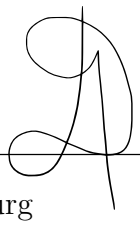
Distribution Agreement

In presenting this thesis as a partial fulfillment of the requirements for a degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis in whole or in part in all forms of media, now or hereafter now, including display on the World Wide Web. I understand that I may select some access restrictions as part of the online submission of this thesis. I retain all ownership rights to the copyright of the thesis. I also retain the right to use in future works (such as articles or books) all or part of this thesis.



Daniel Jacob Roytburg

April 9, 2025

Date

Generative Argument Mining: Pretrained Language Models are Argumentative Text Parsers

By

Daniel Jacob Roytburg

Joyce C. Ho
Advisor

Lauren F. Klein
Co-Advisor

Department of Computer Science

Joyce C. Ho
Advisor

Lauren F. Klein
Co-Advisor

Sandeep Soni
Committee Member

2025

Generative Argument Mining: Pretrained Language Models are Argumentative Text Parsers

By

Daniel Jacob Roytburg

Joyce C. Ho
Advisor

Lauren F. Klein
Co-Advisor

An abstract of
a thesis submitted to the Faculty of the
Emory College of Arts and Sciences of Emory University
in partial fulfillment of the requirements for the degree of
Bachelor of Arts with Honors
Department of Computer Science
2025

Abstract

Generative Argument Mining: Pretrained Language Models are Argumentative Text Parsers
By Daniel Jacob Roytburg

Argument mining is a natural language processing task which imposes a rhetorical structure schema on raw text, assigning labels to argumentative sub-phrases in text and connecting identified sub-phrases together with relations. Such labels may provide text analytics describing functional components of complete arguments like claims and premises, stylistic elements such as testimonies or facts, or some other defined schema. Argument mining is part of the structure prediction task family, using the formal definitions of entity and relation extraction in order to label specific decompositions of rhetorical structure. The task has important implications - not only for applied use-cases in areas such as social media analytics, jurisprudence, and group decision-making - but also for improvement on general structure prediction methods given the unique constraints imposed by the problem.

This thesis adopts the evolving capabilities of pretrained language models to cast argument mining as a generative task. Classical argument mining approaches use discriminative classifiers which produce a distribution of predictions for each individual token or sub-phrase in an input; this requires significant, task-specific architecture to process outputs of autoencoder language models. We consider whether task-agnostic generative language models can use a structured annotation scheme to mimic classification without additional architectural decisions. To this end, we adapt such a scheme which enables models to translate raw inputs to annotated text outputs, allowing efficient parsing and extraction for necessary labels. This decision affords the flexibility to not only introduce generative argument mining systems but also evaluate a wide variety of pretrained models, labeling schemas, training environments, and task configurations.

We explore the limits of these models across four key dimensions: labeling strategies for long-span entities, comparing full token spans, numerical identifiers, and abstractive summaries; encoder-decoder versus decoder-only architectures, contrasting their effectiveness in this structured prediction task; the necessity of fine-tuning for decoder-only models against few-shot in-context learning; and end-to-end extraction versus relation-only extraction, evaluating the impact of providing gold entity boundaries on relation identification. To assess model performance, we supplement traditional classification metrics with a set of criteria based on adherence to an augmented natural language output format, measuring reconstruction, entity, label, and format errors.

We find that generative models outperform current classification-based baselines by 10.41% for argumentative relations and 5.28% for argumentative component. Beyond this, our introduction of compliance allows a granular view of the failure modes of generative models in this context, revealing that while accuracy can be high, compliance errors, particularly in relation to entity coherence and label hallucination, remain significant challenges. Our exploration across model architectures suggests that while

larger decoder-only models exhibit strong in-context learning capabilities, fine-tuned encoder-decoder models can achieve competitive or superior performance, especially when data is limited. Furthermore, our investigation into labeling strategies indicates a trade-off between output length and parsing complexity with accuracy, highlighting the need for more robust methods for representing long-span argumentative units. These findings contribute valuable insights into the application of generative language models for argument mining, outlining both their potential and the key areas requiring further research and development to realize fully end-to-end, high-fidelity argumentative structure prediction.

Generative Argument Mining: Pretrained Language Models are Argumentative Text Parsers

By

Daniel Jacob Roytburg

Joyce C. Ho
Advisor

Lauren F. Klein
Co-Advisor

A thesis submitted to the Faculty of the
Emory College of Arts and Sciences of Emory University
in partial fulfillment of the requirements for the degree of
Bachelor of Arts with Honors
Department of Computer Science
2025

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Generative Structure Prediction: where are we now?

Generative, pretrained language models have exhibited fascinating capabilities in myriad tasks learned through an extensive pretraining process which operates unsupervised tasks over large spans of data. The recent success of large, pretrained language models over a variety of tasks has prompted their use as surrogates for binary or multinomial classification tasks typical to natural language processing.

One such task is known as *structure prediction*. Given a sequence of natural language tokens as input, the task family of structure prediction requires that a system accurately parse and identify pre-specified features of a structure. Such features may be grammatical, such as identifying parts of speech and semantic roles for nouns, verbs, and adjectives in a sentence [9]. More advanced features might incorporate tasks like named entity recognition, where certain noun phrases are identified as proper nouns with classes like "person" or "place." Still others might attempt to tie extracted entities together based on a set of semantically-defined relations. Often, the goal of structure prediction is to generate parseable metadata which can then be constructed

into semantic knowledge graphs such as the Semantic Web or Wikidata [92]. Such structures act as references to index otherwise unstructured text corpora to make information access, extraction and orchestration as efficient as possible.

Historically, such structure prediction tasks have served as benchmarks for the progress of statistical machine learning methods for natural language processing, facilitating the growth of the field through widespread benchmarks and ample training data for future methods. Many structure prediction tasks such as named entity recognition and semantic role labeling were declared as largely 'solved' through complex statistical ensembles that incorporated latent state embeddings for span prediction, such as Long Short-Term Memory Models (LSTMs) or other recurrent models [103]. With the introduction of early embedding models like GloVe and Word2Vec and, later, transformer-based architectures such as BERT [17], these classification tasks saw greater improvements. However, satisfactory performance required the construction of bespoke models which incorporated trainable embedding models as an input to a sequence of classification heads configured in a manner that corresponded to domain requirements.

Thus, as pretrained generative models demonstrated remarkable intrinsic representations of natural language and promising capabilities for use on downstream tasks with transfer learning, their incorporation as potential substitutes for cumbersome classification architectures came to focus. While many approaches attempt to treat generative models as classifiers by posing a classifier prompt and yieldng one- or two-token answers from models, further research sought to employ the long-form generation capabilities of pretrained language models (PLMs) by introducing structure prediction as a machine translation task which required an augmented, annotated instance of natural language [59].

Since then, research has revolved around optimizing the performance of generative models on structure prediction in conventionally defined tasks such as named entity

Input:

"You are a liar! I have been harassed for over 5 years by companies looking for someone I have never heard of because this person put a random string of our area code, local exchange and 4 numbers together (that turned out to be my number) on a loan that he then defaulted on. I have told them to stop calling, they have a wrong number, and guess what " 5 Years later, they are still calling!"

Output:

"[You are a liar! | **value**] [I have been harassed for over 5 years by companies looking for someone I have never heard of | **testimony**] [because this person put a random string of our area code, local exchange and 4 numbers together (that turned out to be my number) on a loan that he then defaulted on. | **testimony** | **reason** = *I have been harassed for over 5 years by companies looking for someone I have never heard of*] [I have told them to stop calling, they have a wrong number, and guess what " | **testimony**] [5 Years later, they are still calling! | **testimony**] !"

Figure 1.1: The Augmented Natural Language Framework, a method for encapsulating argument spans using generative markers, delimiters, and class attributes.

recognition and relation extraction. However, research has not extensively covered the same possibilities for domain-specific structure prediction, especially for tasks with unique properties. One such subtask, dubbed "argument mining," attempts to use foundational theories of rhetorical composition to model entire phrases or sentences as argumentative units which might support one another in a text. Argument mining carries distinct significance as a departure from traditional structure prediction tasks because its identified entities are much longer and its labels less objective than those used for tasks like semantic role labeling. Thus, argument mining presents a unique case study for evaluating the capabilities of generative pretrained language models for structure.

This thesis addresses the question of generative language modeling for argument mining by adopting the frame of translation into augmented natural language. In short, models are expected to re-produce input sequences with modifications that include in the input sequence relevant annotations to delineate the spans of argumentative entities, a class from a set of predefined types, and potential relations which span across argumentative entities.

1.2 Expanding the Scope: Beyond Conventional Benchmarks

The shift from traditional classification-based approaches to generative modeling for structure prediction marks a significant paradigm shift in natural language processing. While benchmarks like named entity recognition and semantic role labeling have historically driven progress, they often fail to capture the complexity and subjectivity inherent in real-world language understanding. Argument mining, with its focus on identifying and analyzing argumentative units, offers a compelling alternative. It pushes the boundaries of structure prediction by demanding the extraction of longer, more abstract entities and the recognition of nuanced relationships between them.

This thesis takes a bold step by applying generative language modeling to argument mining, a domain where the lines between objective classification and subjective interpretation blur. By framing argument mining as a translation task, we leverage the inherent generative capabilities of pretrained language models to produce augmented natural language that encodes argumentative structures. This approach not only challenges the conventional view of structure prediction but also opens up new avenues for analyzing and understanding the persuasive power of language.

1.2.1 The Four Dimensions

Our research is distinguished by its systematic exploration of four critical dimensions that influence the performance of generative models in argument mining. These dimensions provide a structured framework for investigating the strengths and weaknesses of different model configurations:

1. **Annotation Schema/Labeling Strategies**: We recognize that the length and complexity of model outputs can significantly impact performance. Therefore, we investigate strategies to optimize the annotation schema, ensuring that it

strikes a balance between expressiveness and conciseness. This involves exploring different ways to represent argumentative entities and relations, aiming for a schema that is both informative and efficient.

2. **Model Architecture and Family Selection**: The choice of architecture and intra-architectural model families play a pivotal role in determining performance. We consider two generative architectures and five families therein – encoder-decoder models, like T5 [68] and decoder-only models like LLaMA or GPT. This exploration aims to uncover the architectural nuances that contribute to or hinder effective argument mining.

3. **The necessity of Fine-Tuning**: Do pretrained models require careful fine-tuning to behave as structure predictors? We investigate the use of fine-tuning strategies, comparing against in-context learning strategies which might equally be capable of generating schema-compliant outputs and decomposing argumentative structures.

4. **Relaxed Evaluation Metrics**: Recognizing the inherent subjectivity of argument mining, we introduce relaxed evaluation metrics that evaluate model performance exclusively on relation-based subtasks, enabling us to determine models' ability to articulate predictions within defined entity spans and providing a more nuanced understanding of their performance.

1.2.2 Compliance Metrics: Bridging the Gap

A key contribution of this work is the introduction of "compliance metrics." These metrics address a fundamental challenge in using generative models for classification-like tasks: the potential for outputs that deviate from the expected input sequences. By measuring the degree to which model outputs adhere to the annotation schema, compliance metrics provide valuable insights into the models' ability to generate

structured, parseable annotations. These metrics serve as essential heuristics for evaluating the effectiveness of generative models in argument mining and highlight the importance of schema adherence in ensuring the usability of generated annotations.

1.3 Thesis Structure

This thesis proceeds as follows:

Background: first, a cursory history is offered of argument composition from the perspective of literary and philosophical theories of persuasive language composition, exploring the seminal works of the scholars who established formal theories of discourse and language. This background then covers the methodological history of structure prediction as the methodological abstraction behind argument mining by covering recent developments and other important inflection points in natural language processing. Finally, the domain and the method are synthesized through a query into the argument mining literature, considering the tradition of the field in its own right while incorporating its fundamental links to both rhetorical theory and parsed structure prediction.

Approach: drawing from the relevant background, we define the formal definitions necessary to understand argument mining as a structure prediction task, as well as for framing our approach to applying generative models for parsing. We define the scope of our core contributions as a roadmap for experiments, select datasets relevant for the task, and outline the horizon line for future endeavors using our framework. The approach concludes with definitions of the evaluation metrics necessary for our analysis.

Experiments: we list the technical specifications for our experiments to ensure their reproducibility, defining clearly the scope of tests done using the approach and framework outlined.

Analysis: we distill findings across the gamut of our analysis, which incorporates over 50 experiments. We use these findings as the basis for examining future directions in improving generative argument mining capabilities.

1.4 The Bitter Lesson Revisited: Scaling vs. Domain Expertise

Our experimental findings, which demonstrate that pretrained language models can outperform state-of-the-art classification-based approaches, resonate with Richard Sutton's "Bitter Lesson." This lesson underscores the power of scaling computational resources and leveraging general-purpose learning algorithms, even when domain-specific knowledge seems essential. In the context of argument mining, our results suggest that the sheer scale and generalization of pretrained language models can compensate for the lack of specialized architectural designs.

However, we also recognize that domain expertise remains valuable. Our research aims to strike a balance between the benefits of scaling and the insights gained from domain-specific knowledge. By carefully designing our experiments and developing compliance metrics, we seek to harness the power of pretrained language models while acknowledging the unique challenges and opportunities presented by argument mining.

Chapter 2

Background

2.1 Introduction

Automated argument mining presents a fragmented yet increasingly significant field within natural language processing, aiming to scaffold the structure of argumentative discourse from unstructured text. Its foundations trace back to early efforts in discourse analysis, but the field has gained traction in the past decade due to the proliferation of large, diverse text corpora and advances in machine learning. At its core, argument mining seeks to identify both the argumentative components—claims, premises, and conclusions—and the relations that bind them, such as support or attack. The evolution from component identification to full discourse parsing reflects the growing ambition to model entire argumentative structures rather than isolated fragments.

Methodologically speaking, the field currently finds itself in a state of transition towards the wave of methods which exploit the generative pre-trained capabilities of large language models. Current state-of-the-art models employ large language models of many varieties, but often framed in a classification-based environment by isolating one component of argumentative data mining and testing a model's capability to

determine a type of argument or relation between arguments. Consequently, many of the systems produced with this philosophy use pipelines composed of many different models which exhibit cascading errors by sequential accumulation [64] or lack a unified training structure.

Contemporary surveys emphasize the incremental success of large, pretrained language models on disjoint sub-tasks like claim detection and stance classification [81, 16]. However, these works often overlook the broader capacity of language models to jointly generate argumentative components and their interrelations. Even when such strategies are employed, they rarely engage decoding strategies, opting instead to devise task-specific architectures such as bi-affine parsers and dependency trees based on the logits of the outputs of autoencoder architectures.

In light of this, joint argument extraction—a paradigm where models predict argumentative spans and their relations in a unified fashion—has emerged as a natural extension of generative systems' semantic expressiveness. We highlight the paradigm shift towards trainable, generative systems within argument mining research, arguing that these models sidestep the need for a pipeline by implicitly structuring discourse in a single, auto-regressive pass.

In this section, we will broach the subject of argument mining from three distinct angles:

1. **Communication and Rhetorical Theory**: we follow the work of Aristotle and Stephen Toulmin to understand the philosophical foundation of formal argumentative composition, a precursor to contemporary argument mining approaches.

2. **Argument Mining**: we apply this philosophy of rhetoric to approach the existing literature base of computational argument methods, examining the recent growth of the argument mining space as well challenges intrinsic to its murky definition. Here, we can define the goals of our experiments in argument

mining. We also justify their potential to improve existing methods and exemplify the importance of argument mining as a methodological case study in structure prediction tasks.

3. **Structure Prediction in Natural Language Processing**: finally, we introduce the technical abstraction of structure prediction. Here, we describe how the task of predicting structures within natural language evolves through the principles of logical communication. Here, we declare a formal problem definition and explore the necessary sub-tasks known as **entity recognition** and **relation extraction**, along with the contemporary methods which enable performance in these environments.

2.2 Communication and Rhetorical Theory

2.2.1 Introduction

Long before the advent of modern computing systems (and certainly before their application to natural language), philosophers of antiquity such as Aristotle composed rational hierarchies of argumentation. They delineated between overarching claims and specific premises - based in observed reality, logical formulation, or deductive reasoning - that justify a claim.

In our current time, the fundamental principles laid out by Aristotle have been refined and taxonomized by thinkers like Stephen Toulmin. Toulmin's model is of use to computational linguists, as it served to be a prescient ontology for modern annotation structures. Moreover, the model offers an excellent pedagogical framework for appreciating rhetorical composition. Toulmin's research spurred interest in the field of argumentative theory and informs contemporary theories of formal verification and social discourse alike. While most argumentation data has too much noise to

neatly fit into his classification, it serves as a meaningful computational bridge into annotation decisions made by authors in the argument mining space.

2.2.2 Aristotlean Logic

It is Aristotle's work – including *Prior Analytics*, *Posterior Analytics*, and other writings from *The Organon* which established the earliest fundamentals of argumentative theory. Aristotle defined the earliest principles of propositional logic, where claims were broken down into subsequent sub-claims which were either self-evident or justified recursively in deeper decompositions [78]. His statement "All humans are mortal. Socrates is a human. Ergo, Socrates is mortal" is among the first recorded examples of **deductive syllogism**, which contains a major, empirical premise (all humans are mortal) and a minor, particular premise (Socrates is a human) to establish its claim [78].

In addition to his categorical syllogisms, Aristotle developed a nuanced theory of **modal logic**, which accounted for necessity, possibility, and contingency in propositions. In *Prior Analytics*, he distinguished between syllogism-by-assertion, where premises express simple truths, and modal syllogisms, where premises involve modal qualifiers such as necessity or possibility. For example, the statement "It is necessary that all humans are mortal" introduces a layer of modality that affects the logical structure of the argument. This investigation into the temporal and modal dimensions of logic marked one of the earliest systematic treatments of these concepts, influencing later developments in formal logic. These are the predecessors to the modern definitions of inverse, converse, and contrapositive. It is this concept of modality which forms the bedrock of logic across many domains, from mathematical proofs to amicus briefs to systems and decision theory.

Aristotle considered not only formal logic but public persuasion as well. He dubbed the terms "ethos", "pathos", and "logos" as **modes of persuasion** when

seeking the agreement of public audiences. These terms, which refer to an orator's sense of authority, their appeal to emotions, and their appeal to reason, enmesh the fundamental logic described above with key discursive components that connect to the psychology behind persuasion. Aristotle's theory of ethos, pathos, and logos continues to have a significant impact in modern-day communication, especially in areas like advertising, politics, and media. Politicians and public figures use ethos to establish trust and authority, pathos to connect emotionally with voters, and logos to present data and logical arguments to support their positions [65, 57]. In advertising, marketers combine emotional appeals with logical arguments and trustworthy spokespersons to persuade consumers. This enduring framework remains a cornerstone of persuasive communication, helping to shape how we respond to arguments, make decisions, and influence others in today's media-saturated world.

2.2.3 The Toulmin Model

The 20th century philosopher and communication scholar Stephen Toulmin (1922-2009) played a pivotal role in advancing a formal structure of argumentation. His seminal work *The Uses of Argument* establishes his graph-based model for processing any given argument [86]. Toulmin classified arguments as six fundamental entities:

1. **Claim**: a fundamental, falsifiable conclusion proven through the logical support of the related entities in the argument graph. Example: "The city of Atlanta should invest more in public transportation infrastructure."

2. **Grounds/Data**: empirical or factual evidence, usually collected at-scale or through self-evident observation, which suggests the truth of the claim. Example: "Studies prove that cities with efficient public transportation have lower traffic congestion, reduced air pollution, and higher economic output expressed in GDP."

3. **Warrant**: a fundamental and unique component to the Toulmin model, the warrant *connects* the grounds to the claim by offering reasons explaining the evidence that data offers. The warrant provides the argument with a causal component, as without it grounds are often based on correlations and confounded by other factors. Example: "by offering an alternative to motor transport, public transportation decreases the volume of vehicles on the road, offsets carbon emissions, and offers economic mobility for those who cannot afford a car."

4. **Backing**: related to Aristotle's notion of ethos, backing strengthens the believability of the warrant through details such as proof of methodological robustness, qualifications of publishing authors, or new data that isolates the causal mechanisms in the warrant. Example: "Studies from the Georgia Transit Association use tests of statistical significant to isolate increases in GDP for counties that invest in public transportation infrastructure."

5. **Qualifier**: used to limit the scope of the initial claim, qualifiers justify establish distinctions between the general case and the particular claim to prove. Example: "While not all counties in Georgia benefit from public transportation projects, urban and suburban districts in the Perimeter show immense promise for modernization of MARTA train and bus lines."

