ABSTRACT

Title of dissertation: TOWARD ENTITY-CENTRIC UNDERSTANDING OF LONG DOCUMENTS

Abhilasha Sancheti
Doctor of Philosophy, 2024

Dissertation directed by: Professor Rachel Rudinger
Department of Computer Science

Entities and events are the building blocks of language that give the language its richness and expressiveness, be it in everyday conversations, news articles, biographies, legal contracts, or narratives. Documents such as legal contracts and narratives are centered around *people* entities containing rich information about them (such as the rights and duties of the contracting parties, or emotional states of characters in a book), the events they participate in, and their interactions with each other (such as relationships between characters in a book). All this information makes it challenging for the readers to comprehend these documents and find specific information that they need. Understanding such documents from an entity-centric perspective (*i.e., who* is participating in an event, what are its attributes, and relationships), as opposed to an event-centric perspective (*i.e.,* what *happens*), can improve comprehension and extraction of information from these documents enabling the development of practical applications to serve the information needs of readers. Such entity-related information can be conveyed both explicitly and implicitly, and may remain constant or change throughout the document. As a result, certain tasks can be uniquely defined over long contexts. However, limited progress has been made in long document comprehension due to the lack of annotated datasets and challenges in handling extended contexts. This dissertation aims to improve the comprehension of long documents in the

narrative and legal domains by focusing on people entities and addressing these challenges.

First, we systematically investigate the presence and accessibility of implicit script knowledge (*i.e.*, structured commonsense knowledge that humans implicitly share in the form of prototypical sequences of events) in pretrained large language models from a protagonist's perspective via a proposed event sequence description generation task. Based on our findings that these models have limited script knowledge, we propose a script induction framework that is shown to mitigate the issues of mostly omitted, irrelevant, repeated, or misordered events. While scripts are about events that do happen, next, we focus on a setting where multiple entities are involved in an event that may not have happened but is necessary or possible to happen. Taking legal contracts as a test case, we collect a dataset and introduce tasks to identify contracting party-specific obligations, entitlements, prohibitions, and permissions (known as deontic modalities) in lease agreements. We show that transformer-based models trained on this dataset can accurately perform the task demonstrating that the diverse ways of expressing such modalities in natural language are learnable from our dataset. Then, we introduce a task to generate a contracting party-specific extractive summary of the most *important* obligations, entitlements, and prohibitions in a contract that can help monitor compliance, and aid in the contract reviewing process. We collect a dataset of party-specific *importance* ordering (implicit information) among sentences belonging to various modalities in a contract and propose a pipeline-based summarization system to handle the data annotation and long context modeling challenge associated with contract-level summary annotation collection and generation task.

Having designed tasks and entity-centric systems that can generate protagonist-oriented prototypical sequence of events that happen in a scenario, and extract explicit and implicit static information related to entities from unstructured text to a structured form in the legal domain, we then present several strategies to assess large language models' ability to track the fine-grained evolution (dynamic) in social relationship between characters in a book.

Based on the finding that these models fall short in their social reasoning capabilities as they tend to rely on surface-level cues and are sensitive to subtle changes in the context, in our final work, we study the influence of character-related attributes such as gender and race on relationship predictions from a conversation between them to find that these models are prone to heteronormativity biases.

Together, this thesis contributes to the growing field of long context understanding by designing new tasks, collecting datasets, and proposing models covering several aspects (explicit or implicit, static or evolving) of entity-related information to improve comprehension of long documents. By addressing the challenges of handling long context and dataset annotations, this thesis aims to foster the development of entity-centric long document understanding systems to serve the information needs of users.

Toward Entity-Centric Understanding of Long Documents

by

Abhilasha Sancheti

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2024

Advisory Committee:
Professor Rachel Rudinger, Chair
Professor Philip Resnik, Dean's Representative
Professor Hal Daumé III, Member
Professor Abhinav Shrivastava, Member
Dr. Balaji Vasan Srinivasan, Member

Dedicated to
my loving GRANDFATHER,
Late Mr. Kanhaiya Lal Sancheti,
and my FAMILY
for their infinite support, faith, and love.

Acknowledgments

I want to express my deepest gratitude to all my mentors, colleagues, friends, and family members without whose support and belief, I could not have completed my journey to earning a Ph.D. First and foremost, I am immensely grateful to my Ph.D. advisor, Rachel Rudinger, who kindly agreed to advise me. I cannot thank her enough for her consistent support and feedback during my Ph.D. journey. She has helped me grow as a researcher and as a person. I am indebted for the countless hours she has dedicated to brainstorming ideas, offering valuable insights, and helping me refine my ideas. I have not only learned to conduct meaningful and thoughtful, but also high-quality original research from her.

I would also like to extend my sincere gratitude and appreciation to my thesis committee, Philip Resnik, Hal Daumé III, Abhinav Shrivastava, and Balaji Vasan Srinivasan for their invaluable feedback, thought-provoking questions, and suggestions that helped me in improving the quality and depth of my work.

I am extremely grateful to Shriram Revankar (Dolby Laboratories), Nandakishore Kambhatla (Adobe Research), Balaji Vasan Srinivasan (Adobe Research), and Aparna Garimella (Adobe Research) for providing me the opportunity to continue my collaboration with them, and for their support and mentorship. I would also like to thank Dinesh Manocha and Ming Lin for supporting me through assistantship during the initial phase of my Ph.D. program, even though they were not my advisors. I am also thankful to Prof. Marine Carpuat for agreeing to advise me on my first research project as a Ph.D. student. Her thought-provoking questions taught me to carefully design experiments, write clear research questions, and perform solid research. Without their support, I would not have been able to pursue research during my initial years of the program.

I am fortunate to have worked with many researchers throughout my Ph.D. program. Many thanks to Maha Elbayad, David Dale, Artyom Kozhevnikov from FAIR Meta, and

Koustava Goswami and Niyati Chhaya from Adobe Research for the internship and collaboration opportunities.

I would also like to thank Migo Gui, Tom Hurst, and Jodie Gray at UMD for making all the administrative tasks easy.

I want to express my deepest gratitude to Sweta Agrawal for guiding me like her sister and sharing her learning to do good research. I am very grateful to have helpful and motivating colleagues, and friends - Navita Goyal, Haozhe An, Neha Srikanth, Christabel Acquaye, Shramay Palta, Ishani Mondal, Rupak Sarkar, Connor Baumler, Nishant Balepur, Yu Hou, Maharshi Gor, Hyojung Han, Dayeon Ki, and all the members of the CLIP lab. Each of them has played an integral role in providing feedback and sharing ideas that enhanced the quality of my research. Thank you for the friendship and encouragement, and for generously sharing personal insights.

Close friends and family have been a precious source of love and support through the ups and downs of this journey. I feel blessed to have Gaurav Verma, Nishu Gupta, Kushal Chawla, Shivam Lakhotia, Aishwarya Agarwal, Ashita Jain, Pragya Chansoriya, Ritu Yadav, and Ujjwala Gupta in my life who were always there to hear me out and cheer me up during the toughest times, and ensure that I am taking good care of my mental and physical well-being. Thank you Desh Raj, Aditya Gupta, Prakhar Gupta, and Karthik Meher for your guidance, and for sharing job-hunting suggestions and experiences with me.

I am filled with gratitude to have Khushboo Khandelwal, and Divya Kothandaraman for being such great flatmates, and for taking me along to explore D.C.

Finally, I wish to thank my family for their unwavering love, support, understanding, and encouragement to pursue my dreams since the day I was born. I am blessed to have my brother, Abhishek, who always motivates me to do my best. Thank you all for everything.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Overwiew

> "*We use words to talk about the world. Therefore, to understand what words mean, we must have a prior explication of how we view the world.*" – (Hobbs, 1987)

This famous quote by Jerry Hobbs (Hobbs, 1987) highlights a fundamental aspect of language understanding and semantics. It underscores the idea that language is a tool that we use to convey our understanding of the world and our experiences within it. To comprehend the meaning of language, we need to have a shared understanding of the world, entities, events they participate in, and their relationships.

Whether it's in everyday conversations, narratives, news articles, biographies, or legal contracts, entities and events play a critical role as they give the language its richness and expressiveness. Billions of such documents are authored and read by humans on a day-to-day basis for knowledge acquisition, entertainment, work, and communication. Many of these documents (such as novels and legal contracts) are centered around *people entities*, and contain rich information about them, single or multiple (related or unrelated) events they participate in, their roles, relationships, and interactions with other entities, making it challenging for readers to comprehend, and find specific information that they need.

Consider, for instance, a novel that explores the space of human experiences where multiple characters interact with each other over a sequence of events. While reading a novel, a reader may maintain a mental model of different characters, their attributes, and rela-

tionships. This mental model keeps getting updated with any new information as the story progresses. For example, the relationship between *Ron Weasley* and *Hermione Granger* transitions from initial tension and friendship to a deep romantic connection towards the end of the infamous *Harry Potter* series or the vampiric downfall of Lucy and Arthur's engagement in the novel, *Dracula*. Humans use their linguistic capabilities along with world knowledge and contextual information to understand such complex interactions between entities, and events described in the text.

Consider another example of legal contracts that describe the terms and conditions for the entities entering into an agreement. Such documents contain information relevant to each contracting party where these parties may be involved in events that may not have happened but are necessary or possible to happen. When events with different modalities (such as certainty, possibility, permissibility, obligation, and necessity) are expressed in the language, the same information may carry different implications for different entities involved in an event. For example, consider a sentence from a lease agreement, *The tenant shall vacate the property only after a prior notice of at least one month to the landlord*. Here, the *tenant* and the *landlord* are the two entities involved in the event of *vacating the property*. While this sentence *explicitly* mentions an obligation for the tenant towards the landlord, it also *implicitly* mentions an entitlement (entitled to receive prior notice) for the landlord. Therefore, information related to the event of 'vacating the property' has different implications and relevance for each of the parties.

Automatic extraction and identification of such rich information related to entities from these documents can help develop question-answering systems that can reason across facts about entities mentioned in different documents, build analytic models that can aid the readers in understanding constantly evolving social relationships, develop personalized book recommendation systems based on readers' interests, and other practical applications for easier and rapid comprehension of legal or medical documents.

Entity-centric natural language understanding (NLU) involves developing algorithms and models to enable machines to analyze text, both implicitly and explicitly, with an emphasis on real-world entities such as people, and organizations which play a critical role in text (Bamman, 2015; Choi, 2019). Existing technologies may enable automatic extraction and identification of attributes related to entities such as the type of entity (named entity recognition), different expressions used to refer to the entity (coreference resolution), disambiguating entities by linking them to a common source (entity linking), and static relationships with other entities (relation extraction). However, these technologies do not capture *what* information is mentioned about these entities (such as obligations of a tenant in a legal contract), and how different events (these entities participate in) can have different implications on their attributes or relationships (as described in the above examples).

Therefore, this dissertation takes one step ahead in identifying and understanding static or evolving information about *people entities* present in both explicit and implicit form in documents across two domains (*i.e.*, narratives and legal) to enable easier and rapid comprehension of these documents.

1.2 Roadmap

In this thesis, **I argue that entity-centric language understanding systems improve comprehension and extraction of information from long documents, and enable the development of practical applications to serve the information needs of readers**. As we focus on the narrative and legal domain, our work aims to help readers of narratives such as novels (*e.g.*, people who read books), and legal agreements (*e.g.*, the contracting parties and legal professionals) by evaluating entity-centric understanding of large language models for narrative comprehension and building automatic entity-centric information extraction and summarization systems for legal document comprehension.

We introduce new tasks, ideas to collect datasets for them, and design models emphasizing the importance of entity-centric understanding specifically for long documents. These tasks capture various aspects of entity-related information, including static entity-specific deontic modalities, their level of importance for each modality, and the evolving explicitly or implicitly stated characteristics of entities and their relationships across the two domains.

To establish a foundation for language comprehension, we first assess the knowledge of scripts (*i.e.*, structured commonsense knowledge that humans implicitly share in the form of prototypical sequences of events (Schank and Abelson, 1975)) in existing pretrained language models and propose a method that is shown to produce meaningful prototypical sequences of events from a protagonist's perspective (Chapter 3). Next, we introduce new tasks, datasets, and models in the legal domain to extract static and explicitly mentioned information specific to each of the contracting parties (Chapter 4) and produce a structured summary partitioned based on deontic modalities and their importance with respect to each of the parties (Chapter 5). Next, we define the task and propose strategies to track fine-grained evolution in relationships (mentioned either explicitly or implicitly) between the characters in a book (Chapter 6). Finally, owing to the multi-faceted nature of relationships and various causal factors that can affect relationship prediction. We further investigate the influence of such causal factors (such as gender and race) on relationship predictions from LLMs (Chapter 7).

This dissertation expands the scope of information that can be learned about entities, involved in various events, from text in the narrative and legal domains. In all these works, we focus on people entities and bring entity-related information scattered in unstructured form into a structured space. We provide an overview of these contributions below.

1.2.1 Generating Protagonist-oriented Sequence of Events

Script Knowledge (Schank and Abelson, 1975) has long been recognized as crucial for language understanding as it can help in filling in unstated information in a narrative. However, such knowledge is expensive to produce manually and difficult to induce from text due to reporting bias (Gordon and Van Durme, 2013). We assess the knowledge of scripts in existing pretrained language models in Chapter 3. To this end, we introduce a new task of generating full event sequence descriptions (ESDs) from a protagonist's perspective given a scenario in the form of a natural language prompt. Through zero-shot prompting, we find that generative LMs produce poor ESDs with mostly omitted, irrelevant, repeated or misordered events due to the inability to track the state of the protagonist and other entities involved in the events. To address this, we propose a pipeline-based script induction framework (SIF) that is shown to generate good-quality ESDs for unseen scenarios (*e.g., bake a cake*). SIF is a two-staged framework that fine-tunes LM on a small set of ESD examples in the first stage followed by a post-processing stage to filter irrelevant events, remove repetitions, and reorder the temporally misordered events. Through this work, we show that although pretrained language models generate event sequences with mostly omitted, irrelevant, repeated or misordered events in a zero-shot setting, script knowledge can be induced in them via fine-tuning on a small number of example event sequence descriptions followed by our proposed post-processing steps.

1.2.2 Party-Specific Deontic Modality Detection in Legal Documents

While the previous work focuses on events that happen in a scenario from one entity's perspective, in this work, we shift to the legal domain where multiple entities may be involved in an event that may not have happened but is necessary or possible to happen, especially in the case of legal agreements. Deontic modality is frequently used in these

agreements to express obligations, entitlements, permissions, and prohibitions of contracting parties (Ballesteros-Lintao et al., 2016). For instance, modal expressions such as 'shall', 'shall not', and 'may' are respectively used to express 'obligation/entitlement', 'prohibition', and 'permission' in legal agreements. Owing to the non-robustness of prior rule-based approaches to the linguistic diversity of expressing deontic modalities in language and a lack of standard taxonomy for identifying deontic modalities in the legal domain, in Chapter 4, we present a linguistically-informed taxonomy for annotating deontic types in the legal domain. We use this taxonomy to build a corpus, LEXDEMOD, of English contracts consisting of deontic type and their trigger (*i.e*, words or phrases that evoke the deontic type) annotations. We introduce two new tasks of: (1) party-specific multi-label deontic modality classification, and (2) party-specific deontic modality and trigger span detection to extract explicitly mentioned information related to the parties. We handle lengthy agreements by annotating and modeling the deontic modality at a sentence level. We benchmark this dataset on the two tasks using transformer-based models that can accurately perform the task demonstrating that the linguistic diversity of how such modalities are expressed is learnable from our dataset. Transfer learning experiments from lease to employment agreements further indicate the generalizability of the expressions captured in our dataset.

