. *Journalism: Principles and Practice*. 3rd. London, UK: SAGE Publications, 2015.

[407] Bodhisattwa Prasad Majumder et al. "Interview: A large-scale open-source corpus of media dialog". In: *arXiv preprint arXiv:2004.03090* (2020).

[408] Chenguang Zhu et al. "MediaSum: A Large-scale Media Interview Dataset for Dialogue Summarization". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kristina Toutanova et al. Online: Association for Computational Linguistics, June 2021, pp. 5927–5934. DOI: 10.18653/v1/2021.naacl-main.474. URL: https://aclanthology.org/2021.naacl-main.474/.

[409] Hugo Touvron et al. "Llama 2: Open foundation and fine-tuned chat models". In: *arXiv preprint arXiv:2307.09288* (2023). arXiv: 2302.13971 [cs.CL].

[410] Woosuk Kwon et al. "Efficient memory management for large language model serving with pagedattention". In: *Proceedings of the 29th Symposium on Operating Systems Principles*. 2023, pp. 611–626.

[411] Rudolf Flesch. "Flesch-Kincaid readability test". In: *Retrieved October* 26.3 (2007), p. 2007.

[412] Omar Shaikh et al. "Grounding Gaps in Language Model Generations". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 2024, pp. 6279–6296.

[413] Swarnadeep Saha et al. "Branch-Solve-Merge Improves Large Language Model Evaluation and Generation". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 2024, pp. 8345–8363.

[414] Jiao Ou et al. "DialogBench: Evaluating LLMs as Human-like Dialogue Systems". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 2024, pp. 6137–6170.

[415] Robert B Cialdini. *Influence: Science and practice*. Vol. 4. 2009.

[416] Hiromasa Sakurai and Yusuke Miyao. "Evaluating Intention Detection Capability of Large Language Models in Persuasive Dialogues". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 1635–1657. DOI: 10.18653/v1/2024.acl-long.90. URL: https://aclanthology.org/2024.acl-long.90.

[417] Yi Zeng et al. "How Johnny Can Persuade LLMs to Jailbreak Them: Rethinking Persuasion to Challenge AI Safety by Humanizing LLMs". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 14322–14350. DOI: 10.18653/v1/2024.acl-long.773. URL: https://aclanthology.org/2024.acl-long.773/.

[418] Jacob B Hirsh, Sonia K Kang, and Galen V Bodenhausen. "Personalized persuasion: Tailoring persuasive appeals to recipients' personality traits". In: *Psychological science* 23.6 (2012), pp. 578–581.

[419] Dallas Card et al. "With Little Power Comes Great Responsibility". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber et al. Online: Association for Computational Linguistics, Nov. 2020, pp. 9263–9274. DOI: 10.18653/v1/2020.emnlp-main.745. URL: https://aclanthology.org/2020.emnlp-main.745/.

[420] Patrick Lewis et al. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". In: *Advances in Neural Information Processing Systems*. 2020.

[421] *OpenAI Cookbook: Reranking with Cross-Encoders*. https://cookbook.openai.com/examples/search_reranking_with_cross-encoders. Accessed 2024-11-26.

[422] Jason Wei et al. "Chain-of-thought prompting elicits reasoning in large language models". In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NIPS '22. New Orleans, LA, USA: Curran Associates Inc., 2022. ISBN: 9781713871088.

[423] Nils Reimers and Iryna Gurevych. *Cross-Encoders for Ranking*. https://www.sbert.net/examples/applications/cross-encoder-ranking. Accessed 2024-11-26.

[424] Andrew Shinn et al. "Self-Reflective Agents Make Language Models Better Reasoners". In: *arXiv preprint arXiv:2310.06271* (2023).

[425] Minjun Chang et al. "Self-Reflection with Generative Agents". In: *arXiv preprint arXiv:2311.09214* (2023).

[426] Lei Huang et al. "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions". In: *ACM Trans. Inf. Syst.* 43.2 (Jan. 2025). ISSN: 1046-8188. DOI: 10.1145/3703155. URL: https://doi.org/10.1145/3703155.

[427] Ziyi Liu et al. "InterIntent: Investigating Social Intelligence of LLMs via Intention Understanding in an Interactive Game Context". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 6718–6746. DOI: 10.18653/v1/2024.emnlp-main.383. URL: https://aclanthology.org/2024.emnlp-main.383/.

[428] Kushal Chawla et al. "CaSiNo: A Corpus of Campsite Negotiation Dialogues for Automatic Negotiation Systems". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2021, pp. 3167–3185.

[429] Jiwei Li et al. "Deep Reinforcement Learning for Dialogue Generation". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2016, pp. 1192–1202.

[430] AC Bohart. "How clients make therapy work". In: *American Psychological Association* (1999).

[431] AL Brown. "Guided discovery in a community of learners". In: *Classroom lessons: Integrating cognitive theory and classroom practice/press/Bradford Books* (1994).

[432] Aske Plaat et al. "Reasoning with large language models, a survey". In: *CoRR* (2024).

[433] Fengli Xu et al. "Towards large reasoning models: A survey of reinforced reasoning with large language models". In: *arXiv preprint arXiv:2501.09686* (2025).

[434] Melanie Mitchell. "Artificial intelligence learns to reason". In: *Science* 387.6740 (2025), eadw5211. DOI: 10.1126/science.adw5211. eprint: https://www.science.org/doi/pdf/10.1126/science.adw5211. URL: https://www.science.org/doi/abs/10.1126/science.adw5211.

[435] Ling Yang et al. "ReasonFlux: Hierarchical LLM Reasoning via Scaling Thought Templates". In: *ArXiv* abs/2502.06772 (2025). URL: https://api.semanticscholar.org/CorpusID:276250066.

[436] Zeyu Dai, Himanshu Taneja, and Ruihong Huang. "Fine-grained structure-based news genre categorization". In: *Proceedings of the Workshop Events and Stories in the News 2018*. 2018, pp. 61–67.

[437] Christian Ameseder. "Effects of narrative journalism on interest and comprehension: an overview". In: *Journal of Education and Humanities (JEH)* 2.2 (2019), pp. 29–50.

[438] Walter Kintsch and Teun A Van Dijk. "Toward a model of text comprehension and production." In: *Psychological review* 85.5 (1978), p. 363.

[439] Jean M Mandler and Nancy S Johnson. "Remembrance of things parsed: Story structure and recall". In: *Cognitive psychology* 9.1 (1977), pp. 111–151.

[440] Tamara Van Gog. "The signaling (or cueing) principle in multimedia learning". In: *The Cambridge handbook of multimedia learning*. Cambridge University Press, 2014, pp. 221–230.

[441] Steven Feld. *A generative theory of tonal music*. 1984.

[442] Horst Po
ttker. "News and its communicative quality: the inverted pyramid—when and why did it appear?" In: *Journalism Studies* 4.4 (2003), pp. 501–511.

[443] Morton Ann Gernsbacher. *Language comprehension as structure building*. Psychology Press, 1990.

[444] William C Mann and Sandra A Thompson. "Rhetorical structure theory: Toward a functional theory of text organization". In: *Text-interdisciplinary Journal for the Study of Discourse* 8.3 (1988), pp. 243–281.

[445] Qianyu He et al. "Can large language models understand real-world complex instructions?" In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 16. 2024, pp. 18188–18196.

[446] Alexander Spangher et al. "Sequentially Controlled Text Generation". In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. 2022, pp. 6848–6866.

[447] Nellia Dzhubaeva, Katharina Trinley, and Laura Pissani. "Unstructured Minds, Predictable Machines: A Comparative Study of Narrative Cohesion in Human and LLM Stream-of-Consciousness Writing". In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*. 2025, pp. 1079–1096.

[448] Nelson F Liu et al. "Lost in the Middle: How Language Models Use Long Contexts". In: *Transactions of the Association for Computational Linguistics* 12 (2024), pp. 157–173.

[449] Yukyung Lee et al. "Navigating the Path of Writing: Outline-guided Text Generation with Large Language Models". In: *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*. 2025, pp. 233–250.

[450] Lili Yao et al. "Plan-and-write: Towards better automatic storytelling". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 7378–7385.

[451] Cheng-Zhi Anna Huang et al. "Music Transformer: Generating Music with Long-Term Structure". In: *International Conference on Learning Representations*.

[452] Prafulla Dhariwal et al. "Jukebox: A generative model for music". In: *arXiv preprint arXiv:2005.00341* (2020).

[453] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. "Geneval: An object-focused framework for evaluating text-to-image alignment". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 52132–52152.

[454] Yushi Hu et al. "Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 20406–20417.

[455] Katharina Emde, Christoph Klimmt, and Daniela M Schluetz. "Does storytelling help adolescents to process the news? A comparison of narrative news and the inverted pyramid". In: *Journalism studies* 17.5 (2016), pp. 608–627.

[456] Miglena M Sternadori and Kevin Wise. "Men and women read news differently". In: *Journal of media psychology* (2010).

[457] Ruqian Lu et al. "Attributed Rhetorical Structure Grammar for Domain Text Summarization". In: *arXiv preprint arXiv:1909.00923* (2019).

[458] Xinyi Zhou et al. "Fake News Early Detection: A Theory-Driven Model". In: *Digital Threats: Research and Practice* 1.2 (2020), pp. 1–25.

[459] Tom Brown et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[460] Iz Beltagy, Matthew E Peters, and Arman Cohan. "Longformer: The long-document transformer". In: *arXiv preprint arXiv:2004.05150* (2020).

[461] Hannah Rashkin et al. "PlotMachines: Outline-Conditioned Generation with Dynamic Plot State Tracking". In: *arXiv* abs/2004.14967 (2020).

[462] Xiangyu Peng et al. "Guiding Neural Story Generation with Reader Models". In: *arXiv preprint arXiv:2112.08596* (2021).

[463] Prafulla Kumar Choubey et al. "Discourse as a Function of Event: Profiling Discourse Structure in News Articles around the Main Event". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 5374–5386. DOI: 10.18653/v1/2020.acl-main.478. URL: https://www.aclweb.org/anthology/2020.acl-main.478.

[464] Nitish Shirish Keskar et al. "Ctrl: A conditional transformer language model for controllable generation". In: *arXiv preprint arXiv:1909.05858* (2019).

[465] Sumanth Dathathri et al. "Plug and play language models: A simple approach to controlled text generation". In: *arXiv preprint arXiv:1912.02164* (2019).

[466] Kevin Yang and Dan Klein. "FUDGE: Controlled text generation with future discriminators". In: *arXiv preprint arXiv:2104.05218* (2021).

[467] Thomas Wolf et al. "Huggingface's transformers: State-of-the-art natural language processing". In: *arXiv preprint arXiv:1910.03771* (Oct. 2019), pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: https://aclanthology.org/2020.emnlp-demos.6.

[468] Alexis Ross, Ana Marasović, and Matthew Peters. "Explaining NLP Models via Minimal Contrastive Editing (MiCE)". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 3840–3852. DOI: 10.18653/v1/2021.findings-acl.336. URL: https://aclanthology.org/2021.findings-acl.336.

[469] Colin Raffel et al. "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *arXiv preprint arXiv:1910.10683* (2019).

[470] Abigail See et al. "Do massively pretrained language models make better storytellers?" In: *arXiv preprint arXiv:1909.10705* (2019).