6. **Rebuttal**: an answer to potential counterarguments. Example: "Some downplay the economic effects of public transportation, citing the cost of an initial investment. However, higher levels of urban productivity, costs saved due to less vehicular transport and long-term project scaling ensure that such a project could pay for itself."

Due to its clear, diagrammatic structure and graph-like representation, the Toulmin model is a widely taught method of systematic argumentation that has prompted spin-off structures such as the Rogerian argument [83]. The Toulmin model has had a

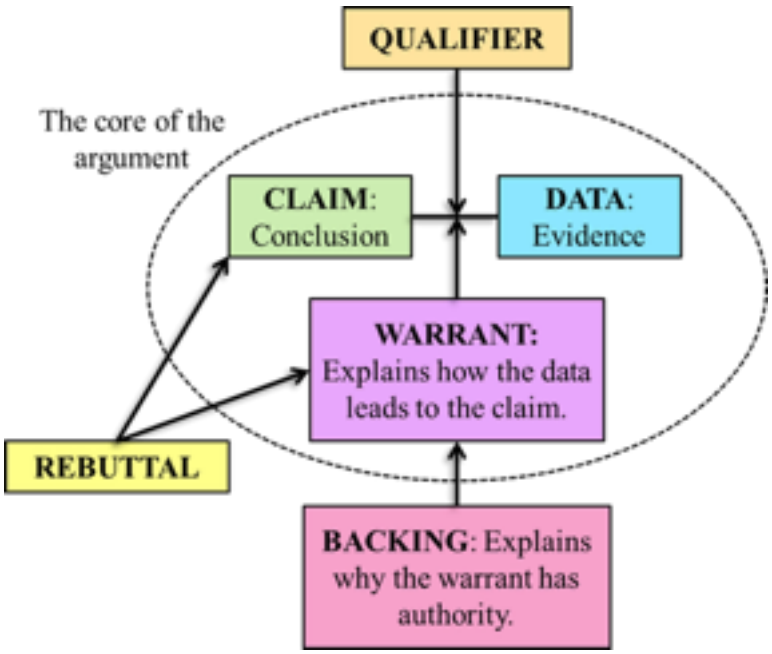


Figure 2.1: A network diagram of the Toulmin model, featuring major components and their relations

significant impact across multiple fields, including law, politics, education, and artificial intelligence. While critics disagree over the necessity of covering each of its components to produce a sound argument, the extensive nature of its classification offers a strong, near-exhaustive ontology for the decomposition of well-reasoned arguments.

In the context of natural language processing, the Toulmin model was identified as a superior hierarchical system for advancing sentence-wide parsers as early as 2009 [91]. These works operated at the intersection of artificial reasoning, formal verification, and argumentation software which precedes modern natural language capabilities. The Toulmin model has become a powerful influence for scholars of computational linguistics as it bridges the gap between rhetorical tradition and parsed structure. It continues to form the basis of annotation schema [93, 45].

2.2.4 Alternative Theorists and the Structured Language Problem

While the models proposed by Toulmin and Aristotle are among the most cited in the argumentation theory literature, they are certainly not the only proposed ontological structures (see [87, 88, 94]). In Antiquity, the statesman and lawyer Cicero also produced a six-point framework: *exordium* (introduction), *narratio* (statement of facts), *divisio* (delineation of arguments), *confirmatio* (proof and evidence), *refutatio* (refuting potential counterarguments), *peroratio* (conclusion)[43]. Due to his background as an orator and consul, his thinking centered on public persuasion at the expense of robust categorization, making his taxonomy less appropriate for formal compositions of argument. Meanwhile, the Rogerian argument [83] evolved from an empathetic background, attempting to bridge the understanding gap through attempting to view arguments dialectically.

2.2.5 Conclusion

As a task in computational linguistics, argument mining derives its philosophical roots from a theory of formal logic and composition millennia in the making. While humanity has long sought to impose a rational hierarchy over the fluid dynamics of rhetorical exchange, the empirical reality of written and spoken communication proves to be more dynamic, with arguments often oscillating between different functions and carrying non-argumentative text as well.

These strategies, however, form a basis with which one can mine the key positions in a paragraph and determine their interrelation. In ensuing sections, we will review argument mining as a distinct computational tradition in natural language processing, before unpacking the methodological abstraction behind argument mining – the task family known as structure prediction.

2.3 Argument Mining

2.3.1 Introduction

At the confluence of rhetorical composition theory and structure prediction (Section 2.4) sits the scholarship known as *argument mining* [42, 77]. With ever-larger collections of argumentative language becoming available due to the predominance of digital communication, ample opportunity exists to interpret the relationships between expressed opinions, factual evidence, and warranted argumentation. Such research enables at-scale interpretation of the major factors that determine opinions exhibited by users online. Argument mining assumes the challenge of long-context entity and relation extraction, where phrase- or sentence-long fragments of argumentative language form each unit. Research in argument mining grapples with the subjectivity inherent to structures mapped onto fluid, dynamic natural language. Successful argument mining systems are capable of typing sentences as premises and claims (or, depending on training data, with deeper labels such as values, policies, facts, references, etc.) and establishing typed relations (reasons, supports, attacks) between entities. As a result, argument mining holds significant potential for applications in fields such as computational social science, legal analysis, and automated fact-checking, where discerning the structure of reasoning is crucial.

The following sections will first differentiate argument mining from relation extraction with its sub-components before outlining a cursory history of explorations in argument mining, the field's adoption of statistical and neural architectures, and the growing interest in designing end-to-end trainable systems instead of pipelines. After covering key datasets that have emerged throughout this history, we will explore some successful approaches in recent history, drawing parallels between the representation encoding and dynamic encoding paradigms from structure prediction.

2.3.2 Subtasks of Argument Mining

Identifying Argumentative Discourse Units

Methods for argument mining historically resemble those for joint entity-relation extraction, following a similar history of pipeline-based subtask modeling and classification environments. Argument mining distinguishes itself from relation extraction by the expansion of entity spans from a few tokens (such as noun or verb phrases) to Argumentative Discourse Units [63] (ADUs), each roughly the span of a sentence, sub-sentence clause [27], or prosodic unit [31] (depending on definition). The broader scale of argument mining also lends itself to greater subjectivity in entity types and relation pairs, evidenced by lower agreement scores between annotators.

Identifying argumentative discourse units begins with a segmentation task to identify beginnings and ends of argumentative spans, followed by a classification task to subsequently identify the entity type of the segmented span. This is equivalent to the entity recognition/extraction task defined by structure prediction 2.4.4.

The classification taxonomies used to type argument entities vary widely by application, model function, and historical context. In classical rhetorical theory, for instance, Reynolds and Reynolds [70] produce a set of argumentative "stases", which may be classified as statistical, testimonial, anecdotal and analogical. Hoeken and Hustinx [32] build a similar taxonomy of individual examples, statistical evidence, causal explanations and expert opinions. Earlier, Fahnestok and Secor [24] type arguments as either fact, definition, cause, value, or action. More recently, argument types depart from classical theories of rhetoric in service of defining functions for computational modeling; these may fall along binary classes, notably verifiable/unverifiable statements [30, 62, 37].

Identifying Relations Between ADUs

Once argumentative discourse units are identified and classified, the next crucial subtask involves relation extraction, which aims to identify how these units interact. Relation extraction in argument mining differs from traditional relation extraction in that relations are inherently more abstract, often reflecting logical or rhetorical connections rather than concrete entity-entity interactions. This abstract nature makes relation annotation more subjective, leading to challenges in inter-annotator agreement.

Again, the types of relations identified between arguments vary by use case. Many relation types operate on a support/attack binary [69, 74, 19], indicating whether an argument supports or undermines another argument in the document. Recent research has also invested in modeling "undercuts"[100], or relations where an argument attempts to disprove the reason that a premise may support a claim, rather than refuting the claim or premise directly. Other relation types have been explored, such as *evidence* and *reason*, used by Park and Cardie [60, 61].

A final stage in argument mining involves discourse-level construction of knowledge graphs, where extracted components and relations are assembled into coherent argument graphs. These structures range from simple tree-like configurations to complex non-hierarchical graphs representing nuanced multi-perspective debates [93]. Some approaches employ pre-defined templates, while others utilize end-to-end machine learning models that infer the most likely graph structure given a corpus.

2.3.3 Key Datasets

Argument mining relies on annotated datasets to train and evaluate models that can identify and extract argumentative structures from text. Many different datasets exist which span across different functions, domain contexts, annotation schemata, and sizes. The scale of argument mining datasets is limited by the difficulty of hand-annotation

in data environments:

Cornell eRulemaking Corpus (CDCP)

Introduced by Park and Cardie [61], the CDCP dataset consists of 731 user comments from an eRulemaking platform concerning Consumer Debt Collection Practices. The corpus includes 4,931 elementary units categorized into five component types: fact, testimony, reference, value, and policy. Additionally, it annotates 1,221 support relations of types reason and evidence. This dataset is valuable for understanding how arguments are structured in public comments on regulatory matters.

Argument Annotated Essays Corpus (AAEC)

Developed by Stab and Gurevych [76], the AAEC comprises 402 persuasive essays written by students. Each essay is annotated with argumentative components such as major claims, claims, and premises, and relations labeled as support or attack. The corpus contains 751 major claims, 1,506 claims, and 3,832 premises, connected by 3,613 support and 219 attack relations. This dataset is widely used for research on argumentation structures in educational contexts.

AbstRCT

The AbstRCT dataset, introduced by Mayer et al. [48], comprises 6,000 abstracts from PubMed, annotated with argumentative components and relations. This dataset is particularly useful for studying argumentation in scientific writing, focusing on how researchers construct and present arguments in biomedical literature.

IBM's Project Debater

The Project Debater Corpus [71] is a large-scale dataset for Context-Dependent Evidence Detection (CDED), consisting of 547 Wikipedia articles across 58 debate

topics. In the 39 training topics, annotators identified 3,057 evidence segments from 274 articles, averaging 2.9 supporting evidence per claim.

The dataset categorizes evidence into Study (empirical results), Expert (authoritative statements), and Anecdotal (narratives). It played a key role in IBM's Project Debater, advancing automated evidence retrieval and argument mining.

Recent research has attempted to unify the various types of entities and relations defined in these datasets [25]. The heterogeneity of different annotation schemata make the task of generalization across datasets particularly challenging. Nevertheless, these datasets have played a crucial role in advancing argument mining research by providing annotated corpora for training and evaluating computational models.

2.3.4 A Brief History

The complexity of argument mining's subtasks necessitates a variety of modeling strategies, from rule-based heuristics to deep learning architectures. Traditional pipeline-based models separate segmentation, classification, and relation extraction, allowing modular improvements but risking cascading errors. End-to-end neural approaches, leveraging transformer-based architectures and structured prediction techniques, increasingly dominate recent efforts by optimizing for joint learning across these tasks. This section provides a historical overview of argument mining research, highlighting key datasets, methodologies, and innovations that have shaped the field.

Early History

The earliest approaches to argument mining precede the formal definition of the task and focus largely on the subtask of entity identification/classification within language. Unlike traditional relation extraction methods, argument mining research has employed statistical and machine learning-based techniques since its inception.

These earliest statistical techniques employ hand-crafted features as vectorized categorical and numerical data as model inputs. Some of the earliest work, done by Moens et. al [52, 51], adopts a simple argumentative/non-argumentative classification task on isolated sentences split from unstructured text corpora. They then apply a Naïve Bayes classifier over word couples, textual statistical features, and a bag-of-words representation of verbs. This follows a series of publications intended for legal audiences regarding the use of argumentative legal agents using rule-based methods extracted from manual annotation interfaces [90, 91]. The initial task definition applies a simple sentence-based segmentation strategy to produce Elementary Discourse Units (EDUs) and employs a binary classifier to filter for those units which are explicitly argumentative. Later, Moens et. al introduce a claim/premise classification task for units which are categorized as argumentative [50].

Parallel to this work, "argumentative zoning" [85] is framed as a segmentation and classification task geared towards scientific publications, using "background", "aim", "method", "result", "conclusion", "discussion" and "future work" as potential labels. This approach allows for the structured analysis of scientific texts by identifying the rhetorical roles of different sections, facilitating tasks such as summarization, citation analysis, and information retrieval.

The consideration of relations between arguments is not considered in the initial argument mining literature. While Mochales and Moens broach the subject of inducing relational structures by parsing classified claims and premises to build structure trees [51], the method relies exclusively on the results of the classification schema and introduces no statistical methodology to construct relations. Later work critiques this approach, introducing the question of relation extraction as an argument mining task and advocating for a relation-first approach [10].

Several early approaches respond to the call for relation-based argument mining (known as Argument Relation Identification). In one case, researchers propose manual

feature extraction to gather pointwise mutual information scores on key relational terms like “however” and “thus”, as well as noun features that count the number of shared nouns between two component structures [76]. These are inputs to a constrained optimization problem solved with Integer Linear Programming, achieving an aggregate F1 score of 75% on relations extracted from known components. Another paper [14] stands out as among the first approaches to explicitly incorporate deep learning. For any two ADUs, Cocarascu and Toni create separate embeddings with GloVe and LSTMs before merging the embeddings, using a dense feedforward network with a softmax classifier head to produce the final, multinomial classification of attack, support, or neither. This architecture produced impressive results, especially when the GloVe embeddings themselves were trained. It serves as a precursor to more recent transformer-based approaches that leverage contextual embeddings for Argument Relation Identification.

Going Deep

As large, pretrained language models demonstrated remarkable improvements in natural language understanding and structure prediction, they were increasingly used to tackle argument mining.

The first deep networks used in an argument mining setting relied on word embeddings generated from GloVe as inputs, processed with a recurrent network architecture, and classified with outputs from a feedforward network. Foundational work relies upon a bidirectional LSTM framework augmented with a convolutional neural network designed to capture dependencies from out-of-vocabulary words with character-level relations (BLC, or Bidirectional LSTM-CNN, for short). In Eger et. al, a BLC is used in two models: one to design a tagging classification scheme which uses cross-task tags to jointly model entities and relations, and another which separately models entities and relations, using TreeLSTM to generate many possible relations [20].

This work optimized the LSTM experiments conducted in [14], proposing the BLCC and LSTM-ER architectures with a set of ablations to determine the contributing factors in improved model architecture [11]. The results demonstrate that while both systems have relative advantages, the pipeline-based, weight-shared LSTM model outperforms in short-context cases due to the advantage of fine-tuning a model on two separate tasks while employing the same parameter sets.

Ye and Teufel [99] incorporate some of the first end-to-end systems with BERT-based biaffine dependency parsing, dividing tasks not only on the basis of entity and relation but also on identifying vs. labeling. The parser treats both component and relation argument tags as a dependency graph, representing the boundaries and labels of entities with BILOU tagging before passing to a biaffine parser to calculate dependency tags on relations as edges. The end-to-end system is jointly trained on a BERT encoder [17] before fine-tuning on separate FNN heads. BERT encodings provide significant performance improvements in both the component and relation classification environment, demonstrating the relevance of task-specific fine-tuning for large, pretrained language models. Further research has demonstrated how improving the underlying language model leads to outsized performance gains on the end-to-end task [53].

Subsequent advances in Argument Relation Identification have leveraged even more sophisticated transformer-based models, incorporating multi-task learning, contrastive loss functions, and external knowledge graphs to refine relational inferences. One key development was the application of cross-attention mechanisms to explicitly model the interdependencies between argumentative discourse units (ADUs). By encoding ADUs separately and then applying a cross-attention layer, models could dynamically weigh the influence of each unit when predicting their relationship, leading to improved classification accuracy.

Increasingly, the focus on pretrained language modeling has sought to eliminate

the need for task-specific architectures, utilizing the strengths of generative language modeling to produce sound results. Some key work begins to tease out the roles that this intersection may play in analysis. Many works use generative modeling in a classification-style setting, where a prompt includes a full text, an extracted portion of said text, and a query to identify a label or argument from a set. In these cases, model outputs are parsed for each generative call for an argumentative component $O(n)$ or for each pair of argumentative components $O(n^2)$, where n is the number of pre-identified segmented components. While such baselines exhibit strong results in component [8, 3, 36] and relation [26, 25] classification, they appear computationally inefficient due to overheads incurred from engaging autoregressive decoders to compute model outputs. The similarity in performance of these results to encoder-based classification methods above challenges the need for generative models for effective relation extraction.

On the other hand, some work has begun to consider efficient, one-pass approaches to generative argument mining. Kawarada et. al [40] incorporate the TANL framework [59] to train T5 [68], an encoder-decoder model, to segment text into ADUs, classify said ADUs, and identify and type relations between them in one pass (see 2.4.4). TANL, or Translation through Augmented Natural Language, frames structure prediction as a machine translation task going from raw text input to augmented natural language - a reiteration of the same text input with structured annotations to incorporate attributes such as entity type and relations. TANL allows the extraction of structure information from output texts with basic text manipulations, assuming that a model adheres to the output requirements and the input text can be reconstructed from its output. This intervention is critical to employing generative models as classifiers, but it also creates the possibility for noisy outputs, as deficits in model accuracy may be the consequence of poor adherence to the structure format, as opposed to attempts to decode correct annotations. We explore TANL and the implications of its adherence

gap, dubbed *errors in compliance*, in Chapter 3).

Another generative argument mining paper by Sun et. al [80] uses a remarkably sophisticated architecture to enable greater prompt interactivity, passing encoder embeddings through a relational Graph Convolution Network before subsequently decoding with an interactive prompt framework. This approach translates the original text to a serialized list of graph entities and triplets, infusing it with structural information.

2.4 Structure Prediction

2.4.1 Introduction

Finally, we consider structure prediction, the formal task family which encompasses computational argument mining. Structure prediction refers to a classification task in natural language processing which automatically applies some predefined semantic schema to labeling components of unstructured text. Examples of structure prediction include tasks like "named entity recognition", the extraction and labeling of proper nouns such as names, places, or geopolitical entities; "relation extraction", where relations between named entities are identified as edges between entities and labeled; "coreference resolution", a subset of relation extraction which merges named entities by identifying similar/identical entities; "dependency parsing", when each word in a sentence is assigned some position in a hierarchy of grammatical relations, and so forth. This review discusses entity and relation extraction at length, at times formulating them as a joint task to be done in one pass by a model.

2.4.2 Named Entity Recognition

"Named entity recognition", or NER, identifies spans in text containing discrete concepts like people or places [1]. Evolution of NER occurred in four phases. First came

naive rule-based systems, which rely exclusively on a knowledge base, predetermined rules and dictionaries in order to flag relevant entities [22, 55]. Traditional statistical/machine learning methods quickly followed, distilling features from text to produce parameterized models of various architectures such as conditional random fields [72], support vector machines [41], and hidden Markov models [54]. As computational capabilities began to match the complex models in deep learning, task-specific recurrent neural networks and long short-term memory models were successfully applied to NER [12]. Most recently, the success of attention-based Pretrained Language Models has encouraged prompt-based applications, where different multi-turn QA environments encourage stronger, data-efficient model results [6, 105].

2.4.3 Relation Extraction

"Relation extraction" (RE) infers a set of relational triplets from unstructured data, often assuming preexisting knowledge of entities. An RE system must parse the unstructured data source (usually, a large body of text) to produce relational triplets in the style of the Resource Description Framework (RDF). Popular early models [21, 5, 23] used extensive dependency parsing systems as well as some limited parameterized/trainable functions to identify strong candidates for relations. In this context, relation extraction is often framed as a classification task with a finite set of given nodes, predicting relations through operations on node-pairwise tensor $D \in \mathbb{R}^{n \times n \times d}$, where n is the number of nodes and d is the size of the relation embedding. Like named entity recognition, relation extraction scaled well with evolution in deep learning thanks to architectures like recurrent neural networks (RNNs) [104] and, more recently, pretrained language models – both as encoders for classifier systems and as generators using the translation paradigm [59]. Relation extraction is the basis for joint entity-relation extraction.

2.4.4 Joint Entity-Relation Extraction

Joint entity-relation extraction (JERE) aims to simultaneously identify entities and their relationships within a given text [104, 106, 105]. This task is more complex than isolated NER or RE as it requires models to both isolate latent entities and declare semantic connections at the same time. Traditional approaches often utilize pipelined systems where NER is performed first, followed by RE. However, these pipelines suffer from error propagation and fail to capture the interdependent nature of entities and relations.