1.2.3 Party-Specific Summarization of Legal Documents

Although previous work has shown the feasibility of extracting various obligations, entitlements, and prohibitions of a party from agreements, understanding and reviewing them can be difficult and tedious due to their sheer number and the complexity of legalese. Having an automated system that can provide an "at a glance" summary of obligations, entitlements, and prohibitions can be useful not only to the parties but also to legal profes-

sionals for reviewing contracts. In Chapter 5, we argue that a single summary may not serve all the parties as they may have different obligations, entitlements and prohibitions (Ash et al., 2020) (*e.g.,* a *tenant* is obligated to timely pay the rent to the landlord whereas, the *landlord* must ensure the safety and maintenance of rented property). Further, each right or duty may not be equally important (*e.g.*, an obligation to pay rent on time (otherwise pay a penalty) is more important for tenant than maintaining a clean and organized premise) and this importance also varies for each party. To address this, we introduce a new task of *party-specific* extractive summarization of *important* obligations, entitlements, and prohibitions in a contract. To facilitate this, we curate a dataset comprising of *party-specific* pairwise importance comparisons annotated by legal experts that include obligations, entitlements, and prohibitions extracted from lease agreements. Using this dataset, we train a pairwise importance ranker and propose a pipeline-based extractive summarization system that generates a *party-specific* contract summary. We establish the need for incorporating domain-specific notions of importance during summarization by comparing our system to various summarization baselines using both automatic and human evaluation methods.

1.2.4 Tracking Evolving Character Relationships in Books

While the ability and feasibility of identifying static information specific to each entity provide a deeper understanding of the information about entities, such information may not always be static when entities are involved in a sequence of events. Specifically, in a broader world such as that of novels which explore the space of human experiences, multiple entities could interact with each other in a sequence of events. As a result, their behaviors, motivations, and relationships could evolve throughout the novel. In Chapter 6, we characterize evolution in the relationship between characters in terms of a trajectory of predefined relationship and evolution types taking inspiration from studies in social sci-

ences (Deri et al., 2018). Tracking such evolution in a book-length context poses two main challenges: (1) handling long context, and (2) lack of annotated datasets. To address these challenges, we use recent large language models (Jiang et al., 2023; Team et al., 2024; Dubey et al., 2024) having large context window size, and their zero-shot reasoning capabilities to propose various strategies based on the granularity of information used to track the evolution in the relationship between characters in a book. In the absence of annotated datasets, we thoroughly analyze the agreements between predictions from different families of LLMs and manually examine the consensus predictions. Low-to-moderate agreement scores between predictions from different LLMs as well as between humans reflect the complex nature of the task. Qualitative analysis reveals that predictions from LLMs are sensitive to subtle changes (such as pronoun replacement) and the presence of multiple characters in the provided context. Additionally, they tend to adopt heuristics based on surface-level cues and lack a proper understanding of the context. Humans may interpret a context differently due to the multi-faceted nature of relationships resulting in disagreement between their annotations.

1.2.5 Investigating the Influence of Implicit Character Attributes on Relationship Predictions from Large Language Models

Relationship prediction is a complex task because relationships are multi-faceted (Hovy and Yang, 2021) and are often expressed indirectly through actions and need to be inferred from the conversation between characters and their emotional states. Similar actions and conversations may have different interpretations depending upon the context, and appropriateness (Jurgens et al., 2023). Further, relationship predictions may be influenced by several causal factors such as the tone of the speakers, perceived demographic attributes related to the speakers (such as age and gender), and location or setting (such as, in a cafe

vs in a school). In Chapter 7, we study the influence of such causal factors, specifically the gender and race of the characters, on the relationship predictions from LLMs to understand if LLMs resort to such heuristics for performing the task. To this end, we perform controlled experiments by replacing the names of characters in a conversation with names having different gender and race associations keeping the conversation the same. We show that LLMs are less likely to predict romantic relationships for (a) same-gender character pairs than different-gender pairs; and (b) intra/inter-racial character pairs involving Asian names as compared to Black, Hispanic, or White names. Further analysis of contextual embeddings reveals that gender is less discernible for Asian names as compared to non-Asian names. These findings suggest that LLMs use spurious correlations between a relationship type and these causal factors for the prediction task which could either result in incorrect predictions or correct predictions for the wrong reasons.

1.3 Contributions

This dissertation makes the following contributions:

- **Design Tasks:** We design a novel task of protagonist-oriented prototypical script generation (Chapter 3) and formalize the task of tracking the evolution in the relationship between characters in books (Chapter 6) to improve comprehension of information in the narrative domain. Further, we introduce entity-specific tasks to extract entity-relevant information (Chapter 4) and entity-specific importance to develop a summarization system (Chapter 5) in the legal domain. These tasks aim to understand both implicit or explicit, and static or dynamic entity-centric information contained in long documents to improve its comprehension.

- **Introduce Datasets:** We introduce two entity-centric datasets (Chapter 4 and Chapter 5) in the legal domain to enable the development of an entity-specific extractive

summarization system to serve the information needs of the stakeholders.

- **Propose Entity-centric Systems:** We develop entity-centric language understanding systems that capture different aspects of entity-centric information contained in long documents for improving comprehension (Chapter 3, 6, and 7), extraction (Chapter 4), and enabling the development of a practical application (Chapter 5) to serve the information needs of readers in the narrative and the legal domain.

Chapter 2: Background

In this chapter, we first provide an overview of linguistic theories that define how entities and their relationships are represented in the discourse in Section 2.1. Then, we provide details on the existing entity-centric natural language understanding tasks ranging from information extraction to generation, and the basics of modeling concepts used for solving these tasks in Section 2.2. We then review existing works in the field of narrative comprehension (Section 2.3) and legal language understanding (Section 2.4.1) as these narratives and legal documents contain multiple entities and information relating to them, making them a good fit for entity-centric understanding.

2.1 Entity-centric Natural Language Understanding

The entity-centric understanding of language plays a critical role in long document comprehension by helping systems track, disambiguate, and relate entities across extended texts. The well-known concepts of Discourse Representation Theory (Kamp, 1981a,b; Kamp et al., 2010) and Centering Theory (Grosz et al., 1995; Poesio et al., 2004) provide valuable insights into how entities and their relationships are managed in discourse (*i.e.*, across sentences). Discourse Representation Theory (DRT), introduced by Kamp (1981a), offers a formal framework for modeling how discourse entities are introduced, tracked, and updated as discourse unfolds. DRT focuses on the representation of entities in the context of an ongoing discourse, where each new piece of information is integrated into a

dynamic model of the discourse world. Unlike traditional sentence-based semantics, DRT views meaning across sentences as evolving over time, with each sentence contributing to a larger discourse representation with referents (*i.e.*, entities or concepts introduced in the discourse) and conditions that specify how those entities relate to each other. As discourse unfolds, new referents are added, and the conditions are updated, reflecting the dynamic nature of language understanding.

While DRT focuses on creating a representation of the entire discourse, Centering Theory (Grosz et al., 1995) specifically analyzes the local coherence of discourse by identifying the most salient entity that links each sentence or phrase to the previous one within a discourse. The core idea behind Centering Theory is that humans maintain a mental representation of entities that are relevant to the ongoing discourse (such as characters while reading a book), which helps to maintain coherence and facilitate the interpretation of pronouns and other referring expressions. According to this theory, each sentence in a discourse can be centered around a particular entity, called the “centering focus". This focus is typically the most salient or prominent entity in the discourse at a given point. It maintains *forward-looking centers* (entities which can be referenced in the future) and a *backward-looking center* (salient entity in the previous sentence that is realized in the future sentences) to explain how the coherence of a discourse is maintained through coreference relations and other discourse structures. Centering Theory has proven influential in NLP, particularly in the development of algorithms for coreference resolution (Bagga and Baldwin, 1998; Soon et al., 2001) (see next section for details) and discourse coherence modeling (Barzilay and Lapata, 2008; Li and Jurafsky, 2017).

2.2 Entity-centric Tasks in NLP

Understanding entities, such as names of people, locations, organizations, and their interactions, lies at the core of human language comprehension. While the previous section focuses on theories behind the representation of entities and their relationships in discourse, entity-centric tasks provide the necessary tools to operationalize such theories into practice. Together, these approaches are essential for developing advanced NLP systems capable of comprehending complex, and lengthy documents with semantic depth.

Early works in Natural Language Processing (NLP) recognized the significance of entity-centric tasks. For instance, the development of Named Entity Recognition (NER) (Grishman and Sundheim, 1995) systems was catalyzed by the CoNLL-2003 Shared Task (Tjong Kim Sang and De Meulder, 2003), marking a milestone in recognizing and categorizing named entities. Moreover, advances in Coreference Resolution (Bagga and Baldwin, 1998; Soon et al., 2001) played a pivotal role in understanding how entities mentioned in the text are related and ensure coherent discourse. Linking these recognized entities to structured knowledge bases (such as Wikipedia, DBpedia) and ontologies (Mann and Yarowsky, 2003; Bunescu and Paşca, 2006; Cucerzan, 2007) enabled advanced semantic reasoning and knowledge-driven applications such as question answering. Relation Extraction (muc, 1991) is another task that focuses on identifying and categorizing relationships between entities mentioned in the text. It deals with understanding how different entities interact with each other in textual data. Therefore, the ability to understand entities and their relationships within textual data is fundamental in advancing various domains of NLU.

In recent years, the emergence of pretrained language models, such as GPT (Radford et al., 2019), and BERT (Devlin et al., 2019a), has revolutionized entity-centric NLU. These models, pretrained on vast text corpora, have been fine-tuned for various downstream tasks, achieving competitive results. Entity-centric models, like ERNIE (Zhang et al., 2019) and

SpanBERT (Joshi et al., 2020), have extended these architectures to better capture entity information. Zhang et al. (2019) utilize both large-scale textual corpora and entity knowledge graphs to train an enhanced language representation model, which can take full advantage of lexical, syntactic, and external knowledge simultaneously resulting in significant improvements in entity-typing and relation extraction tasks. Joshi et al. (2020), on the other hand, introduce a masked span prediction pretraining objective to learn better discourse representations that are shown to achieve substantial gains in coreference resolution and question answering. Apart from these, prior works have also looked at a variety of entity-centric generation tasks. One of the fundamental entity-centric generation tasks is summarization (Fan et al., 2018a; Maddela et al., 2022) which focuses on creating concise and coherent summaries of longer texts, with an emphasis on given entities and their information in the input. For example, generating a summary of the performance of a particular player from a sports article. Another line of research focuses on story generation with an emphasis on specific entities that create narratives that are engaging and contextually relevant. These narratives can be fictional or based on real-world events. Entity-centric storytelling enhances narrative coherence and relevance (Fan et al., 2018b; Zhang et al., 2022). Entity-based question-answering tasks involve generating responses that contain information about specific entities (Onishi et al., 2016; Sciavolino et al., 2021).

While the existing entity-centric tasks aid in document comprehension, and generation, the information extraction-related tasks such as NER, entity linking, coreference resolution, and relation extraction provide meta-information about entities. In this thesis, we focus on novel entity-centric tasks that help in filling-in the information needs of readers for easier consumption of long documents such as legal contracts and aid in better comprehension of documents such as narratives by identifying and extracting both static and dynamic, explicit or implied information.

2.2.1 Modeling Primer

In this section, we provide a background on the modeling backbones that are ubiquitously used in the field of NLP and the following chapters of this thesis. We start with an overview of the Transformer model (Vaswani et al., 2017) which is the foundation of existing large language models. We touch upon the paradigms of pretraining, fine-tuning, and prompting, followed by task-specific modeling, and their training and inference details.

Transformers and Large Language Models (LLMs)

The Transformer model (Vaswani et al., 2017) is a neural network architecture which has revolutionized the field of natural language processing (NLP) and has become the foundation for many state-of-the-art NLP models. It consists of a stack of encoders and a stack of decoders of the same number. The encoder aims at converting the input tokens into a semantic representation. It consists of a self-attention (Bahdanau et al., 2014) layer followed by a feed-forward neural network. Self-attention layers allow the encoder to focus on other tokens in the input sequence of tokens as it encodes a specific token. The outputs from the self-attention layer are fed to a shared feed-forward neural network to obtain the final representation for each input token. The decoder aims at generating a sequence based on the input sequence. In addition to self-attention and feed-forward network layers, a cross-attention layer helps focus on relevant parts of the input sequence. The key innovation of the Transformer is its attention mechanism, which allows the model to weigh the importance of different parts of the input sequence when making predictions. This architecture has proven highly effective for a wide range of NLP tasks relying entirely on self-attention to compute representations of its input and output without using sequence-aligned recurrent neural networks (RNNs) (Rumelhart et al., 1986; Hochreiter and Schmidhuber, 1997) or convolution neural networks (LeCun et al., 1989). Transformer models have been shown

to perform strongly on various NLP tasks (such as, machine translation, document generation, and syntactic parsing) as they provide more structured memory for handling long-term dependencies in text, compared to RNNs, resulting in robust transfer performance across diverse tasks. Therefore, Transformers make it feasible to train very deep neural architectures with the help of language modeling objective function as defined below.

$$L(W) = \sum_i logP(w_i|w_{<i}; \theta)$$

where $W = \{w_1, w_2, \ldots, w_n\}$ denotes the text corpus, and $w_{<i}$ refers to the tokens generated until timestep t. The conditional probability P is modeled using a network with parameters θ.

A large number of language models have been introduced since then depending on whether it uses a decoder-only auto-regressive model (GPT Radford (2018); Radford et al. (2019); Brown et al. (2020); OpenAI (2022), LlaMa Touvron et al. (2023a,b); Dubey et al. (2024)), a bi-directional encoder-only model (BERT Devlin et al. (2019a), and RoBERTa Liu et al. (2019)) or an encoder-decoder-based model (BART Lewis et al. (2020) and T5 Raffel et al. (2020)). For a detailed overview of these models, we refer the readers to these survey papers (Qiu et al., 2020; Raffel et al., 2020; Han et al., 2021).

Pretraining and Post-training

Language model training involves pre- and post-training stages as described below.

Pretraining involves unsupervised training of a neural network on a vast corpus of text data, such as Wikipedia articles, to capture general language understanding including syntactic and semantic knowledge. During pretraining, a language model learns to predict the next word given the previous words in a sequence (language modeling objective, as

mentioned above) or to predict missing words in a sequence (masked language modeling objective, as used in BERT). This process enables the model to capture a broad understanding of language. Pretraining creates a versatile language model that can be further fine-tuned (Radford, 2018) or instruction-tuned (Wei et al.) for various downstream tasks (such as, entity recognition, sentiment classification, and summarization).