[471] Bente Kalsnes and Anders Olof Larsson. "Understanding news sharing across social media: Detailing distribution on Facebook and Twitter". In: *Journalism studies* 19.11 (2018), pp. 1669–1688.

[472] Nguyen Minh Ngoc. "Journalism and Social Media: The Transformation of Journalism in the Age of Social Media and Online News". In: *European Journal of Social Sciences Studies* 7.6 (2022).

[473] Tianxiao Shen et al. *Style Transfer from Non-Parallel Text by Cross-Alignment*. 2017. arXiv: 1705.09655 [cs.CL]. URL: https://arxiv.org/abs/1705.09655.

[474] Zhiting Hu et al. "Toward controlled generation of text". In: *International conference on machine learning*. PMLR. 2017, pp. 1587–1596.

[475] Abigail See, Peter J. Liu, and Christopher D. Manning. *Get To The Point: Summarization with Pointer-Generator Networks*. 2017. arXiv: 1704.04368 [cs.CL]. URL: https://arxiv.org/abs/1704.04368.

[476] Jingqing Zhang et al. "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization". In: *International conference on machine learning*. PMLR. 2020, pp. 11328–11339.

[477] Junxian He et al. "Ctrlsum: Towards generic controllable text summarization". In: *arXiv preprint arXiv:2012.04281* (2020).

[478] Chao Zhao et al. "Read Top News First: A Document Reordering Approach for Multi-Document News Summarization". In: *Findings of the Association for Computational Linguistics: ACL 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 613–621. DOI: 10.18653/v1/2022.findings-acl.51. URL: https://aclanthology.org/2022.findings-acl.51/.

[479] Max Grusky, Mor Naaman, and Yoav Artzi. *Newsroom: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies*. 2020. arXiv: 1804.11283 [cs.CL]. URL: https://arxiv.org/abs/1804.11283.

[480] Zeyu Dai, Himanshu Taneja, and Ruihong Huang. "Fine-grained structure-based news genre categorization". In: *Proceedings of the Workshop Events and Stories in the News*. 2018, pp. 17–23.

[481] Michel De Montaigne. *Essays*. Self-published, 1580.

[482] Kota Shamanth Ramanath Nayak. "Does ChatGPT Measure Up to Discourse Unit Segmentation? A Comparative Analysis Utilizing Zero-Shot Custom Prompts". In: ().

[483] Jihao Zhao et al. "Meta-chunking: Learning efficient text segmentation via logical perception". In: *arXiv preprint arXiv:2410.12788* (2024).

[484] Yaxin Fan et al. "Uncovering the Potential of ChatGPT for Discourse Analysis in Dialogue: An Empirical Study". In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 2024, pp. 16998–17010.

[485] Feng Jiang et al. "Advancing Topic Segmentation and Outline Generation in Chinese Texts: The Paragraph-level Topic Representation, Corpus, and Benchmark". In: *arXiv preprint arXiv:2305.14790* (2023).

[486] Alexander Spangher et al. "Sequentially Controlled Text Generation". In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 6848–6866. DOI: 10.18653/v1/2022.findings-emnlp.509. URL: https://aclanthology.org/2022.findings-emnlp.509.

[487] Bruce T. Lowerre. "The HARPY speech recognition system". In: 1976. URL: https://api.semanticscholar.org/CorpusID:61409851.

[488] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013/.

[489] Wojciech Kryscinski et al. "Evaluating the Factual Consistency of Abstractive Text Summarization". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber et al. Online: Association for Computational Linguistics, Nov. 2020, pp. 9332–9346. DOI: 10.18653/v1/2020.emnlp-main.750. URL: https://aclanthology.org/2020.emnlp-main.750/.

[490] Vladimir I. Levenshtein. "Binary codes capable of correcting deletions, insertions, and reversals". In: *Soviet physics. Doklady* 10 (1965), pp. 707–710. URL: https://api.semanticscholar.org/CorpusID:60827152.

[491] David Caswell and Konstantin Dörr. "Automated Journalism 2.0: Event-driven narratives: From simple descriptions to real stories". In: *Journalism practice* 12.4 (2018), pp. 477–496.

[492] Alexander Spangher et al. "NewsEdits: A News Article Revision Dataset and a Novel Document-Level Reasoning Challenge". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 127–157. DOI: 10.18653/v1/2022.naacl-main.10. URL: https://aclanthology.org/2022.naacl-main.10.

[493] David Caswell. "Telling every story: Characteristics of systematic reporting". In: *Journalism and Reporting Synergistic Effects of Climate Change*. Routledge, 2024, pp. 266–283.

[494] Alexander Quinn Nichol et al. "GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 16784–16804.

[495] Jonathan Ho and Tim Salimans. "Classifier-Free Diffusion Guidance". In: *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*. 2021.

[496] Shanchuan Lin et al. *Common Diffusion Noise Schedules and Sample Steps are Flawed*. 2023. arXiv: 2305.08891 [cs.CV].

[497] Prafulla Dhariwal and Alexander Nichol. "Diffusion models beat gans on image synthesis". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 8780–8794.

[498] Rinon Gal et al. "Stylegan-nada: Clip-guided domain adaptation of image generators". In: *arXiv preprint arXiv:2108.00946* (2021).

[499] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. "Diffusionclip: Text-guided diffusion models for robust image manipulation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 2426–2435.

[500] Katherine Crowson et al. "Vqgan-clip: Open domain image generation and editing with natural language guidance". In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVII*. Springer. 2022, pp. 88–105.

[501] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.

[502] Yilun Du, Shuang Li, and Igor Mordatch. "Compositional visual generation with energy based models". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6637–6647.

[503] Stable Diffusion Documentation. *How does negative prompt work?* https://stable-diffusion-art.com/how-negative-prompt-work/.

[504] Katherine Crowson et al. "Vqgan-clip: Open domain image generation and editing with natural language guidance". In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVII*. Springer. 2022, pp. 88–105.

[505] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2021. arXiv: 2112.10752 [cs.CV].

[506] Huan Ling et al. "EditGAN: High-Precision Semantic Image Editing". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2021.

[507] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In: *International Conference on Learning Representations*. 2020.

[508] Andrew Brock et al. "Neural Photo Editing with Introspective Adversarial Networks". In: *International Conference on Learning Representations*. 2016.

[509] Tomas Mikolov et al. "Efficient Estimation of Word Representations in Vector Space". In: *International Conference on Learning Representations*. 2013.

[510] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *ArXiv* abs/1810.04805 (2019).

[511] Nora Belrose et al. "LEACE: Perfect linear concept erasure in closed form". In: *arXiv preprint arXiv:2306.03819* (2023).

[512] Nikolay Malkin, Zhen Wang, and Nebojsa Jojic. "Coherence boosting: When your pretrained language model is not paying enough attention". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8214–8236. DOI: 10.18653/v1/2022.acl-long.565. URL: https://aclanthology.org/2022.acl-long.565.

[513] Jonathan Pei, Kevin Yang, and Dan Klein. *PREADD: Prefix-Adaptive Decoding for Controlled Text Generation*. 2023. arXiv: 2307.03214 [cs.CL].

[514] Weijia Shi et al. "Trusting Your Evidence: Hallucinate Less with Context-aware Decoding". In: *arXiv preprint arXiv:2305.14739* (2023).

[515] Maxwell Nye et al. "Show Your Work: Scratchpads for Intermediate Computation with Language Models". In: *Deep Learning for Code Workshop*. 2022.

[516] Jason Wei et al. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 24824–24837. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

[517] Leo Gao et al. *A framework for few-shot language model evaluation*. Version v0.0.1. Sept. 2021. DOI: 10.5281/zenodo.5371628. URL: https://doi.org/10.5281/zenodo.5371628.

[518] Sören Auer et al. "The SciQA Scientific Question Answering Benchmark for Scholarly Knowledge". In: *Scientific Reports* 13.1 (2023), p. 7240.

[519] Mandar Joshi et al. "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension". In: *arXiv preprint arXiv:1705.03551* (2017).

[520] Rowan Zellers et al. "HellaSwag: Can a machine really finish your sentence?" In: *arXiv preprint arXiv:1905.07830* (2019).

[521] Keisuke Sakaguchi et al. "Winogrande: An adversarial winograd schema challenge at scale". In: *Communications of the ACM* 64.9 (2021), pp. 99–106.

[522] Christopher Clark et al. "BoolQ: Exploring the surprising difficulty of natural yes/no questions". In: *arXiv preprint arXiv:1905.10044* (2019).

[523] Yonatan Bisk et al. "Piqa: Reasoning about physical commonsense in natural language". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 05. 2020, pp. 7432–7439.

[524] Karl Cobbe et al. "Training verifiers to solve math word problems". In: *arXiv preprint arXiv:2110.14168* (2021).

[525] Kinjal Basu, Farhad Shakerin, and Gopal Gupta. "Aqua: Asp-based visual question answering". In: *Practical Aspects of Declarative Languages: 22nd International Symposium, PADL 2020, New Orleans, LA, USA, January 20–21, 2020, Proceedings 22*. Springer. 2020, pp. 57–72.

[526] Peter Clark et al. "Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge". In: *arXiv:1803.05457v1* (2018).

[527] Denis Paperno et al. "The LAMBADA dataset: Word prediction requiring a broad discourse context". In: *arXiv preprint arXiv:1606.06031* (2016).

[528] Ari Holtzman et al. "The curious case of neural text degeneration". In: *arXiv preprint arXiv:1904.09751* (2019).

[529] Stella Biderman et al. *Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling*. 2023. arXiv: 2304.01373 [cs.CL].

[530] Hugo Touvron et al. "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971* (2023).

[531] Jack W. Rae et al. *Scaling Language Models: Methods, Analysis & Insights from Training Gopher*. 2021. DOI: 10.48550/ARXIV.2112.11446. URL: https://arxiv.org/abs/2112.11446.

[532] Karl Cobbe et al. "Training Verifiers to Solve Math Word Problems". In: *arXiv preprint arXiv:2110.14168* (2021).

[533] Wang Ling et al. "Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 158–167. DOI: 10.18653/v1/P17-1015. URL: https://aclanthology.org/P17-1015.

[534] Xuezhi Wang et al. "Self-Consistency Improves Chain of Thought Reasoning in Language Models". In: *ICLR 2023*. 2023. URL: https://arxiv.org/abs/2203.11171.

[535] Can Xu et al. *WizardLM: Empowering Large Language Models to Follow Complex Instructions*. 2023. arXiv: 2304.12244 [cs.CL].

[536] Tim Dettmers et al. *QLoRA: Efficient Finetuning of Quantized LLMs*. 2023. eprint: arXiv:2305.14314.

[537] Ben Wang and Aran Komatsuzaki. *GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model*. https://github.com/kingoflolz/mesh-transformer-jax. May 2021.

[538] Erik Nijkamp et al. "CodeGen: An Open Large Language Model for Code with Multi-Turn Program Synthesis". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=iaYcJKpY2B_.

[539] Mark Chen et al. "Evaluating Large Language Models Trained on Code". In: (2021). arXiv: 2107.03374 [cs.LG].

[540] Rohan Taori et al. *Stanford Alpaca: An Instruction-following LLaMA model*. https://github.com/tatsu-lab/stanford_alpaca. 2023.

[541] Kai Greshake et al. "More than you've asked for: A Comprehensive Analysis of Novel Prompt Injection Threats to Application-Integrated Large Language Models". In: *arXiv preprint arXiv:2302.12173* (2023).