Recent advances in deep learning have enabled the development of end-to-end models that jointly learn to extract entities and relations [35, 46, 49]. These models often leverage shared representations and joint decoding mechanisms to improve performance. For instance, sequence-to-sequence models have shown promise in generating structured outputs that capture both entities and their relationships [59]. Additionally, graph-based models have been used to explicitly model the dependencies between entities and relations, leading to improved accuracy [97, 15, 47].

The shift towards generative models, particularly those based on pretrained language models, has further enhanced the capabilities of JERE systems. By framing JERE as a sequence generation task, these models can leverage their strong language understanding and generation abilities to produce coherent and accurate entity-relation triplets. This approach not only simplifies the model architecture but also improves robustness and generalization.

2.4.5 Challenges

While much progress has been made in automated methods for joint entity-relation extraction, several key roadblocks persist:

1. **Data Scarcity**: there are no natural data generation processes for abstract

relation ontologies, placing the burden on manual annotators to produce labels by hand. Key datasets use noisy or PLM-generated data to address this, which might propagate existing errors or flaws in the annotations available [79].

2. **Inter-annotator Agreement**: Relatedly, since a given input yields many acceptable node and edge designations, agreement between annotators is low. To resolve this, limited re-annotation efforts have been made to include several right answers. This is especially the case in the context of argument mining, per the Structured Language Problem.

3. **Long-Tail Sparsity**: Audits of key datasets TACRED[4] and DocRED[34] reveal a sparsity of long-tail or rare relation types, due to gaps in annotation quality along specific types. This causes training data to over-fit to easy relations and miss triplets obvious to human readers. To address this, researchers propose re-annotation to include rarer entities, or direct instruction in the case of synthetic data. However, such sparsity depends on the domain and context of the dataset.

4. **Heterogeneous schema**: Relation types are not uniform and reflect the context of their data. This makes it more difficult for PLMs to scale from multiple sources or datasets. Unified schema representations seek to harness this heterogeneity to encourage greater model generalization [47, 96].

2.4.6 Language Models for Joint Entity-Relation Extraction

Researchers have successfully employed two primary strategies to leverage language models in their methods:

- **Representation Encoding**: LLMs yield latent embedding or attention weights which are aggregated to produce relation scores, in a matter akin to binary candidate classification.

- **Direct Decoding**: LLMs, typically encoder-decoders, generate entities and/or relations themselves. The output is a linearization of an edgelist $T \in \mathbb{R}^{(h,r,t)\times m}$ as E and R can be reconstructed from T.

Table 2.1: A survey of featured models

Model	PLM	Type	Notes
Encoding-based models			
MaMa [95]	BERT, GPT2	Enc.	Only uses attention matrix
ATG [101]	DeBERTa*	Enc.	Encodes spans in a limited vocabulary matrix.
Grapher [49]	T5	Enc./Dec.	Modular inputs, end-to-end differentiable. The only node-first approach listed.
ReLik [58]	E5	Enc.	Two encoders – one for relations, one for "reading"
Generation-based models			
TANL [59]	T5	Enc./Dec.	Original seq2seq problem formulation
KnowCoder [46]		Enc./Dec.	Code Generation instruct-based LLM from scratch
DeepStruct [96]		Enc./Dec.	Pretraining on schema generation and prediction
REBEL [35]	BART	Enc./Dec.	TANL-like on BART with fine-tuning
ReGen [18]	T5	Enc./Dec.	Fine-tuning with RL-inspired loss functions
GenIE [39]	BART	Enc./Dec.	Beam search during inference for constrained generation
UIE [47]	T5	Enc./Dec.	Universal schema framework and model-generated schemas
EDC [102]	Several	Dec.	Explicit "schema" and "canonicalization" definitions
PiVE [29]	T5, GPT	Enc.-Dec.	Iterative Verification

Representation Encoding Encoding-based models, such as MaMa [95], utilize LLM attention matrices for relation extraction. Other frameworks, like ATG [101] and ReLik [58], optimize latent state representations for classification. Grapher [49] offers a hybrid approach, generating nodes and classifying edges.

Direct Decoding The generative paradigm, exemplified by TANL [59], reframes JERE as a sequence-to-sequence task, directly generating entity-relation triplets. REBEL [35] and BT5 [2] further this approach with fine-tuned BART and T5 models. Recent innovations focus on optimizing generation through customized loss functions, like the self-critical sequence training (SCST) used in ReGen [18], GenIE [39], and PiVE [29], which reduces gradient variance. Creative pretraining strategies, such as DeepStruct [96] and KnowCoder [46], aim to enhance LLM understanding of structured prediction by pretraining on diverse tasks and converting schemas to code generation environments, respectively. These methods address data scarcity and improve model generalization.

The self-critical sequence training loss function is defined as:

$$\nabla_\theta \mathcal{L}_{\text{SCST}} \propto -\left(R(\hat{x}_T) - R(x_T^*)\right) \nabla_\theta \log p_\theta(\hat{x}_T,) \tag{2.1}$$

where $R(\hat{x}_T)$ is the generated text, $R(x_T^*)$ is the greedy-max optimized baseline, and θ refers to model parameters.

2.5 Conclusion

We have thus laid the groundwork for understanding argument mining by exploring its philosophical origins in rhetorical theory, its technical parallels in structure prediction, and its unique challenges as a field of study. By examining the contributions of Aristotle and Toulmin, we established a theoretical framework for argumentative discourse. We then transitioned to the technical aspects of structure prediction, detailing the evolution of methods for entity recognition and relation extraction, culminating in the complexities of joint entity-relation extraction. Finally, we explored the specific domain of argument mining, highlighting its distinctions and the ongoing shift towards generative models. The challenges inherent in argument mining, such as

data scarcity, inter-annotator agreement, and the nuanced nature of argumentative language, underscore the need for innovative approaches. The progression towards end-to-end trainable systems, particularly those leveraging large language models, represents a promising direction for overcoming these obstacles. Subsequently, we will surface key evaluation criteria for understanding a post-language model paradigm for argument mining, focusing on efficient and scalable solutions without compromising the formal requirements of a classification-based task.

Chapter 3

Approach

3.1 Introduction

Here, we define the scope of our project as an audit of pretrained language model capabilities for end-to-end, joint entity-relation extraction of arguments. After formally defining the problem, we explore four key dimensions of argument mining to determine the limits of language models in this domain:

- **Labeling Strategies**: traditionally, relation extraction tasks assume entities that comprise very few words. As our task uses entities that span one or more sentences, referencing entities with full token spans could increase latency and even reduce performance due to longer context windows. We consider the impact of instead using numerical labels as well as abstractive summarization techniques.
- **Encoder-Decoder v. Decoder-only Models**: the necessity of encoders in effective language modeling has been a subject of debate given the popularization of attention-based auto-regressive modeling [89]. We experiment with leading models under these two architectures to compare their effectiveness and offer theories for performance imbalances.
- **The Necessity of Fine-Tuning**: large language models exhibit an emergent

capability of in-context learning, leveraging knowledge gained in the pretraining process to incorporate dense embedding and attention structures. While sequence-to-sequence models typically require fine-tuning for maximal effectiveness, we measure the impact of fine-tuning on performance of decoder-only models.

- **End-to-End Extraction v. Relation-Only**: while we strive for an end-to-end system, entity recognition is an upper bound on relation performance. We conduct experiments where a model is given a segmented and labeled instance of text to evaluate improvements on relation extraction.

Then, we turn to the key criteria necessary to evaluate performance. Metrics for performance fall under two categories.

First, we consider **compliance**, which relates to a model's capability to adhere to the structure of the problem without guided decoding. Across our four dimensions, we consider the integrity of an input after removing annotations (reconstruction error), adherence to label categories for entities and relations (label error), erroneous formats on tags (format error), and relations with imprecise, non-existent, or otherwise irretrievable entities (entity error).

Second, we consider **accuracy**. Assuming high compliance with the augmented natural language, we measure the proximity of model outputs to ground truth labels with traditional metrics such as precision, recall, and F_1 scores. As we are especially interested in a model's ability to produce relations, we consider a relaxed criterion for relations which removes labels from entities and relations, requiring only that the spans of head and tail entities match.

We conclude with reflections on directions for future audits. Potential experiments could center model size, comparison across datasets or open, out-of-distribution argumentative relation extraction, as well as the impact of cascading errors in an end-to-end generation environment such as ours.

3.2 Problem Formulation

We define the task of jointly mining entities and relations in arguments as an end-to-end process.

3.2.1 Joint Entity-Relation Extraction

Entity Labeling

Entity labeling is defined as the process of identifying and classifying specific spans of text that correspond to entities within a given document. An entity can be defined as any mention of a concrete or abstract object, such as a person, organization, location, or date. More formally, given a text $T = \{t_1, t_2, ..., t_n\}$, entity labeling consists of identifying a set of token spans $S = \{s_1, s_2, ..., s_m\}$, where each span $s_i = [t_a, t_b]$ represents the boundaries of an entity in the text, and assigning a label L_i to each span such that:

$$\text{EntityLabel}(s_i) = L_i, \quad s_i \in S, \quad L_i \in \mathcal{L}_{\text{entity}},$$

where $\mathcal{L}_{\text{entity}}$ represents the set of all possible entity labels (e.g., Policy, Value, Fact, Testimony, etc.).

Relation Extraction

Relation extraction is the task of identifying the semantic relationship between pairs of entities within a given text. Formally, given two entity spans s_1 and s_2 from the set S defined earlier, the relation extraction task can be represented as identifying a relation r from a set of predefined relations $\mathcal{L}_{\text{relation}}$ between the two entities such that:

$$\text{Relation}(s_1, s_2) = r, \quad r \in \mathcal{L}_{\text{relation}},$$

where $\mathcal{L}_{\text{relation}}$ represents the set of possible relations (e.g., "reason for", "evidence of"). This task involves not only the identification of entity spans but also the correct identification of the relation that connects the entities.

Joint Problem Formulation

In the joint entity-relation extraction task, we aim to simultaneously extract both entities and their corresponding relations from a given text. Formally, the goal is to extract a set of entity spans $S = \{s_1, s_2, ..., s_m\}$ and a set of relations $R = \{r_1, r_2, ..., r_p\}$ such that for each relation r_i in R, there exist two entities s_a and s_b in S where:

$$r_i = \text{Relation}(s_a, s_b), \quad \text{for} \quad s_a, s_b \in S, \quad r_i \in \mathcal{L}_{\text{relation}}.$$

The joint task can be approached as a sequence-to-sequence problem, where the model must generate both the entity spans and their relations in a single end-to-end pass. The final output consists of a set of entity-span pairs, along with the relations that exist between them, as structured entities.

3.2.2 Generation v. Classification

A key distinction in both argument mining and joint entity-relation extraction literature is the use of generative modeling to produce entities and relations. Previously, classification-based approaches defined the entity extraction component as a classification task determining the probability that a token was at the start, middle, or end of an entity span using BILOU or IOB tags. Similarly, relation extraction was defined given predetermined entities, and assessing if a relation exists between them with a null class to signify no relation.

However, we employ an *augmented natural language* framework. This means that we attempt to reproduce an input text with separators and delimiters, such that

we can parse our output automatically. This is what Paolini et. al[59] refer to as *translation through augmented natural language* (TANL).

3.2.3 The TANL Framework

Translation through augmented natural language (TANL) is an abstract framework framing structured prediction as a sequence-to-sequence translation. TANL takes a document as input and returns a similar output sequence which adds brackets around identified entity spans and appends task-specific metadata through delimiters within the bracket spans:

$$s_i = [\text{entity text} \,|\, \text{attribute}_1 = \text{value}_1 \,|\, \text{attribute}_2 = \text{value}_2 \,|\, \dots]$$

In the context of joint entity-relation extraction, the attributes refer to entity types, and relation types to tails (Table 3.1). Generated outputs are expected to follow this schema for efficient parsing.

Entity Text	**Label**	**Relations**
`I disagree.`	`value`	
`Keeping a paper trail protects everyone.`	`value`	`(reason, I disagree.)`

Table 3.1: Parse structure for `[I disagree.|value][Keeping a paper trail protects everyone.|value|reason=I disagree.]`
The original sentence is `I disagree. Keeping a paper trail protects everyone.` Abridged sample of CDCP dataset.

3.3 Key Questions

We reiterate the key research questions which motivate our experiments. While there are many dimensions to consider in the open problem space of argument mining, we select those which appear most task-intrinsic and crucial to a case study of long-context joint entity-relation extraction.

3.3.1 Argument Annotation Strategy

How does the representation of entities and relations in the output affect model performance, especially in long-context arguments? Specifically, we investigate whether alternative annotation strategies can improve efficiency without sacrificing accuracy.

We compare three output formats:

- **Full Token Spans:** Standard TANL output, using actual text spans.
- **Numerical IDs:** Shortening outputs with numerical identifiers.
- **Abstractive Summaries:** Condensing entities with generated summaries.

The core questions are: Can numerical IDs reduce latency, and at what cost to accuracy? Will abstractive summaries maintain TANL compliance? We hypothesize that numerical IDs may trade accuracy for efficiency, and abstractive summaries will likely introduce TANL compliance errors. We expect full token spans to yield the highest accuracy, assuming long-context handling is not a limitation.

3.3.2 Encoder-Decoder v. Decoder-only

Since the publication of "Attention is All You Need"[89], researchers have steadily departed from including encoding architectures when designing generative language models. The auto-regressive, decoding-based transformer has taken a monopoly in natural language processing as an optimal configuration for its efficiency, scaling capabilities, and representative capacity in pretraining. While decoder-only models have proven highly effective, encoder-decoder models such as BART[44] or T5[68] carry certain asymmetric advantages by framing problems as sequence-to-sequence pretraining with masked language modeling, rather than through next-word prediction. Their bidirectionality, encoding module, and adaptability to downstream fine-tuning

render them exceptionally capable, often outperforming state-of-the-art autoregressive LLMs[66].

We explore how this dynamic plays out in the specific context of joint entity-relation extraction, particularly within the TANL framework and with long-context arguments. Specifically, we aim to determine whether the inherent advantages of encoder-decoder models, such as their ability to capture long-range dependencies through bidirectional encoding, translate into superior performance in this structured prediction task. Conversely, we investigate if the generative capabilities and efficiency of decoder-only models provide a competitive edge.

To this end, we will compare the performance of encoder-decoder models (T5 and BART) with decoder-only models (Llama 3.2 3B, Mistral 7B, Qwen 2.5 7B, and GPT-4o (few-shot)). We would like to compare the performance of seq2seq models with decoder-only models, determining if recent innovations in decoder-only architectures can outpace the intrinsic benefits of older models.

3.3.3 Few-Shot v. Fine-Tuning

Here we consider how fine-tuning compares to few-shot, in-context learning for joint entity-relation extraction with decoder-only models. Specifically, we aim to answer: Can fine-tuning smaller models achieve higher accuracy than few-shot learning with larger models?

We investigate the performance differences between these two training paradigms, focusing on accuracy. We compare different decoder-only model families, evaluating their performance with a baseline few-shot prompting strategy and fine-tuning. Fine-tuning methodologies will be detailed in the experiments section.

We are primarily concerned with accuracy and measure performance using accuracy and compliance metrics.

3.3.4 End-to-End v. Relation-only

This ablation relaxes the nature of the task by eliminating the need to correctly segment relations. The correctness of relation extraction in the joint entity-relation environment is upper bounded by the correct segmentation of entities. If an entity is improperly defined, either in terms of its starting or ending spans or by type, then there is no chance that it correctly identifies ground truth relations.

In light of this, we consider an environment where a model input already has entities defined, and the desired output is the same input with relations included as tags on existing entities. This lowers the standard for performance in two ways. First, it removes the need for precision when labeling and spanning entity arguments. Second, the relation-only nature of the task allows loss functions to purely reflect capability in identifying other labels, as opposed to incorporating entity spans.

Naturally, model performance in this space would improve as a function of resolving the entity problem. Here, we aim to directly understand a model's ability to recognize claims and premises.

3.4 Evaluation Criteria

To measure the impact of these levers on model performance, we evaluate performance based on **compliance** and **accuracy**. Compliance, unique to framing entity/relation extraction in a generative environment, measures how well model generations accord to the expected parameters of the format. This includes whether a model output can be correctly parsed, whether it resembles the original sentence, whether related entities exist or are hallucinations, and whether the entity and relation types are a part of the dataset schema.

3.4.1 Compliance

Argument mining and relation extraction systems which use generative models encounter the particular challenge of ensuring that model outputs comply with their expected format. Such questions are crucial when engaging language models as parsing agents or for structured language tasks like SQL generation [73, 75] and tabular understanding [82]. In the context of structured prediction for relation extraction, Paolini et. al [59] study the effect of different testing environments on the quality of model outputs, isolating four error criteria.

The quick brown fox jumps over the lazy dog.
The [quick brown fox | animal | jumps = dog] jumped over the [lazy dog | dog |].

Figure 3.1: An example of the four error types
, where *animal* is the only valid entity type and *jump* is the only valid relation type. Entity error in red, reconstruction error in yellow, label error in blue, and format error in green.

Reconstruction error: When a generative model re-produces an input text, it may incorrectly do so, omitting, hallucinating, or modifying punctuation and semantically meaningful content. Paolini et. al refer to this as *reconstruction error*, as it is determined after processing a model output to extract tags and value information. Reconstruction error is determined by matching sequences to original input. Paolini et. al introduce a post-processing alignment algorithm based on Needleman-Wunsch's dynamic programming solution [56], which computes a minimal-cost adjustment schema based on a matrix of previous adjustments.

Entity error: When generating relation tags, the related entity (or the tail) may be attributed to an entity which does not exist. Entities are defined by the bracketed text which appears in the input, or may be mapped to a shorthand numerical/summary label which presents in the first tag (the entity class). In the example above, "dog" is

not a valid entity, while "lazy dog" would be.

Label error: Sometimes, models hallucinate labels which are not part of the defined entity or relation types, such as "dog" above. These are label errors as they map entities and relations to types not given.

Format error: If the format of the augmented natural language (namely the square brackets and | delimiter) are incorrectly formed, a sentence produces a format error. The extra delimiter in Figure **??** is an example.

Under our evaluation framework, we measure each error type as a simple true/false value for each example. For any example, a model which makes more than one error of any type will receive an error score of at most 1 for that type.

3.4.2 Accuracy

Structured prediction tasks such as named entity recognition and relation extraction are ultimately forms of span classification. As classification tasks, the most descriptive evaluation frameworks come from the standard precision, recall, and F1 scores. However, the question becomes which unit to evaluate under. We follow the work of other argument mining researchers [76, 53, 20, 99, 40, 80] to consider each Argumentative Discourse Units as the basis for analysis, as opposed to token-level or sentence-level classification. This creates some gray area as the accuracy of a relation is upper-bounded by the accuracy of the entities which it connects. We consider three distinct measurements of accuracy:

1. **Entity Classification**: Models are evaluated on the start token, end token, and class of each ADU.

2. **Relation Identification**: Models are evaluated on the start and end tokens for the head and tail of each relation between ADUs.

3. **Relation Classification**: Models are evaluated on start, end, and class for the

head and tail, as well as the correct class of relation type.

3.5 Conclusion

This chapter outlines the key tenets of our contribution towards literature in generative structure prediction and argument mining, focusing on four dimensions of analysis and introducing a set of compliance metrics to measure the successful transition of generative models to classification tasks using augmented natural language. In the chapters to follow, we articulate the explicit experimental approaches used to test performance along these dimensions and novel metrics, focusing on two key datasets in argument mining as a litmus test for structure prediction and improvements.

Chapter 4

Experiments

4.1 Introduction

In this section we apply accumulated domain knowledge and the framework of our evaluation to a suite of language models which employ different strategies to learn the terrain of argument mining-based knowledge extraction. Here, we articulate datasets and pretrained models used in our experiments, the technical details of the training paradigms that we used for fine-tuning language models, prompting set-ups for in-context learning environments, various relaxations applied to the scoring of model performances and different evaluation paradigms of most interest. The results of these experiments are to be addressed in 5.