Post-training involves adapting a general pretrained language model to a desired task or a capability by updating its weights using supervised training. Post-training is done either via fine-tuning or a combination of fine-tuning and reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022). **Fine-tuning** involves further training of a pretrained model on a smaller, task-specific dataset (such as that for entity recognition, sentiment classification, and summarization) using task-specific objective functions. The pretrained model's weights act as a starting point, and further training is done to adapt the model's parameters to the nuances of the target task. More recent language models (such as Instruct-GPT (Ouyang et al., 2022)) use **instruction-tuning**, a form of fine-tuning that aims at adapting a pretrained model to perform specific tasks by providing a set of instructions (guidelines) in natural language (Mishra et al., 2022; Wei et al.; Xu et al., 2023), followed by RLHF to align the generation to human preferences. Such models can also be fine-tuned for dialogue use cases resulting in chat-based models such as ChatGPT (OpenAI, 2022), LlaMa (Touvron et al., 2023b), Mistral (Jiang et al., 2023), and Gemini (Team., 2023). **RLHF** involves learning a reward model based on human preferences about an output for an input instruction and then using reinforcement learning algorithms such as proximal policy optimization (Schulman et al., 2017) or direct preference optimization (Rafailov et al., 2024) to align the fine-tuned model with these human preferences.

Prompting Large Language Models

With scaling (an increased number of parameters or amount of pretraining data), pretrained models can even perform tasks in a zero or few-shot setting (Radford et al., 2019) without task-specific fine-tuning. Owing to these capabilities, prompting LLMs is a widely adopted technique used to interact with and elicit specific responses from these models that they have learned during pretraining (Liu et al., 2023). It involves providing a text "prompt" or input to the model to generate a desired output. Large language models are designed to take text inputs and generate text outputs. Prompting involves giving the model a starting point in the form of a prompt sentence or question to guide its subsequent responses. For example, to generate a sequence of events that Mike undergoes when he goes to a restaurant we can use the prompt: *Describe the sequence of events that Mike undergoes when he goes to a restaurant:* (Chapter 3). The output can be generated given the probability distribution of the next word (token) $P(w_i|prompt)$ using different decoding algorithms as below:

- **Greedy Search**: This algorithm converts a distribution over the vocabulary of tokens to a single token by simply selecting the token that has the maximum probability, *i.e.*, $w = \text{argmax} P(w_i|prompt)$. We use this algorithm in Chapter 6 and 7.

- **Beam Search**: This algorithm (Graves, 2012) is based on the observation that the most likely tokens at each timestep may not result in a global best output at the sequence level. Therefore, in beam search, at each timestep a set of k partial hypotheses is maintained and eventually, the hypothesis with the highest sentence level probability is chosen as the final output.

- **Top-k Sampling**: Instead of maximizing the probability, sampling a token from the probability distribution can result in diverse outputs. However, sampling from the entire distribution often results in disfluent and degenerate outputs. In Top-k sam-

pling, the next token is instead sampled from the top-k most probable options (Fan et al., 2018b).

- **Nucleus Sampling:** Building on the observation that the probability mass at each timestep is often concentrated on a few vocabulary tokens, in this algorithm (Holtzman et al.) the next token is sampled from a top-p portion of the probability mass instead of relying on a fixed number of top-k tokens. We use this strategy in Chapter 3 and 6 along with the top-k sampling.

Training Objectives for Fine-tuning on Downstream Tasks

Pretrained LLMs are fine-tuned for various downstream tasks such as relation classification, named entity recognition, and summarization using task-specific training objectives as described below.

Training Objectives Supervised training requires labeled data (X,Y), where $x \in X$ is an input (sentence or a sentence-pair) and $y \in Y$ is the corresponding label (such as, deontic modal labels, relationship type, sequence of BIO (begin-inside-out)labels, or a reference summary). For a binary sequence classification task (Chapter 3 and 5), binary cross-entropy is a standard objective used to fine-tune an LLM. It is defined as below:

$$\mathcal{L} = -\left(y \log(p) + (1-y) \log(1-p)\right),$$

where y is the ground-truth class (0 or 1), and p is the predicted probability. In case of multi-label classification task (when one input can belong to multiple classes) (Chapter 4),

the below variant of cross-entropy loss function is used for training purpose.

$$\mathcal{L} = -\sum_{i=1}^{N} \left(y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \right),$$

where N is the number of labels, y_i and p_i is the ground-truth label and predicted probability for the i-th class, respectively.

For a sequence labeling task (token-classification) (such as in Chapter 4), each input word has an associated output label encoded in the form of BIO (B: Begin, I: Inside, O: Outside) labels (Ramshaw and Marcus, 1995). The training objective for this task is

$$\mathcal{L} = -\sum_{t=1}^{T} \sum_{i=1}^{N} y_{t,i} \cdot \log(p_{t,i}),$$

where N is the number of possible labels, $y_{t,i}$ is a binary indicator (0 or 1) that represents whether label i is the true label for timestep t. and $p_{t,i}$ is the predicted probability of label i at timestep t.

For a generation task such as language modeling (Chapter 3) and summarization, the training objective is defined in the same way as the sequence labeling task, but with the variable descriptions as: T is the length of the generated sequence, N is the vocabulary size, $y_{t,i}$ is a binary indicator (0 or 1) that represents whether token i is the true token at timestep t in the output, and $p_{t,i}$ is the predicted probability of token i at timestep t in the output sequence.

2.3 Narrative Understanding and Comprehension

A “narrative" refers to a story or account of events, whether real or fictional, that unfolds in a sequence with a beginning, middle, and end. It often includes characters, a setting, a plot, and a central theme, which can be analyzed to understand the meaning and message

conveyed within the story. The goal of computational narrative understanding is to algorithmically represent, understand, and generate narratives. Early computational studies on narratives have focused on learning procedural scripts and event sequences (Schank and Abelson, 1977; Manshadi et al., 2008; Regneri et al., 2010), narrative chains or schemas (Chambers and Jurafsky, 2008, 2009), and plot units (Lehnert, 1981; Goyal et al., 2010; McIntyre and Lapata, 2010; Elsner, 2012). There also exists work on character-centric modeling of narratives (Barzilay and Lapata, 2008; Chambers, 2013) as characters are critical components of a narrative. In the following sections, we provide an overview of these two fields, and contribute by closing the gaps in Chapter 3, 6, and 7.

2.3.1 Scripts, and Narrative Schema Induction

Scripts are structured commonsense knowledge in the form of event sequences that characterize commonplace scenarios, such as eating at a restaurant (Schank and Abelson, 1975). Scripts are fundamental pieces of commonsense knowledge that humans share and assume to be tacitly understood by each other. They have been shown to help understand narratives by providing expectations, resolving ambiguity, and filling in unstated information (Chambers and Jurafsky, 2008; Rahman and Ng, 2012; Modi et al., 2017). When someone says *"I went to a restaurant for lunch"*, our script knowledge allows us to infer that the speaker would have arrived at the restaurant, got herself seated, a waiter would have taken the order, the speaker would have eaten the lunch, paid for it, and tipped the waiter, even if these events are not explicitly mentioned. Knowledge of scripts, whether implicit or explicit, has been recognized as important for language understanding tasks (Miikkulainen, 1995; Mueller, 2004; Modi et al., 2017) as it can enable an AI system to understand and fill-in the implicit events in any narrative.

Though Schank and Abelson's work involved encoding these scripts by hand, manual

construction of scripts is not a scalable approach. Automatic learning of Shank-style script structures at scale, broadly known as script induction, was done using co-occurrence and mutual-information-based methods, following the development of association rule mining techniques, such as word associations (Church and Hanks, 1990), and inference rules (Lin and Pantel, 2001). Seminal work by Chambers and Jurafsky (2008) describe a number of simple event co-occurrence-based systems that infer (verb, dependency) pairs (known as narrative events) with partial-ordering related to one or multiple participants (Balasubramanian et al., 2013; Pichotta and Mooney, 2014) in discourse (known as narrative chains). For example, the sentence *"John drove to the university and attended his class.*" would generate two narrative events corresponding to the protagonist John: (drive, nsubj) followed by (attend, nsubj). In the follow-up work Chambers and Jurafsky (2009) extend narrative events to fill their roles with a cluster of nouns representing participants rather than just a governing syntactic dependency. Jans et al. (2012) evaluate different counting and ranking methods for evaluating narrative chain models under the narrative cloze test (predict a removed narrative event, given all the other events). Modi and Titov (2014) train a neural model that composes distributional event and participant representations for an event ordering task. Rudinger et al. (2015) demonstrate that a simple neural language model (the log-bilinear model of (Mnih and Hinton, 2007)) outperforms all prior count-based methods on the narrative cloze task. As statistical co-occurrences cannot capture long-range dependencies between events, Pichotta and Mooney (2016a) represent events using LSTM (Hochreiter and Schmidhuber, 1997) leading to improved narrative cloze task performance.

However, much of the information about events is usually left implicit in text. Moreover, narrative events are highly abstracted (Ostermann, 2020), manual creation of such knowledge resources is challenging due to the wide coverage and complexity of relevant scenario knowledge, and inefficiency of close task to evaluate script knowledge (Chambers,

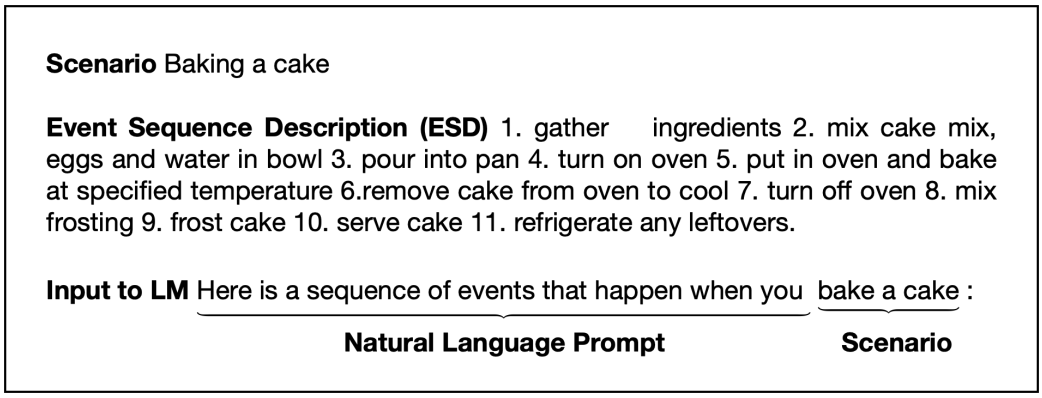


Figure 2.1: Sample event sequence description (ESD) from Wanzare et al. (2016) for BAKING A CAKE scenario. We use prompts (Table 3.2) to *generate completely ordered* ESDs for evaluating the extent of script knowledge accessible through LMs in Chapter 3.

2017). Therefore, efforts have been made to acquire crowdsourced ESDs (Event Sequence Descriptions; see Figure 2.1) (Singh et al., 2002; Regneri et al., 2010; Modi et al., 2016; Wanzare et al., 2016; Ostermann et al., 2018, 2019) and to learn similar events in a scenario using unsupervised (Regneri et al., 2010) and semi-supervised (Wanzare et al., 2017) approaches. However, the collected datasets are small, domain-specific, and crowdsourcing is not scalable due to cost. With the success of pretrained language models (PLM) (Devlin et al., 2019b; Liu et al., 2019; Radford et al., 2019) in various natural language understanding tasks, In Chapter 3, we investigate the extent and accessibility of explicit script knowledge present in PLMs by introducing a protagonist-oriented script generation task.

2.3.2 Character-centric Narrative Comprehension

Plots and characters are the two key components of a narrative (among others) that contribute to a good piece of fiction (Kennedy and Gioia, 1983; McKee, 1997; Card, 1999). Particularly, characters are central to narratives since their traits, motivations, relationships, and actions determine the flow of the plot. Therefore, character comprehension has been widely recognized as important to understanding narratives in psychological, literary,

and educational research (Bower and Morrow, 1990; Paris and Paris, 2003; Currie, 2009; Kennedy, 2015). Humans build mental models for characters and keep updating them as they read a narrative. Such mental models help in explaining character's identity, emotional status (Gernsbacher et al., 1998), future behaviors (Mead, 1990), and even making counterfactual inferences for stories about that character (Fiske et al., 1979).

The field of character-centric narrative comprehension focuses on understanding character's personas, roles, goals, relationships, and emotions. Character "schemas" have played a key role in narrative understanding, including Propp's functions, protagonist/antagonist relations derived from classical tragedy (Moretti, 2013), Girard and Hamerton-Kelly's theory of mimetic desire, and (Woloch, 2009)'s distinction between major and minor characters. There exist several works that aim at identifying characters (Chen and Choi, 2016; Brahman et al., 2021; Sang et al., 2022b) and inferring their personalities (Flekova and Gurevych, 2015; Sang et al., 2022a; Yu et al., 2023) in movie plot summaries and fictional novels (Bamman et al., 2013, 2014a; Vala et al., 2015; Flekova and Gurevych, 2015). Others have considered modeling inter-character relationships (Iyyer et al., 2016a; Srivastava et al., 2016; Chaturvedi et al., 2016, 2017; Kim and Klinger, 2019b; Zhao et al., 2024) and emotions (Brahman and Chaturvedi, 2020). Another line of research aims at constructing social networks of characters (Agarwal et al., 2014; Labatut and Bost, 2019) and understanding information propagation (Sims and Bamman, 2020) between characters from novels (Elson et al., 2010; Elsner, 2012) and films (Krishnan and Eisenstein, 2015). In this section, we focus on the task of character relationship modeling.

Characters and their relationships are one of the basic building blocks of narratives that make the narrative engaging and interesting. Therefore, automatic identification of these relationships is an important task for analyzing narrative text, and interpreting and justifying character's actions in a narrative. While there are many methods for extracting character networks (Labatut and Bost, 2019), these often provide networks without

relationship information. Existing works that model relationships between characters in narratives either presume a fixed set of coarse-grained relations, such as cooperative or non-cooperative (Srivastava et al., 2016; Chaturvedi et al., 2016) and familial or professional (Makazhanov et al., 2014; Massey et al., 2015; Azab et al., 2019) or learn a set of relationship descriptors (Iyyer et al., 2016a). There also exist works that classify emotional relationships between characters in fan-fiction stories (Kim and Klinger, 2019b) and harry potter novel (Zehe et al., 2020) following Kim and Klinger (2019a). Another line of research analyze the polarity and intensity of emotions of characters towards each other (Nalisnick and Baird, 2013) in Shakespearean plays, or classify interpersonal relationships from dialogues in TV series (Chen et al., 2020), movie scripts (Jia et al., 2021) or detective narratives (Zhao et al., 2024).

However, assuming that the relationship between characters is static, is limiting because characters and their relationships keep evolving as a narrative progresses. Such evolving relationships play a critical role in making a narrative interesting (*e.g.*, Ron and Hermoine's relationship in the *Harry Potter* series transitions from an initial tension, and a friendship to a deep romantic connection). Chaturvedi et al. (2016) model the evolution of interpersonal relationships in novels in a supervised setting, requiring manually annotated data, and model relationships as binary polarities. Iyyer et al. (2016a), on the other hand, introduce an unsupervised method, RMN (Relationship Modeling Network), to model this evolving nature of the relationship. Given a narrative text and a character pair, RMN (a variation of a deep recurrent autoencoder) learns a sequence of discrete states depicting the relationship between the two characters. Chaturvedi et al. (2017) propose unsupervised methods to learn relationship sequences from plot summaries. More recently, Qamar et al. (2021) employ psychological models to classify movie dialogues into four attachment styles (*e.g.*, friend, family, pleasant, unpleasant) and four association types (*e.g.*, secure, fearful, preoccupied, dismissing) (Smith et al., 2014) to analyze the transformation between relationships. How-

ever, their analysis is relationship-focused rather than evolution in character relationships.