[542] Andrew Rutherford. *ANOVA and ANCOVA: a GLM approach*. John Wiley & Sons, 2011.

[543] Ebtesam Almazrouei et al. "Falcon-40B: an open large language model with state-of-the-art performance". In: (2023).

[544] Victor Sanh et al. "Multitask Prompted Training Enables Zero-Shot Task Generalization". In: *International Conference on Learning Representations*. 2021.

[545] Andreas Köpf et al. "OpenAssistant Conversations–Democratizing Large Language Model Alignment". In: *arXiv preprint arXiv:2304.07327* (2023).

[546] Albert Webson and Ellie Pavlick. "Do prompt-based models really understand the meaning of their prompts?" In: *arXiv preprint arXiv:2109.01247* (2021).

[547] Weizhe Yuan et al. "Self-Rewarding Language Models". In: *arXiv preprint arXiv:2401.10020* (2024).

[548] Andrew L. Maas et al. "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. URL: http://www.aclweb.org/anthology/P11-1015.

[549] Tao Meng et al. "Controllable Text Generation with Neurally-Decomposed Oracle". In: *arXiv preprint arXiv:2205.14219* (2022).

[550] José M Chenlo, Alexander Hogenboom, and David E Losada. "Rhetorical Structure Theory for Polarity Estimation: An Experimental Study". In: *Data & Knowledge Engineering* 94 (2014), pp. 135–147.

[551] Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. "Evaluation of Text Generation: A Survey". In: *arXiv preprint arXiv:2006.14799* (2020).

[552] Masaru Isonuma, Junichiro Mori, and Ichiro Sakata. "Unsupervised Neural Single-Document Summarization of Reviews via Learning Latent Discourse Structure and its Ranking". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 2142–2152. DOI: 10.18653/v1/P19-1206. URL: https://aclanthology.org/P19-1206.

[553] Georg Rehm, Karolina Zaczynska, and Julián Moreno-Schneider. "Semantic Storytelling: Towards Identifying Storylines in Large Amounts of Text Content". In: (2019).

[554] Ali Haif Abbas. "Politicizing the Pandemic: A Schemata Analysis of COVID-19 News in Two Selected Newspapers". In: *International Journal for the Semiotics of Law-Revue internationale de Sémiotique juridique* (2020), pp. 1–20.

[555] Rashmi Prasad et al. "The Penn Discourse TreeBank 2.0." In: *LREC*. Citeseer. 2008.

[556] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. "Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory". In: *Current and new directions in discourse and dialogue*. Springer, 2003, pp. 85–112.

[557] W Victor Yarlott et al. "Identifying the Discourse Function of News Article Paragraphs". In: *Proceedings of the Workshop Events and Stories in the News 2018*. 2018, pp. 25–33.

[558] Ronan Collobert and Jason Weston. "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning". In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 160–167.

[559] Debopam Das, Manfred Stede, and Maite Taboada. "The Good, the Bad, and the Disagreement: Complex Ground Truth in Rhetorical Structure Analysis". In: *Proceedings of the 6th Workshop on Recent Advances in RST and Related Formalisms*. 2017, pp. 11–19.

[560] Yu Zhang and Qiang Yang. "A Survey on Multi-Task Learning". In: *arXiv preprint arXiv:1707.08114* (2017).

[561] Xiangci Li, Gully Burns, and Nanyun Peng. "Scientific Discourse Tagging for Evidence Extraction". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 2550–2562. URL: https://aclanthology.org/2021.eacl-main.218.

[562] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL].

[563] Ashish Vaswani et al. "Attention is All You Need". In: *Advances in neural information processing systems* 30 (2017), pp. 5998–6008.

[564] David Craig. *The Ethics of the Story: Using Narrative Techniques Responsibly in Journalism*. Rowman & Littlefield, 2006.

[565] Kathy Roberts Forde. "Discovering the Explanatory Report in American Newspapers". In: *Journalism Practice* 1.2 (2007), pp. 227–244.

[566] Catherine A Steele and Kevin G Barnhurst. "The Journalism of Opinion: Network News Coverage of US Presidential Campaigns, 1968–1988". In: *Critical Studies in Media Communication* 13.3 (1996), pp. 187–209.

[567] Robert F Bales. "Interaction Process Analysis; a Method for the Study of Small Groups." In: (1950).

[568] Robert Freed Bales. "Personality and Interpersonal Behavior." In: (1970).

[569] Peter C Austin and Elizabeth A Stuart. "Moving Towards Best Practice When Using Inverse Probability of Treatment Weighting (IPTW) Using the Propensity Score to Estimate Causal Treatment Effects in Observational Studies". In: *Statistics in medicine* 34.28 (2015), pp. 3661–3679.

[570] Matthew E Peters et al. "Deep contextualized word representations". In: *Proceedings of NAACL-HLT*. 2018, pp. 2227–2237.

[571] Jaejun Lee, Raphael Tang, and Jimmy Lin. "What Would Elsa Do? Freezing Layers During Transformer Fine-Tuning". In: *arXiv preprint arXiv:1911.03090* (2019).

[572] Terrance DeVries and Graham W Taylor. "Dataset Augmentation in Feature Space". In: *arXiv preprint arXiv:1702.05538* (2017).

[573] Jesper E Van Engelen and Holger H Hoos. "A Survey on Semi-Supervised Learning". In: *Machine Learning* 109.2 (2020), pp. 373–440.

[574] Sergey Edunov et al. "Understanding Back-Translation at Scale". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 489–500. DOI: 10.18653/v1/D18-1045. URL: https://www.aclweb.org/anthology/D18-1045.

[575] Myle Ott et al. "fairseq: A Fast, Extensible Toolkit for Sequence Modeling". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. 2019, pp. 48–53.

[576] Jiaao Chen, Zichao Yang, and Diyi Yang. "MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 2147–2157. DOI: 10.18653/v1/2020.acl-main.194. URL: https://www.aclweb.org/anthology/2020.acl-main.194.

[577] Qizhe Xie et al. "Unsupervised Data Augmentation for Consistency Training". In: *Advances in Neural Information Processing Systems* 33 (2020).

[578] David Berthelot et al. "MixMatch: A Holistic Approach to Semi-Supervised Learning". In: *NeurIPS*. 2019.

[579] Minsung Hyun, Jisoo Jeong, and Nojun Kwak. "Class-Imbalanced Semi-Supervised Learning". In: *arXiv preprint arXiv:2002.06815* (2020).

[580] Joachim Bingel and Anders Søgaard. "Identifying Beneficial Task Relations for Multi-Task Learning in Deep Deural Networks". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 164–169. URL: https://aclanthology.org/E17-2026.

[581] Connor Shorten and Taghi M Khoshgoftaar. "A Survey on Image Data Augmentation for Deep Learning". In: *Journal of Big Data* 6.1 (2019), pp. 1–48.

[582] Nitesh V Chawla et al. "SMOTE: Synthetic Minority Over-Sampling Technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[583] Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. "On the Stability of Fine-tuning BERT: Misconceptions, Explanations, and Strong Baselines". In: *9th International Conference on Learning Representations*. CONF. 2021.

[584] Ruize Wang et al. "K-adapter: Infusing knowledge into pre-trained models with adapters". In: *arXiv preprint arXiv:2002.01808* (2020).

[585] Lucien Mehl. *Automation in the legal world*. National Physical Laboratory, 1958.

[586] Robert Dale. "Law and word order: NLP in legal tech". In: *Natural Language Engineering* 25.1 (2019), pp. 211–217.

[587] Samuel Gibbs. "Chatbot lawyer overturns 160,000 parking tickets in London and New York". In: *The Guardian* (June 2016). URL: https://www.theguardian.com/technology/2016/jun/28/chatbot-ai-lawyer-donotpay-parking-tickets-london-new-york.

[588] Neel Guha et al. *LegalBench: A Collaboratively Built Benchmark for Measuring Legal Reasoning in Large Language Models*. 2023. arXiv: 2308.11462 [cs.CL].

[589] Daniel Martin Katz et al. "Gpt-4 passes the bar exam". In: *Available at SSRN 4389233* (2023).

[590] Niklas Dehio, Malte Ostendorff, and Georg Rehm. "Claim Extraction and Law Matching for COVID-19-related Legislation". In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. 2022, pp. 480–490.

[591] Rishi Bommasani et al. “On the opportunities and risks of foundation models”. In: *arXiv preprint arXiv:2108.07258* (2021).

[592] Haoxi Zhong et al. “JEC-QA: a legal-domain question answering dataset”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05. 2020, pp. 9701–9708.

[593] Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. “A dataset for statutory reasoning in tax law entailment and question answering”. In: *arXiv preprint arXiv:2005.05257* (2020).

[594] Yuta Koreeda and Christopher D Manning. “ContractNLI: A dataset for document-level natural language inference for contracts”. In: *arXiv preprint arXiv:2110.01799* (2021).

[595] Dan Hendrycks et al. “Cuad: An expert-annotated nlp dataset for legal contract review”. In: *arXiv preprint arXiv:2103.06268* (2021).

[596] Shomir Wilson et al. “The creation and analysis of a website privacy policy corpus”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2016, pp. 1330–1340.

[597] Sebastian Zimmeck et al. “Maps: Scaling privacy compliance analysis to a million apps”. In: *Proc. Priv. Enhancing Tech.* 2019 (2019), p. 66.

[598] Steven H Wang et al. “MAUD: An Expert-Annotated Legal NLP Dataset for Merger Agreement Understanding”. In: *arXiv preprint arXiv:2301.00876* (2023).

[599] Judith Eckle-Kohler, Roland Kluge, and Iryna Gurevych. “On the role of discourse markers for discriminating claims and premises in argumentative discourse”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 2236–2242.

[600] Jiaao Chen and Diyi Yang. “Structure-aware abstractive conversation summarization via discourse and action graphs”. In: *arXiv preprint arXiv:2104.08400* (2021).

[601] Karl Engisch, Thomas Würtenberger, and Dirk Otto. *Einführung in das juristische Denken*. Kohlhammer Verlag, 2018.

[602] Anne von der Lieth Gardner. *Artificial intelligence approach to legal reasoning*. Tech. rep. Stanford Univ., 1984.

[603] Kevin P Tobia. “Testing ordinary meaning”. In: *Harv. L. Rev.* 134 (2020), p. 726.

[604] Christopher D Manning et al. "The Stanford CoreNLP natural language processing toolkit". In: *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 2014, pp. 55–60.

[605] Mariana Neves and Jurica Ševa. "An extensive review of tools for manual annotation of documents". In: *Briefings in bioinformatics* 22.1 (2021), pp. 146–163.

[606] Pontus Stenetorp et al. "brat: a Web-based Tool for NLP-Assisted Text Annotation". In: *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Avignon, France: Association for Computational Linguistics, Apr. 2012, pp. 102–107. URL: https://www.aclweb.org/anthology/E12-2021.

[607] Jie Yang et al. "YEDDA: A lightweight collaborative text span annotation tool". In: *arXiv preprint arXiv:1711.03759* (2017).

[608] Seid Muhie Yimam et al. "Webanno: A flexible, web-based and visually supported system for distributed annotations". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 2013, pp. 1–6.

[609] Megumi Kameyama. "Recognizing referential links: An information extraction perspective". In: *arXiv preprint cmp-lg/9707009* (1997).