4.2 Datasets

As covered in Chapter 2, several datasets exist which segment text into argumentative units, classify said units under a given label scheme, and relate units together with some heterogeneous edge label. For the sake of our experiments, we pick two benchmark datasets: the Cornell e-Rulemaking CDCP Corpus [60] and the UKP Lab's Argument Annotated Essays Corpus (AAEC) [76]. We selected these datasets over other strong

candidates for several reasons:

	CDCP	**AAEC**
Documents	731	402
Components	4,931	6,089
Relations	1,220	3,832
Component Types	5	3
Relation Types	2	3
Avg. Components/Doc.	6.74	15.15
Avg. Relations/Doc.	1.6	9.53
Avg. Tokens/Doc.	121.85	368.42

Table 4.1: Comparison of CDCP and AAEC Datasets

	CDCP
Policy	815
Fact	786
Value	2,182
Testimony	1,117
Reference	32
Reason	1,353
Evidence	73

	AAEC
Major Claim	751
Claim	1,506
Premise	3,832
Support	4,841
Attack	497
Semantically Similar	349

Figure 4.1: Entity/Relation Counts by Type

1. **Diverse node-edge compositions**: The two benchmarks represent the most relation-rich and structurally complex instances of data annotated to include both entities and relations. The Cornell e-Rulemaking Corpus on Consumer Debt Collection Practices (CDCP), for instance has 4,931 components and 1,220 relations across 731 comments; this equates to an average of 6.74 entities and 1.6 relations per comment. CDCP also includes a significant quantity of components which have more than one edge in their composition, suggesting the existence of a more complicated annotation schema which does not automatically render directed acyclic graphs. Additionally, the entity classes in CDCP are more descriptive – describing facts, values, policies, testimonies, and references – while

the relation classes are only reason and evidence, with no attack/undermine sort of edge. Meanwhile, the AAEC dataset accomplishes a complementary set of objectives. It incorporates 6,089 components and 3,832 relations over 402 essays; these are averages of 15.15 entities and 9.53 relations per essay, significantly higher than CDCP. AAEC assumes an acyclic graph annotation schema, meaning that nodes have at most one edge. AAEC has three entity classes: claim, major claim, and premise, and its relation types are attack, support, and semantically same. As a caveat, it is possible to use the "semantically same" category to group entities together as co-references, thus inducing cyclic component relationships as in CDCP. It is also worth noting that the average entry AAEC is about three times as long as CDCP (368.42 v. 121.85) [1].

2. **Content relevance**: The content of the two datasets carries important stylistic variations within themselves and between eachother; however, their supervised curation prevents them from devolving to noise typically collected online. AAEC is sourced from English students (many learning English as a second language) forming arguments, with each data point taking its own subject, such as travel, friendship, criminal justice, housing, environmentalism, economic inequality, urban life, etc.. It has a much wider gamut than CDCP, which focuses squarely on potential regulations to impose upon debt collectors, specifically with respect to contacting behaviors. While the CDCP dataset includes many documents produced by fluent English speakers squarely discussing consumer protection practices, it does not have the same caliber of domain-specific jargon seen in other similar datasets like the biomedical AbstRCT [61] or for argumentative zoning [84].

3. **Comparison against other baselines**: AAEC and CDCP are the most widely used datasets in terms of benchmarking from other state-of-the-art

[1] measured in tokens generated with the LLaMa tokenizer

"Nowadays, as the world is getting smaller, we need to be able to cooperate. Working in team is a necessary skill that every individual must master for success. Despite the importance of working in team, I still believe that the capability of working independly is more significant..."

Figure 4.2: An excerpt of a sentence from AAEC.

"You are a liar! I have been harassed for over 5 years by companies looking for someone I have never heard of because this person put a random string of our area code, local exchange and 4 numbers together (that turned out to be my number) on a loan that he then defaulted on. I have told them to stop calling, they have a wrong number, and guess what " 5 Years later, they are still calling!"

Figure 4.3: An excerpt from CDCP.

reference systems for argument mining. While each piece of literature describes experiments using a suite of data sources, most data aligns on using these two datasets as standard benchmarks. CDCP owes this to its status as a relatively older dataset, while AAEC serves as the largest gold-standard datasets in argument mining in terms of raw token count. As such, it comes as no surprise that many recent papers use these two works as benchmarks [53, 99, 40, 7].

4.3 Baselines

To ground our work in current argument mining literature, we also articulate key works which claim state-of-the-art performance as baselines for evaluation.

Morio et. al 2022 [53] employ a bi-affine dependency architecture inherited from [99], which uses autoencoder embeddings as inputs to a multilayer perceptron classifier operation to identify and classify spans. Then, classified spans are related to one another with two separate classification heads: one for link detection and a second for classification. Weights for the base encoder as well as the classification heads are updated through backpropagation on spans and ground truth relations. The autoencoder used is Longformer, which iterates over the BERT base model to support longer context encoding.

Bao et. al 2022 [7] implement a generative system to jointly model argumentative components and relations, decoding classic relational triples with an encoder-decoder model. They employ several techniques during encoding, namely a pointer mechanism which marks the start and end tokens of a sequence. The pointer distribution is constrained through the validity of token matches, such as predicting end tokens after start tokens; predicting relations using only the distribution of the last token, and ensuring that relation label pointers can only sample from the predicted relation space.

Kawarada et. al 2024 [40] introduce a generative framework that embraces translation through augmented natural language. They introduce a basic approach which uses encoder-decoder models trained on input-output representations constructed in TANL style.

With these baselines guiding our evaluation of argument mining capabilities with modern language models, we specify the procedures used to explore the directions defined in 3.

4.4 Experimental Details

As a reminder, this analysis incorporates a four-dimensional analysis of language modeling capabilities for argument mining. We are interested in different ablations to truncate the labels of Argumentative Discourse Units, the distinction between in-context learning and fine-tuning strategies, the use of different model architectures (with specification about exact models used), and the "oracle" setting where entities are pre-segmented and classified to facilitate end-to-end experiments.

4.4.1 Argument Labeling Strategy

There are three annotation strategies necessary for this analysis. Experiments with these annotation strategies each use the same input (a raw, unstructured sentence),

Regular:
"[You are a liar! | **value**] [I have been harassed for over 5 years by companies looking for someone I have never heard of | **testimony**] [because this person put a random string of our area code, local exchange and 4 numbers together (that turned out to be my number) on a loan that he then defaulted on. | **testimony** | **reason** = *I have been harassed for over 5 years by companies looking for someone I have never heard of*] [I have told them to stop calling, they have a wrong number, and guess what " | **testimony**] [5 Years later, they are still calling! | **testimony**] !"

Number:
"[You are a liar! | **value = 0**] [I have been harassed for over 5 years by companies looking for someone I have never heard of | **testimony = 1**] [because this person put a random string of our area code, local exchange and 4 numbers together (that turned out to be my number) on a loan that he then defaulted on. | **testimony = 2** | *reason = 1*] [I have told them to stop calling, they have a wrong number, and guess what " | **testimony = 3**] [5 Years later, they are still calling! | **testimony = 4**]"

Summary:
"[You are a liar ! | **value = You are a liar !**] [I have been harassed for over 5 years by companies looking for someone I have never heard of | **testimony = I've been the victim of harassment**] [because this person put a random string of our area code , local exchange and 4 numbers together (that turned out to be my number) on a loan that he then defaulted on . | **testimony = Defendant's phone number was used in a default** | **reason** = *I've been the victim of harassment*] [I have told them to stop calling , they have a wrong number , and guess what " | **testimony = They have a wrong number**] [5 Years later , they are still calling ! | **testimony = Defendants continue to harass**]"

Figure 4.4: An excerpt from CDCP, with full, number, and summary annotations in respective order.

but expect different outputs (exposed through few-shot prompts or in fine-tuned label ids). Examples of each of these labels are presented as figures at the end of the subsection.

The first two strategies are quite self-explanatory. The first strategy, referred to as "full" representation, repeats an argumentative entity if it appears as a tail node in relation in other argumentative units. The second "number"-based strategy assigns a numeric id to each entity which is used to relate entities together.

The third, "summary"-based strategy attempts to merge the semantic richness of full sentences with the efficiency associated with shorter labels by teaching models trained under this schema to use a "language decomposition," or summary of an ADU, as a label, following research that such decompositions can bring out implicit arguments in models and potentially improve their relation identification using metrics of semantic proximity [33]. To do this, a large, autoregressive language model is passed through each entity in each example. The model is exposed to the entire document as well as the individual sentence, prompted to produce a summary of the argument relative to its text with an upper limit on word counts. Arguments less than 10 tokens long are not included in this abstractive summarization process, and their original representation is used in a manner similar to the first "full" strategy.

Interestingly, the summary based strategy does not necessarily yield shorter sen-

tences than full outputs. Because each argument entity must include an entity, there is potential to create much longer sequences for decoding. Using a LLaMa tokenizer, the maximum token length of any sequence is 1,670 for sentences with summaries, followed by 1,123 tokens for "full" labels and 897 tokens for "number" labels. AAEC also shows little difference: 1,185 tokens is the maximum for summary labels, only 22 tokens shorter than the full label maximum of 1,207. The number label strategy sits at around 860 tokens.

4.4.2 Encoder-Decoder v. Decoder-only

As with dataset selection decisions, model family selection is quite important for our experimental environment. We select two sequence-to-sequence (encoder-decoder) models and three decoder-only models to make comparisons.

Encoder-Decoder Models

For our sequence-to-sequence (seq2seq) experiments, we selected BART-large [44] and FLAN-T5-large [13].

BART (Bidirectional and Auto-Regressive Transformers) is a denoising autoencoder that is trained to reconstruct corrupted input text. It's pre-trained by corrupting documents with an arbitrary noising function and then learning to reconstruct the original text. This pre-training enables it to excel at tasks like text summarization, translation, and text generation.

FLAN-T5 (Fine-tuned LAnguage Net T5) builds upon the T5 architecture, which is pre-trained on a massive text corpus using a text-to-text framework. However, FLAN-T5 distinguishes itself by its instruction tuning approach. It's fine-tuned on a collection of datasets that are reformulated as natural language instructions. This process trains the model to follow instructions, making it particularly effective at zero-shot and few-shot generalization. The instruction tuning allows FLAN-T5 to

adapt to a wider range of tasks, including those not explicitly seen during training, by interpreting and following natural language instructions.

While the broader field of language models is increasingly dominated by decoder-only architectures, the encoder-decoder paradigm remains valuable for tasks requiring bidirectional context understanding. We selected BART and FLAN-T5 due to their established effectiveness and the lack of readily available, similarly capable seq2seq alternatives. These models' ability to process and generate coherent text, along with FLAN-T5's instruction-following capabilities, makes them well-suited for the complex argument mining tasks we aim to address.

Decoder-Only Models

We use three models as our baseline for decoder-only models.

LLaMA 3.2-3B-Instruct is a mid-size model developed by Meta, belonging to the LLaMA series of large language models. Its training involves pre-training on a massive text corpus to learn general language patterns, followed by instruction tuning to enhance its task-specific capabilities, emphasizing efficiency and instruction-following within a smaller model footprint.

Mistral-7B-Instruct, developed by Mistral [38], is a mid-size, language model that also utilizes a decoder-only transformer architecture. Notably, Mistral-7B incorporates architectural innovations such as Grouped-query attention (GQA) and Sliding Window Attention (SWA), which contribute to improved efficiency and the ability to handle longer context windows. As with the LLaMA Instruct variant, Mistral-7B-Instruct has been fine-tuned on instruction datasets, enhancing its ability to accurately follow user instructions. Its training process involves pre-training on extensive text data, followed by fine-tuning to optimize instruction-following capabilities, making it particularly well-suited for tasks demanding both efficiency and long-context processing.

QWen-2.5-7B-Instruct , developed by Alibaba Cloud [67], is another mid-size

language model. Pre-training occurs on a large, multilingual dataset, with a strong focus on both Chinese and English language data. Similar to the other 'Instruct' models, QWen-2.5-7B-Instruct undergoes fine-tuning on instruction datasets, enabling it to better respond to user instructions.

These models were selected due to their instruction tuning, allowing them to closely resemble the goals of encoder-decoder models which require instruction-based fine-tuning for optimal function. Additionally, each model is the mid-size edition of the latest open-source releases in each family. The mid-size edition was chosen to demonstrate the capabilities of fine-tuning and to compare against bart-large and flan-t5-large, both of which are in the middle relative to XL and XXL models.

4.4.3 End-to-End v. Relation-only

Many argument mining papers which attempt to construct an end-to-end system also include "oracle" settings [7] where the ground-truth annotations for one pipeline component – typically the segmentation and classification of Argumentative Discourse Units – is given. This enables research on the relation extraction or linking component to be isolated and evaluated independently. Without such settings, the evaluation of relation extraction becomes entangled with the performance of ADU segmentation and classification, making it difficult to pinpoint the source of errors. Therefore, oracle settings provide a crucial benchmark for understanding the isolated performance of relation models, though they may not fully reflect real-world scenarios.

In these circumstances, we evaluate models on their ability to identify relations as well as their classes. We conduct such experiments on both datasets for all fine-tunes of auto-regressive models as well as for GPT-4o-mini's few-shot capabilities.

4.4.4 Few-Shot v. Fine-Tuning

While the previous dimension alters the intended output string, this dimension suggests two different approaches to encoding a learning process into auto-regressive language models. Recent studies and empirical investigations have suggested that auto-regressive language models are capable of incorporating hard rules from patterns through prompting strategies. To this end, we prepend the 10 inputs with the most relations from the CDCP dataset to evaluate the outputs of GPT-4o-mini (which we do not use in fine-tuning experiments) and LLaMA 3.2 3B in a 10-shot setting.

We use the examples with the most relations in order to increase the statistical distribution of relations within model selection outputs. CDCP is relatively relation-sparse, which can lead to circumstances in random sampling where no samples demonstrate a relation between two argumentative units.

The samples for prompting are taken from the test split, so evaluation results for the few-shot case are out of 140 examples instead of 150.

4.5 Implementation Details

Below, we describe some necessary implementation details to reproduce the results in this work. In the spirit of open research, code will be available to review and run model experiments independently.

For all fine-tuned models, we use unsloth [28], a multi-tool acceleration suite to attach low rank adapters and flash attention in quantized model training environments to accelerate the computations necessary for train- and test- time inference without impeding performance (see subsection 4.5.2).

4.5.1 Encoder-Decoder Models

Encoder-decoder models were trained on adaptations of the source code for the TANL paper [59]. We trained all parameters in each model over 20 epochs with a learning rate of 1e-5 with a dropout rate of 0.1, AdamW optimizer hyper-parameters set to the default values of 0.9 and 0.99 for α and β, respectively. Models trained on CDCP had maximum input sizes of 1024 tokens, with unlimited output size. Mixed precision training was enabled using FP16. All other specifications follow the default values used for FLAN and BART.

4.5.2 Decoder-Only Models

We loaded the models above with 4-bit quantization for efficient memory usage. We configured the model with appropriate maximum sequence length and data type settings to suit our experimental requirements. To enable parameter-efficient fine-tuning, we employed Low-Rank Adaptation (LoRA). LoRA was applied to specific projection matrices within the model's architecture, with a LoRA rank of 16 and corresponding alpha value. We optimized performance by disabling bias and enabling gradient checkpointing to minimize memory consumption. We used a per-device training batch size of 2, with gradient accumulation to effectively increase the batch size. A linear learning rate scheduler with a peak learning rate of 2e-4 was utilized, with a short warmup period. The model was trained for a maximum of 500 steps. We used the AdamW 8-bit optimizer with weight decay. Mixed precision training was enabled using either FP16 or BF16, depending on hardware support. The training seed was fixed at `3407` for reproducibility.

4.6 Conclusion

With the framework of our experiment outlined, we detail limitations to our experimental environment, which offer opportunities for future directions of exploration.

4.6.1 Future Directions

Further few-shot testing

There are several areas throughout the four dimensions of this analysis which could be made complete with more experimentation on different cross-sections. For instance, our few-shot setting makes limited use of LLaMA-3.2-3B-Instruct in comparison against GPT-4o-mini and does not attempt to evaluate against Mistral and QWen models. We found that the preliminary results of the few-shot environment demonstrated insufficient capability for open-source models of this size to compete against larger, proprietary models served by APIs, especially given asymmetric latency expectations when working with open-source models served locally on GPUs. Future research would benefit from optimizing the latency of locally-served open-source models prior to experimentation in order to increase throughput and allow more opportunity for cross-model analysis. This also applies to the limitation of working with the CDCP dataset, as poor latency conditions were exacerbated by the relative size of the AAEC dataset.

Greater model variety

We acknowledge the limitation of using only three decoder-only models and two encoder-decoder models in terms of finding strong candidates for argument mining systems. While the models selected represent the state of the art for language modeling paradigms, they are certainly not the only ones claiming top performance on reasoning and structured prediction benchmarks. Future research should incorporate a wider set

of model families to challenge the results set forth in Chapter 5.

The size of the models used for training are quite limited as well, due to constrained compute resources. Further experimentation would make judicious use of larger models through quantization, deeper mixed-precision training, larger cloud clusters, and further optimizations beyond flash attention for reducing the number of TFLOPS needed per full training/inference pass.

Chapter 5

Analysis

5.1 Introduction

In the following section, we will unpack the distribution of model performance results by defined dimension, comparing against key baselines along the accuracy- and compliance- based metrics. We find a variety of interesting results across two datasets, four dimensions of model configuration, and two sets of metrics. We explore the implications of compliance failures as a potential upper bound on model accuracy, delving into failure case studies as demonstrations of this effect.

Overall, we find that the dimension of *model architectures* bears the largest impact on accuracy, namely differentiating between encoder-decoder and decoder-only models. While a relative dearth of open-source encoder-decoder models and the stark difference in output quality might preclude a uniform declaration of the superiority of encoder-decoder models, our case studies demonstrate the power of this effect. We conjecture several possible reasons that encoder-decoder models offer stronger performance than decoder-only models. This distinction is particularly defined when comparing these classes of models in terms of their relation scores. We find that the discrepancy in relation performance is closely related to the number of predicted relations offered by

each model. T5 models were much more likely to produce relations than fine-tuned decoder-only models, though GPT produces more relations than T5 in the experiments conducted. This suggests that potential remains for decoder-only language models of sufficient size and training to outperform encoder-decoder models in generative argument mining, though the gap has not yet closed.

Such a hypothesis is corroborated by findings on compliance-based metrics. We unsurprisingly discover a discrepancy between models fine-tuned on the argument mining structure prediction task and models which engage in-context learning to produce few-shot answers. Fine-tuned models are more likely to follow the conventions of the annotation schema, producing fewer label and entity errors. The compliance behavior of fine-tuned models is equivalent for both encoder-decoder and decoder-only language models; both perform quite well at conforming to the basic standards of parsed language.

Labeling strategies demonstrate relatively smaller performance differences across models, with full-text label models generally outperforming numerical labels, which in turn outperform summary labels. Different labels also carry different implications for compliance – for instance, numerical labels yield far fewer entity identification errors than those for summaries or full text.

Comparing end-to-end models with relation-only model environments, we see inconsistent results. Some, but not all, models improve their relation-F1 score when given ground truth entity labels.

This section proceeds by identifying overall performance metrics for the two datasets investigated. We offer an overview table measuring the accuracy and compliance of models across these four dimensions before comparing across different dimensions of analysis and finally exhibiting a curated selection of performance successes and failures.

5.2 Accuracy

In terms of accuracy metrics, we compare F1 scores across three settings: **(i) Entity classification**, where models are evaluated on the start token, end token, and class of each ADU; **(ii) Relation identification**, where models are evaluated on the start and end tokens for the head and tail of each relation between ADUs; and **(iii) Relation classification**, where models are evaluated on start, end, and class for the head and tail, as well as the correct class of relation type. The full set of results for CDCP is exhibited in Table 5.1 and for AAEC in Table 5.2, shown against major baselines in classical and generative argument mining.

The following section will describe overall accuracy on the two datasets before considering the datasets jointly and analyzing results across dimensions. We will internally compare encoder-decoder models as well as decoder-only models, both with the full gamut of labeling strategies. We will then compare relation scores of full-text models in the "oracle setting" where entities are given as input.

5.2.1 CDCP Overview

Across models of different architectures and training environments, we see that FLAN-T5-Large is the model capable of the strongest performance, matching or beating the state of the art across all three metrics of analysis (Table 5.1). Our implementation of TANL using a full labeling strategy and minimal augmentations to the original formatting scheme, outperforms models of similar scale by Kawarada et. al [40] (E: +5.24, RI: +4.17). In addition, LLaMA 3.2-3B-Instruct performs unusually well in the unlabeled case, competing with but not outperforming baseline models.