While there exist works that track the evolution in the relationship between characters, the set of relationships is either coarse-grained or unsupervised. Further, they do not consider the fine-grained evolution (such as, ups and downs in a friendship) in the relationship. In Chapter 6, we define the task of tracking fine-grained evolution in relationships in novels and propose several strategies that differ in the granularity of information used to predict the trajectory of evolution in the relationship from large language models (LLM). Owing to the absence of gold annotations for this task, we thoroughly analyze the predictions using agreement analysis and manual examination.

Existing works have shown that character relationship prediction is challenging even with the improved reasoning capabilities of recent LLMs (Zhao et al., 2024). This is because relationships are multi-faceted (Hovy and Yang, 2021) and are often expressed indirectly through actions. Similar actions and conversations may have different interpretations depending upon the context, and appropriateness (Jurgens et al., 2023). Further, relationship predictions may be influenced by the tone of the speakers, perceived demographic attributes related to the speakers (such as age and gender), and location or setting (such as, in a cafe vs in a school). In Chapter 7, we study the impact of changing the implicit attributes (such as gender and race) as identified by the names of the characters involved in a conversation on the relationship predictions from LLMs to understand if LLMs adopt such spurious correlations.

2.4 Legal Language Understanding

Legal documents have peculiar characteristics such as frequent use of jargons and complex language, too long as compared to other domains, and the existence of ambiguity in terms of interpretation of the same content (Allen and Saxon, 1994; Bernheim and Whin-

ston, 1998; Tiersma, 1999), requiring specialized NLP tools to extract critical information, identify legal entities, and comprehend contractual nuances. The unique intricacies and stakes of the legal domain demand precision and efficiency. Accurate legal understanding is vital for tasks like contract analysis, due diligence, legal research, and regulatory compliance. NLP tools in this specialized domain have the capacity to not only save time but also enhance access to justice, and empower legal professionals to make informed decisions. Legal NLP Document Understanding encompasses a range of techniques and tools designed to analyze, interpret, and extract relevant information from legal documents such as contracts, court cases, statutes, and legal opinions. In the following sections, we provide an overview of the existing works in the field of legal language understanding and then shift focus to two specific tasks, namely, rights and obligations detection, and summarization in the legal domain.

2.4.1 Legal Natural Language Processing

Prior works have investigated a number of tasks (Zhong et al., 2020a) in legal NLP domain including legal judgement prediction (Aletras et al., 2016; Luo et al., 2017; Zhong et al., 2018; Chen et al., 2019; Chalkidis et al., 2019a), legal entity recognition and classification (Cardellino et al., 2017; Chalkidis et al., 2017; Angelidis et al., 2018; Leitner et al., 2019; Kalamkar et al., 2022; Chalkidis et al., 2019b), legal question answering (Duan et al., 2019; Zhong et al., 2020b; Holzenberger and Van Durme, 2021), and legal summarization (Hachey and Grover, 2006; Bhattacharya et al., 2019; Manor and Li, 2019a; Balachandar et al., 2021; Keymanesh et al., 2020). While legal NLP covers a wide range of tasks, limited efforts have been made for contract review despite it being one of the most time-consuming and tedious tasks. Leivaditi et al. (2020) introduced a benchmark for lease contract review for detecting named entities and red flags. Hendrycks et al. introduced a

large expert-annotated dataset and Tuggener et al. (2020) a large semi-automatically annotated dataset for provision type classification across a variety of contract types. In this thesis, we contribute by advancing the research in the space of contract review from each contracting party's perspective by introducing new tasks, datasets, and models, with a focus on rights and obligations extraction and summarization.

2.4.2 Rights and Obligations Extraction from Legal Contracts

Contracts are legally binding agreements that define and govern the rights, duties, and responsibilities of all parties involved in it. As contracts are considered mutually binding documents, the obligation for one party is at the same time a right for the other party and vice versa. Obligation, prohibition, and permission in contracts are frequently expressed by modal verbs and other exponents of deontic modality (Matulewska, 2010; Ballesteros-Lintao et al., 2016). *Modality* refers to the linguistic ability to describe alternative ways the world could be and is commonly expressed by modal auxiliaries such as *shall*, *will*, *must*, *can*, and *may*. There are three main types of modalities (Van Linden, 2012): epistemic (relating to knowledge and beliefs, expresses more an opinion than a fact), deontic (expresses the necessity or probability of action presented by morally responsible agents), and dynamic (expresses the strength and ability of the subject participant).

Extracting rights and duties is an important task for legal firms and legal departments, especially when they process large numbers of contracts to monitor the compliance of each party. Matulewska (2010) analyzed contracts from different countries and types considering fine-grained deontic modalities such as, obligation, permission, and prohibition with temporal constraints. Existing works either propose rule-based methods (Kiyavitskaya et al., 2008; Wyner and Peters, 2011; Peters and Wyner, 2016a) or use a combination of NLP approaches such as syntax and dependency parsing (Dragoni et al., 2016) for

extracting rights and duties from legal documents such as Federal code regulations, European directives or customer protection codes. Another line of works (Bracewell et al., 2014; Neill et al., 2017; Chalkidis et al., 2018a) use machine learning and deep learning approaches to predict deontic types with the help of small datasets. However, rule-based approaches are not robust due to the rich linguistic variety and ambiguity of modal expressions, and the annotated datasets do not consider multiple deontic types for a sentence and their association with the parties which is important for contract understanding. Ash et al. (2020) propose a rule-based unsupervised approach to identify deontic types with respect to a party and compute statistics for the rights and duties for a party. However, rule-based approaches have limitations as mentioned above. Recently, Funaki et al. (2020) curated an annotated corpus of contracts for recognizing rights and obligations along with the parties using LegalRuleML (Athan et al., 2013). However, the corpus is not publicly available, does not annotate for modal triggers, and does not cover all the deontic types expressed in a contract. In Chapter 4, we address all these limitations of the existing works by introducing party-specific deontic modality detection tasks, dataset, and models.

2.4.3 Summarization in Legal Domain

Another important task in the legal domain is that of summarization as these documents are extremely long, making them difficult to consume and seek the required information. Existing works focus on summarizing legal case reports (Galgani and Hoffmann, 2010; Galgani et al., 2012; Bhattacharya et al., 2021; Agarwal et al., 2022; Elaraby and Litman, 2022), civil rights (Shen et al., 2022), terms of service agreements (Manor and Li, 2019b), and privacy policies (Tesfay et al., 2018; Zaeem et al., 2018; Keymanesh et al., 2020). Hybrid approaches combining different summarization methods (Galgani et al., 2012) or word frequency (Polsley et al., 2016) with domain-specific knowledge have been used for

legal case summarization task. Other works either propose to identify sections of privacy policies with a high privacy risk factor to include in the summary by selecting the riskiest content (Keymanesh et al., 2020) or directly generate section-wise summaries for employment agreements (Balachandar et al., 2021) using existing extractive summarization methods (Kullback and Leibler, 1951; Mihalcea and Tarau, 2004; Erkan and Radev, 2004; Ozsoy et al., 2011). However, all these works consider generating a single summary for a document. In Chapter 5, we argue that a single summary may not serve the purpose of all the contracting parties instead, we introduce a new task, dataset, and model for generating *party-specific* summaries of what must, can, and cannot be done under an agreement by learning to rank sentences belonging to each of these categories based on their importance as decided by legal experts.

2.4.4 Entity-Conditioning

A standard way to specify a condition or constraint in transformer-based models to pay extra attention to the input (both during training and inference) is to represent it as a special token (*e.g.*, $\langle[Tenant]\rangle$) pre-pended to the input X, which acts as a side constraint. The encoder learns a hidden representation for this token as for any other vocabulary tokens, and the final representation of the input is influenced by this token. This simple strategy has been used in sequence generation tasks such as machine translation (Sennrich et al., 2016) to control second-person pronoun forms when translating into German and to control style, content, and task-specific behavior for conditional language models (Keskar et al., 2019). To realize these side constraints, special tokens need to be added to the vocabulary of the model (*i.e.*, to the tokenizer).

2.5 Summary

In this chapter, we provided a detailed overview of the field of existing entity-centric NLU tasks, basic modeling concepts, an overview of the tasks in the field of narrative comprehension, and legal language understanding. We identify the following gaps in the literature that we aim to close with our work as detailed below:

1. **Evaluating and Inducing protagonist-oriented script knowledge in LLMs.** While LLMs have been shown to perform well on several natural language understanding tasks, the extent of implicit script knowledge present in these models is never systematically evaluated. To close this gap, unlike cloze-based script evaluation, we evaluate this aspect by introducing the task of generating full ESDs from natural language prompts. Based on the findings, we propose a script induction framework that is able to generate protagonist-oriented ESDs for a given scenario (Chapter 3). This work contributes towards modeling knowledge that humans implicitly use in the form of protagonist-oriented sequences of events for a scenario, required for narrative comprehension.

2. **Identifying different party-specific deontic modalities in legal documents.** Existing works either do not cover all the deontic modalities that are present in legal documents, or associate parties with the corresponding modalities. We present a linguistically-informed taxonomy of deontic modalities, use it to collect a dataset that is released publicly, and benchmark the dataset on two party-specific tasks, that aim to identify deontic modalities with respect to each of the parties, using transformer-based models. (Chapter 4). We contribute by designing a task, dataset, and party-specific legal document understanding systems that help in improving the identification and extraction of static and explicit information specific to each contracting

party.

3. **Party-specific summarization of legal documents.** Existing works generate a single summary for a legal document however, each party has different importance for the content of the legal documents. We introduce a task of party-specific summarization of important deontic modalities, collect party-specific pairwise-importance annotations for sentences in legal documents, and propose a pipeline-based model for the introduced task (Chapter 5). Through this work, we show how entity-centric understanding enables developing a practical application, *i.e.*, contracting party-specific summarization, that can make complex contracts more accessible, and aid in contract review or negotiations.

4. **Tracking fine-grained evolution in entity relationships.** Prior works that track evolution in character relationships either use a coarse-grained (cooperative vs non-cooperative) or unsupervised relationship ontology. Further, they either use book summaries or a subset of passages from the book owing to the limited context window of models at that time. To address these gaps, in Chapter 6, we consider modeling evolution in relationships between characters as a task of predicting relationship and evolution types from a predefined set of fine-grained categories. We propose several strategies varying in terms of the granularity of information used to do the prediction task using large language models with large context window sizes and improved reasoning abilities. Owing to the difficulty of obtaining annotated datasets at a book-length, we thoroughly analyze the predictions by computing agreements and a manual examination. We show that tracking evolution in relationships is a challenging task even with improved modeling capacity due to the multi-faceted, complex, and uncertain nature of the prediction task. Our manual analysis reveals that language models may adopt surface-level heuristics, and lack a proper understanding of

the context raising questions on their *true* understanding of social relationships.

5. **Investigating the influence of implicit character attributes on relationship predictions.** Prior works have shown that relationship prediction is challenging due to its multi-faceted nature, and that large language models may adopt heuristics and surface-level cues for such predictions. We believe that there may be several other causal factors such as the gender, age, and race of the characters, their tone, and the location or setting of the story that might influence predictions from language models. To this end, in Chapter 7, we aim to investigate the impact of changing implicit character attributes such as gender and race (as identified by their name) on relationship predictions from large language models.

Chapter 3: Generating Protagonist-oriented Sequence of Events[1]

In Section 2.3.1, we introduced the concept of scripts central to language understanding research. Scripts are fundamental pieces of commonsense knowledge that humans share and assume to be tacitly understood by each other. When someone says *"I went to a restaurant for lunch"*, our script knowledge allows us to infer that the speaker would have arrived at the restaurant, got herself seated, a waiter would have taken the order, the speaker would have eaten the lunch, paid for it, and tipped the waiter, even if these events are not explicitly mentioned. Knowledge of scripts, whether implicit or explicit, has been recognized as important for language understanding tasks (Miikkulainen, 1995; Mueller, 2004; Modi et al., 2017) as it can enable an AI system to understand and fill-in the implicit events in any narrative. Consider this example narrative: *John drove to the Olive Garden. He ordered lasagna. He left a big tip and went home.* When asked the question 'What did John eat?', although the event of eating is not explicitly mentioned in the narrative, using the restaurant script knowledge humans can infer that food is ordered to be eaten and therefore John ate lasagna.

As pretrained language models (Devlin et al., 2019a; Liu et al., 2019; Radford et al., 2019) have revolutionized natural language understanding (NLU) with their impressive results in various tasks such as named entity recognition, and sentiment analysis, in this chapter, we aim to investigate the extent and accessibility of explicit script knowledge present

[1] Abhilasha Sancheti, and Rachel Rudinger. 2022. What do large language models learn about scripts? In *Proceedings of 11th Joint Conference on Lexical and Computational Semantics*, Seattle, Washington. Association for Computational Linguistics.

Scenario Baking a cake

Event Sequence Description (ESD) 1. gather ingredients 2. mix cake mix, eggs and water in bowl 3. pour into pan 4. turn on oven 5. put in oven and bake at specified temperature 6.remove cake from oven to cool 7. turn off oven 8. mix frosting 9. frost cake 10. serve cake 11. refrigerate any leftovers.