[610] Tianyu Liu et al. "Autoregressive structured prediction with language models". In: *arXiv preprint arXiv:2210.14698* (2022).

[611] María Granados Buey et al. "The AIS project: Boosting information extraction from legal documents by using ontologies." In: *ICAART (2)*. 2016, pp. 438–445.

[612] Erik F Sang and Fien De Meulder. "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition". In: *arXiv preprint cs/0306050* (2003).

[613] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. "Kernel methods for relation extraction". In: *Journal of machine learning research* 3.Feb (2003), pp. 1083–1106.

[614] Qi Li and Heng Ji. "Incremental joint extraction of entity mentions and relations". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2014, pp. 402–412.

[615] Zexuan Zhong and Danqi Chen. "A frustratingly easy approach for entity and relation extraction". In: *arXiv preprint arXiv:2010.12812* (2020).

[616] Colin Raffel et al. "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *The Journal of Machine Learning Research* 21.1 (2020), pp. 5485–5551.

[617] Zhen Wan et al. "Gpt-re: In-context learning for relation extraction using large language models". In: *arXiv preprint arXiv:2305.02105* (2023).

[618] Erik F Sang and Sabine Buchholz. "Introduction to the CoNLL-2000 shared task: Chunking". In: *arXiv preprint cs/0009008* (2000).

[619] Lorenda A Naylor. "Counting an invisible class of citizens: The LGBT population and the US census". In: *Public Integrity* 22.1 (2020), pp. 54–72.

[620] Jeffrey Mervis. "Census citizenship question is dropped, but challenges linger". In: *Science* 365.6450 (2019), pp. 211–211. ISSN: 0036-8075. DOI: 10.1126/science.365.6450.211. eprint: https://science.sciencemag.org/content/365/6450/211.full.pdf. URL: https://science.sciencemag.org/content/365/6450/211.

[621] RaJade M Berry-James, Susan T Gooden, and Richard Greggory Johnson III. "Civil Rights, Social Equity, and Census 2020". In: *Public Administration Review* 80.6 (2020), pp. 1100–1108.

[622] Andrew Reamer. "Counting for dollars 2020: the role of the decennial census in the geographic distribution of federal funds". In: *Initial Analysis* 16 (2018).

[623] BV Elasticsearch. "Elasticsearch". In: *software], version* 6.1 (2018).

[624] Jonathan E Vespa, David M Armstrong, Lauren Medina, et al. *Demographic turning points for the United States: Population projections for 2020 to 2060*. US Department of Commerce, Economics and Statistics Administration, US . . ., 2018.

[625] Marcos Eduardo Kauffman and Marcelo Negri Soares. "AI in legal services: new trends in AI-enabled legal services". In: *Service Oriented Computing and Applications* 14.4 (Dec. 2020), pp. 223–226. ISSN: 1863-2394. DOI: 10.1007/s11761-020-00305-x. URL: https://doi.org/10.1007/s11761-020-00305-x.

[626] Jason Morris. *Making Mischief With Open-Source Legal Tech: Radiant Law*. Oct. 2019. URL: https://www-proquest-com.libproxy2.usc.edu/blogs-podcasts-websites/making-mischief-with-open-source-legal-tech/docview/2307652320/se-2?accountid=14749.

[627] Grant Vergottini. *To go Open Source or Not?* 2011. URL: https://xcential.com/to-go-open-source-or-not/.

[628] Katrina June Lee, Susan Azyndar, and Ingrid AB Mattson. "A New Era: Integrating Today's Next Gen Research Tools Ravel and Casetext in the Law School Classroom". In: *Rutgers Computer & Tech. LJ* 41 (2015), p. 31.

[629] Adam Z Wyner and Wim Peters. "On Rule Extraction from Regulations." In: *JURIX*. Vol. 11. 2011, pp. 113–122.

[630] Nicola Zeni et al. "GaiusT: supporting the extraction of rights and obligations for regulatory compliance". In: *Requirements engineering* 20.1 (2015), pp. 1–22.

[631] Borja Espejo-Garcia et al. "End-to-end sequence labeling via deep learning for automatic extraction of agricultural regulations". In: *Computers and Electronics in Agriculture* 162 (2019), pp. 106–111.

[632] Lucy Lu Wang et al. "CORD-19: The COVID-19 Open Research Dataset". In: *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*. Online: Association for Computational Linguistics, July 2020. URL: https://www.aclweb.org/anthology/2020.nlpcovid19-acl.1.

[633] Mohammad Golam Sohrab et al. "BENNERD: A Neural Named Entity Linking System for COVID-19". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 182–188. DOI: 10.18653/v1/2020.emnlp-demos.24. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.24.

[634] Tom Hope et al. "SciSight: Combining faceted navigation and research group detection for COVID-19 exploratory scientific search". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 135–143. DOI: 10.18653/v1/2020.emnlp-demos.18. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.18.

[635] Alexander Spangher et al. "Enabling Low-Resource Transfer Learning across COVID-19 Corpora by Combining Event-Extraction and Co-Training". In: *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*. Online: Association for Computational Linguistics, July 2020. URL: https://www.aclweb.org/anthology/2020.nlpcovid19-acl.4.

[636] Karin Verspoor et al., eds. *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*. Online: Association for Computational Linguistics, July 2020. URL: https://www.aclweb.org/anthology/2020.nlpcovid19-acl.0.

[637] Karin Verspoor et al., eds. *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*. Online: Association for Computational Linguistics, Dec. 2020. URL: https://www.aclweb.org/anthology/2020.nlpcovid19-2.0.

[638] Markus Hartung, Micha-Manuel Bues, and Gernot Halbleib. *Legal tech*. CH Beck, 2017.

[639] Jeongsub Lim. "THE MYTHOLOGICAL STATUS OF THE IMMEDIACY OF THE MOST IMPORTANT ONLINE NEWS An analysis of top news flows in diverse online media". In: *Journalism Studies* 13 (Feb. 2012), pp. 71–89. DOI: 10.1080/1461670X.2011.605596.

[640] Neil J. Thurman. "How Live Blogs are Reconfiguring Breaking News". In: 2013. URL: https://api.semanticscholar.org/CorpusID:153429039.

[641] Michael Karlsson and Jesper Strömbäck. "Freezing the Flow of Online News : Exploring Approaches to Study the Liquidity of Online News". In: 2009. URL: https://api.semanticscholar.org/CorpusID:15653949.

[642] Andreas Widholm. "Tracing Online News in Motion". In: *Digital Journalism* 4 (2016), pp. 24–40. URL: https://api.semanticscholar.org/CorpusID:62063121.

[643] Michael Karlsson, Christer Clerwall, and Lars Nord. "Do Not Stand Corrected: Transparency and Users Attitudes to Inaccurate News and Corrections in Online Journalism". In: *Journalism Mass Communication Quarterly* 94 (June 2016). DOI: 10.1177/1077699016654680.

[644] Sydney L. Forde, Robert E. Gutsche, and Juliet Pinto. "Exploring "ideological correction" in digital news updates of Portland protests & police violence". In: *Journalism* 24 (2022), pp. 157–176. URL: https://api.semanticscholar.org/CorpusID:248910766.

[645] Michael Karlsson. "RITUALS OF TRANSPARENCY". In: *Journalism Studies* 11 (2010), pp. 535–545. URL: https://api.semanticscholar.org/CorpusID:142571133.

[646] Neil J. Thurman and Nic Newman. "The Future of Breaking News Online?" In: *Journalism Studies* 15 (2014), pp. 655–667. URL: https://api.semanticscholar.org/CorpusID:143987445.

[647] Damien Neadle, Elisa Bandini, and Claudio Tennie. "Testing the individual and social learning abilities of task-naıve captive chimpanzees (Pan troglodytes sp.) in a nut-cracking task". In: *PeerJ* 8 (2020), e8734.

[648] Doreen Thompson and James A. Russell. "The ghost condition: imitation versus emulation in young children's observational learning." In: *Developmental psychology* 40 5 (2004), pp. 882–9. URL: https://api.semanticscholar.org/CorpusID:10208845.

[649] Victoria Horner and Andrew Whiten. "Causal knowledge and imitation/emulation switching in chimpanzees (Pan troglodytes) and children (Homo sapiens)". In:

Animal Cognition 8 (2005), pp. 164–181. URL: https://api.semanticscholar.org/CorpusID:1949770.

[650] Carl Bereiter and Marlene Scardamalia. "The psychology of written composition". In: 1987. URL: https://api.semanticscholar.org/CorpusID:143781031.

[651] Nancy I. Sommers. "Revision Strategies of Student Writers and Experienced Adult Writers". In: *College Composition & Communication* (1980). URL: https://api.semanticscholar.org/CorpusID:42322944.

[652] Dirk Van Hulle. "Introduction: The draft in literary history". In: *Drafts in Literary History*. Open-access introduction. John Benjamins / Association Internationale de Littérature Comparée, 2024. DOI: 10.1075/chlel.xxxv.int.

[653] Ronald A Finke, Thomas B Ward, and Steven M Smith. *Creative Cognition: Theory, Research, and Application*. MIT Press, 1992.

[654] Alexander Spangher et al. "NewsEdits 2.0: Learning the Intentions Behind Updating News". In: *arXiv preprint arXiv:2411.18811* (2024). Submitted on 27 Nov 2024.

[655] Roman Grundkiewicz and Marcin Junczys-Dowmunt. "The wiked error corpus: A corpus of corrective wikipedia edits and its application to grammatical error correction". In: *International Conference on Natural Language Processing*. Springer. 2014, pp. 478–490.

[656] Manaal Faruqui et al. "WikiAtomicEdits: A Multilingual Corpus of Wikipedia Edits for Modeling Language and Discourse". In: *arXiv preprint arXiv:1808.09422* (Oct. 2018), pp. 305–315. DOI: 10.18653/v1/D18-1028. URL: https://www.aclweb.org/anthology/D18-1028.

[657] Fan Zhang and Diane Litman. "Annotation and Classification of Argumentative Writing Revisions". In: *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*. Denver, Colorado: Association for Computational Linguistics, June 2015, pp. 133–143. DOI: 10.3115/v1/W15-0616. URL: https://aclanthology.org/W15-0616.

[658] Darsh Shah, Tal Schuster, and Regina Barzilay. "Automatic fact-guided sentence modification". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05. 2020, pp. 8791–8798.

[659] Tazin Afrin et al. "Annotation and Classification of Evidence and Reasoning Revisions in Argumentative Writing". In: *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*. 2020, pp. 75–84.

[660] Manaal Faruqui et al. "WikiAtomicEdits: A Multilingual Corpus of Wikipedia Edits for Modeling Language and Discourse". In: *Proceedings of EMNLP*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 305–315. DOI: 10.18653/v1/D18-1028.

[661] Aurélien Max and Guillaume Wisniewski. "Mining Naturally-occurring Corrections and Paraphrases from Wikipedia's Revision History (WiCoPaCo)". In: *arXiv preprint arXiv:2202.12575* (2022).

[662] Irshad Ahmad Bhat and Talita Anthonio. "Towards Modeling Revision Requirements in wikiHow Instructions". In: *Conference on Empirical Methods in Natural Language Processing*. 2020. URL: https://api.semanticscholar.org/CorpusID:226262307.

[663] Suzanne M Kirchhoff. *US newspaper industry in transition*. DIANE Publishing, 2010.