On the other hand, BART-Large, an encoder-decoder model, performs the worst of all testing environments. As we will see later, this is due to compliance errors that BART experiences even after fine-tuning for extended periods of time. BART

Model	Entity	Relation ID	Relation Class.
GPT4o-few-shot			
Full	50.35	8.29	16.20
Number	50.03	6.32	14.22
Summary	48.05	2.82	7.04
LLaMA few-shot			
Full	41.50	4.63	7.36
LLaMA 3.2-3B-Instruct			
Full	65.44	26.09	21.12
Number	54.33	22.56	11.97
Summary	50.39	6.28	3.86
Mistral 7B-Instruct			
Full	70.42	8.47	6.78
Number	70.63	10.11	7.98
Summary	70.25	12.66	11.61
Qwen 2.5-7B-Instruct			
Full	58.66	16.09	7.36
Number	61.33	8.48	4.99
Summary	57.84	2.82	1.88
FLAN-T5-Large			
Full	72.21	**32.59**	**26.67**
Number	71.20	28.35	20.97
Summary	**74.18**	28.80	21.10
BART-Large			
Full	34.92	0.0	0.0
Number	29.02	5.74	0.52
Summaries	36.5	0	0
Other Baselines			
Bao et al. 2022 [7]	57.72	16.57	
Morio et al. 2022 [53]	68.90	31.94	16.26
Kawarada et al. 2024 Large [40]	68.94	28.42	
Kawarada et al. 2024 XL [40]	72.12	31.01	

Table 5.1: CDCP Performance Metrics by model and output type, in percentage points. Top results are in **bold**.

struggles to reproduce input sentences correctly, often replacing large swaths of text with semantically distinct content. This creates a low ceiling on the potential for

accurate performance.

We find small but consistent patterns in performance degradation along the different labeling strategies. In the relation identification/classification task, we see that full text labels tend to outperform numerical labels (avg. E: +3.02%, RI: +4.82%, RC: +5.01%), and in turn that numerical labels narrowly outperform summary-based labels on relation-based tasks, while summaries outperform for entity tasks (avg. E: −8.36%, RI: +1.91%, RC: +1.24%). This data affirms the applicability of Occam's Razor to generative argument mining – the simplest approach works best, especially as foundation models continue to expand context windows.

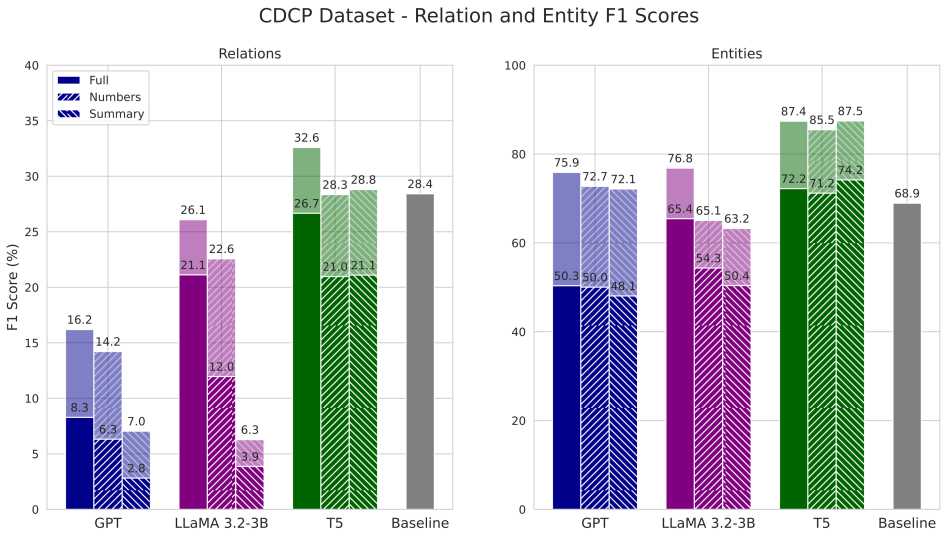


Figure 5.1: CDCP dataset, comparison of four models across three dimensions (few shot, seq2seq v. decoder-only, and labeling strategies). Relation/entity identification, the class-agnostic relaxation of relation/entity classification, is pictured with `alpha`=0.5. The baseline is Kawarada et. al 2024's FLAN-T5-Large finetune [40].

5.2.2 AAEC Overview

For the AAEC dataset, we see general improvements in the relation identification and classification scores due to an increased prevalence of relations compared to CDCP, aiding stronger performance for models trained on this dataset. As in the CDCP

Model	Entity	Relation ID	Relation Class.
GPT4o-few-shot			
Full	43.22	17.99	13.68
Number	44.34	16.41	12.88
Summary	39.62	14.65	9.89
LLaMA 3.2-3B-Instruct			
Full	68.29	41.77	36.04
Number	70.67	46.38	41.76
Summary	68.32	41.19	33.77
Mistral 7B-Instruct			
Full	70.43	48.83	43.18
Number	68.60	38.10	33.05
Summary	72.30	46.50	41.46
Qwen 2.5-7B-Instruct			
Full	73.01	46.99	41.97
Number	70.35	38.87	34.29
Summary	72.24	42.53	38.66
FLAN-T5-Large			
Full	75.19	<u>53.30</u>	<u>48.05</u>
Number	74.50	45.42	41.29
Summary	<u>76.14</u>	46.46	42.60
BART-Large			
Full	28.52	0.00	0.00
Number	28.55	9.58	7.91
Summaries	28.54	0.00	0.00
Other Baselines			
Bao et al. 2022 [7]	75.94	50.08	
Morio et al. 2022 [53]	75.54	55.66	42.30
Kawarada et al. 2024 Large [40]	77.75	56.06	
Kawarada et al. 2024 XL [40]	**78.51**	**56.80**	

Table 5.2: AAEC Performance Metrics by model and output type, in percentage points. Top results are in **bold**, and top performance in our experiments is <u>underlined</u>.

dataset, we again observe that FLAN-T5 fine-tuned with a full label strategy performs the best on relation identification and classification, while FLAN-T5 fine-tuned on summary labels performs the best on entity classification (see Table 5.2. However, we do not see performance improvements relative to established baselines.

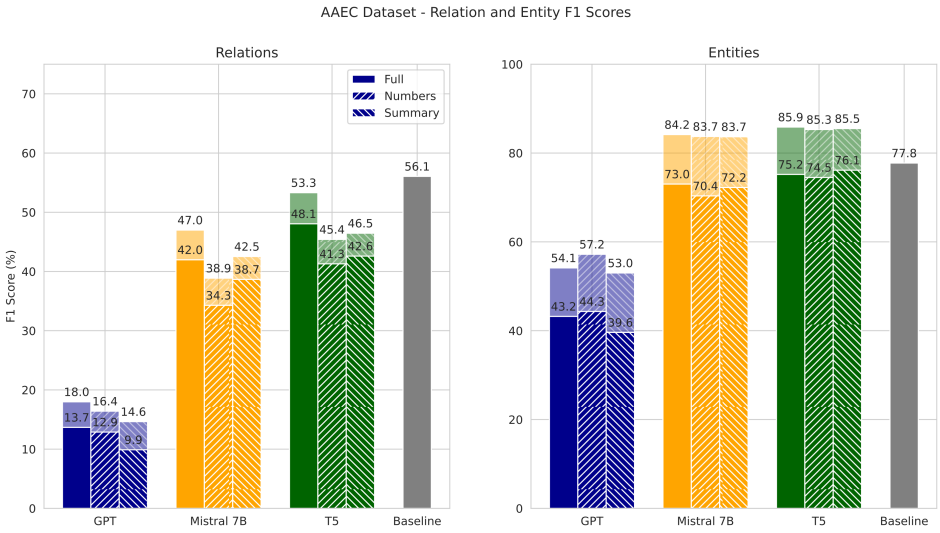


Figure 5.2: AAEC dataset, comparison of four models across three dimensions (few shot, seq2seq v. decoder-only, and labeling strategies). The baseline is Kawarada et. al 2024's FLAN-T5-Large finetune [40].

Along labeling strategies, we observe a slightly different pattern. While full text labels continue to perform the best, summary labels appear to narrowly outperform numerical labels, excluding BART (E: +.03%, RI: +1.23%, RC: +0.62%).

5.2.3 Comparing Encoder-Decoder Models

Between the two encoder-decoder models, T5 consistently outperforms BART on both relation and entity-based tasks. In the relation setting, BART models struggle to produce relations. As we will see in compliance metrics, this discrepancy occurs not because BART does not attempt to produce relations, but because the imprecision of its sentence reconstruction and entity references is such that sentences cannot be parsed.

The gap between the T5 experiments, the strongest-performing models, and BART, the worst-performing, suggests a discrepancy produced by instruction tuning. BART and FLAN-T5 are pretrained on similar, but distinct objectives – BART is trained to de-noise corrupted data, while T5 is trained to do masked language modeling

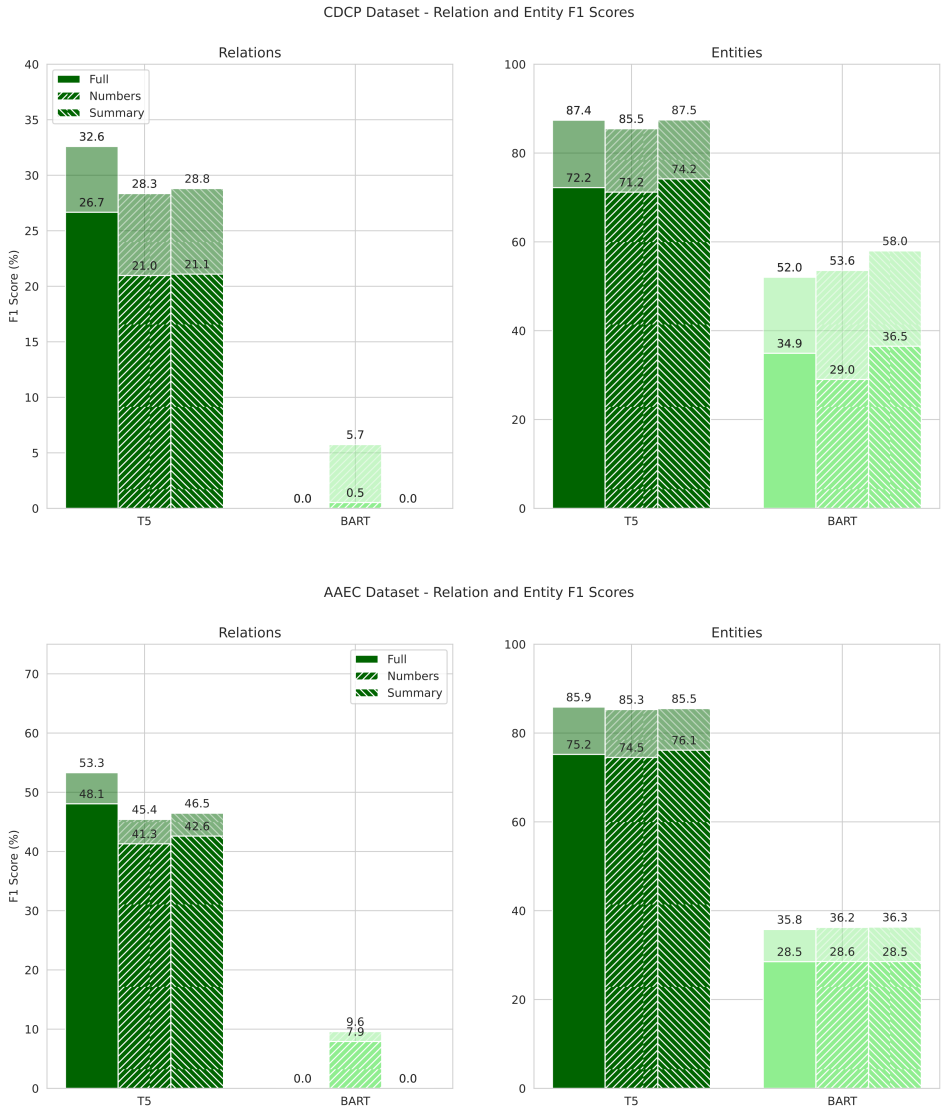


Figure 5.3: Comparing F1 Scores between BART and T5, CDCP (top) and AAEC (bottom)

on cloze-style reconstruction. Another key architectural difference is BART's use of absolute position encoding, as opposed to relative positional encoding. However, the vast divergence of performance on this task is likely due to FLAN-T5's fine-tuning on a variety of different instruction tasks. Instruction-following is a crucial step for tasks which require high-fidelity reconstructions such as generative structure prediction.

5.2.4 Comparing Decoder-Only Models

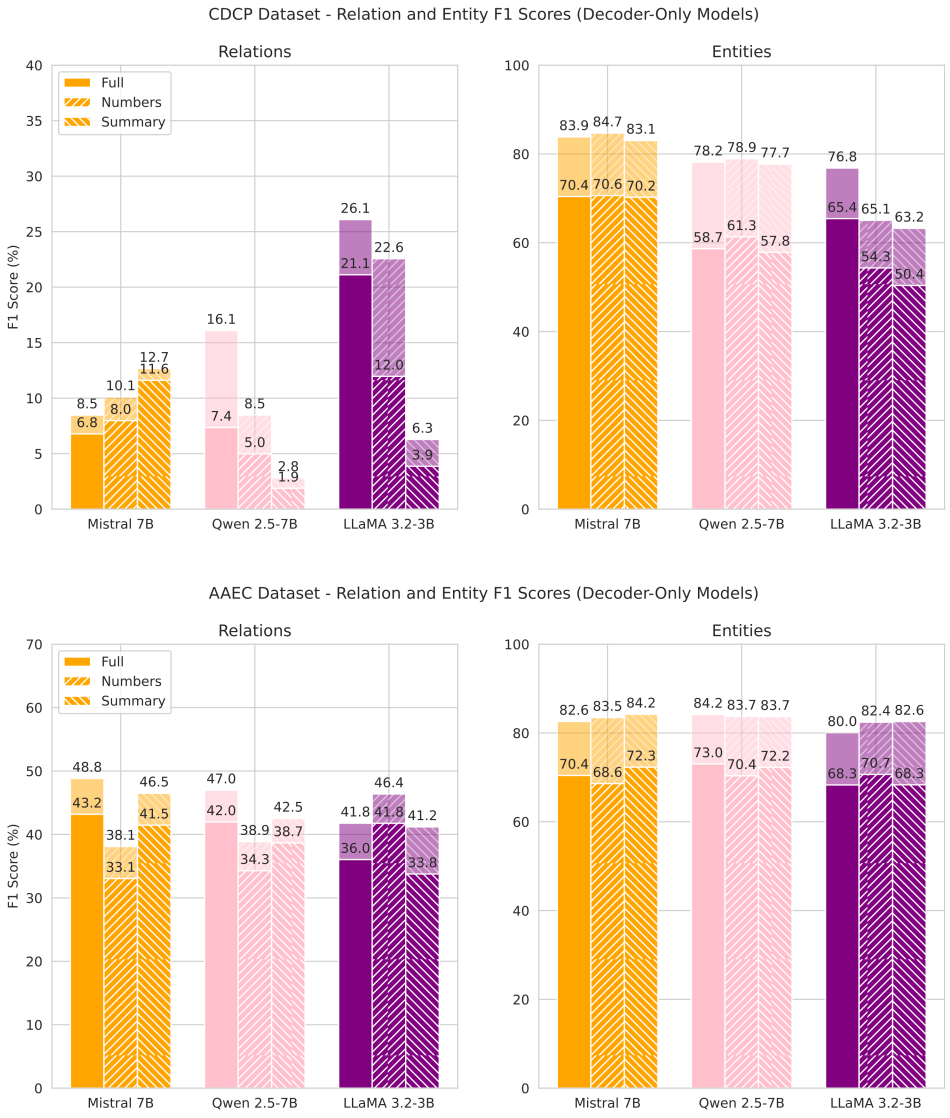


Figure 5.4: Comparing F1 Scores between Mistral, Qwen, and LLaMA on CDCP (top) and AAEC (bottom)

Among fine-tuned auto-regressive models, the picture is more nuanced. When comparing relation scores, we see that Mistral 7B acts as the strongest performer - and, in the case of CDCP labeling strategies, exhibits performance inverse to the general trend; summaries perform better than numbers, which in turn perform better than full labels.

In CDCP, model selection appears to have a large impact on F1 accuracy. LLaMA shows the strongest relation-based performance with full labels, but the effect does not carry to summary labels or entity-related tasks. Performance moderates substantially when compared to Qwen and Mistral (RI: +9.18%, +7.91%, respectively). Choice of labeling strategy also has a clear impact on relation-based task performance. For relation identification, full labels achieve an average of 3.16% higher F1 score than numeric labels and 9.26% higher than summary labels. The diversity of these impacts suggests that the bottleneck in CDCP, which is the relative scarcity of relations in the dataset distribution, can have diverse impacts on individual models as well as on labeling strategies. It is worth noting that Mistral completely inverts the labeling strategy expectations, performing better with numerical labels and best with summaries.

CDCP Relation Identification F1 Scores				
	Full	Num.	Sum.	**Avg.**
Mistral	8.47	10.11	12.66	10.41
Qwen	16.09	8.48	2.82	9.13
LLaMA	26.09	22.56	6.28	18.31
Avg.	16.88	13.72	7.25	

AAEC Relation Identification F1 Scores				
	Full	Num.	Sum.	**Avg**
Mistral	48.80	38.10	46.50	44.47
Qwen	47.00	38.90	42.50	42.80
LLaMA	41.80	46.40	41.20	43.13
Avg	45.87	41.13	43.40	

Table 5.3: Relation Identification F1 scores, averaged by model and strategy, for both CDCP and AAEC.

In CDCP, model selection appears to have a large impact on F1 accuracy. LLaMA shows the strongest relation-based performance with full labels, but the effect does not carry to summary labels or entity-related tasks. Performance moderates substantially when compared to Qwen and Mistral (RI: +9.18%, +7.91%, respectively). Choice of labeling strategy also has a clear impact on relation-based task performance. For relation identification, full labels achieve an average of 3.16% higher F1 score than numeric labels and 9.26% higher than summary labels. The diversity of these impacts suggests that the bottleneck in CDCP, which is the relative scarcity of relations

in the dataset distribution, can have diverse impacts on individual models as well as on labeling strategies. It is worth noting that Mistral completely inverts the labeling strategy expectations, performing better with numerical labels and best with summaries.

In AAEC, Mistral exhibits the strongest average performance across strategies, but the differences seem to be distributed across labeling strategies as opposed to across models (see Table 5.3). Performance of full labels beats summary labels (RI: +2.45%, RC: +2.43%) which in turn beats numerical labels (RI: +2.29%, RC: +1.60%). LLaMA curiously defies this, achieving the strongest relation performance in the numerical setting.

5.2.5 Oracle Setting

In the oracle setting, we compare the full text labeling strategy against an environment where a model input is half-augmented – the entities in the document are given and classified, and a model is only expected to produce relations. This ablation tests the power of the span identification upper bound on our datasets, attempting to quantify the deficit created by error propagation in the end-to-end nature of joint entity-relation argument mining.