Input to LM Here is a sequence of events that happen when you bake a cake :

Natural Language Prompt **Scenario**

Figure 3.1: Sample event sequence description (ESD) from Wanzare et al. (2016) for BAKING A CAKE scenario. We use prompts (Table 3.2) to *generate completely ordered* ESDs for evaluating the extent of script knowledge accessible through LMs in Chapter 3.

in pretrained language models. In this chapter, we want to evaluate pretrained language models on the ability to *fully generate* event sequence descriptions (ESDs) (Regneri et al., 2010) from a protagonist's point of view, given a scenario, in free-form natural language (Figure 3.1). This is a challenging task as scripts are complex structures with varied granularity of describing a scenario (*e.g.*, starting from going to grocery store to buy ingredients or starting with finding a recipe for BAKING A CAKE scenario), and the requirement to produce all the scenario-*relevant* events in the correct temporal *order* ensuring that the state of the protagonist and properties of different entities involved in a scenario are tracked properly.[2]

3.1 Probing for Script Knowledge

We design a zero-shot probing experiment to evaluate pretrained language models' ability to generate ESDs by carefully selecting natural language prompts, which LMs are known to be sensitive to (Bouraoui et al., 2020). We experiment with 16 manually crafted

[2]Code and dataset are available at https://github.com/abhilashasancheti/script-generation

1	**these are the things that happen when you bake a cake:** 3 tablespoons (30ml) of unsalted butter, at room temperature 2 tablespoons (30ml) of granulated sugar 2 tablespoons (30ml) of brown sugar, sifted 1 large egg, at room temperature 1 teaspoon vanilla extract 3 large egg yolks 1 cup (350ml) buttermilk To the bowl of an electric mixer fitted with a paddle attachment, add the butter, sugar, brown sugar, egg yolks, and vanilla extract and beat until light and fluffy. Add the buttermilk, then continue beating until fully incorporated. Sift the flour, baking powder, and salt into the bowl of the mixer and beat
2	**here is an ordered sequence of events that occur when you bake a cake:** 1. The cake is baked. 2. The cake is cooled. 3. The cake is served at the table. 4. The dessert is placed on the table. 5. The cake is placed in a baking pan. 6. The cake is transferred to the oven. 7. The cake is put to the middle of the oven and baked. 8. The cake is left in the oven until it is done. 9. The cake is removed from the oven and placed on a plate. 10. The cake is transferred to a cake rack. 11. The cake is placed on a cake plate.
3	**describe baking a cake in small sequences of short sentences:** "I want my cake to be soft and moist." (And, you know, don't do this: "If I use a pastry bag, I can get it back to me in less than two minutes.") You may be looking for the most simple recipe for cake, but baking requires skill and a lot of practice. You don't need to learn how to make a cake recipe. You don't even need to know the difference between the cake and the sponge. All you need is a great recipe and a couple of ovens and an oven thermometer. 2. Use a mixer. Mixers are really, really good. (A great idea is to invest in one of those mini, little, portable ones.
4	**here is a sequence of events that happen while baking a cake: 1.** The cake goes into the oven. 2. The cake cools down in the oven, because the water vapor from the oven condenses onto the top of the cake. 3. The cake is cooled and the frosting set (the sugar crystals are not allowed to fall off). 4. The frosting is set (a little more water vapor condenses onto the top of the frosting). 5. The frosting is set (more water vapor condenses onto the top of the frosting). 6. The cake cools down (the water vapor cools off). 7. The cake comes out of the oven and it is ready to eat. How is this cake not a cake? The first step is that the cake is a cake.
5	**here is an ordered sequence of events that occur when you bake a cake: 1.** You have a cake. 2. You have a batch of flour (or other ingredient, such as water, sugar or corn syrup) 3. You have your flour prepared 4. You have your flour mixture 5. You have your flour mixture pre-floured 6. You have your batter prepared 7. You have your batter made 8. You have your batter placed on a baking sheet 9. You have your batter removed from the oven 10. You have your cake cooled in the tray (you must place your cake in the tray before you take it out of the oven) 11. You take your cake out of the tray and remove it from the oven 12. You eat your cake, or leave it out to cool.
6	**here is a sequence of events that happen while baking a cake: 1. get a cake mix** 2. pour the cake mix into the pan 3. let it bake for 10 minutes 4. get out the oven lid and turn it on 5. let it bake for another 10 minutes 6. get out the oven lid and flip it on 7. let it bake for another 10 minutes 8. get out the oven lid and switch it back on 9. let it bake for another 10 minutes 10. get out the oven lid, turn it back on, and let it bake again 11. get out the oven lid and get out the knife 12. bake the cake again, and then let it cool on a rack 13. use the cake mix to make cupcakes 14. cut the cake, and then use the cupcake for serving!
7	**these are the things that happen when you bake a cake: 1. get a cake mix 2. gather together other ingredients** that you'll need for your cake (coconut oil, powdered sugar, eggs, etc.) 3. preheat oven to 350 4. pour on flour and mix by hand 5. pour in egg and mix by hand 6. add in powdered sugar and mix by hand 7. add in coconut oil and mix by hand 8. let it all sit for a minute 9. add in your flour and mix by hand 10. add in your sugar and mix by hand 11. add in your flour and mix by hand 12. add in your flour and mix by hand 13. add in your flour and mix by hand 14. add in your flour and mix by hand 15. add in your egg and mix by hand 16. add in your salt
8	**describe baking a cake in small sequences of short sentences: 1. get a cake mix 2. gather together other ingredients** 3. start to bake 4. bake a cake 5. take it to the oven 6. the cake is done 7. is it done? 8. is it still hot? 9. what is that crust? 10. what does the cake look like? If we think of a sequence of a few hundred or so words, we can take the basic idea of this example and say that we could write an equivalent, yet shorter, sentence: 1. Get a cake mix 2. Get together the ingredients 3. Bake a cake 4. Take it to the oven 5. The cake is done 6. Is it done? 7. is it hot? 8. what is that crust? 9. what does the cake look like

Table 3.1: Scripts generated from GPT2-L for BAKING A CAKE scenario with **bold-faced** prompts.

Prompt Beginnings		Continuations
here is a sequence of events that happen while baking a cake:		None
these are the things that happen when you bake a cake:	⊕	1.
describe baking a cake in small sequences of short sentences:		1. get a cake mix
here is an ordered sequence of events that occur when you bake a cake:		1. get a cake mix 2. gather together other ingredients

Figure 3.2: Different prompt formulations for BAKING A CAKE scenario for probing. 16 prompts are created by combining a prompt beginning with a continuation.

prompts[3] (Figure 3.2) with different phrasing and levels of conditioning to enquire large versions of GPT2, BART, and T5 for script knowledge. The intuition behind these prompts is similar to asking questions (prompts) to a knowledge source in various ways to get the required answer (ESD for a scenario).

BART and T5 were not able to output anything except the input prompt or start, end, and pad tokens and hence we only present qualitative outputs from GPT2, when probed with various prompts for BAKING A CAKE scenario, in Table 3.2. We observe that the quality of generated ESDs vary for different prompts. Although GPT2 is able to generate some scenario-relevant events with just the prompt beginnings and no continuations (e.g., 1 and 2 in Table 3.2), the ESDs are incomplete with many auxiliary details, and incorrect event ordering (e.g., '3. The cake is served at the table' before '6. The cake is transferred to the oven.' in 2). It sometimes outputs (e.g., 4) narrations rather than procedural descriptions. As generation from scratch is an open-ended task, we use a prompt with a number to guide GPT2 to generate a procedural script. Although 4 and 5 are more procedural, the events are still at a coarse-grained level with most of the intermediate events missing. To further guide the generation towards a fine-grained level, we condition the generation on a few events (manually curated by authors looking at sample ESDs) along with the prompt

[3]We also experiment with capitalized prompts but did not find any significant change in the quality of generations.

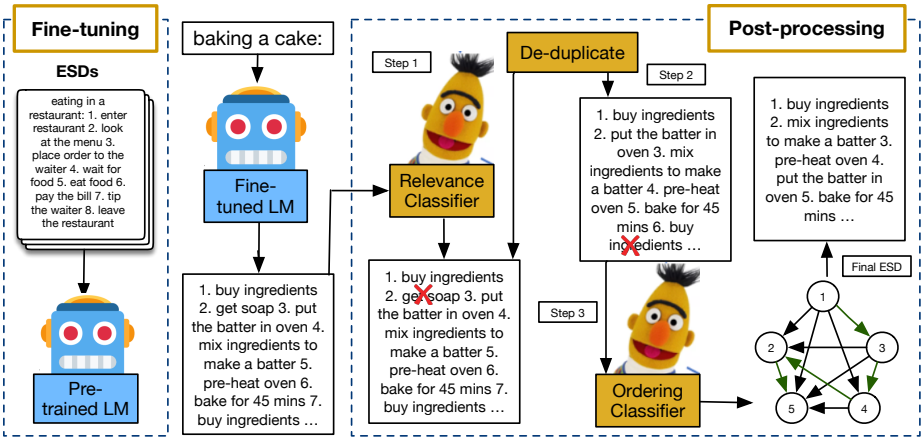


Figure 3.3: SIF: Pretrained language model is fine-tuned on DeScript (Wanzare et al., 2016). Generated scripts are then post-processed with RoBERTa-based classifiers to correct for event relevance (Step 1), repetition (Step 2), and temporal ordering (Step 3).

beginning. This helps us in examining whether GPT2 has temporal knowledge about the events related to a scenario. Conditioning on the events results in a better quality ESD (e.g., 6, 7, 8). However, there is a repetition of events ('let it cool for another 10 minutes' in 6, 'add in your flour and mix by hand' in 7) in addition to wrong event ordering, irrelevant (e.g., 'is it hot?' in 8) and missing events. As GPT2 produces poor quality ESDs in this zero-shot setting with BART and T5 not even being able to output any events, we propose a script induction framework detailed in the following section.

3.2 SIF: Script Induction Framework

In this section, we provide details on our pipeline-based script induction framework, SIF (Figure 3.3), which addresses the limitations of zero-shot ESD generation. SIF is a two-staged framework that fine-tunes LM on a small set of ESDs in the first stage. In the second stage, ESDs generated using the fine-tuned LM are passed through a sequence of RoBERTa-based classifiers Liu et al. (2019) to identify relevant events, remove redundant events, and predict pair-wise temporal ordering between the events. These pair-wise or-

SEQUENCE here is a sequence of events that happen while baking a cake: 1. e_1 2. e_2
EXPECT these are the things that happen when you bake a cake: 1. e_1 2. e_2
ORDERED here is an ordered sequence of events that occur when you bake a cake: 1. e_1 2. e_2
DESCRIBE describe baking a cake in small sequences of short sentences: 1. e_1 2. e_2
DIRECT baking a cake: 1. e_1 2. e_2
TOKENS ⟨SCR⟩ baking a cake ⟨ESCR⟩: 1. e_1 2. e_2
ALLTOKENS ⟨SCR⟩ baking a cake ⟨ESCR⟩: ⟨BEVENT⟩ e_1 ⟨EEVENT⟩ ⟨BEVENT⟩ e_2 ⟨EEVENT⟩

Table 3.2: Different prompt formulations for BAKING A CAKE scenario with two events (e_1 and e_2).

derings are then used to create a full event ordering using topological sorting on a directed graph created from the predicted orderings.

3.2.1 Stage I: Fine-tuning pretrained language models

Pretrained language models fine-tuned on commonsense datasets like ATOMIC (Sap et al., 2019) can generalize beyond the scenarios observed during fine-tuning (Bosselut et al., 2019). Hence, we investigate the learning capability of LMs when a small number of script examples are available. We fine-tune LMs on ESDs using different natural language and pseudo-natural language prompt formulations for encoding ESDs (Table 3.2) to study the effect of prompt formulations on this task as observed during the probing experiments. We fine-tune LMs using negative log-likelihood objective.

3.2.2 Stage II: Post-processing Generated ESDs

We sample ESDs for an unseen (*i.e.*, not seen during fine-tuning but we acknowledge that pretrained language models might have seen these scenarios during their pretraining stage due to the use web-scraped data from a variety of sources) scenario using the fine-tuned LMs and employ a 3-step post-processing method to correct them for relevance, repetitions, and ordering.

Step 1: Irrelevant Events Removal. The first post-processing step is to remove non-scenario-relevant events from an ESD. An event is not relevant for a scenario if it is not a part of the scenario (e.g., 'tipping a waiter' is not a part of BAKING A CAKE scenario). For irrelevant events removal, we first need to identify irrelevant events for a scenario. We pose this identification problem as a binary classification task to predict if a given event belongs to a given scenario. For training purpose, a positive example is constructed by pairing a scenario with an event belonging to that scenario; negative samples are drawn from another scenario in the training data. Using this data, we train a RoBERTa-L-based Liu et al. (2019) classifier and remove those events from an ESD which are predicted as irrelevant by this classifier.

Step 2: Event De-duplication. The second step involves the identification and removal of repeated events. Repetition of events can occur by an exact copy of an event or by a paraphrase of an event (e.g., '6. You have your batter prepared' and '7. You have your batter made' in 5 of Table 3.1). To identify such de-duplications, we train a RoBERTa-L-based paraphrase identification system using MRPC (Dolan and Brockett, 2005) dataset. However, we observe many false-positives (e.g., 'open a faucet' and 'close a faucet' were identified as paraphrases) with this system. Since false-positives can lead to unnecessary removal of events, we employ a conservative approach of only identifying repeated events. We find edit distance between each pair of events in an ESD and remove multiple occurrences of an event from the ESD, as identified by the edit distance score of 0.

Step 3: Temporal Order Correction. The final step is to correct the order of events in an ESD. We correct the ESDs for ordering by first obtaining pair-wise event orderings and then using a graph-based approach to get the final overall ordering. We pose the problem of pair-wise event ordering as a binary classification task to predict if the order of a given

pair of events is correct with respect to the given scenario. We sample event pairs from gold ESDs to construct positive (sequence order) and negative (reverse order) examples to train a RoBERTa-L-based classifier. Topological sort is then used to get the final ESD for a scenario from the ordering predictions for all the $\binom{N}{2}$ pairs of events in an ESD. We construct a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of events in a scenario with events as nodes ($\mathcal{V}$) of the graph and a directed edge from node $v_1 \in \mathcal{V}$ to $v_2 \in \mathcal{V}$ if event represented by v_2 is predicted to occur after the event represented by v_1. We keep the original ordering of events in case the constructed graph is cyclic[4] due to incorrect predictions from the classifiers.

3.2.3 Implementation Details

Dataset preproccessing. We fine-tune LMs on ESDs from DeScript (Wanzare et al., 2016) dataset which consists of 100 ESDs each for 40 scenarios, collected via crowdsourcing. The scenarios are randomly partitioned into 8 folds with each fold consisting of ESDs from 5 scenarios to perform 8-fold cross-validation of `SIF` for each of the prompt formulations. We lowercase and enclose each ESD within a begin of scenario $\langle$`BOS`$\rangle$ and an end of scenario $\langle$`EOS`$\rangle$ token for fine-tuning. The input to the relevance classifier is: `scenario` $\langle$`/s`$\rangle$ e and to the temporal classifier is `scenario name` $\langle$`/s`$\rangle$ e_1 $\langle$`/s`$\rangle$ e_2, where $\langle$`/s`$\rangle$ is a separator token and e, e_1, e_2 are events.

Training details. We use huggingface's transformers library (Wolf et al., 2020) to fine-tune LMs on each of the 7 prompt formulations, leading to 7 variations for each LM, for 1 epoch with a batch size of 1, gradient accumulation per 16 steps, and block size of 150. Train time for fine-tuning GPT2-L (774M parameters) on each of the folds for each of the variant was 13 ($\pm$1) minutes on an average. It took 10 minutes to train BART-L (407M parameters) on each of the folds for each of the variants. At inference time, 5

[4] $66 \pm 15\%$ (averaged across all the input variants and folds) of the complete graphs are acyclic for GPT2.

Fold	Scenarios	Held-out
1	baking a cake, borrowing a book from the library, flying in an airplane, going on a train, riding on a bus	cooking pasta
2	getting a hair cut, going grocery shopping, planting a tree, repairing a flat bicycle tire, taking a bath	going bowling
3	eating in a fast food restaurant, paying with a credit card, playing tennis, going to the theater, taking a child to bed	planting a tree
4	washing dishes, making a bonfire, going to the sauna, making coffee, going to the swimming pool	going grocery shopping
5	taking a shower, ironing laundry, taking a driving lesson, going to the dentist, going to a funeral	taking the underground
6	washing one's hair, fueling a car, sending food back (in a restaurant), changing batteries in an alarm clock, checking in at an airport	paying with a credit card
7	having a barbecue, ordering a pizza, cleaning up a flat, making scrambled eggs, taking the underground	eating in a fast food restaurant
8	renovating a room, cooking pasta, sewing a button, doing laundry, going bowling	getting a hair cut

Table 3.3: Dataset partitioning details. Held-out refers to the scenarios kept aside from training folds for validation purposes of relevance and temporal ordering classifiers.