[664] Johannes Daxenberger and Iryna Gurevych. "A Corpus-Based Study of Edit Categories in Featured and Non-Featured Wikipedia Articles". In: *Proceedings of COLING 2012*. Mumbai, India: The COLING 2012 Organizing Committee, Dec. 2012, pp. 711–726. URL: https://www.aclweb.org/anthology/C12-1044.

[665] Johannes Daxenberger and Iryna Gurevych. "Automatically classifying edit categories in Wikipedia revisions". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 2013, pp. 578–589.

[666] Peter Kin-Fong Fong and Robert P Biuk-Aghai. "What did they do? deriving high-level edit histories in wikis". In: *Proceedings of the 6th International Symposium on Wikis and Open Collaboration*. 2010, pp. 1–10.

[667] Chip Scanlan. "Writing from the top down: Pros and cons of the inverted pyramid". In: *Poynter Online., Erişim tarihi* 14 (2003).

[668] Zhe Quan et al. "An efficient framework for sentence similarity modeling". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.4 (2019), pp. 853–865.

[669] Sheikh Abujar, Mahmudul Hasan, and Syed Akhter Hossain. "Sentence similarity estimation for text summarization using deep learning". In: *Proceedings of the 2nd International Conference on Data Engineering and Communication Technology*. Springer. 2019, pp. 155–164.

[670] Haipeng Yao, Huiwen Liu, and Peiying Zhang. "A novel sentence similarity model with word embedding based on convolutional neural network". In: *Concurrency and Computation: Practice and Experience* 30.23 (2018), e4415.

[671] Qingyu Chen et al. "Sentence similarity measures revisited: ranking sentences in PubMed documents". In: *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. 2018, pp. 531–532.

[672] Tomoyuki Kajiwara and Mamoru Komachi. "Building a Monolingual Parallel Corpus for Text Simplification Using Sentence Similarity Based on Alignment between Word Embeddings". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 1147–1158. URL: https://aclanthology.org/C16-1109.

[673] Xiaoqi Jiao et al. "TinyBERT: Distilling BERT for Natural Language Understanding". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4163–4174. DOI: 10.18653/v1/2020.findings-emnlp.372. URL: https://aclanthology.org/2020.findings-emnlp.372.

[674] Harold W Kuhn. "The Hungarian method for the assignment problem". In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.

[675] Kishore Papineni et al. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[676] Kathleen A Hansen et al. "Local breaking news: Sources, technology, and news routines". In: *Journalism Quarterly* 71.3 (1994), pp. 561–572.

[677] Justin Lewis and Stephen Cushion. "The thirst to be first: An analysis of breaking news stories and their impact on the quality of 24-hour news coverage in the UK". In: *Journalism Practice* 3.3 (2009), pp. 304–318.

[678] Mats Ekström, Amanda Ramsälv, and Oscar Westlund. "The Epistemologies of Breaking News". In: *Journalism Studies* 22.2 (2021), pp. 174–192.

[679] Nikki Usher. "Breaking news production processes in US metropolitan newspapers: Immediacy and journalistic authority". In: *Journalism* 19.1 (2018), pp. 21–36.

[680] Mingyu Derek Ma et al. "EventPlus: A Temporal Event Understanding Pipeline". In: *arXiv preprint arXiv:2101.04922* (2021).

[681] Alexander Spangher et al. ""Don't quote me on that": Finding Mixtures of Sources in News Articles". In: *Proceedings of Computation+Journalism Conference*. 2020.

[682] Rujun Han, Qiang Ning, and Nanyun Peng. "Joint Event and Temporal Relation Extraction with Shared Representations and Structured Prediction". In: *Proceedings*

of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 434–444. DOI: 10.18653/v1/D19-1041. URL: https://aclanthology.org/D19-1041.

[683] Nasrin Mostafazadeh et al. "A corpus and cloze evaluation for deeper understanding of commonsense stories". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2016, pp. 839–849.

[684] Nathanael Chambers and Dan Jurafsky. "Unsupervised learning of narrative event chains". In: *Proceedings of ACL-08: HLT*. 2008, pp. 789–797.

[685] Te-Lin Wu et al. "Understanding Multimodal Procedural Knowledge by Sequencing Multimodal Instructional Manuals". In: *Proceedings of the Conference of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2022.

[686] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan (Mar. 2003), pp. 993–1022. ISSN: 1532-4435.

[687] Xiaoya Li et al. "Dice Loss for Data-imbalanced NLP Tasks". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 465–476. DOI: 10.18653/v1/2020.acl-main.45. URL: https://aclanthology.org/2020.acl-main.45.

[688] Norm Goldstein. *The Associate Press Rules Regulations and General Orders*. 1953. URL: https://www.apstylebook.com/.

[689] Ryan L Boyd, Kate G Blackburn, and James W Pennebaker. "The narrative arc: Revealing core narrative structures through text analysis". In: *Science advances* 6.32 (2020), eaba2196.

[690] Nasrin Mostafazadeh et al. "Lsdsem 2017 shared task: The story cloze test". In: *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*. 2017, pp. 46–51.

[691] Zhixing Tian et al. "Scene Restoring for Narrative Machine Reading Comprehension". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 3063–3073.

[692] Nanyun Peng et al. "Towards controllable story generation". In: *Proceedings of the First Workshop on Storytelling*. 2018, pp. 43–49.

[693] Jiaao Chen, Jianshu Chen, and Zhou Yu. "Incorporating structured commonsense knowledge in story completion". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 6244–6251.

[694] Motti Neiger and Keren Tenenboim-Weinblatt. "Understanding journalism through a nuanced deconstruction of temporal layers in news narratives". In: *Journal of Communication* 66.1 (2016), pp. 139–160.

[695] Diyi Yang et al. "Identifying semantic edit intentions from revisions in wikipedia". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017, pp. 2000–2010.

[696] Kostas Saltzis. "Breaking news online: How news stories are updated and maintained around-the-clock". In: *Journalism practice* 6.5-6 (2012), pp. 702–710.

[697] George R Doddington et al. "The automatic content extraction (ace) program-tasks, data, and evaluation." In: *Lrec*. Vol. 2. 1. Lisbon. 2004, pp. 837–840.

[698] Teun A Van Dijk. *News as discourse*. Lawrence Erlbaum Associates, 1998.

[699] Sha Li, Heng Ji, and Jiawei Han. "Document-Level Event Argument Extraction by Conditional Generation". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kristina Toutanova et al. Online: Association for Computational Linguistics, June 2021, pp. 894–908. DOI: 10.18653/v1/2021.naacl-main.69. URL: https://aclanthology.org/2021.naacl-main.69.

[700] Kung-Hsiang Huang, Sam Tang, and Nanyun Peng. "Document-level Entity-based Extraction as Template Generation". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens et al. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 5257–5269. DOI: 10.18653/v1/2021.emnlp-main.426. URL: https://aclanthology.org/2021.emnlp-main.426.

[701] I Hsu et al. "DEGREE: A data-efficient generation-based event extraction model". In: *arXiv preprint arXiv:2108.12724* (2021).

[702] Yixin Nie et al. "Adversarial NLI: A New Benchmark for Natural Language Understanding". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020.

[703] Rujun Han, Xiang Ren, and Nanyun Peng. "Econet: Effective continual pretraining of language models for event temporal reasoning". In: *arXiv preprint arXiv:2012.15283* (2020).

[704] Qingyu Tan, Hwee Tou Ng, and Lidong Bing. "Towards benchmarking and improving the temporal reasoning capability of large language models". In: *arXiv preprint arXiv:2306.08952* (2023).

[705] Siheng Xiong et al. "Large language models can learn temporal reasoning". In: *arXiv preprint arXiv:2401.06853* (2024).

[706] Jungo Kasai et al. "RealTime QA: What's the Answer Right Now?" In: *arXiv preprint arXiv:2207.13332* (2022).

[707] Talita Anthonio, Irshad Bhat, and Michael Roth. "wikiHowToImprove: A Resource and Analyses on Edits in Instructional Texts". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 5721–5729. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.702.

[708] Ildiko Pilan et al. "A dataset for investigating the impact of feedback on student revision outcome". In: *12th Conference on Language Resources and Evaluation (LREC 2020)*. European Language Resources Association (ELRA). 2020, pp. 332–339.

Glossary

Mathematical Notation

- x**:** Event / context under judgment (e.g., a policy item from SFBOS or article text being evaluated) that conditions the decision or trajectory; serves as the input whose properties and surroundings drive π. (Sections 1.2, 2.2, 2.2.1, 2.2.3, 2.3, 2.3.3, 2.3.1, 2.3.3.1, 2.2.4, 2.2.2, 2.3.3.4, 3.2, 3.2.1, 3.2.3, 3.4.1.2, 3.3, 3.3.2, 3.4.3.2, 3.4.3, 3.4.3.3, 4.3.1.2, 3.4, 4.1, 4.2, 4.2.3.1, 4.4, 4.2.2, 4.3.2, 4.3.2.1, 4.3, 4.3.4, 5.1, 5.2, 5.2.1.1, 5.2.1.3, 5.3, 5.2.3.1, 5.2.3.2, 5.2.3, 5.2.3.6, 5.3.4, 5.2.3.5, 5.3.5, 5.3.3, 4.3.4.1)
- $g = s_n$**:** Goal-state artifact (e.g., a published article or realized homepage layout) observable at the end of a trajectory and used by inverse models (e.g., $q_\theta(\cdot \mid g)$) to infer latent actions; supervision signal when actions are hidden. (Sections 1.2, 2.2, 2.3.1, 2.3, 2.2.3, 3.2.1, 3.2, 3.4, 4.1, 4.2.3.1, 3.3.2, 5.1, 5.2)
- a**,** a_t**:** (Latent) action / decision variable indicating what an expert does at a step (e.g., *cover/ignore*, *retrieve source*, *place lead*, *edit sentence*); instantiated per chapter as selection, sourcing, structuring, or editing moves. (Sections 2.2, 2.2.1, 2.3.1, 3, 3.2, 3.4, 4.1, 4.2.3.1, 5.1, 5.2, 5.3)
- $\mathbf{a} = a_1, a_2, \ldots$**:** A full sequence of actions (the decision sequence realized by the expert or a model policy). (Sections 2.2.1, 3, 3.4, 5.2)
- s_t**:** State at step t (e.g., a draft state or version-t of an article) aggregating history and constraining feasible next actions. (Sections 2.2.1, 4.1, 4.2, 5.2, 5.3)
- $\mathbf{s} = s_1, s_2, \ldots$**:** A full sequence of states (intermediate artifacts along the trajectory). (Sections 4.1, 5.2)
- $\tau = [(a_1, s_1), (a_2, s_2), \ldots]$**:** State–action trajectory whose realization yields the observed artifact $g = s_n$. (Sections 2.2.1, 3, 4.1)
- $\tau = [(a_{1,1}, s_{1,1}), \ldots, (a_{1,2}, s_{1,2}), \ldots, (a_{2,1}, s_{2,1}), \ldots]$**:** Trajectory in the *NewsEdits* experiment where i indexes published versions and j drafts within a version; s_{i,n_j} (final draft of version i) is observable, intra-version drafts are unobserved. (Sections 5.2, 5.3)
- $\pi(\cdot)$**,** $\hat{\pi}(\cdot)$**:** Policy (true) and learned policy mapping contexts/states to actions/trajectories. Variations include: $\pi(a \mid x)$ (binary selection), $\pi(a \mid x, C)$ (selection with competitor set), $\hat{\pi}(a \mid x)$ and $\hat{\pi}(a \mid x, C)$ (learned classifiers/rankers), $\pi(\tau \mid x)$ (trajectory distribution), and $\pi^*(a_{t+1} \mid s_t)$ (greedy/optimal next-action under a planner). (Sections 2.2, 2.3.1, 2.3, 3, 3.4, 4.1, 5.3)
- $q_\theta(\cdot)$**:** Inverse model mapping observables back to latent actions/labels for supervision/analysis. Variations

include: $q_\theta(a \mid g)$ (single action from end-state), $q_\theta(\mathbf{a} \mid g)$ (multi-action set), $q_\theta(\tau \mid g)$ (full trajectory from g), and $q_\theta(\cdot \mid s_t, s_{t+1})$ (inferring edit intentions from version deltas). (Sections 2.2.3, 3.2, 3.2.1, 4.2.3.1, 5.3)