Our findings are quite clear. We reveal that in most cases, providing entities improves model performance. However, the effect of improvement is different between the two datasets. In AAEC, providing entity scores nearly doubles the relation F1 scores (RI: average +38.1%). Understanding the difference of this effect requires an understanding of the AAEC dataset; in AAEC, not every span is argumentative, while in CDCP every word is accounted for in some argumentative discourse unit. Thus, using a simple sentence-level heuristic to segment CDCP would work much better than for AAEC. In short, the entity spanning problem is slightly harder in AAEC. Thus, the oracle setting removes the bottleneck for accurate relation identification, as

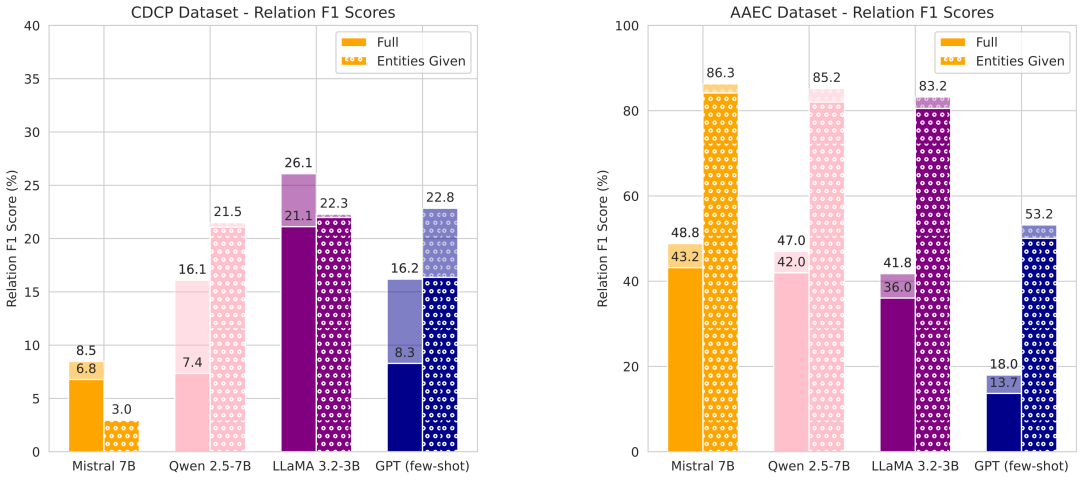


Figure 5.5: Relation classification/identification F1 scores for various models on CDCP (left) and AAEC (right). Relation identification in `alpha`=0.5 behind classification.

the relations defined by AAEC are clearer both in terms of relation types than for CDCP and more abundant in distribution (see Table 4.1). For CDCP, however, oracle-trained models show substantial improvements for only two of four models. In fact, for Mistral and LLaMA, performance degrades by 5.5% and 3.8% respectively. This curious finding reveals that CDCP's relation sparsity problem cannot be alleviated by exposing a model to relations. Given that 100% of the text in the dataset is featured in one ADU or another, it should be unsurprising that clarifying entity spans would not necessarily resolve fundamental performance gaps, even if performance on entity identification is roughly equivalent for CDCP and AAEC.

5.3 Compliance

Now that we have considered the accuracy of model predictions using F1 scores as analogues of general classification performance, we ought to consider the second class of "compliance" metrics defined in Chapter 3. As a reminder, we incorporate four metrics in our analysis of compliance: reconstruction error, label error, entity error,

and format error. Each metric is a simple 0/1 value for each output documented evaluated, signifying whether such an error type was present.

The quick brown fox jumps over the lazy dog.
The [quick brown fox | animal | jumps = dog] jumped over the [lazy dog | dog |].

Figure 5.6: An example of the four error types
, where *animal* is the only valid entity type and *jump* is the only valid relation type. Entity error in red, reconstruction error in yellow, label error in blue, and format error in green.

5.3.1 CDCP Dataset

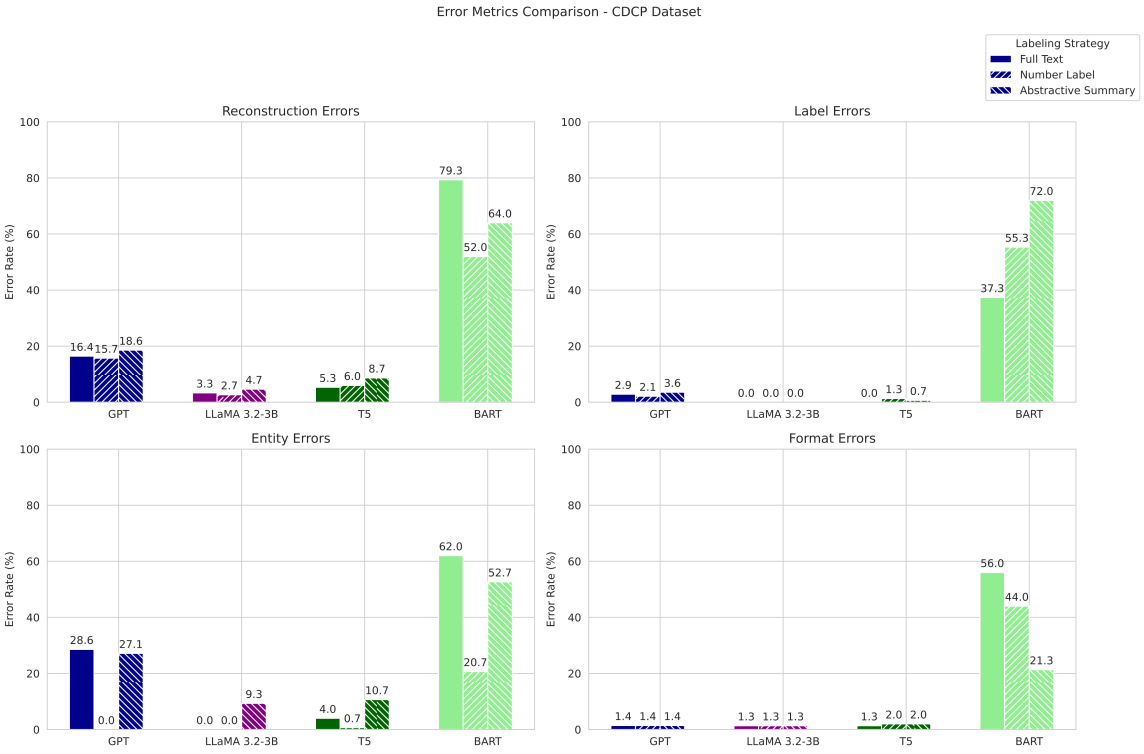


Figure 5.7: Analysis of Error Metrics across top-performing CDCP models of each type, with BART included for reference.

The results depicted in the CDCP dataset establish a set of intuitions regarding compliance metrics. We will see a repetition of these patterns compared against the AAEC dataset. Some crucial observations:

CDCP Original (ex. #86)	CDCP GPT Reconstructed Output (ex. #86)
... If is the former, some basic information relative to the account – contract, itemized statements, or any other identifying information that only the creditor would have access to would suffice. If CFPB wants "verify" to answer every single objection a debtor can cook up, **a collector could never move forward without spending hundreds of hours normally reserved for litigation**. As to a time frame on answering, I think the current system of stopping collection action until verification is provided is appropriate...	... If is the former, some basic information relative to the account – contract, itemized statements, or any other identifying information that only the creditor would have access to would suffice. If CFPB wants "verify" to answer every single objection a debtor can cook up, [**MISSING**] As to a time frame on answering, I think the current system of stopping collection action until verification is provided is appropriate...

Figure 5.8: Example of Reconstruction Error in `gpt-4o-mini` full text outputs, CDCP. [MISSING] indicates where the bold text is not included.

1. **Fine-tuned models exhibit higher compliance**. Unsurprisingly, one of the most consistent signals a model might learn in from cross-entropy loss is the exact replication of input sequences and other annotation schema. This makes compliance a principal initial goal to achieve during the supervised fine-tuning. The only exception to this is BART.

2. **Label strategies do not impact compliance**, except when looking at entity error. In this case, numerical labels are much less likely to yield entity errors because the entities referenced at the tail of relations are simple number tags as opposed to repetitions of an entire input sentence or summary labels. If the parsing algorithm requires strict adherence to labels for a functioning lookup table, it becomes much easier to index entities by number labels – even if on accident.

3. **BART is quite bad at TANL.** These figures give some explanation for the poor performance of fine-tuned BART on structure prediction and generative argument mining. As we will explore in case studies, BART frequently "remixes" inputs, returning semantically related sentences which are never exact and rarely even preserve the same semantic meanings. Per empirical experiments, increasing

the number of fine-tuning epochs does not necessarily encourage discovery of replication for loss reduction. This begs the question of fundamental pretraining research around structure prediction, such as that done by KnowCoder and DeepStruct [46, 96] to teach non-instruct-tuned models how to comply with the format of TANL.

CDCP Original (ex. #86)	CDCP BART Reconstructed Output (ex. #86)
... It contained nothing I recognized. You are presuming they are actually validating. This is the biggest mistake. They do none of that. Any rule that defines what qualifies as a dispute would help. The clearer, the better. **The** vagueness now has led to abuses. Such as "frivolous and irrelevant" disputes. **That is a license to abuse consumers.** ...	... " It contained nothing I recognized. You are presuming they are actually validating. **This is the process** They do none of that. Any rule that defines what qualifies as a dispute would help. The clearer, the better. **THE** vagueness now has led to abuses. Such as "frivolous and irrelevant" disputes. **No, I was not being sarcastic. You are correct. The phrase "frivolous and irrelevant" suggests the author is expressing frustration with the process, but I didn't receive the actual passage from you. If you'd like to share the passage, I'd be happy to help you understand its meaning and provide a label for the correct reason why it's not clear what constitutes a dispute, specifically the use of a process server to deliver the summons and complaint, which is a requirement in most jurisdictions to ensure that the person providing the notice is properly served.** ...

Figure 5.9: Example of Reconstruction Error in `bart-large` full text outputs, CDCP. Bold text signifies differences from the original.

5.3.2 AAEC Dataset

When evaluating compliance under AAEC training environments, we see across-the-board performance gaps, likely attributable to the difference in size between AAEC and CDCP (a factor of about 3.02:1 tokens). Longer sequences increase risk of compliance errors as more tokens inherently increase the likelihood of lower performance.

Beyond this finding, we see similar performance patterns as shown in CDCP, with

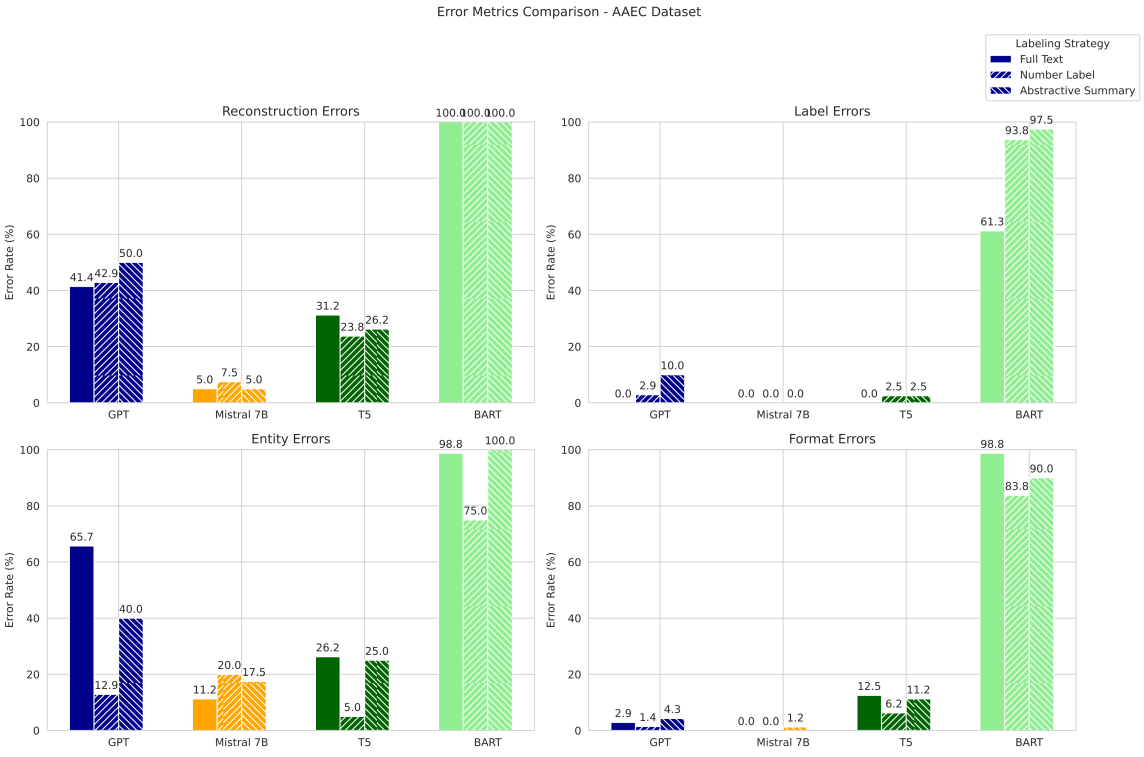


Figure 5.10: Analysis of Error Metrics across top-performing AAEC models of each type.

the curious exception of T5. T5 has a higher (avg. 21%) degree of reconstruction errors than Mistral, as well as notable entity and format errors. Despite this, we see that T5 achieves the highest accuracy on AAEC. We proceed with a case-by-case analysis of compliance failures in order to examine the different negative implications of model performance.

T5

When parsing examples of failed reconstruction in T5, some consistencies emerge. For instance, most reconstruction errors involve sentence- and phrase-level omissions and repetitions. These errors are more likely to happen at the end of a sentence than at the beginning. In Figure 5.11, the last sentence is repeated several times instead of continuing to copy the original input. Many examples of this are present in AAEC towards the end of documents. Other times, the copy will end abruptly in the middle

AAEC Original	Reconstructed Output
... Anyone can meet this problem once in any conversation. E - mail helps me arrange my idea perfectly and make it persuasive. On the whole, to compare to all ways of communication, I really think that e - mail is the greatest way for major people to use. Its convenience saves our time. Additionally, honesty is necessary for every conversation and also fluency can't be missing for a persuasive expression.	... Anyone can meet this problem once in any conversation. E - mail helps me arrange my idea perfectly and make it persuasive. On the whole, **E - mail helps me arrange my idea perfectly and make it persuasive. On the whole, E - mail helps me arrange my idea perfectly and make it persuasive**

Figure 5.11: Example of Reconstruction Error in `FLAN-T5-large` full text outputs, AAEC. (ex. #50)

and simply exclude the rest of the original document. However, when all content is included it is unlikely to be modified, which cannot be said of BART. This perhaps explains why T5 could have both high compliance errors and relatively strong accuracy scores. Since mutations do not significantly modify individual text, the compliance gaps are not zero-sum with effective parsing and might be improved with greater fine-tuning or pretraining on structure prediction.

BART

On the other hand, BART experiences cascading errors in the AAEC dataset that make the text difficult to parse 5.12. BART is more prone to reconstruction errors, which are semantic mutations of the text as well as omissions or repetitions of earlier text. These semantic mutations are magnified when entities are repeated as relation tails, with entities more likely than not to have several words replaced, sometimes changing the meaning of the entity entirely. These entity errors make it difficult to reconstruct relations, and as such BART is logged as making 0 predicted relations even when many are attempted in the output format. Beyond this, BART also creates a significant quantity of format errors, which make the text difficult to parse and often cause the reconstructed sentence to include many artifacts of the parsing process.

While each problem might be treated in isolation with relaxations of the parsing algorithm, together these make BART outputs difficult to work with.

Again, such an issue could arise in the lack of instruction tuning in BART models, or it could be a consequence of BART's training objective, which requires denoising inputs from corrupted text as opposed to cloze-style masked language modeling.

5.4 Discussion and Analysis

The model fine-tuning experiments conducted on the CDCP and AAEC datasets reveal curious findings regarding the right "knobs" to turn to improve model performance in generative argument mining. The composite of results suggest some trends along the dimensions identified in our approach, contesting the general literature direction in natural language processing which favors decoder-only models.

1. **Labeling Strategies**: we have established that full labels tend to produce higher accuracy scores for the relation classification and identification tasks, with limited effects on the entity classification task. Past this, however, a pattern does not emerge between the two datasets. CDCP data suggests that numerical labels clearly outperform summary labels, while AAEC muddles such a distinction, suggesting that summary labels outperform, albeit by a narrow margin.

2. **Encoder-Decoder v. Decoder-Only Models**: on both datasets, we find that the strongest performance comes from encoder-decoder model architectures, something to unpack further in the discussion and analysis. Among decoder-only models, CDCP would suggest that LLaMA is superior among models tested, while AAEC demonstrates very little distinction along model family. Further testing with other decoder-only models might be necessary to establish definitive claims. Additionally, a crucial prerequisite to effective performance is instruction fine-tuning. As we see with BART, despite having the supposedly superior

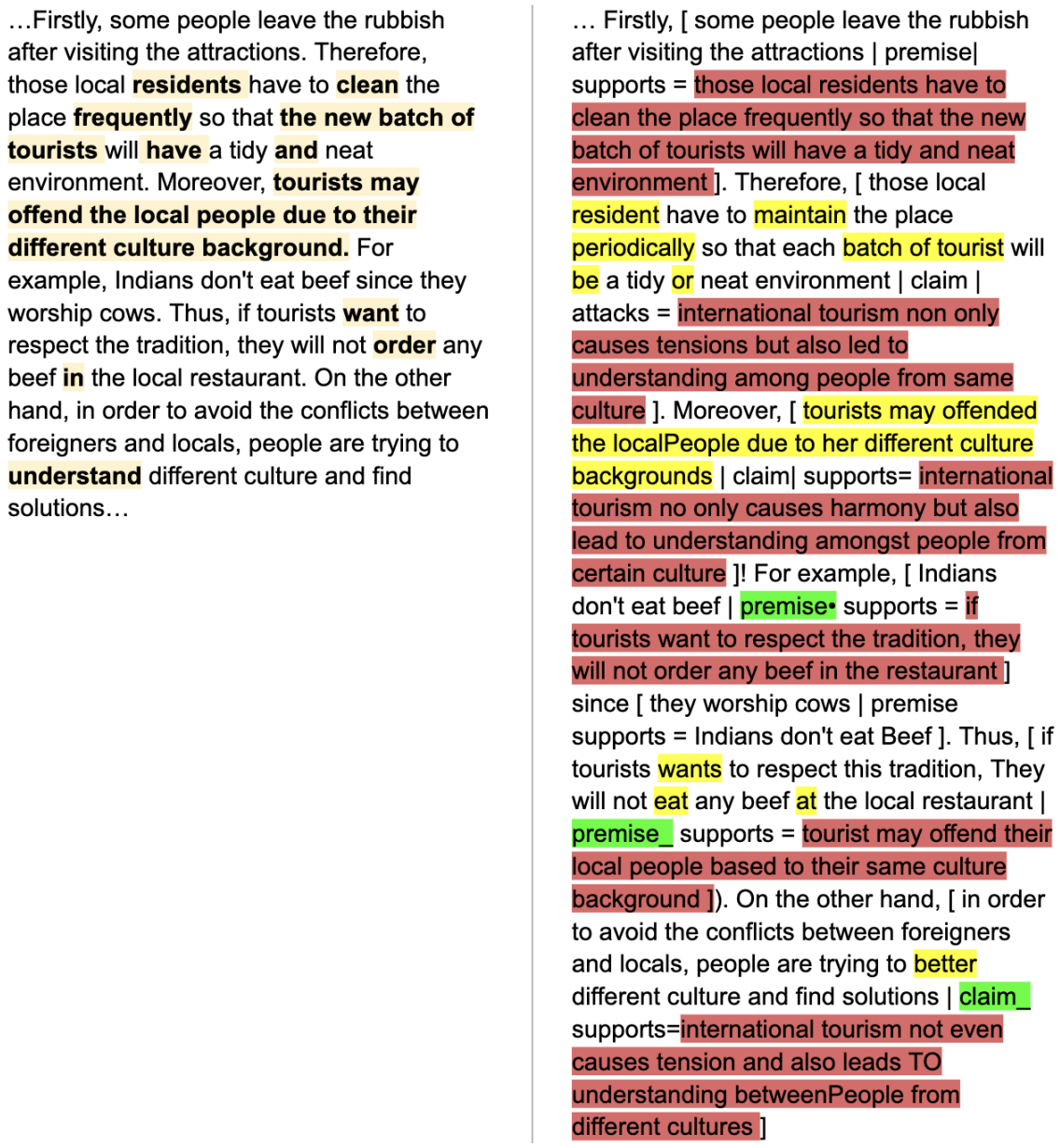


Figure 5.12: Multiple error types are present in attempting to reconstruct AAEC ex. 73 outputs with BART (left: original, right: reconstructed). Reconstruction errors are highlighted in yellow, format errors in green, and entity errors in red.

architecture, the pretraining paradigm does not prepare the model to comply with augmented natural language formats and reproduce input texts, making outputs too dissimilar from inputs for coherent analysis.

3. **Fine-Tuning v. Few-Shot**: we find a clear case for supervised fine-tuning

compared to few-shot use of pretrained language models. Decoder-only models are more suited for few-shot contexts, as encoder-decoder models require additional configurations and tuning for few-shot applicability. Among decoder-only models, most fine-tuned models outperform 10-shot `GPT-4o-mini` across metrics and labeling strategies. This is despite `GPT-4o-mini` training on more data and with more parameters than open-source, mid-size equivalents. We see this effect especially when comparing compliance of models

4. **End-to-end v. Relations-only**: Clearly, making the problem easier for generative models by including spanned entities as input would at least not degrade performance significantly in models. However, we found that the effect of the "oracle setting" was moderated for one dataset relative to the other. While models compared under AAEC consistently saw F1 scores double in the oracle setting, the CDCP dataset did not have the same effects, and some models in fact performed worse in oracle setting. This suggests differences in the problem spaces defined by the datasets and the bottlenecks they introduce, revealing the key differences in these two benchmarks.