ESDs are sampled for each of the given scenarios with top 50 probable tokens, nucleus sampling (Holtzman et al.) probability of 0.9, and maximum length set at 150. We use RoBERTa-L architecture (355M parameters) from the transformers library for relevance and temporal order classifiers. Relevance (Temporal) classifier is trained for 10 (5) epochs with average validation accuracy of 84.50% (83.87%) across the folds. It took 282 minutes on an average to train relevance classifier for each of the folds while 294 minutes for temporal ordering classifier. Relevance and temporal ordering classifiers were trained on each of the training fold by keeping aside all the ESDs from a randomly chosen scenario for validation purposes as shown in Table 3.3. The model with the best accuracy on the valid split is used in the post-processing stage. We use python's *editdistance* library to compute edit distance for the de-duplication step. We use Adam optimizer with an initial learning rate of $2e^{-5}$, warm-up steps set at 0.06 of total steps, batch size of 16, and maximum input

Models	TOKENS	EXPECT	SEQUENCE	ALLTOKENS	DESCRIBE	DIRECT	ORDERED
(1) Zero-shot	03.1 (5.2)	03.6 (5.5)	05.4 (2.8)	03.1 (5.2)	03.2 (3.6)	03.9 (5.1)	06.2 (6.6)
(2) GPT2-$L_{SCRATCH}$	17.2 (3.1)	19.3 (3.7)	16.8 (2.9)	18.6 (4.5)	17.6 (2.6)	14.4 (3.9)	17.7 (3.2)
(3) BART-FT	15.5 (6.0)	20.8 (3.5)	19.6 (3.5)	19.7 (9.2)	19.2 (3.9)	18.0 (6.6)	11.7 (4.8)
(4) GPT2-FT	30.7 (5.1)	31.3 (5.5)	32.4 (6.3)	30.7 (6.6)	32.3 (5.9)	31.4 (5.8)	31.0 (4.8)
(5) BART-**SIF**	16.8 (5.1)	21.1 (4.2)	19.9 (3.7)	20.5 (11.1)	20.0 (3.8)	19.6 (7.2)	13.7 (5.0)
(6) GPT2-**SIF**	**33.6** (5.4)	**33.9** (5.6)	**35.2** (6.9)	**32.5** (6.9)	**34.2** (5.3)	**33.6** (5.7)	**33.2** (5.5)

Table 3.4: Automatic evaluation results: Mean BLEU scores (and std. dev.) over 8 folds of held-out scenarios are reported. (1) is pretrained GPT2 (no fine-tuning or post-processing); (2) is randomly initialized GPT2 with fine-tuning; (3-4) are fine-tuned BART and GPT2; (5-6) are **SIF** applied to BART and GPT2.

Models	TOKENS	EXPECT	SEQUENCE	ALLTOKENS	DESCRIBE	DIRECT	ORDERED
(1) GPT2-FT	30.7 (5.1)	31.3 (5.5)	32.4 (6.3)	30.7 (6.6)	32.3 (5.9)	31.4 (5.8)	31.0 (4.8)
(2) GPT2-FT+Relevance (R)	33.1 (5.1)	33.1 (4.9)	34.7 (6.9)	31.9 (6.7)	33.7 (5.0)	32.6 (5.8)	33.2 (5.2)
(3) GPT2-FT+R+De-duplicate (D)	33.5 (5.2)	33.6 (5.2)	35.1 (6.9)	32.1 (6.7)	**34.3** (5.0)	32.9 (5.7)	**33.6** (5.5)
(4) GPT2-FT+R+D+Reorder (GPT2-SIF)	**33.6** (5.4)	**33.9** (5.6)	**35.2** (6.9)	**32.5** (6.9)	34.2 (5.3)	**33.6** (5.7)	33.2 (5.5)

Table 3.5: Ablation analysis of each step in the proposed pipeline for GPT2. Mean BLEU scores (and std. dev.) over 8 folds of held-out scenarios are reported. (1) fine-tuned GPT2; (2-4) are fine-tuned GPT2 with successive post-processing steps.

length 150 for both the classifiers. All the models are trained and tested on NVIDIA Tesla V100 SXM2 16GB GPU machine.

3.3 Evaluation

We use SIF to induce script knowledge in GPT2, BART, and T5, and evaluate full ESDs generated for a given unseen scenario using **BLEU** metric (Papineni et al., 2002), following Pichotta and Mooney (2016b) who use BLEU to score individual LM-generated events. As BLEU is a precision-based metric, we measure n-gram overlap of the sampled ESDs against multiple gold-reference ESDs[5] for each scenario in the test fold.

Additionally, for deeper analysis of the generated ESDs, two of the authors evaluate a subset of the generated ESDs (blinded to the identity of the models and prompt variants)

[5]We use NLTK python library to calculate BLEU score with add-1 smoothing function and n-grams upto $n = 4$. We convert the outputs of different variants & gold references into numbered form, 1. e_1 2. e_2 ... n. e_n for a fair comparison.

on three levels – individual events (**Relevance (R)**), pairwise events (**Order (O)**), and the overall sequence (**Missing (M)**). **R** measures the % of generated events relevant to a scenario; **O** measures the % of consecutive event pairs correctly ordered given a scenario; and **M** measures the degree to which important events are missing on a 4-point Likert scale defined as (1) no or almost no missing events, (2) some insignificant missing events, (3) notable missing events, and (4) severe missing events. As scripts are complex structures and require an understanding of scenarios, we chose not to resort to a crowdsourcing platform for manual analysis. We manually analyze the outputs to evaluate SIF as well as perform an error analysis to identify opportunities for future research directions.

We evaluate our framework on scenarios in each of the eight folds as well as novel scenarios from Regneri et al. (2010), and day-to-day activities. As we do not have access to gold-reference ESDs for the novel scenarios, we demonstrate our framework's performance only using manual evaluation.

3.4 Results and Analysis

3.4.1 Automatic Evaluation

We present the automatic evaluation results on held-out scenarios in Table 3.4. As baselines, we report scores from non-fine-tuned GPT2-L (Zero-shot), a randomly-initialized GPT2-$L_{SCRATCH}$ model fine-tuned on DeScript ESDs, and BART-FT and GPT2-FT models which are fine-tuned in the first stage of SIF. We do not report any results for T5 as it was even struggling to learn the input ESD formulations during fine-tuning. We explain the findings from automatic evaluation below.

SIF significantly outperforms fine-tuning baselines. Both GPT2-SIF and BART-SIF have higher BLEU scores as compared to their corresponding fine-tuned (GPT2-FT and

BART-FT) models across all the prompt variants. This clearly reflects the advantage of the post-processing stage in SIF framework. Improvement across different LMs reinforces the LM-agnostic nature of our framework. Variation in the extent of induction across prompt variants indicates the sensitivity of LMs to prompt formulations.

Script knowledge is best accessible through GPT2 than other LMs. As previously mentioned in probing experiments, BART and T5 were not able to output anything useful in the zero-shot setting while GPT2 could produce ESDs, although erroneous and of poor quality. We observe same trends even after fine-tuning these LMs or using SIF to induce script knowledge in these LMs. Interestingly, a randomly initialized and fine-tuned GPT2 (GPT2-$L_{SCRATCH}$) is able to perform comparable to a pretrained BART fined-tuned using DeScript (BART-FT), and even better for TOKENS and ORDERED variants. Overall, GPT2 is found to be better than BART in terms of the presence and accessibility of script knowledge through them. One possible explanation for this is that GPT2 is a generative language model while BART and T5 are encoder-decoder-based language models making it challenging to encode complete script knowledge within a scenario name.

Performance across LMs is sensitive to prompt formulation and scenario. We consistently observe variation in performance across prompt variants. Moreover, this variation is also observed across LMs. For BART, EXPECT outperforms other prompt variants while SEQUENCE performs the best for GPT2. High variance across folds also shows that different prompts perform differently depending upon a scenario. This indicates the sensitivity of LMs to prompt formulations and thus justifies our experiments with different prompt formulations to study the extent of script knowledge that can be accessed through pretrained language models.

Variants	BLEU↑	Manual Evaluation		
		R↑	O↑	M↓
TOKENS	19.2/**22.8**	77.2/**84.3**	72.3/**89.3**	**2.6**/**2.6**
EXPECT	22.8/**26.0**	81.9/**82.7**	74.5/**86.5**	**3.0**/**3.0**
SEQUENCE	27.8/**33.4**	73.3/**83.2**	74.0/**87.5**	**2.5**/**2.5**
ALLTOKENS	33.5/**35.0**	83.5/**85.7**	82.7/**89.5**	**2.6**/**2.6**
DESCRIBE	27.1/**28.6**	80.7/**86.3**	83.9/**85.9**	**2.8**/**2.8**
DIRECT	30.9/**34.1**	81.2/**84.2**	**88.5**/86.1	**2.6**/**2.6**
ORDERED	**31.9**/31.5	84.9/**86.2**	78.6/**86.8**	**2.6**/**2.6**

Table 3.6: Manual and BLEU scores on fine-tuned GPT2 (GPT2-FT) SIF applied to GPT2 (FT/SIF), computed for a stratified sample of outputs (one ESD per scenario across two folds). Mean scores across two annotators are reported. Annotator agreement is measured with Cohen's Kappa (Cohen, 1960) (κ=0.61 for **O**, κ=0.56 for **R**) and Spearman's correlation (ρ=0.64 for **M**). Underline and **bold** denotes the best across variants, and between FT and Ours, respectively. O scores are calculated only when both the events are marked as relevant by the two annotators.

3.4.2 Ablation Analysis of **SIF**

We next analyze the contribution of each the stage of SIF and each step of stage II leading to improvement in the performance via an ablation study, on GPT2, in Table 3.5. As expected stage I contributes maximum to the performance boost and there is a consistent improvement in BLEU after each of the post-processing steps except in the case of DESCRIBE and ORDERED wherein, reordering leads to a slight decrease in BLEU as the trained classifiers are not perfectly accurate. We present qualitative outputs when SIF is used to induce script knowledge in GPT2 in Table 3.8.

3.4.3 Manual Evaluation and Error Analysis

We manually evaluate a total of 140 ESDs (for M) comprising 652 individual events (for R) and 582 consecutive pair of events (for O) generated from GPT2-FT and GPT2 SIF across all the prompt variants (Table 3.6). BLEU scores are also reported for the same set of

ESDs to study the correlation between manual and automatic metrics. We find that outputs from SIF have higher BLEU, R, and O scores than FT across all prompt variants (except O for DIRECT and BLEU for ORDERED). M scores do not change, which shows that significantly important events are not dropped during the irrelevant events removal step. Different prompts perform well in different aspects. DESCRIBE generates most relevant events, ALLTOKENS has the best temporal ordering knowledge, and SEQUENCE leads to least severe missing events after Stage II of SIF. To our surprise, we find no statistically significant correlation between BLEU and any of the manual evaluation metrics (pearson correlation between BLEU and R, O and M was $r = 0.23, -0.06, -0.49$ with p> 0.1, respectively), emphasizing a need for more sophisticated automatic metrics than BLEU for evaluating full ESDs, having a complex structure. The best performing variant as per BLEU score differs from the best one in Table 3.4 due to variance in performance across scenarios as well as different sampled ESDs of the same scenario in Table 3.6. Manual examination reveals that a model can miss significant events, even though it can generate many relevant ones. As we only de-duplicate multiple occurrences of exactly the same events in a scenario, we observe repeated paraphrases (4.6% across all prompt variants) of the same event, such as 'pour some milk in the pot' and 'pour the milk into the coffee pot' (MAKING COFFEE scenario). 23.9% of the irrelevant events (13.5% across all prompt variants) are incoherent ('take the flat to the bathroom' for CLEANING A FLAT), 11.4% mixed ('sit in front of coffee shop' for MAKING COFFEE), 61.4% unrelated ('add shampoo' for WASHING DISHES), and rest ungrammatical.

We present a manual evaluation of novel scenarios to gauge the generalizability of our framework in Table 3.7. The framework generalizes to most novel scenarios except for those involving very granular events like MAKING GINGER PASTE or TYING SHOE LACES. Although GPT2 is a contextualized model, it confuses BUYING FROM VENDING MACHINE with buying from a store, SURFING THE INTERNET with the 'surfing' activity,

Scenario	R↑	O↑	M↓
Order fastfood online	81.5	84.6	2.6
Cook in a microwave	89.5	92.0	2.4
Answer telephone	65.5	91.7	2.0
Buy from vending machine	77.1	81.3	3.4
Tie shoe laces	65.8	66.7	3.6
Brush teeth	75.9	71.4	2.6
Make ginger paste	41.5	85.7	3.4
Attend a wedding	71.9	100.0	2.4
Wash a car	85.7	90.0	3.0
Take out trash	88.5	92.3	2.2
Take a taxi	85.7	76.2	2.0
Surf the internet	73.3	62.5	2.8
Watch television	77.4	73.7	3.0
Go to a club to dance	100.0	93.5	1.4
Average Score	77.1	83.0	2.6

Table 3.7: Manual evaluation of ESDs for novel scenarios. Averaged across 5 sampled ESDs per scenario generated using the best performing SEQUENCE variant of GPT2-SIF as per automatic measure.

or ATTENDING A WEDDING with 'getting married'.

Additionally, we provide a few good and bad quality outputs from GPT2 models for held-out (Table 3.8) and novel (Table 3.9) scenarios to identify the avenues for improving script induction in LMs. Qualitative outputs from BART-FT and BART-SIF are shown in Table 3.10. We observe that the outputs are of poor quality. Although SIF is able to filter out a few irrelevant events in all the examples and correct the ordering (e.g., GOING ON A TRAIN), there are many mistakes like the events are not coherent 'board the train Borders Boarders', or repetitions such as 'enjoy the train', 'enjoy train'. One reason for poor performance is that BART is an encode-decoder-based language model and thus encoding script knowledge in a scenario is a challenging task. On the other hand, GPT2 is a decoder-based language model that is trained with a language modeling objective and is able to generate the following events given a scenario in the form of a natural language prompt.

		Good quality generations
GOING ON A TRAIN	FT	1. get dressed 2. go to station 3. buy ticket 4. get on train 5. wait for train 6. get on train 7. sit in seat 8. read newspaper 9. wait for train 10. get off train 11. get dressed 12. leave station
	SIF	1. go to station 2. buy ticket 3. wait for train 4. get on train 5. sit in seat 6. get off train 7. leave station
FLYING IN AN AIR-PLANE	FT	1. go to airport 2. get boarding pass 3. go to the airline counter 4. present boarding pass and boarding pass id to attendant 5. present ticket and id to attendant 6. sit down in seat 7. make sure you are in seat 8. board airplane
	SIF	1. go to airport 2. go to the airline counter 3. present ticket and id to attendant 4. present boarding pass and boarding pass id to attendant 5. get boarding pass 6. board airplane 7. make sure you are in seat 8. sit down in seat
		Poor quality generations
CLEANING UP A FLAT	FT	1. get the flat 2. take the flat to the bathroom 3. take some water 4. take the flat to the sink 5. turn on the water 6. let it wash the flat 7. then put in some dryer sheet 8. get a sponge 9. scrub the flat 10. put the flat in the dryer 11. when done dry the flat with a towel 12. put away the flat
	SIF	1. get the flat 2. take the flat to the bathroom 3. take some water 4. take the flat to the sink 5. turn on the water 6. let it wash the flat 7. then put in some dryer sheet 8. get a sponge 9. scrub the flat 10. put the flat in the dryer 11. when done dry the flat with a towel 12. put away the flat
TAKING THE UNDER-GROUND	FT	1. arrive at the designated location 2. get in car 3. park in car 4. walk into the designated area 5. enter the room 6. find a chair 7. sit on a chair 8. watch movie
	SIF	1. walk into the designated area

Table 3.8: Scripts generated using SEQUENCE variant of GPT2 for held-out scenarios. FT denotes output from the fine-tuned model and SIF refers to outputs from our framework applied to GPT2.