- $R(\cdot)$ **vs.** $R(x)$**:** Reward/utility in decision modeling vs. observation-channel recall (probability that a true decision leaves detectable evidence); disambiguated by argument and chapter. (Section 2.2)
- $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$**:** MDP tuple for general framing (states, actions, dynamics, reward, discount), including the horizon-1 specialization. (Section 2.2.1)
- c**,** C**:** External context and competitor set (e.g., other homepage items considered jointly) that modulate preferences/utilities and make prominence judgments set-dependent. (Section 2.3.1)
- $M_\psi(x, g)$**:** Learned linking/alignment function in the PRM-based observation channel estimating coverage ($x \leftrightarrow g$) from auxiliary attributes; supervises binary publishing decisions. (Section 2.2.3)
- l**:** Binary link indicator in the PRM implementing $M_\psi(x, g) = p(l \mid x, g, \mathbf{h})$, denoting whether g covers x. (Section 2.2.3)
- h_i**:** Auxiliary PRM attributes (content/source-derived features) used in $M_\psi(x, g) = p(l \mid x, g, \mathbf{h})$, where $\mathbf{h} = h_1, h_2, \ldots$. (Section 2.2.3)
- $p_o(x > x')$**:** Observed pairwise homepage preference for outlet o, derived from layout cues (position/-size/graphics); used to recover latent utilities that rank items. (Section 2.3.3)
- $u_\theta(x, C)$**:** Latent utility consistent with observed pairwise preferences over a competitor set C (e.g., Bradley–Terry/Thurstone/Plackett–Luce formulations) used to supervise $\hat{\pi}$. (Section 2.3.1)
- $D_{\text{train}}, D_{\text{test}}$**:** Time-based splits for forward-generalization evaluation under temporal drift. (Section 2.2.4)
- q_i **(source):** The i-th element of the source set $\mathcal{Q}$—a person, document, dataset, record, or observation used in sentence→source mapping α—with channel type and discourse metadata (role, centrality, stance). (Sections 3.2, 3.2.1, 3.4.1.2)
- $\mathcal{D}$=$\mathcal{D}$**,** $\mathcal{Q}$=$\mathcal{Q}$**:** Universes of sentences and sources for detection/identification tasks. (Section 3.2)
- $\alpha(x)$**:** Sentence→source subset mapping α permitting multiple sources/channels per sentence; supports evaluation of attribution quality. (Section 3.2)
- π_{plan}**,** π_{exec}**:** High-level planner and lower-level executor policies in hierarchical reporting; the planner selects discourse/narrative needs, the executor issues concrete retrieval actions. (Section 3.4)
- $\nu(g)$=$\nu(g)$**:** Narrative needs (e.g., context, countervoice, data) that a completed story should satisfy; used in schema-level planning/evaluation. (Section 3.4)
- $\widehat{R}$**:** NLI-derived document-level score proxy (e.g., aggregation of entail/contradict signals) used to operationalize "covers" vs. "challenges." (Section 3.3.2)

- $\vec{a}$**:** Target sentence-level discourse codes (control signals) specifying desired structural roles for planning/-generation. (Section 4.2.1.1)
- $H,\ \hat{H}$**:** LM hidden states and their controlled perturbations under Hidden-State Control (HSC) to emphasize desired roles during generation. (Section 4.2.3.2)
- α, β**:** Sentence-level beam search mixture weights trading off generator likelihood (fluency) and labeler guidance (structural conformity). (Section 4.3.2)
- γ**,** $\overline{a}$**:** Classifier-Free Guidance (CFG) strength and complementary "negative" labels for steering generation away from undesired properties. (Section 4.4)
- $m(i,j) = b^{|i-j|}$**:** Discount prior attenuating label influence by positional distance for localized structural control. (Section 4.2.2)
- w**:** Sliding-window half-width controlling the local context considered when enforcing structure. (Section 4.2.2)
- s_{t+1}**,** $\Delta(s_t, s_{t+1})$**:** Successor version and observed change summary used by emissions/intention models to infer edit types/intentions. (Sections 5.2, 5.2.1.3)
- $\mathcal{A}$**,** $a_{t,ij}$**:** Edit-intention label space and per-(version/sentence-pair) latent intentions in revision modeling. (Section 5.3)
- $E(\Delta_t \mid s_t, s_{t+1})$**:** Emission/observation estimator mapping version deltas to measurable edit types for supervision. (Section 5.2.1.3)
- $y^{(1..3)},\ c_{t,k},\ b(\cdot)$**:** Task targets for three predictive setups (existence of next version; edit count bins; local outcomes), per-type edit counts, and the binning function. (Section 5.2.3.1)
- $\mathrm{Sim}_{\mathrm{asym}}$**:** Asymmetric sentence similarity robust to merges/splits used in sentence alignment graphs for revision pairing. (Section 5.2.1.3)
- $\phi(\cdot)$**:** Word/embedding similarity function employed in alignment/scoring modules for linking sentences across versions. (Section 5.2.1.3)
- $p(l{=}\textbf{FactUpdate} \mid s_i, D)$**:** Per-sentence factual-update likelihood within document D, enabling selective behaviors (e.g., abstention when evidence may be stale). (Sections 5.3.4, 5.3.5)

Definitions

- **Emulation Learning (EL)**: Framework for modeling complex, creative tasks outlined in this dissertation. The approach is to learn policies by recovering latent actions/trajectories from end-states g via an *inverse model* $q_\theta(a|g)$, and then to train a policy model $\pi(a|x)$ from starting states to: (1) learn human rewards and objectives, preserving them in agentic processes (2) reach goal states g that are similar or improved from observational data; (3) match human distributional signatures across tasks (selection, sourcing, structuring, editing). (Sections 1.2, 2.2, 3, 4.1, 5.1)

- **Horizon-1 setting**: In reinforcement learning, the horizon refers to the number of steps an agent considers into the future when making decisions. This can be a fixed, finite number of steps or an infinite duration. In *news-finding*, we reduce of state-action trajectory planning to *one* step – predicting the newsworthiness of a piece of text, enabling simpler inverse modeling and evaluation. (Section 2.2)

- **Observation / emission channel**: The measurable "footprints" of actions (links, pairwise layout preferences, sentence operations) used to supervise inverse modeling and to evaluate learned policies. Specifically, an *emissions channel* is the mechanism by which an underlying, unobserved (or hidden) state generates an observable output. An observation channel is a broader term for how a piece of information or data is collected. The concepts are central to models that analyze systems where the direct cause is hidden, and only its effects can be seen, notably the Hidden Markov Model (HMM).(Sections 2.2.3, 2.3.3.1, 5.2.1.3)

- **Reward Function**: Clarifies the latent desirability of actions. $R(s_t, a_t, s_{t+1})$ specifies the immediate numerical feedback an agent receives from the environment after taking an action a_t in state s_{t+1} and transitioning to a new state s_{t+1}. Under maximum-entropy views classifier log-odds can act as affine proxies when true rewards are unavailable. (Section 2.2)

- **Utility Function**: Clarifies the latent desirability of actions. The utility function, often represented by value functions like $V(s)$ (state-value function) or $Q(s, a)$ (action-value function), quantifies the long-term desirability of a state or a state-action pair. Unlike the immediate reward, utility considers future rewards, often discounted by a factor γ to prioritize immediate rewards over delayed ones. (Section 2.2)

- **Policy learning / prediction** Training $\hat{\pi}$ to imitate inferred actions or predicting action likelihoods (e.g., which sentences will be updated). Policy learning and prediction involve learning optimal actions (policies) or predicting outcomes under a given policy, often in reinforcement learning or decision-making contexts.

Policy prediction determines the value or outcome of a specific policy, while policy learning aims to find the best policy to achieve a goal, sometimes by first learning a world model to predict future outcomes and then using those predictions to optimize the policy.(Sections 2.2, 5.3.4)

- **Compositionality / Predictability**: Hypothesis that sources and discourse moves co-occur in structured, learnable patterns (set-level coupling), enabling models to predict missing/next sources and assess structured dependence. Evaluated with (i) an *ablation probe* that removes sentences attributed to a source and tests detection of the removal vs. a matched no-op, and (ii) a *NewsEdits probe* that predicts whether a new source will be added in the next version; these findings motivate set-aware selection objectives (e.g., submodular gains, DPPs) when choosing sources jointly. (Sections 3, 3.2.3; Fig. 3.4, Fig. 3.4b)

- **Pairwise preference model**: a statistical or machine learning framework used to predict the outcome of head-to-head comparisons between pairs of items. In our work, we convert homepage layout features into $p_o(x > x')$ and recovers a consistent latent ordering via $u_\theta(x, C)$. (Section 2.3.3.1)

- **Transitive utilities assumption**: a core principle from economics that is sometimes applied in reinforcement learning (RL), particularly in multi-objective or preference-based settings. It posits that an agent's preferences are consistent and can be represented by a single, real-valued utility function. For instance, assumes pairwise comparisons factorize a global ranking usable for supervision and evaluation. (Section 2.3.3.1)

- **Planner / executor** this paradigm in reinforcement learning (RL) involves a hierarchical approach where a high-level planner and a low-level executor collaborate to achieve complex tasks. This architecture is particularly beneficial for long-horizon problems and sparse reward environments, where a single agent might struggle with credit assignment and exploration. In our case, the planner sets discourse/narrative goals and an executor issues concrete retrieval or writing actions. (Section 3.4, 4.2, 4.3)

- **Options / Semi-MDP** Options, as a form of temporal abstraction in reinforcement learning, create a Semi-Markov Decision Process (SMDP) where decisions are made over temporally extended actions ("options") rather than single steps. High-level steps are represented as single abstract actions (e.g., the *Get-Source* action, a_t, which constitutes actions: *Identify Need*→*Retrieve Source*→*Obtain Information*). (Sections 3.2.3, 3.4)

- **Submodular maximization** involves finding the best subset of items from a larger set to maximize a function that exhibits a "diminishing returns" property, meaning the benefit of adding an item decreases

as more items are included. Used when selecting multiple sources jointly to balance coverage and diversity. (Section 3.2.3)

- **Determinantal Point Processes (DPPs)** are probabilistic models that define a probability distribution over all possible subsets of items, specifically favoring diverse subsets over redundant ones by using determinants to model negative correlations. Connected with submodular maximization because the problem of finding the most likely subset in a DPP—known as the Maximum a Posteriori (MAP) inference—can be formulated and approximately solved as a submodular maximization problem. (Section 3.2.3)