5.4.1 Prediction of Relations v. Accuracy

Between encoder-decoder and decoder-only models, a performance in relation-based tasks might arise from the sheer number of relation predictions a model may or may not make. We find that, when training on the CDCP dataset, there is a strong correlation between the number of predictions which are made and the overall F1 score of relations. This suggests that some models, in particular those in the decoder-only family, are less likely to produce a relation annotation when decoding an output sequence, placing an overall cap on their performance (see Figure 5.13.)

Such a hypothesis requires further testing at the logit-level of predictions. Namely, an experiment could compare the next-token probability of the token which comes

before a closing bracket. If the next-token probability is, on aggregate, more likely to be a closing bracket] than a delimiter |, this would suggest that language models are less likely to assign a relation to a tag, perhaps due to the relative sparsity of relations compared to entities. This would present a problem unique to the generative argument mining context as the next-token probabilities might be controlled by the distribution present at train-time. A potential solution could devise a custom loss function which encourages higher production of relations.

5.5 Conclusion

Our analysis confirms the promising capabilities of pretrained language models to act as annotators for structure prediction and argument mining. Such generative, end-to-end systems require only one pass in order to produce annotations akin to classification. Our investigations build upon previous research in generative modeling for structure prediction, advocating for the unique position of argument mining as a case study for long-context entity-relation extraction as well as an intrinsically beneficial system. We define four dimensions of possible configurations upon which to test further models. Experiments along these four dimensions establish that encoder-decoder model architectures outperform newer, larger decoder-only models, despite being more culpable of incorrectly reconstructing input sequences. Additionally, we confirm that fine-tuning models remains an essential component to ensure their success in behaving as classifiers to encourage the production of schema-adherent outputs.

Our results demonstrate performance superior or comparable to several state-of-the-art models which use classification-based systems in order to identify relations between arguments. Further research is necessary to validate our empirical findings along the four dimensions identified, with promising research directions for training and synthetic data techniques to improve bottlenecks to performance in decoder-only

Model	CDCP	AAEC
GPT4o-few-shot		
Full	297	939
Number	399	1134
Summary	192	1100
Relations-only	475	884
LLaMA 3.2-3B-Instruct		
Full	159	1117
Number	26	1065
Summary	140	1130
Relations-only	123	313
Mistral 7B-Instruct		
Full	30	1176
Number	52	1071
Summary	55	1115
Relations-only	14	1184
Qwen 2.5-7B-Instruct		
Full	111	1125
Number	77	1124
Summary	101	1090
Relations-only	141	1184
FLAN-T5-Large		
Full	216	1099
Number	191	1139
Summary	169	1091
BART-Large		
Full	0	0
Number	59	129
Summaries	0	0
True	324	1186

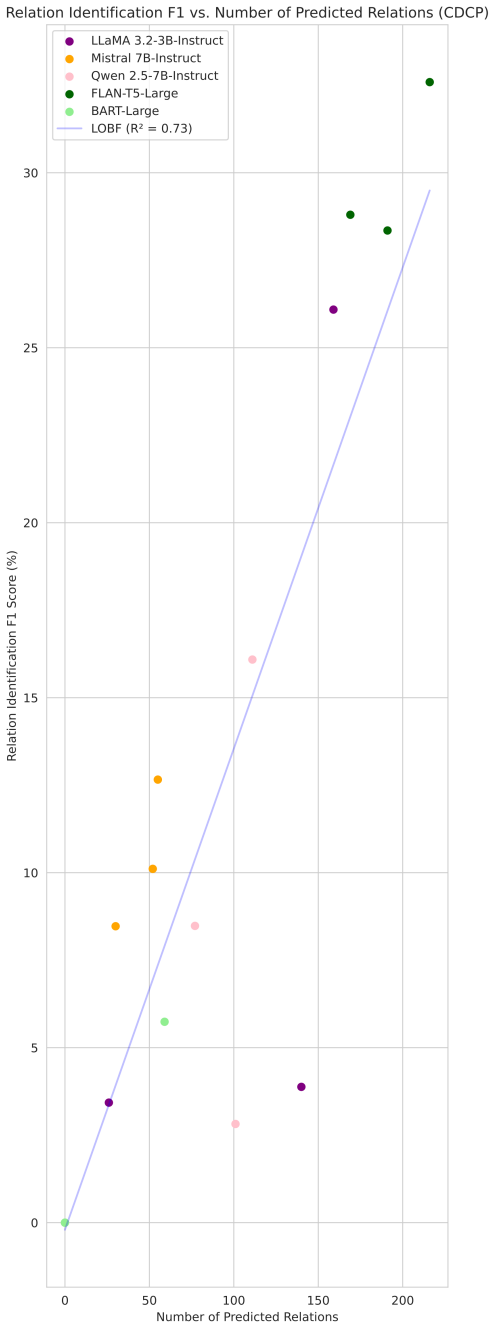


Figure 5.13: Left: a table of the number of predicted relations in each dataset. Right: a plot comparing relation identification scores with number of predicted relations (CDCP)

models.

Chapter 6

Conclusion

6.1 Across Four Dimensions

6.1.1 Research Goals

In this work, we investigate the subtask of argument mining as a case study for scaling generative modeling capabilities towards end-to-end systems for predicting structure in natural languages. Contemporary research is increasingly focused on using large, pretrained language models to address these challenges. This evolution marks a shift from traditional pipeline systems towards end-to-end trainable models, which aim to capture the complexities of argumentative discourse in a more unified and efficient manner. Such work has critical implications for natural language understanding, unstructured text mining, knowledge graph construction, and a litany of other fields. Argument mining is a crucial context to situate such goals; it requires models that are robust to scaled sizes of documents and entities, as well as models that quickly learn subjective categorization taxonomies and reproduce them on unseen data samples. Pretrained language models exhibit such characteristics and have begun to see applications in argument mining, but such an investigation has yet to find grounding on key empirical considerations - what this work dubs as "dimensions" of configuration.

We take as our starting point previous research that establishes the framework for generative structure prediction [59], create perturbations of the framework based on different annotation strategies useful for argument prediction, and retrofit the training environment for applicability to newer model architectures.

The key contribution of this work is the declaration and investigation of four such dimensions: labeling strategies for argument annotations, model architectures, fine-tuning versus in-context learning, and end-to-end versus relation-based settings. We argue that these dimensions form a span of future research trajectories, laying the groundwork for further exploration of generative argument mining, and by extension, structure prediction. As defined in this work, we use three different labeling strategies, five open-source pre-trained models across two architectures, one proprietary model for in-context learning, and two settings for task formulation. This presents 36 combinations for exploration, of which 25 configurations were tested.

Another key contribution is to identify a new set of evaluation metrics for understanding generative structure prediction. Structure prediction is usually framed as a classification task, and thus results are exclusively presented through conventional classification metrics like precision, recall and F1 (dubbed "accuracy" metrics). We introduce a set of metrics designed to capture how well generative models behave as classifiers, or their "compliance" with structure prediction. This set of metrics allows us to understand the impact that our four dimensions and tested datasets might implicate the performance of models, offering greater visibility into performance failures which are a consequence of incorrect *generations* as opposed to incorrect *classifications*.

6.1.2 Key Findings

Across our experimental configurations, we find a variety of results which suggest avenues for further research. We sort the dimensions by their relevance to performance

based on our experiments.

Dimension 1: Fine-Tuning v. In-Context Learning

Along our four dimensions, we identify fine-tuning as the most significant determinant of generative argument mining capabilities. Models expected to do structure prediction in a few-shot, in-context environment tend to exhibit worse performance on both accuracy and compliance metrics, suggesting that they are not only incapable of meeting the prerequisite requirements of the task but also poor classifiers even when producing coherent outputs. Smaller pretrained language models fine-tuned on existing data samples beat larger, proprietary, API-served systems by a considerable margin. Future research ought to challenge this empirical finding by scaling in-context learning to incorporate more examples or contrastive learning.

Dimension 2: Choice of Model Architecture

The dimension of model architecture serves as the next significant determinant of generative argument mining capabilities. Specifically, we find that encoder-decoder models (T5 in particular) outperform larger decoder-only models when fine-tuned in similar training environments. This suggests that the architecture of encoder models enables better bi-directional attention. Decoder-only models tend to produce fewer relations than encoder-decoder models, and we find some correlation between the number of relations predicted and the relation-based scores of models. This effect might be remedied with loss functions designed to encourage production of relations or the inclusion of a trainable reward model. Surprisingly, we find that encoder-decoder models exhibit lower compliance, with higher error rates for reconstruction of original inputs and pointers to non-existent entities in relation tags. These errors appear towards the end of model outputs, suggesting that encoder-decoder models struggle to scale relative to decoder-only models. While one encoder-decoder model (T5) still

exhibits higher accuracy despite these errors, another model (BART) yields outputs which cannot be parsed, bleeding into accuracy and preventing almost every generated relation from even counting as a prediction.

Choice of model does not have the same staggering effect when focusing exclusively on decoder-only, open-source models. Between the three tested, `LlaMA-3.2-3B-Instruct` performs the best on the CDCP dataset, while `Mistral-7B-Instruct` narrowly outperforms others on AAEC. These models are pretrained with a similar next-token prediction task, which might explain their relative uniformity when fine-tuned on specific datasets. Some experimental configurations reveal large differences in decoder-only models, while others do not.

Dimension 3: End-to-End v. Relation Only

One of our dimensions relaxes the requirements of the problem by introducing an "oracle setting" with segmented, classified argument entities as inputs. This is key to testing the impact of cascading errors from an end-to-end system. If an entity is incorrectly spanned, it would be impossible for a model to correctly identify any of its relations by the definition of relation accuracy we use. Naturally, we expect such a relaxation would improve performance by some margin; indeed, performance on relation-based tasks tends to improve when entities are given as input, though the impact of improvements are clearly partitioned by dataset. This reveals important implications regarding the relative bottlenecks in each benchmark which make the argument mining task difficult in different ways.

Further research might continue to explore this as an end-to-end system by using two calls to a model as opposed to one, incorporating gradients from both components of the task individually as opposed to the joint composition we use.

Dimension 4: Labeling Strategies

We anticipated that the axis of labeling strategies would bear large implications for model performance as they truncate augmented natural language outputs, creating shorter contexts that might improve model performance or more parse-able text. However, we find that truncated labeling strategies such as numbers and summaries perform worse than full text labels. In terms of compliance, all strategies perform similarly, with numerical labels yielding lower entity error because numerical labels are the easiest to index.

6.2 Implications for Argument Mining

This work situates itself among emerging research in the argument mining space which incorporates large, pretrained models in structure prediction [20, 53, 7, 25]. However, the models used are often encoder-only models which are positioned in bespoke architectures with classification heads in order to improve performance. Even in instances where generative models are used, they are often configured such that many model calls must be made for any document to first identify entities, then classify them, then do pairwise comparisons between identified entities to determine what relation might exist between them. In comparison, our approach uses a simple one-pass framework that enables efficient structure prediction that scales easily to long documents with diverse entities and relations. We show that such a framework can beat state-of-the-art systems through certain experimental findings.

While some research has applied similar augmented natural language frameworks to argument mining [40], it does not establish the four dimensions of configuration which serve as horizons for future experiments. In contrast, our work lays out clear directions for exploration by incorporating a variety of different model types and label strategies which invite deeper investigations bridging the gap between generative

structure prediction models and the argument mining subtask. Additionally, our introduction of compliance-based metrics for generative argument mining highlights important considerations for treating generative models as classifiers, providing a more nuanced understanding of their performance.

6.3 Implications for Structure Prediction

Our work carries significant implications for the broader task of structure prediction in natural language. By framing argument mining as a structure prediction problem and exploring the capabilities of generative language models within this context, we offer valuable insights into how these models can be effectively leveraged for other tasks that involve the identification and extraction of structured information from text. The dimensions that we investigate – labeling strategies, model architectures, fine-tuning versus in-context learning, and end-to-end versus relation-based settings – are not only relevant to argument mining but also have broader applicability to other structure prediction tasks. Our work is inherited from and contributes to the rich legacy of methods and approaches in the context of named entity recognition and relation extraction. In this sense, argument mining is understood as a joint entity-relation extraction problem with unique characteristics due to its size and relatively subjective nature. The case study provides important implications that challenge conventional literature.

Considering the compliance scores of encoder-decoder models, we can see why such a case study would be important. We isolate examples of poor compliance in long-context documents and observe their frequency towards the end of entity inputs. The context length of reconstructed outputs tends to exceed that of usual benchmarks in joint entity-relation extraction, and as such the argument mining context exposes breakdowns in long-context limits for the older encoder-decoder models. Further

research ought to explore the scaling limits of such models in order to make the case for using long-context decoder-only models to generate augmented natural language.

On the other hand, the choice of labeling strategy significantly impacts how entities are identified and classified. Our research suggests that while truncated labeling strategies might seem advantageous for solving the long context problem, full text labels often lead to better performance even in the face of compliance problems. This insight can help guide the development of more effective NER systems that rely on generative models. Similarly, the comparison between fine-tuning and in-context learning has direct implications for NER. Our findings indicate that fine-tuning generally outperforms in-context learning, highlighting the necessity of task-specific training for achieving high accuracy in NER tasks.

Furthermore, our emphasis on compliance-based metrics has significant implications for all structure prediction tasks. Traditional evaluation metrics, such as precision, recall, and F1-score, may not fully capture the nuances of generative models' performance in structure prediction. Our introduction of compliance metrics, which assess a model's adherence to the structural requirements of the task, provides a more comprehensive evaluation framework. This framework can be applied across various structure prediction tasks to gain a deeper understanding of model strengths and weaknesses.

In conclusion, our research provides a valuable contribution to the broader field of structure prediction. By thoroughly examining the impact of labeling strategies, model architectures, fine-tuning versus in-context learning, and end-to-end versus relation-based settings, we offer insights that can be applied to various tasks involving the extraction of structured information from text. Our emphasis on compliance-based metrics further enhances the evaluation of generative models, providing a more nuanced understanding of their performance in structure prediction.

6.4 Future Work

The nature of our work is such that it implies a wide gamut of future directions for research. Here, we articulate those directions which we find critical for further exploration.

6.4.1 Encoder-Decoder v. Decoder-Only Models

We find that the choice of model architecture carries significant implications for both accuracy and compliance of generated outputs. A key limitation of decoder-only models appears to be the under-representation of relations in auto-regressive prediction of entity tags. Without a cross-attention module which considers the latent state representation of a full input, it appears as if these models exaggerate the relative sparsity of relations. This is reflected in Figure 5.13, where the number of relations predicted is correlated to F1 scores of relations themselves. The under-prediction problem ought to be diagnosed at greater depth and resolved with creative training strategies.

Diagnostically, a logits-based analysis can determine whether under-prediction of relations is a consequence of next-token probabilities. Future work might isolate the next-token likelihood of a closing bracket] or a delimiter | after a tag has been completed. At that position, a delimiter would imply a relation as opposed to a closing bracket, so aggregating logit probabilities in different circumstances might yield an empirical confirmation of the effect observed.

Procedurally, improvements for decoder-only models could be made in a variety of contexts. Creative sampling procedures with configuration of hyperparameters like temperature might also help to improve inference-time performance without changing the supervised fine-tuning procedure. Alternatively, train-time modifications might be considered, such as a custom loss function which considers the number of

relations expected versus generated could help improve under-prediction, or a different training scheme entirely which employs a reward model and reinforcement learning techniques for controllable generation. The ensemble of these methods could improve the performance of decoder-only models. Finally, we consider the use of synthetic data, or language-model-generated data samples with relevant annotations, as a proxy for simulating increased relation patterns along generative models [34, 98]. Such an approach would shift the distribution of relations as present in hand-annotated benchmarks like CDCP, addressing a potential bottleneck which occurs in certain language environments.

Beyond under-prediction, ablation experiments using T5 might be used to determine how necessary the encoder module truly is to T5's performance. For instance, an experiment might freeze the weights of the encoder during fine-tuning to determine how crucial it is for overall model performance.

Finally, novel pretraining techniques might be incorporated in order to improve model compliance across the board as a pre-requisite to argument mining. While instruct models trained on following user requests perform best for the structure prediction task, the instruct pretraining rarely considers structure prediction or augmented natural language as part of its objective, due to their scarcity. Thus, incorporating ANL as a pretraining task might be an important consideration.

6.4.2 Further Model Dimensions

There are other dimensions of model selection which might be considered when expanding the search for optimal models within a certain architecture.

For instance, current literature recognizes that model size has important implications for performance, citing empirical effects as well as theoretical scaling laws. Model size is not extensively tested in this work due to limitations in the experiment environment, as some models with hundreds of billions of parameters are cost-prohibitive to

download, train and test. This direction might be necessary to establish new directions in model search.

Expanding the choice of model families is another area of consideration. While we test two encoder-decoder models and three decoder-only models, many other model families exist which might have a unique configuration suiting them for structure prediction.

6.5 Concluding Remarks

Our research has rigorously examined the potential of pretrained language models (PLMs) to serve as effective annotators for complex tasks such as structure prediction and argument mining. The generative, end-to-end systems we developed demonstrate a significant advantage: they can produce annotations that rival traditional classification-based approaches, but with a single, streamlined pass. This eliminates the need for multiple iterative steps, simplifying the annotation process and potentially accelerating it. Our investigation builds upon the burgeoning field of generative modeling for structure prediction, but we specifically highlight argument mining as a compelling and unique case study. This domain presents a challenging yet valuable arena for exploring long-context entity-relation extraction, given the intricate and often lengthy dependencies between argumentative components. Moreover, the intrinsic benefits of an automated argument mining system—such as enhanced critical thinking analysis and improved information retrieval—further underscore its importance.

To systematically evaluate the performance of PLMs in this context, we defined four critical dimensions of possible configurations. These dimensions encompass variations in model architecture, fine-tuning strategies, input representation, and output decoding methods. By meticulously exploring these dimensions, we aimed to uncover the optimal configurations for achieving high-quality annotations. Our experimental findings

revealed a notable trend: encoder-decoder model architectures, such as T5, consistently outperformed newer and larger decoder-only models like GPT-3, particularly when tasked with generating structured outputs. This result is significant, as it suggests that the bidirectional encoding capabilities of encoder-decoder models are crucial for capturing the complex relationships inherent in argument structures. Interestingly, we observed that encoder-decoder models were more prone to incorrectly reconstructing input sequences, a phenomenon that warrants further investigation. Despite this, their superior performance in relation extraction and annotation generation was undeniable. Furthermore, our experiments provided strong evidence that fine-tuning remains an indispensable step in adapting PLMs for classification-like tasks. This process is essential for guiding the models to produce outputs that adhere to predefined schemas, ensuring the validity and usability of the generated annotations.

Our empirical results demonstrate that our generative, end-to-end systems achieve performance levels that are either superior or comparable to several state-of-the-art models that rely on traditional classification-based systems for identifying relations between arguments. This achievement underscores the potential of PLMs to revolutionize annotation processes in argument mining and related fields. However, we acknowledge that further research is necessary to validate our findings across a broader range of datasets and model architectures. Specifically, we advocate for continued exploration of the four dimensions we identified, with a focus on refining model architectures, developing more effective fine-tuning techniques, and optimizing input and output representations. Promising research directions include the development of novel training methodologies and the exploration of synthetic data generation techniques to address the performance bottlenecks observed in decoder-only models. By pursuing these avenues, we can unlock the full potential of PLMs for automated annotation, paving the way for more efficient and accurate argument mining systems.

Bibliography

[1] Appendix C: Named entity task definition (v2.1). In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*, 1995. URL https://aclanthology.org/M95-1024.

[2] Oshin Agarwal, Mihir Kale, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. Machine Translation Aided Bilingual Data-to-Text Generation and Semantic Parsing. In Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina, editors, *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 125–130, Dublin, Ireland (Virtual), December 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.webnlg-1.13.

[3] Abdullah Al Zubaer, Michael Granitzer, and Jelena Mitrović. Performance analysis of large language models in the domain of legal argument mining. *Frontiers in artificial intelligence*, 6:1278796, 2023.

[4] Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1558–1569, Online, July 2020. Association for Com-

putational Linguistics. doi: 10.18653/v1/2020.acl-main.142. URL https://aclanthology.org/2020.acl-main.142.