3.5 Summary

In this chapter, we investigate whether pretrained language models are capable of generating full event sequence descriptions with minimal prompting and find that pretrained GPT2 has an incomplete understanding of scripts, while BART and T5 did not even produce

		Good quality generations
BRUSHING TEETH	FT	1. go into bathroom 2. turn on faucet 3. remove any dirt or debris 4. grab a brush 5. scrub and floss the teeth 6. leave the bathroom
	SIF	1. go into bathroom 2. grab a brush 3. scrub and floss the teeth 4. leave the bathroom
GOING TO A CLUB TO DANCE	FT	1. choose which club to attend. 2. drive or park your car. 3. get in your car. 4. go to the club. 5. enter the club. 6. get up and dance.
	SIF	1. choose which club to attend. 2. get in your car. 3. go to the club. 4. drive or park your car. 5. enter the club. 6. get up and dance.
TAKING A TAXI	FT	1. get in car 2. get into car 3. wait for taxi 4. enter the car 5. pay the fare 6. get out the driver 7. get out the door 8. exit car
	SIF	1. get into car 2. get in car 3. wait for taxi 4. enter the car 5. pay the fare 6. get out the driver 7. exit car 8. get out the door
		Poor quality generations
MAKING GINGER PASTE	FT	1. get your hot water 2. get your bowl 3. turn on the hot water 4. whisk a bowl of sugar into a paste 5. put the bowl on the stove 6. turn on the hot water 7. boil the paste 8. add salt to the paste 9. turn off the water 10. put the bowl on a rack 11. pour the hot water into a saucepan 12. put some salt and sugar in the saucepan 13. turn the heat on 14. pour the sauce on to the bowl 15. eat the paste
	SIF	1. get your hot water 2. get your bowl 3. turn on the hot water 4. whisk a bowl of sugar into a paste 5. put the bowl on the stove 6. boil the paste 7. add salt to the paste 8. put the bowl on a rack 9. pour the hot water into a saucepan 10. put some salt and sugar in the saucepan 11. turn the heat on 12. pour the sauce on to the bowl 13. eat the paste
SURFING THE INTERNET	FT	1. get in the car. 2. drive to the beach. 3. rent a surfboard. 4. go to the beach. 5. put on my sunglasses. 6. get in my car. 7. drive to my beach. 8. go to the surfing.
	SIF	1. 1. rent a surfboard. 2. get in the car. 3. go to the surfing.
ATTENDING A WEDDING	FT	1. schedule an appointment for the reception 2. drive to the venue 3. show your guest information 4. present your wedding ceremony/ reception to the guests 5. invite the friends/ family/ coworkers/ etc. to the wedding 7. present your wedding present to the guests 8. make the guest list 9. take the invitation 10. go to the reception 11. pay the fee/ gratuity 12. give your guest his/her ticket 13. leave the venue
	SIF	1. schedule an appointment for the reception 2. drive to the venue 3. show your guest information 4. present your wedding ceremony/ reception to the guests 5. invite the friends/ family/ coworkers/ etc. to the wedding 6. present your wedding present to the guests 7. make the guest list 8. take the invitation 9. go to the reception 10. leave the venue
BUYING FROM A VENDING MACHINE	FT	1. enter the shop 2. look for the item you wish to purchase 3. go into the register to purchase the item 4. pay for the item 5. take your item to your car 6. take the item to the car and remove the card 7. leave the shop 8. return the card 9. leave the shop
	SIF	1. enter the shop 2. look for the item you wish to purchase 3. go into the register to purchase the item 4. pay for the item 5. take your item to your car 6. take the item to the car and remove the card 7. leave the shop 8. return the card

Table 3.9: Scripts generated using SEQUENCE variant of GPT2 for novel scenarios. FT denotes output from the fine-tuned model and SIF refers to outputs from our framework applied to GPT2.[6]

anything useful through zero-shot probing experiments. We propose SIF, an LM-agnostic script induction framework, that is shown to produce meaningful ESDs for unseen scenarios and mitigate errors (such as scenario-irrelevant, repeated, and misordered events) that were observed during probing experiments, as measured by automatic and manual evaluation. Additionally, we provide evidence for the generalization capability of our framework

GOING ON A TRAIN	FT	1 2. go to the train station 3. board 4. get off 5. board the train Borders Borders 7. get a ticket 8. find a seat 9. enjoy train Borders 10. get dressed 11. ride the train 12. get home barriers 13. enjoy the train 14. pay for travel Alpha Alpha Omega Omega Omega Alpha Omega Delta Alpha Omega Alpha Alpha Alpha Delta Omega Beta Delta Delta Gamma Delta Delta Delta DDDDFDFEE bundles baggage transfer WTBSD and luggage transfer 14. get on board 15. get luggage assigned to correct location Bold TGBS Framework 6. ride to terminal DWDSD
	SIF	1. go to the train station 2. get a ticket 3. get luggage assigned to correct location Bold TGBS Framework 4. board the train Borders Borders 5. get on board 6. board 7. ride to terminal DWDSD TGBS 8. find a seat 9. enjoy the train 10. ride the train 11. enjoy train Borders 12. get off 13. get home barriers
HAVING A BARBEQUE	FT	1. set up the grill 2. put some water in the water 3. bring the grill out 4. throw a large amount of charcoal in the pit 5. open the fire 6. throw the lighter fluid into the fire 7. add the newspaper 8. wait for the flame to get hot 9. put a small amount of grease on the wood 10. enjoy it
	SIF	1. set up the grill 2. bring the grill out 3. throw a large amount of charcoal in the pit 4. open the fire 5. throw the lighter fluid into the fire 6. wait for the flame to get hot 7. put a small amount of grease on the wood 8. enjoy it
BORROWING A BOOK FROM A LIBRARY	FT	1. call the library 2. find a book to borrow 3. sign the book release paper 4. obtain the book 5. place the book in the book shelf 6. read the book reaching your full potential 7. go home autoritarian reaching the goal 8. pay for the book 9. start the car the goal 10. continue until you reach it 11. put the book away 12. check the cover of the book 13. read your favorite book 14. put a receipt in your wallet demonstrating that you read the correct book aut autoritarian style of aut aut
	SIF	1. call the library 2. find a book to borrow 3. obtain the book 4. pay for the book 5. check the cover of the book 6. read your favorite book 7. continue until you reach it 8. sign the book release paper 9. place the book in the book shelf 10. read the book reaching your full potential 11. put the book away

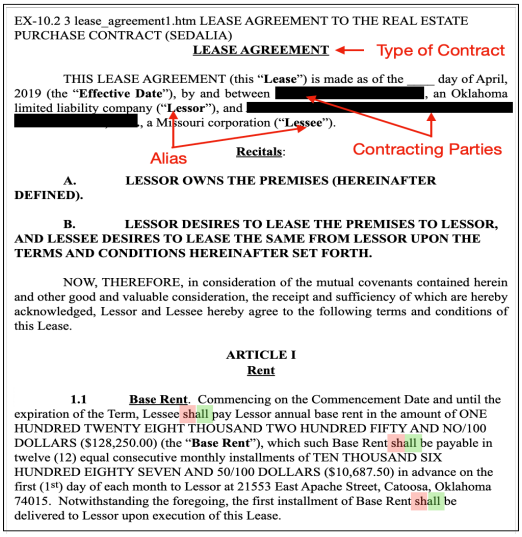
Table 3.10: Scripts generated using EXPECT variant of BART for held-out scenarios. FT denotes output from the fine-tuned model and SIF refers to outputs from our framework when applied to BART. We filter extra tokens (⟨s⟩, ⟨/s⟩, ⟨SEP⟩) generated in between the events to present a clean output. These extra tokens were not generated from GPT2.

to novel scenarios. However, manual error analysis and qualitative outputs suggest avenues for future work that may focus on developing more sophisticated automatic metrics and an end-to-end system for script induction which might help in mitigating cascading of errors, due to each component, common to any pipeline-based approaches.

While we consider events that happen in a prototypical scenario from a protagonist's perspective in this chapter, multiple entities may be involved in events that may not have happened but are necessary or possible to happen. Under such scenarios, same information may carry different implications for different entities involved in the same event. To explore this aspect, in the next chapter, we shift to the legal domain where deontic modalities are frequently used to express the rights and duties of entities involved in legally binding contracts.

Chapter 4: Party-specific Deontic Modality Detection in Legal Documents[1]

In this chapter, we shift to the legal domain wherein the legally binding contracts act as a good example of written communication where multiple contracting parties (instead of a single protagonist) are involved in events that may not have happened (in contrast to the previous chapter where we consider events that do happen in a scenario) but are necessary or possible to happen resulting in the same information carrying different implications for different parties. Deontic modality is frequently used to express obligations, entitlements, permissions, and prohibitions of various contracting parties (henceforth, "party") (Ballesteros-Lintao et al., 2016) involved in a contract. Layperson often signs contracts (*e.g.*, lease agreements, terms of services, privacy policies, EULA, etc.) without even reading them (Cole, 2015; Obar and Oeldorf-


EX-10.2 3 lease_agreement1.htm LEASE AGREEMENT TO THE REAL ESTATE PURCHASE CONTRACT (SEDALIA)

LEASE AGREEMENT ← Type of Contract

THIS LEASE AGREEMENT (this "**Lease**") is made as of the ____ day of April, 2019 (the "**Effective Date**"), by and between [redacted], an Oklahoma limited liability company ("**Lessor**"), and [redacted], a Missouri corporation ("**Lessee**").

Alias Contracting Parties

Recitals:

A. LESSOR OWNS THE PREMISES (HEREINAFTER DEFINED).

B. LESSOR DESIRES TO LEASE THE PREMISES TO LESSOR, AND LESSEE DESIRES TO LEASE THE SAME FROM LESSOR UPON THE TERMS AND CONDITIONS HEREINAFTER SET FORTH.

NOW, THEREFORE, in consideration of the mutual covenants contained herein and other good and valuable consideration, the receipt and sufficiency of which are hereby acknowledged, Lessor and Lessee hereby agree to the following terms and conditions of this Lease.

ARTICLE I
Rent

1.1 Base Rent. Commencing on the Commencement Date and until the expiration of the Term, Lessee shall pay Lessor annual base rent in the amount of ONE HUNDRED TWENTY EIGHT THOUSAND TWO HUNDRED FIFTY AND NO/100 DOLLARS ($128,250.00) (the "**Base Rent**"), which such Base Rent shall be payable in twelve (12) equal consecutive monthly installments of TEN THOUSAND SIX HUNDRED EIGHTY SEVEN AND 50/100 DOLLARS ($10,687.50) in advance on the first (1st) day of each month to Lessor at 21553 East Apache Street, Catoosa, Oklahoma 74015. Notwithstanding the foregoing, the first installment of Base Rent shall be delivered to Lessor upon execution of this Lease.

Figure 4.1: Sample contract[2] indicating the terminologies used to refer to the elements of a contract. 'shall' triggers obligation for Lessee and entitlement for Lessor. Contracting party is referred to via an "alias" (such as Lessor or Lessee) throughout the contract.

[1] Abhilasha Sancheti, Aparna Garimella, Balaji Vasan Srinivasan, and Rachel Rudinger. 2022a. Agent-specific deontic modality detection in legal language. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[2] Party names redacted for anonymity purposes.

Hirsch, 2020) due to their length and the complexity of legalese used. Having an automated system that can extract such party relevant obligations, entitlements, prohibitions, and permissions, can be of great help not only to the parties but also to legal professionals for contract reviewing.

Consider example (1) where deontic modals such as, 'shall', 'shall not', and 'may' are explicitly used to express 'obligation/entitlement', 'prohibition', and 'permission', respectively.

(1) a. Tenant **shall** pay the rent to the Landlord.

b. Landlord **shall not** obtain financing or enter into any agreement affecting the Property.

c. Landlord **may** continue this Lease in effect after Tenant's abandonment and recover Rent as it becomes due.

(2) a. Tenant **agrees** to pay the rent.

b. Landlord **is responsible for** maintaining the structural soundness of the house.

Existing rule-based (Wyner and Peters, 2011; Peters and Wyner, 2016b; Dragoni et al., 2016; Ash et al., 2020) or data-driven approaches (Neill et al., 2017; Chalkidis et al., 2018b) cannot be used as they do not (in practice) capture the rich linguistic variety (*e.g.*, use of non-modal expressions in (2)) and ambiguity of modal expressions (*e.g.*, 'shall' in (1a)). Furthermore, annotated datasets used in the data-driven approaches do not consider multiple deontic types for a sentence and their association with the party (*e.g.*, (1a) is an instance of 'obligation' for the Tenant and an 'entitlement' for the Landlord).

In this chapter, we introduce two entity-centric tasks for legal document understanding: (i) party-specific multi-label deontic modality classification, and (ii) party-specific deontic modality and trigger span detection. To build automatic systems for these tasks, we present

a linguistically-informed taxonomy for annotating deontic types in the legal domain and use it to collect a corpus (LEXDEMOD[3]) of English contracts with two types of annotations: (i) all deontic types expressed in a sentence with respect to an party, and (ii) spans of modal *triggers*, *i.e.*, expressions (*e.g.*, bold-faced phrases in examples (1) and (2)) that evoke the modal meaning; . Transfer learning experiments show that the linguistic diversity of modal expressions in LEXDEMOD generalizes reasonably from lease to employment and rental agreements. A short case study indicates that a model trained on LEXDEMOD can detect red flags with high recall.

4.1 LEXDEMOD Dataset Curation

We first describe the dataset source (§4.1.1) followed by preprocessing (§4.1.2), annotation protocol (§4.1.3), and the quantitative and qualitative analysis (§4.1.4) of the collected dataset.

4.1.1 Dataset Source

We use the contracts available in the LEDGAR corpus (Tuggener et al., 2020) which comprises material contracts (Exhibit-10), such as agreements (*e.g.*, shareholder/ employment/ lease/ non-disclosure), crawled from Electronic Data Gathering, Analysis, and Retrieval (EDGAR) system. EDGAR is maintained by the U.S. Securities and Exchange Commission (SEC[4]). The documents filed on SEC are public information and can be redistributed without a further consent.[5]

[3]This dataset is released publicly for use to the research community at https://github.com/adobe-research/LexDeMod

[4]https://www.sec.gov/

[5]https://www.sec.gov/privacy.htm#dissemination

4.1.2 Contract Preprocessing

The raw contracts in LEDGAR are available in html format. We extract all the paragraphs (henceforth, "provisions") from the html (identified by `<p>` or `<div>` tags) of a contract, and heuristically filter the provisions defining any terminologies (identified by presence of phrases such as 'shall mean', 'means', 'shall have the meaning', 'has the meaning', etc.). As contracts have a hierarchical structure (*e.g.*, bullets and sub-bullets), we prepend the higher level context with the lower level (*e.g.*, combining sub-bullets with its context in the main bullet). After this, we heuristically extract the type of the contract (*e.g.*, lease or employment contract) and the alias (*e.g.*, "Lessee" in Figure 4.1) used to refer the contracting parties from the content of the contracts.