- **KL divergence**, also known as relative entropy, is a measure from information theory that quantifies how one probability distribution differs from a reference distribution. In our case, we used it to compare discourse mixtures or roles across policies π^* and $\hat{\pi}$. (Section 3.4.3.3)

- **Bootstrap significance** a method for conducting hypothesis tests that does not rely on assumptions about the data's underlying distribution. Instead, it uses resampling with replacement from the observed data to generate a simulated sampling distribution for a test statistic. Used here to estimate confidence in retrieval or prediction gains. (Section 3.4.3)

- **Ablation study / probe** A kind of hypothesis testing that removes experimental conditions, factors or modalities (e.g., policy text, meetings) to quantify their contribution to performance in the overall task. (Sections 2.2.4, 3.2.3)

- **Inverse RL / Offline RL concerns** Identifiability issues (multiple rewards explain behavior) and support mismatch when learning only from logged data. (Section 3.2.3)

- **Sparsity / locality / stability (assumptions)** Assumptions that a small, local set of cues often determines intentions and that labels remain stable enough to be learnable. (Section 5.3.2.2)

- **Multitask learning** a subfield of machine learning that trains a single model to learn multiple related tasks at the same time. By simultaneously learning tasks with a shared representation, the model leverages common knowledge and correlations among them, which can lead to better performance and more efficient learning than training separate models for each task. (Section 5.2.3.2)

Glossary Terms

Model-Specific Terms

- **PRM (Probabilistic Relational Model)** Factorization of $P(l \mid g, x)$ through auxiliary attributes to improve linking. (Section 2.2.3)
- **Linking function** a function, $M_\psi(x, g)$, to determine if nodes on a graph should be linked. In our case, we aligned events and artifacts; aggregated non-matches imply $a{=}0$. (Section 2.2.3)
- **Recall of channel** $R(x)$ Ability of an observation channel to capture *all* true positives; if the recall is high, this means if an event is *not* measured positive, it is negative. In our case, we used this to measure the probability that coverage of event x yields a detectable g. (Section 2.2)
- **Model abstention** Strategy of an LLM answering a question to *not* answer a question if it's evidence is likely stale or contradicted by imminent updates. (Section 5.3.5)
- **Homepage layout parsing** Extract article "cards" from screenshots/HTML. An "article card" is all the text (e.g. headline, summary, picture, link) associated with a single article on a homepage. (Sections **??**, 2.3.3)
- **DOM-tree bootstrapping** A heuristic for detecting *full* article cards on a homepage. We detected all <a>'s, and traversed up the HTML to obtain the maximal-sized subtree still containing a single <a>. (Section 2.3.3)
- **Detectron2 (ResNet-101+FPN)** Classic computer vision detector model. In our case, trained on bootstraps for robust card localization. (Section 2.3.3)
- **L1 loss** Sparisity-inducing loss for linear/logistic regression. (Section 2.3.3)
- **OCR + YOLO screening** Quality control for screenshots and HTML conformance. (Section 2.3.3)
- **SingleFile** a library that captures an HTML page as a single file (e.g. all associated style sheets, assets, are included).
- **Internet Archive / Wayback** Web archives that snapshot pages online and preserve them. (Section 2.3)
- **TF–IDF, BM25** Two classical methods for embedding text as sparse vectors. Used for sparse retrieval

baselines. (Sections 2.2.3, 3.4.3)

- **SBERT / OpenAI embeddings** Two modern methods for generating dense embeddings for linking and retrieval. (Section 2.2.3)
- **DPR** Supervised dense passage retrieval for training embedding spaces to retrieve relevant documents from queries. (Section 3.4.3)
- **Coreference resolution** Canonicalization of entity mentions/pronouns. (Section 3.2.1)
- **Doc-level NLI** A method developed to aggregate sentence-pair NLI measurements into document-level signals (*entails/contradicts/neutral*). (Section 3.3.2)
- **BigBird / Longformer / LED** Long-context transformer-based embedding architectures for detection and intention tagging. (Sections 3.2.1, 5.3.2.2)
- **Interleaved retrieval** Using LLMs to generate queries to retrievers, analyze documents, and issue followup queries. (Section 3.4.3.2)
- **Planned interleaved retrieval** Interleaving with discourse-aware planning; ways of projecting queries into the future. (Sections 3.4.3, 3.4.3.2)
- **Re-ranking** Reordering retriever results based on discourse intent. (Section 3.4.3.2)
- **SFR-Embedding-2_R** A very large transformer-based embedding model used for retrieval. (Section 3.4.3)
- **Sentence alignment** Determining when two sentences contain substantially the same facts, information and intent. In our case, used to link sentences across *article versions*. (Section 5.2.1.3)
- **Bipartite matching graph** After linking sentences, the bipartite graph over article versions determines when sentences were edited/added/removed in article updates. (Section 5.2.1.3)
- **Asymmetric matching similarity** A sentence-matching algorithm we developed with optimal performance. (Section 5.2.1.3)
- **Hungarian matching** an efficient combinatorial optimization algorithm that finds an optimal assignment in a weighted bipartite graph. (Section 5.2.1.3)
- **BLEU / n-gram overlap** Sentence overlap by measuring exact word-level matches. (Section 5.2.1.3)

- **Word embeddings** Dense vectors generated per word for similarity and prediction (via models like RoBERTa). (Sections 5.2.1.3, 5.2.3.2).
- **Contextualization layer** A lightweight Transformer over sentence embeddings to add contextual information. (Section 5.2.3.2)
- **Event / quote detection** Pipelines to detect events and quotations/sources in sentences. (Section 5.2.2)
- **News discourse model** Assigns MAIN/CAUSE/DISTANT roles. (Section 5.2.2)
- **Argumentation features** Capture argumentative structure. (Section 5.3.2.2)
- **NLI (textual entailment)** Features capturing entailment/contradiction. (Section 5.3.2.2)
- **LDA (topic modeling)** A classic, unsupervised approach to latent variable modeling that discover latent "topics" underlying a collection of documents *topics*. (Section 5.2.3.5)
- **Logistic regression (TF–IDF)** A classical and simple text classification approach based on frequency-weighted word-counts. Sparse baseline for $\pi(a \mid x)$. (Section 2.2.4)
- **GPT3-Babbage (fine-tuned)** Early GPT3 model, `Babbage` was smaller than `Curie` and `Da Vinci`. We used them to study pretraining as well as finetuning in multiple experiments. (Section 2.2.4)
- **GPT-3/4 variants** A modern GPT model, available for performing zero-/few-shot and fine-tuned variations. (Sections 3.2.1, 3.3)
- **LLaMA-3-8B / Llama 3.1 / Mixtral / Command-R** Open-source large language models. Used primarily to generate text, process text data by making decisions, or, in our case, as planning/normalization models. (Sections 3.3, 4.3.1.2, 4.3.4)
- **DistilBERT / RoBERTa / FLAN-T5** Standard text models that project text into high dimensional embeddings. They are used for many different tasks, including retrieval, classification (with a head) and, in our case, pairwise comparisons. (Sections 2.3.3.1, 4.2.3.3)
- **PEFT** Parameter-efficient fine-tuning; an efficient method for tuning large language models with limited data and minimal computing power. (Sections 2.3.3.1, 4.3.4)
- **PTLM, GPT-2-base** Pretrained Language model. In our case, used to provide naive word likelihoods and

embeddings. (Sections 4.2.3.2, 4.2)

- **Oracle trial** A trial of a multi-task experiment where at least step is presolved with ground truth. Used to provide an upper bound. In our case, we gave gold supervision indicating whether an update happened, used to bound abstention strategies. (Section 5.3.5)

- **Human upper bound** Expert performance used to contextualize model scores and task difficulty. (Sections 5.2.3.6, 5.3.4)

- **Control codes / discourse labels** $\vec{a}$, or actions in our structural generation experiments (Chapter 4). A set of sentence-level structure control codes (e.g. "Provide Background"). (Section 4.2.1.1)

- **Emulation loss** L_{emul} A distributional distance between schema-level summaries of trajectories (e.g., role mixtures) for model vs. human behavior. (Section 3.4)

- **Local-only / Past-aware / Full-sequence** Structural awareness regimes for label modeling: specifies how much of the control code/action trajectory the state-transition model is made aware of before generating s_t. (Section 4.2.2)

- **Hidden-State Control (HSC)** A method for sequentially controlled generation. Perturb hidden states $H \to \hat{H}$ to upweight desired labels. (Section 4.2.3.2)

- **Direct-Probability Control (DPC)** A method for sequentially controlled generation. Multiply LM and labeler scores to steer next tokens. (Section 4.2.3.2)

- **Editing (mask-and-infill)** A method for controlling text generation. Performing masking and infilling (i.e. editing) on generated text to increase the likelihood that a label applies to the text. (Section 4.2.3.3)

- **Sentence-level beam search** Mix generating and label-scoring each sentence-generation step to optimize the sequence of sentences. (Section 4.3.2)

- **CFG (Classifier-Free Guidance)** A constrative sampling method for upweighting the effect of the prompt on the generation. Subtract from the prompt-conditioned next-token distribution the unconditioned distribution (Section 4.4)

- **Negative prompting** Use $\bar{a}$ to steer away from undesired attributes/actions. (Section 4.4)

- **Bradley–Terry / Thurstone / Plackett–Luce** Classical pairwise/listwise preference formulations. (Section

2.3.1)

- **DAgger** Interactive aggregation of expert corrections to counter compounding errors in sequential decision-making. (Section 3.2.3)

- **Temporal hold-out** Train/test split by time to emulate deployment and reduce leakage from future events. (Section 2.2.4)

Journalism & Newsroom Practice Terms

- **Newsworthiness** Human judgments on how important, relevant and interesting a piece of information is to a reader.