[5] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In Chengqing Zong and Michael Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1034. URL https://aclanthology.org/P15-1034.

[6] Dhananjay Ashok and Zachary C. Lipton. PromptNER: Prompting For Named Entity Recognition, June 2023. URL http://arxiv.org/abs/2305.15444. arXiv:2305.15444 [cs].

[7] Jianzhu Bao, Yuhang He, Yang Sun, Bin Liang, Jiachen Du, Bing Qin, Min Yang, and Ruifeng Xu. A generative model for end-to-end argument mining with reconstructed positional encoding and constrained pointer mechanism. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10437–10449, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.713. URL https://aclanthology.org/2022.emnlp-main.713/.

[8] Jérémie Cabessa, Hugo Hernault, and Umer Mushtaq. In-context learning and fine-tuning gpt for argument mining, 2024. URL https://arxiv.org/abs/2406.06699.

[9] Xavier Carreras and Lluís Màrquez. Introduction to the CoNLL-2005 shared

task: Semantic role labeling. In Ido Dagan and Daniel Gildea, editors, *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL https://aclanthology.org/W05-0620.

[10] Lucas Carstens and Francesca Toni. Towards relation based argumentation mining. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 29–34, 2015.

[11] Artem Chernodub, Oleksiy Oliynyk, Philipp Heidenreich, Alexander Bondarenko, Matthias Hagen, Chris Biemann, and Alexander Panchenko. TARGER: Neural argument mining at your fingertips. In Marta R. Costa-jussà and Enrique Alfonseca, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 195–200, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-3031. URL https://aclanthology.org/P19-3031/.

[12] Jason P.C. Chiu and Eric Nichols. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4: 357–370, 2016. doi: 10.1162/tacl_a_00104. URL https://aclanthology.org/Q16-1026.

[13] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling

instruction-finetuned language models, 2022. URL https://arxiv.org/abs/2210.11416.

[14] Oana Cocarascu and Francesca Toni. Identifying attack and support argumentative relations using deep learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1374–1379, 2017.

[15] Enyan Dai, Tianxiang Zhao, Huaisheng Zhu, Junjie Xu, Zhimeng Guo, Hui Liu, Jiliang Tang, and Suhang Wang. A Comprehensive Survey on Trustworthy Graph Neural Networks: Privacy, Robustness, Fairness, and Explainability, September 2023. URL http://arxiv.org/abs/2204.08570. arXiv:2204.08570 [cs].

[16] Zhenyun Deng, Michael Schlichtkrull, and Andreas Vlachos. Document-level Claim Extraction and Decontextualisation for Fact-Checking, June 2024. URL http://arxiv.org/abs/2406.03239. arXiv:2406.03239 [cs].

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[18] Pierre L. Dognin, Inkit Padhi, Igor Melnyk, and Payel Das. ReGen: Reinforcement Learning for Text and Knowledge Base Generation using Pretrained Language Models, August 2021. URL http://arxiv.org/abs/2108.12472. arXiv:2108.12472 [cs].

[19] Ryo Egawa, Gaku Morio, and Katsuhide Fujita. Annotating and analyzing

semantic role of elementary units and relations in online persuasive arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 422–428, 2019.

[20] Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. Neural end-to-end learning for computational argumentation mining. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11–22, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1002. URL https://aclanthology.org/P17-1002/.

[21] Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Web-scale information extraction in knowitall: (preliminary results). In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, page 100–110, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 158113844X. doi: 10.1145/988672.988687. URL https://doi.org/10.1145/988672.988687.

[22] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134, 2005. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2005.03.001. URL https://www.sciencedirect.com/science/article/pii/S0004370205000366.

[23] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, page 1535–1545, USA, 2011. Association for Computational Linguistics. ISBN 9781937284114.

[24] Jeanne Fahnestock and Marie Secor. The stases in scientific and literary argument. *Written communication*, 5(4):427–443, 1988.

[25] Debela Gemechu, Ramon Ruiz-Dolz, and Chris Reed. ARIES: A general benchmark for argument relation identification. In Yamen Ajjour, Roy Bar-Haim, Roxanne El Baff, Zhexiong Liu, and Gabriella Skitalinskaya, editors, *Proceedings of the 11th Workshop on Argument Mining (ArgMining 2024)*, pages 1–14, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.argmining-1.1. URL https://aclanthology.org/2024.argmining-1.1/.

[26] Deniz Gorur, Antonio Rago, and Francesca Toni. Can large language models perform relation-based argument mining? In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8518–8534, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.569/.

[27] Joseph E Grimes. *The thread of discourse*, volume 2. Mouton The Hague, 1975.

[28] Daniel Han and Michael Han. unsloth, 2023. URL https://github.com/unslothai/unsloth.

[29] Jiuzhou Han, Nigel Collier, Wray Buntine, and Ehsan Shareghi. PiVe: Prompting with iterative verification improving graph-based generative capability of LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics ACL 2024*, pages 6702–6718, Bangkok, Thailand and virtual meeting, August 2024. Association for

Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.400. URL https://aclanthology.org/2024.findings-acl.400.

[30] Naeemul Hassan, Chengkai Li, and Mark Tremayne. Detecting check-worthy factual claims in presidential debates. In *Proceedings of the 24th acm international on conference on information and knowledge management*, pages 1835–1838, 2015.

[31] Julia Hirschberg and Diane Litman. Empirical studies on the disambiguation of cue phrases. *Computational Linguistics*, 19(3):501–530, 1993. URL https://aclanthology.org/J93-3003/.

[32] Hans Hoeken and LGMM Hustinx. The relative persuasiveness of anecdotal, statistical, causal, and expert evidence. 2003.

[33] Alexander Hoyle, Rupak Sarkar, Pranav Goel, and Philip Resnik. Natural language decompositions of implicit content enable better text representations, 2025. URL https://arxiv.org/abs/2305.14583.

[34] Quzhe Huang, Shibo Hao, Yuan Ye, Shengqi Zhu, Yansong Feng, and Dongyan Zhao. Does recommend-revise produce reliable annotations? an analysis on missing instances in docred, 2022. URL https://arxiv.org/abs/2204.07980.

[35] Pere-Lluís Huguet Cabot and Roberto Navigli. REBEL: Relation Extraction By End-to-end Language generation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.204. URL https://aclanthology.org/2021.findings-emnlp.204.

[36] Arman Irani, Ju Yeon Park, Kevin Esterling, and Michalis Faloutsos. Wiba: What is being argued? a comprehensive approach to argument mining. In *International Conference on Advances in Social Networks Analysis and Mining*, pages 337–354. Springer, 2024.

[37] Israa Jaradat, Pepa Gencheva, Alberto Barrón-Cedeño, Lluís Màrquez, and Preslav Nakov. Claimrank: Detecting check-worthy claims in arabic and english. *arXiv preprint arXiv:1804.07587*, 2018.

[38] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL `https://arxiv.org/abs/2310.06825`.

[39] Martin Josifoski, Nicola De Cao, Maxime Peyrard, Fabio Petroni, and Robert West. GenIE: Generative Information Extraction. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4626–4643, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.342. URL `https://aclanthology.org/2022.naacl-main.342`.

[40] Masayuki Kawarada, Tsutomu Hirao, Wataru Uchida, and Masaaki Nagata. Argument Mining as a Text-to-Text Generation Task. In Yvette Graham and Matthew Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Pa-*

pers), pages 2002–2014, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.eacl-long.121.

[41] Jun'ichi Kazama, Takaki Makino, Yoshihiro Ohta, and Junichi Tsujii. Tuning support vector machines for biomedical named entity recognition. In *ACL Workshop on Natural Language Processing in the Biomedical Domain*, 2002. URL https://api.semanticscholar.org/CorpusID:10262770.

[42] John Lawrence and Chris Reed. Argument Mining: A Survey. *Computational Linguistics*, 45(4):765–818, January 2020. ISSN 0891-2017, 1530-9312. doi: 10.1162/coli_a_00364. URL https://direct.mit.edu/coli/article/45/4/765-818/93362.

[43] Michael C. Leff. The topics of argumentative invention in latin rhetorical theory from cicero to boethius. *Rhetorica: A Journal of the History of Rhetoric*, 1(1): 23–44, 1983. ISSN 07348584, 15338541. URL http://www.jstor.org/stable/10.1525/rh.1983.1.1.23.

[44] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461, 2019. URL http://arxiv.org/abs/1910.13461.

[45] Jiaxing Li. Moving beyond text: Multi-modal expansion of the toulmin model for enhanced ai legal reasoning. In *BenchCouncil International Symposium on Intelligent Computers, Algorithms, and Applications*, pages 299–308. Springer, 2023.

[46] Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Lixiang Lixiang, Zhilei Hu, Long Bai, Wei Li, Yidan

Liu, Pan Yang, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. KnowCoder: Coding structured knowledge into LLMs for universal information extraction. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8758–8779, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.475. URL https://aclanthology.org/2024.acl-long.475.

[47] Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. Unified Structure Generation for Universal Information Extraction, March 2022. URL https://arxiv.org/abs/2203.12277v1.

[48] Tobias Mayer, Elena Cabrio, and Serena Villata. Transformer-based argument mining for healthcare applications. In *ECAI 2020*, pages 2108–2115. IOS Press, 2020.

[49] Igor Melnyk, Pierre Dognin, and Payel Das. Knowledge Graph Generation From Text, November 2022. URL http://arxiv.org/abs/2211.10511. arXiv:2211.10511 [cs].

[50] Raquel Mochales and Marie-Francine Moens. Argumentation mining. *Artificial intelligence and law*, 19:1–22, 2011.

[51] Raquel Mochales Palau and Marie-Francine Moens. Automatic argumentation detection and its role in law and the semantic web. In *Law, Ontologies and the Semantic Web*, pages 115–129. IOS Press, 2009.

[52] Marie-Francine Moens. Summarizing court decisions. *Information processing & management*, 43(6):1748–1764, 2007.

[53] Gaku Morio, Hiroaki Ozaki, Terufumi Morishita, and Kohsuke Yanai. End-to-

end argument mining with cross-corpora multi-task learning. *Transactions of the Association for Computational Linguistics*, 10:639–658, 2022.

[54] Sudha Morwal, Nusrat Jahan, and Deepti Chopra. Named entity recognition using hidden markov model (hmm). *International Journal on Natural Language Computing*, 1:15–23, 12 2012. doi: 10.5121/ijnlc.2012.1402.

[55] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticæ Investigationes*, 30(1):3–26, 2007. ISSN 0378-4169. doi: https://doi.org/10.1075/li.30.1.03nad. URL `https://www.jbe-platform.com/content/journals/10.1075/li.30.1.03nad`.

[56] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[57] Neni Nurkhamidah, Raihana Ziani Fahira, and Ayu Ratna Ningtyas. Rhetorical analysis of joe biden's inauguration address. *JL3T (Journal of Linguistics, Literature and Language Teaching)*, 7(2):73–82, 2021.

[58] Riccardo Orlando, Pere-Lluís Huguet Cabot, Edoardo Barba, and Roberto Navigli. ReLiK: Retrieve and LinK, Fast and Accurate Entity Linking and Relation Extraction on an Academic Budget. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics ACL 2024*, pages 14114–14132, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. URL `https://aclanthology.org/2024.findings-acl.839`.

[59] Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. Structured Prediction as Translation between Augmented Natural Languages,

December 2021. URL http://arxiv.org/abs/2101.05779. arXiv:2101.05779 [cs].

[60] Joonsuk Park and Claire Cardie. Identifying appropriate support for propositions in online user comments. In *Proceedings of the first workshop on argumentation mining*, pages 29–38, 2014.

[61] Joonsuk Park and Claire Cardie. A corpus of erulemaking user comments for measuring evaluability of arguments. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[62] Ayush Patwari, Dan Goldwasser, and Saurabh Bagchi. Tathya: A multi-classifier system for detecting check-worthy statements in political debates. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2259–2262, 2017.

[63] Andreas Peldszus and Manfred Stede. Rhetorical structure and argumentation structure in monologue text. In Chris Reed, editor, *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 103–112, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2812. URL https://aclanthology.org/W16-2812/.

[64] Isaac Persing and Vincent Ng. End-to-end argumentation mining in student essays. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1384–1394, 2016.

[65] Wilma Prafitri and Muhammad Alim Akbar Nasir. Persuasive strategies in donald trump's political speeches. *EBONY: Journal of English Language Teaching, Linguistics, and Literature*, 3(1):33–44, 2023.

[66] Muhammad Qorib, Geonsik Moon, and Hwee Tou Ng. Are decoder-only language models better than encoder-only language models in understanding word meaning? In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16339–16347, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.967. URL https://aclanthology.org/2024.findings-acl.967/.

[67] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

[68] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019. URL http://arxiv.org/abs/1910.10683.

[69] Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. Classification and Clustering of Arguments with Contextualized Word Embeddings, June 2019. URL http://arxiv.org/abs/1906.09821. arXiv:1906.09821 [cs].

[70] Rodney A Reynolds, J Lynn Reynolds, James Price Dillard, and M Pfau. The persuasion handbook: Developments in theory and practice. 2002.

[71] Ruty Rinott, Lena Dankin, Carlos Alzate Perez, Mitesh M. Khapra, Ehud Aharoni, and Noam Slonim. Show me your evidence - an automatic method for context dependent evidence detection. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 440–450, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1050. URL https://aclanthology.org/D15-1050/.

[72] Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In Nigel Collier, Patrick Ruch, and Adeline Nazarenko, editors, *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 107–110, Geneva, Switzerland, August 28th and 29th 2004. COLING. URL https://aclanthology.org/W04-1221.

[73] Jiawei Shen, Chengcheng Wan, Ruoyi Qiao, Jiazhen Zou, Hang Xu, Yuchen Shao, Yueling Zhang, Weikai Miao, and Geguang Pu. A study of in-context-learning-based text-to-sql errors, 2025. URL https://arxiv.org/abs/2501.09310.

[74] Edwin Simpson and Iryna Gurevych. Finding Convincing Arguments Using Scalable Bayesian Preference Learning. *Transactions of the Association for Computational Linguistics*, 6:357–371, 2018. doi: 10.1162/tacl_a_00026. URL https://aclanthology.org/Q18-1026. Place: Cambridge, MA Publisher: MIT Press.

[75] Oleg Somov. The generalization and error detection in llm-based text-to-sql systems. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, WSDM '25, page 1077–1079, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400713293. doi: 10.1145/3701551.3707416. URL https://doi.org/10.1145/3701551.3707416.

[76] Christian Stab and Iryna Gurevych. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659, 2017.

[77] Manfred Stede, Jodi Schneider, and Graeme Hirst. *Argumentation mining*. Springer, 2019.

[78] Gisela Striker et al. *Aristotle's Prior Analytics book I: Translated with an introduction and commentary*. Oxford University Press, 2009.

[79] Michael Strobl, Amine Trabelsi, and Osmar Zaïane. Freda: Flexible relation extraction data annotation. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, pages 902–910, 2023.

[80] Yang Sun, Muyi Wang, Jianzhu Bao, Bin Liang, Xiaoyan Zhao, Caihua Yang, Min Yang, and Ruifeng Xu. PITA: Prompting Task Interaction for Argumentation Mining. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5036–5049, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.275. URL https://aclanthology.org/2024.acl-long.275/.

[81] Megha Sundriyal, Atharva Kulkarni, Vaibhav Pulastya, Md. Shad Akhtar, and Tanmoy Chakraborty. Empowering the Fact-checkers! Automatic Identification of Claim Spans on Twitter. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7701–7715, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.525. URL https://aclanthology.org/2022.emnlp-main.525.

[82] Xiangru Tang, Yiming Zong, Jason Phang, Yilun Zhao, Wangchunshu Zhou,

Arman Cohan, and Mark Gerstein. Struc-bench: Are large language models really good at generating complex structured data?, 2024. URL `https://arxiv.org/abs/2309.08963`.

[83] Nathaniel Teich. Rogerian problem-solving and the rhetoric of argumentation. *Journal of advanced composition*, pages 52–61, 1987.

[84] Simone Teufel and Marc Moens. Argumentative classification of extracted sentences as a first step towards flexible abstracting. *Advances in automatic text summarization*, 155:1–171, 1999.

[85] Simone Teufel, Jean Carletta, and Marc Moens. An annotation scheme for discourse-level argumentation in research articles. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 110–117, 1999.

[86] Stephen E. Toulmin. *The Uses of Argument*. Cambridge University Press, 2 edition, 2003.

[87] Frans H Van Eemeren, Rob Grootendorst, Ralph H Johnson, Christian Plantin, and Charles A Willard. *Fundamentals of argumentation theory: A handbook of historical backgrounds and contemporary developments*. Routledge, 2013.

[88] Frans H Van Eemeren, Rob Grootendorst, and Tjark Kruiger. *Handbook of argumentation theory: A critical survey of classical backgrounds and modern studies*, volume 7. Walter de Gruyter GmbH & Co KG, 2019.

[89] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[90] Bart Verheij. *Virtual arguments: on the design of argument assistants for lawyers and other arguers*, volume 6. Springer, 2005.

[91] Bart Verheij. The toulmin argument model in artificial intelligence. In Guillermo Simari and Iyad Rahwan, editors, *Argumentation in Artificial Intelligence*. Springer, Boston, MA, 2009. ISBN 978-0-387-98196-3. doi: 10.1007/978-0-387-98197-0_11.

[92] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, September 2014. ISSN 0001-0782. doi: 10.1145/2629489. URL https://doi.org/10.1145/2629489.

[93] Henning Wachsmuth, Nona Naderi, Yufang Hou, Yonatan Bilu, Vinodkumar Prabhakaran, Tim Alberdingk Thijm, Graeme Hirst, and Benno Stein. Computational argumentation quality assessment in natural language. In Mirella Lapata, Phil Blunsom, and Alexander Koller, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 176–187, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-1017/.

[94] Douglas Walton. *Methods of argumentation*. Cambridge University Press, 2013.

[95] Chenguang Wang, Xiao Liu, and Dawn Song. Language Models are Open Knowledge Graphs, October 2020. URL http://arxiv.org/abs/2010.11967. arXiv:2010.11967 [cs].

[96] Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. DeepStruct: Pretraining of language models for structure prediction. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/

2022.findings-acl.67. URL https://aclanthology.org/2022.findings-acl.67.

[97] Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, and Bo Long. Graph Neural Networks for Natural Language Processing: A Survey, October 2022. URL http://arxiv.org/abs/2106.06090. arXiv:2106.06090 [cs].

[98] Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. DocRED: A large-scale document-level relation extraction dataset. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1074. URL https://aclanthology.org/P19-1074.

[99] Yuxiao Ye and Simone Teufel. End-to-end argument mining as biaffine dependency parsing. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 669–678, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.55. URL https://aclanthology.org/2021.eacl-main.55/.

[100] Yuxiao Ye and Simone Teufel. Computational modelling of undercuts in real-world arguments. In *Proceedings of the 11th Workshop on Argument Mining (ArgMining 2024)*, pages 59–68, 2024.

[101] Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. An autoregressive text-to-graph framework for joint entity and relation extraction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19477–

19487, Mar. 2024. doi: 10.1609/aaai.v38i17.29919. URL https://ojs.aaai.org/index.php/AAAI/article/view/29919.

[102] Bowen Zhang and Harold Soh. Extract, Define, Canonicalize: An LLM-based Framework for Knowledge Graph Construction, April 2024. URL http://arxiv.org/abs/2404.03868. arXiv:2404.03868 [cs].

[103] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45, 2017. URL https://nlp.stanford.edu/pubs/zhang2017tacred.pdf.

[104] Xiaoyan Zhao, Yang Deng, Min Yang, Lingzhi Wang, Rui Zhang, Hong Cheng, Wai Lam, Ying Shen, and Ruifeng Xu. A comprehensive survey on relation extraction: Recent advances and new frontiers. *ACM Comput. Surv.*, 56(11), July 2024. ISSN 0360-0300. doi: 10.1145/3674501. URL https://doi.org/10.1145/3674501.

[105] Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. A comprehensive survey on automatic knowledge graph construction. *ACM Comput. Surv.*, 56(4), nov 2023. ISSN 0360-0300. doi: 10.1145/3618295. URL https://doi.org/10.1145/3618295.

[106] Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. Llms for knowledge graph construction and reasoning: recent capabilities and future opportunities. *World Wide Web*, 27(5):58, aug 2024. ISSN 1573-1413. doi: 10.1007/s11280-024-01297-w. URL https://doi.org/10.1007/s11280-024-01297-w.