- **Newsworthiness prediction** A machine learning designed to test a model's ability to predict how newsworthy event x is. (Sections 2.2, 2.3)

- **News values** Normative criteria guiding coverage decisions made by journalists. (Section 2.2)

- **News-finding** Applying newsworthiness predictions across many events x in order to find candidate events/policies for coverage. (Section 2.2)

- **Homepage preference signals** Layout decisions made by homepage editors (position/size/graphics) that encode how newsworthy or salient they believe a story is. (Section 2.3)

- **Context / competitor set** C Editorial choices are relative (e.g. one day may have more news stories, or more important breaking news). Modeling C captures set-effects where prominence depends on co-present items. (Section 2.3.1)

- **Page One / homepage meetings** Editorial meetings at *the New York Times* to set daily priorities. (Section 2.3)

- **Above the fold / Page-A1** High-importance positions in the newspaper for the most newsworthy articles. (Section 2.3)

- **Article card** Visual block housing a story, summary, picture and link; on homepages. (Sections 2.3, 2.3.3)

- **San Francisco Board of Supervisors (SFBOS)** Local government we studied; produces policies and announcements and outlet for labels. (Sections 2.2.2, 2.2)

- **San Francisco Chronicle (SFChron)**: a local News outlet in San Francisco, California (Sections 2.2.2, 2.2)
- **Public comment** A period during a city council meeting when members of the public are allowed to ask questions and make comments about legislation. (Section 2.2.4)
- **Policy item** A motion, bill, amendment, settlement, law; any other piece of text conveying decisions about government. (Section 2.2.3)
- **Source-finding** A task testing a machine's ability to identify narrative needs in a story → find relevant sources →, obtain information from the source. (Section 3)
- **Source (informational)** Person, document, record, observation or database contributing facts (includes explicitly mentioned and implicit sources). (Section 3.2)
- **Attribution** Linking a sentence in a news article to one/more sources that provided information for that sentence (explicit/implicit). (Section 3.2)
- **Source channels** The style in which the information is provided by the source and how it is conveyed in the news article. IncludeS: Direct/Indirect Quote, Statement/Speech, Email/Social, Published Work, Lawsuit/Court, Proposal/Order/Law, Price Signal, Direct Observation, etc. (Section 3.2.1)
- **Press release (PR)** An announcement or document authored by a public or private organization designed to be covered by a news outlet. (Sections 3.3, 3.3.2)
- **Contrastive summarization** News coverage that both contextualizes and challenges a PR. (Section 3.3.2)
- **Angle** The lens/idea a journalist pursues on a PR. (Sections 3.3, 3.3.3.2)
- **Creativity (1–5)** In our case, defined as how *different* sourcing or angle decisions are from the original PR. More creative news articles are more different from PRs. (Section 3.3.3.2)
- **Primary/secondary sources** How important a source is to a central narrative in a news article. (Section 3.2.2)
- **Article versions / updates** Every time an article is *republished* to the same URL, we can collect a new article version. (Section 3.2.2)
- **Beats / coverage types** Different topics or areas of coverage in a newsroom; usually with a dedicated

reporter, or a consistent tempo of coverage (e.g. a "police beat" covers police activities). (Sections 3.2.3, 3.4.2)

- **Breaking news** news articles that cover events that are updating very quickly. We study norms on uncertainty, updating, verification under time-pressure. (Section 5.2.2)
- **Section / beat effects** Topic/section patterns in predictability and edit mix. (Sections 5.2.3.5, 5.3.3)

Communication / Discourse & Narrative Terms

- **Discourse structure** Functional organization of sentences toward an argumentative purpose. (Sections 4.1, 4.2.1.1)
- **Macro-structure** Global organization aiding compression/navigation/recall. (Section 4.1)
- **Narrative schemata** Canonical arrangements improving recall/coherence. (Section 4.1)
- **Topicality** Degree of on-topic content relative to headline/source. (Section 4.2.6)
- **Introductory elements** Opening/scene-setting roles (DiscoSum schema). (Section 4.3.2.1)
- **Contextual details** Background elaboration (DiscoSum schema). (Section 4.3.2.1)
- **Event narration** Core event description (DiscoSum schema). (Section 4.3.2.1)
- **Engagement directive** Reader-engaging/eliciting role (DiscoSum schema). (Section 4.3.2.1)
- **Discourse roles (sources)** Functions such as *Main Actor*, *Background*, *Counter*, *Expert*, *Data*, *Confirmation*, *Analysis*, *Broadening* (plus anecdotal/subject variants). (Section 3.4.1.2)
- **Centrality** High/Medium/Low importance of a source. (Section 3.4.1.2)
- **Stance** Support/oppose/neutral posture of a source. (Section 3.4.2)
- **Contextualization vs. challenge** "References/entails" vs. "contradicts" relations for effective coverage. (Section 3.3.2)
- **Partial order** $\prec$ Weak source ordering induced from structure/priors. (Section 3.2)

- **Equifinality** Many trajectories can yield the same g. (Section 3.4)
- **Narrative arc** Structured progression of a story over versions. (Sections 5.1, 5.2.2)
- **News discourse roles (EDA)** Main, Cause, Distant roles in news prose. (Section 5.2.2)

Cognitive Science Terms

- **Emulation (learning)** Reproducing outcomes/goals without copying actions step-by-step. (Section 5.1)
- **Imitation (learning)** Copying observed actions directly. (Section 5.1)
- **Ghost condition** Agent hidden; only apparatus changes observed—isolates emulation. (Section 5.1)
- **Planning–translating–reviewing** Classical cyclical model of writing. (Section 5.1)
- **Genetic criticism** Studying drafts/revisions as traces of the creative process. (Section 5.1)
- **Spatial organization as preference** Readers scan top-left; editors guide attention via spatial hierarchy. (Section 2.3)
- **Visual salience cues** Position, size, typography, imagery signal importance. (Section 2.3)

Evaluation & Metrics

- **ROC (AUC-ROC)** Area under ROC curve. (Section 2.2.4)
- **F1 / Micro–Macro F1** Harmonic mean of precision/recall; class- and instance-averaged variants. (Sections 2.2.4, 5.2.3.2)
- **Recall@10** Fraction of truly newsworthy items among top-10. (Section 2.2.4)
- **MRR** Mean reciprocal rank of first relevant item. (Section 2.2.4)
- **Cohen's κ** Inter-annotator agreement. (Section 4.2.6)
- **Kendall's τ** Rank correlation across outlets. (Section 2.3.3.4)
- **Human preference / ID accuracy** Expert judgments on recommendations and origin identification. (Section

2.2.4)

- **Label accuracy** Match of sentence to target discourse label. (Section 4.2.6)
- **Grammar** Human grammaticality/local coherence (1–5). (Section 4.2.6)
- **Logical flow** Human rating of story progression (1–5). (Section 4.2.6)
- **On-topic** Human topical relevance (1–5). (Section 4.2.6)
- **Perplexity (PPL)** Automatic fluency proxy and selection signal. (Sections 4.2.3.3, 4.2.7)
- **Diversity (n-grams)** Automatic diversity indicator. (Section 4.3)
- **ROUGE-L** Summary overlap metric. (Section 4.3.4.1)
- **FactCC** Factual consistency classifier. (Section 4.3.4.1)
- **AlignScore** Factual correspondence metric. (Section 4.3.4.1)
- **Match Score (MS)** Position-wise label match between predicted vs. target sequences. (Section 4.3.4.1)
- **LCS / Levenshtein distance** Longest common subsequence / edit distance over label sequences. (Section 4.3.4.1)

Discourse Schemas Introduced

VD3 News Discourse Schema (Section 4.5)

1. Lede: An opening hook that engages the reader and sets up the main event (may be an anecdote, question, or scene).
2. Main Event: The focal event or subject of the report (the precipitating, most recent, or central phenomenon).
3. Consequence: An event or outcome directly caused by, or immediately following, the Main Event.
4. Previous Event: A specific prior event that directly leads to or explains the Main Event.
5. Circumstances: The immediate world-state or situational context preceding the Main Event, not tied to a single event.
6. Secondary Event: An event occurring in parallel with the Main Event, often illustrative of a broader pattern or trend.
7. Historical Event: A more distal past event (e.g., weeks or longer prior) that remains causally or thematically relevant.
8. Expectation: Projected or anticipated future developments and their likelihood.
9. Evaluation: Journalist or source commentary assessing significance, quality, or implications of events.
10. Explanation: Causal or justificatory reasoning about why events occur or how they relate.

11. Verbal Reaction: Reported remarks or quotes that do not take on another discourse role.

Source Discourse Roles (Section 3.4)

1. Main Actor: The central entity driving or experiencing the focal event; supplies the core claims or actions.
2. Background: Context, history, definitions, or timelines that help interpret the event.
3. Counter: Opposing or alternative perspectives that challenge or complicate the main narrative.
4. Expert: Domain expertise offering technical explanation or informed interpretation.
5. Data: Quantitative evidence (statistics, records, indicators) substantiating claims.
6. Confirmation: Independent corroboration of previously asserted facts.
7. Analysis: Synthesis that draws connections and articulates causal or thematic takeaways.
8. Broadening: Framing that situates the case within larger geographies, domains, or trends.
9. Anecdotes: Illustrative first-person or vignette-style accounts.
10. Subject: A directly affected person or group embodying the story's stakes.

Source Centrality (Section 3.4)

1. High: Indispensable; removing it renders the article incomplete or misleading.
2. Medium: Important but not indispensable; the article remains coherent without it.
3. Low: Replaceable support; adds color or redundancy but is not required.

Stance (Section 3.4)

1. Authoritative: Provides first-hand or central knowledge to affirm a central claim.
2. Supporting: Affirms or strengthens the main claim or action.
3. Opposing: Disputes or undermines the main claim or action.
4. Neutral: Describes or contextualizes without taking a side.
5. Informative: Provides information without taking a stance.

Document-Level NLI (Section 3.3)

1. Entailment (Reference): Article content is consistent with, or directly supported by, the press release.
2. Contradiction (Challenge): Article content conflicts with or refutes the press release.
3. Neutral: No semantic commitment with respect to the press release claim.

Source Information Channels (Section 3.2)

1. Direct Quote: Verbatim speech attributed to a person or document.
2. Indirect Quote: Paraphrased content attributed to a person or document.
3. Statement / Public Speech: Formal remarks, briefings, or official statements.
4. Email / Social Media Post: On-the-record statements via email or platform posts.
5. Published Work / Press Report: Prior reporting or publications used as sources.
6. Proposal / Order / Law: Legal or policy instruments (bills, orders, regulations).
7. Court Proceeding: Filings, rulings, complaints, dockets, or courtroom statements.

8. Price Signal: Market or economic indicators used as evidence.
9. Direct Observation: Reporter's first-hand witnessing or recordings.
10. Other: Sourced content not covered by the above categories.

Sentence-Level State-Change Types (Section 5.2)

1. Addition: A sentence appears in the new version but not in the prior version.
2. Deletion: A sentence appears in the prior version but not in the new version.
3. Edit: A sentence changes surface form or specifics while preserving core meaning.
4. Refactor: A sentence is repositioned (moved up/down) to change emphasis or flow.

Edit-Intentions Ontology — Coarse Families (Section 5.3)

1. Factual: Alters represented world-state (event/source updates, corrections).
2. Style: Modifies presentation (clarity, tone, syntax) without changing substance.
3. Narrative/Contextual: Reshapes framing via background, analysis, or anecdotes.
4. Other: Housekeeping or non-semantic cases (e.g., alignment issues).

Edit-Intentions Ontology — Fine-Grained Elements (Section 5.3)

1. Event Update: Revises an event mention or its attributes (status, timing, details).
2. Quote/Source Added: Introduces a new source or quotation.
3. Correction: Fixes previously published factual information.
4. Style-Guide / Copyedit: Conforms to house style or improves readability.
5. Emphasis / De-emphasis: Adjusts salience, often via position or summarization.

6. Add Background: Adds contextual or historical information.
7. Add Analysis: Adds interpretation that connects facts or explains implications.
8. Add Anecdote: Adds illustrative, case-based narrative material.
9. Unchanged: No substantive intention beyond persistence across versions.
10. Incorrect Link: Alignment/linking error between versions (bookkeeping).

Local Update Outcomes (Per-Sentence Prediction Targets)

1. Deletion: Target sentence will be removed in the next version.
2. Edit: Target sentence will be revised while preserving core meaning.
3. Unchanged: Target sentence will remain the same.
4. Refactor: Up: Target sentence will move upward in position.
5. Refactor: Down: Target sentence will move downward in position.
6. Refactor: Unchanged: Target sentence will not change position.
7. Addition Above: New sentence(s) will be inserted above the target.
8. Addition Below: New sentence(s) will be inserted below the target.