Contract Type Extraction. We heuristically scan the first 20 provisions[6] to identify the type of the contract using regular expressions (all uppercase characters and presence of 'AGREEMENT').

Party Alias Extraction. Party in a contract can be either a person or a company. Therefore, we scan the first 20 provisions of a contract to find company mentions using lexnlp (Bommarito II et al., 2021)[7] and named entities with 'person' tag using spaCy (Honnibal et al., 2020) library. We then use a regular expression (alias is mentioned in parenthesis (see Figure 4.1) following the party mention) to extract the alias used to refer to the found parties in the provisions. For each type of contract, we manually select the most frequently occurring aliases extracted after using the regular expression.

We collect all the sentences of provisions belonging to a contract wherein alias for an

[6]We found that the structure of contracts is not fixed and table of contents sometimes precedes the actual contract.

[7]lexnlp was better at extracting company names than spaCy as it is specifically trained for legal text processing.

party is found. We posit that if a sentence does not contain an alias, then deontic type is not expressed for an party. For instance, '*Any such month-to-month tenancy shall be subject to every other term, covenant and agreement contained herein.*' is a rule and does not specifically mention any deontic type for an party.

4.1.3 Annotation Protocol

Annotation task description. We propose party-specific deontic modality detection tasks that address the following issues: (i) non-robustness of rule-based extraction of rights and duties as it cannot capture the rich linguistic variety and ambiguity of modal expressions; (ii) lack of standard taxonomy for annotating fine-grained deontic types; (iii) non-association of deontic type with the party during annotation, and (iv) considering deontic type detection as a single class classification task. Consider, for instance, the following:

(3) a. [*Tenant*] Tenant **shall**$_{obl}$ pay the rent to the Landlord and **may**$_{per}$ use the parking space.

b. [*Landlord*] Tenant **shall**$_{ent}$ pay the rent to the Landlord and may use the parking space.

In these examples, the words in bold evoke the modal expression, which we call a *trigger*. For Tenant as the [*Party*], an obligation (obl) and a permission (per) are expressed in the sentence (3a), and an entitlement (ent) for the Landlord (3b).

Our data collection is performed via crowdsourcing on Amazon Mechanical Turk (AMT). We ask the annotators to provide two types of annotations for each sentence with respect to an party (referred to via an alias): (i) select all the deontic types expressed, and (ii) select trigger word(s) (as span) for each selected deontic type. If a sentence contains more than one party, we duplicate it to get separate annotations with respect to each party so that the annotators focus their understanding with respect to one party at a time. This task design

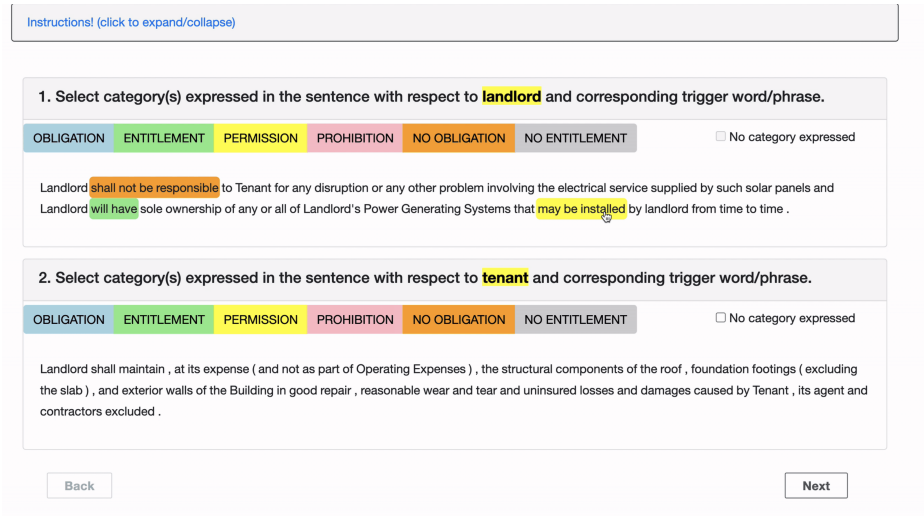


Figure 4.2: Annotation Interface.

choice helps in better estimation of the time taken to do each HIT (Human Intelligence Task) as the number of party mentions in a sentence can vary. This also simplifies the custom annotation interface (see Figure 4.2) built to get the annotations. We present the instructions, and provide the correctly and incorrectly annotated examples with explanations to the annotators as in Figure 4.6, Figure 4.7, and Figure 4.8.

Deontic Type	**Description**
Obligation (Obl)	Party is required to have or do something
Entitlement (Ent)	Party has the right to have or do something
Prohibition (Pro)	Party is forbidden to have or do something
Permission (Per)	Party is allowed to have or do something
No Obligation (Nobl)	Party is not required to have or do something
No Entitlement (Nent)	Party has no right to have or do something

Table 4.1: Taxonomy[8] for deontic type.

Taxonomy for deontic type. We base our taxonomy for deontic type annotation (Table 4.1) on the deontic logic theory of Von Wright (1951). Von Wright's categorical modals

[8]We also provide an additional option 'None' in case none of the deontic types is expressed or it is a rule.

are best suited for legal contracts which talk about rights and duties of contracting parties (Ballesteros et al., 2020; Matulewska, 2010) than other categorizations (Chung, 1985; Palmer, 2001; Jespersen, 2013) which are not found in contracts (*e.g.*, desiderative, hortative). We also include no-obligation and no-entitlement categories to cover all possible modalities which were found on manual inspection of contracts.

Annotation process and requirements. As legalese is syntactically complex and difficult to understand, the annotation task is quite intricate in nature. To ensure that the annotators properly understand the task, we first conduct a qualification test which explains the task with the help of right and wrong annotation examples along with explanations. The qualification test contains 10 questions, including 5 questions to test the understanding of identifying the correct deontic modal type and 5 to test their understanding of trigger span selection for a modal type. The test is open to annotators with $\geq 95\%$ approval rate and $\geq 1,000$ approved HITs. Finally, we select 25 annotators who answered all the qualification questions correctly.

The main annotation task consists of 12 sentences per HIT, including 2 quality check questions (from 50 manually annotated sentences) to ensure annotators are sincerely and correctly annotating each HIT. We publish 3 pilot HITs, with revised guidelines in each of them. We also manually check the annotations (selected randomly) to ensure quality and provide feedback to the annotators. We observe a learning curve for the task and considerable variation in the time taken per HIT (7.5 ± 1.5 mins). After the pilots, the annotations were majorly performed by 3 annotators. We publish a batch of 50 HITs with 3 annotations for each HIT from the 3 annotators. As we found good inter-annotator agreement between the 3 annotators (see §4.1.4), we collect only one annotation per HIT for the remaining HITs to get more sentences annotated within reasonable time.

Split	#Sent.	#Spans	Obl	Ent	Pro	Per	Nobl	Nent	None
Train	4282	5279	1841	1231	343	289	265	239	1071
Dev	330	421	176	86	20	18	21	22	78
Test	1777	1952	575	418	64	167	101	88	539

Table 4.2: Dataset Statistics.

Obl	Ent	Pro	Per	Nobl	Nent	None
0.82	0.68	0.44	0.82	0.76	0.41	0.65

Table 4.3: Deontic type-wise inter-annotator agreement (α) for the test set.

4.1.4 Annotated Dataset Statistics and Analysis

Each contract contains 202.6 (± 162.4) provisions on average (standard deviation in parentheses), with 2.2 (± 1.7) sentences per provision; each contract consists of 306.4 (± 235.8) sentences on an average. Among these, 75.8 $(\pm 14.4)\%$ of sentences per contract have atleast one party mentioned in them, with an average length of 65 (± 47).

We collect a total of $8,230$ trigger span annotations for $7,092$ sentences from 23 lease contracts after considering HITs for which both the quality questions are correctly answered. For duplicate sentences, in case of annotation disagreements, we retain those annotations that are inline with one of the authors (and discard 14.1% of duplicated ones). The disagreement could occur because of any missing modality in case multiple modalities expressed are in a sentence, incorrect interpretation of the sentence, or human error in terms of annotating with respect to a tenant or a landlord. For instance, the sentence: "*[landlord] After final approval of the Final Plans by applicable governmental authorities, no further changes may be made thereto without the prior written approval of both Landlord and Tenant.*", was annotated as 'prohibition' for landlord by one of the annotators and 'none' by another annotator. As the prohibition mentioned in the sentence is not for the landlord, the correct annotation is 'none'. Therefore, we retain the correct annotation for the example

and discard the sentence with the incorrect annotation. Another example is "*[landlord] All conditions and agreements under the Lease to be satisfied or performed by Landlord have been satisfied and performed.*" which was incorrectly annotated as an 'obligation'.

The test set comprises of sentences from 5 contracts including those for which we have 3 annotations per sentence, and rest of the sentences are divided into train and development sets such that the sentences from the same contract belong to the same set. We combine the 3 annotations for a subset of sentences in the test set using majority voting[9] for deontic type and by taking a union[10] of annotated trigger spans for the majority deontic types. The average inter-annotator agreement for each deontic type computed with Krippendorff's α (Krippendorff, 2018) is substantial ($\alpha = 0.65$) given the complexity of the task (see Table 4.3 for type-wise agreement). For trigger span annotation, the token-level inter-annotator agreement for the majority deontic types for a sentence is also substantial ($\alpha = 0.71$). The fine-grained dataset statistics after filtering and resolving disagreements are presented in Table 4.2.

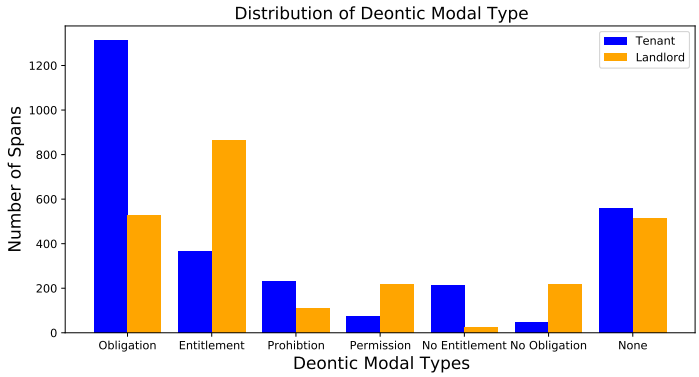


Figure 4.3: Distribution of deontic type with respect to Tenant and Landlord for lease agreements.

[9]We discard the sentence in case no majority is found.

[10]Union was performed in 15.54% of sentences and we manually corrected 9.4% of these sentences where union lead to incorrect triggers.

Type	Top 10 triggers
Obl	shall, will, agrees, agree, acknowledges, acknowledge, represents and warrants, shall be responsible for, undertakes, will be responsible for
Ent	shall, will, agrees, shall have the right to, shall be entitled to, represents and warrants, acknowledges, waives no rights, shall not, retains all other rights, will be entitled to
Pro	shall not, will not, may not, nor shall, not to be, neither lessor nor lessee may, in no event shall, nor will, will not allow, nor may
Per	may, is permitted, will allow, has the right, shall, or at landlord's option, shall be permitted to, shall be allowed
Nobl	shall not be liable, shall not be obligated to, shall not be required to, shall, shall have no obligation to, in no event shall landlord be obligated to, waives, shall not, shall have no liability
Nent	shall, shall have no right to, waives no rights, shall not, shall have no obligation to, waives, shall not be required, shall not be obligated, waive the right, shall not have the right to

Table 4.4: Top 10 triggers for each deontic type in decreasing order of frequency.

Qualitative analysis. Figure 4.3 shows the distribution of annotated spans[11] over deontic types with respect to each party (Landlord and Tenant for lease agreements[12]) in the train set. Interestingly, tenants have more obligations and prohibitions, and fewer entitlements or permissions than landlords. 17.3% of the sentences have multiple trigger annotations, 48.6% of these sentences express multiple deontic types. 24.8% of the sentences do not express any deontic type with respect to a given party. The dataset contains 383 unique triggers across all the deontic types. Table 4.4 lists the top 10 triggers for each deontic type, and Figure 4.4 shows the frequency-based wordcloud of the annotated triggers. 'Shall' constitutes 44.6% of the annotated triggers used to express not only obligation but entitlement, no-obligation, and no-entitlement as well. Prohibitions may be expressed using negation words (14.9%) between the context (*e.g.*, 'neither lessor nor lessee may') of a sentence. While modal auxiliaries (*e.g.*, *shall*, *will*, *may*) are more frequently used, 45.2% of the total unique triggers are non-modal expressions (*e.g.*, *agrees*, *represents*) covering 20.3% of the anno-

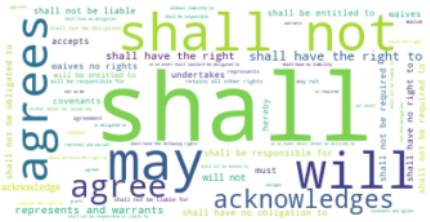


Figure 4.4: Frequency-based wordcloud of all the triggers.

[11]For 'None' type, no span is annotated so the bar denotes the number of sentences labeled as none.

[12]We add the statistics for tenant, subtenant & lessee under Tenant, and landlord, sublandlord & lessor under Landlord.

Type	Examples
Obl	[tenant] Tenant **shall** repair any damage resulting from such removal and **shall** restore the Property to good order and condition. [tenant] Tenant **acknowledges** and **agrees** that Landlord shall have the right to adopt reasonable rules and regulations for the use and/or occupancy of the Leased Premises and Tenant **agrees** that it shall at all times observe and comply with such rules and regulations.
Ent	[tenant] Tenant **shall also have the right to** use the roof riser space of the Building. [lessor] Rent **shall** be payable at Lessor 's place of business , or such other place as Lessor may direct from time to time. [landlord] Landlord **reserves the right to** modify Common Areas, provided that such modifications do not materially adversely affect Tenant's access to or use of the Premises for the Permitted Use.
Pro	[lessee] Lessee **shall not** commit or allow waste to be committed on the Premises, and Lessee **shall not** allow any hazardous activity to be engaged in upon the Premises. [lessor] **Neither Lessor nor Lessee may** record this Lease nor a short-form memorandum thereof.
Per	[tenant] Tenant **may**, without Landlord's consent, before delinquency occurs, contest any such taxes related to the Personal Property. [lessor] Additional keys **may** be furnished at a charge by Lessor.
Nobl	[tenant] For the avoidance of doubt, to the extent there is a bank vault in the Premises, Tenant **shall have no obligation to** remove such vault on surrendering the Premises. [lessor] Further, **in no event shall Lessor have any obligation to** repair any damage to, or replace any of Lessee's furniture, trade fixtures, equipment or other personal property.
Nent	[landlord] Landlord hereby **waives the right to** any revenue that may be generated as a result of the use of the roof by Tenant or any other third - parties pursuant to the terms of the Lease during the Term. [lessee] The Lessee **will not be entitled to** a reimbursement of any part of the Rent, even if in practice the Building Capacity for which it has paid has not been utilized.
None	[lessor] For the avoidance of doubt, it is hereby clarified that wherever the word Lessor is written this means: "the Lessor and/or anyone acting on its behalf". [landlord] Other than the Purchase Agreement, this Lease represents the entire agreement and understanding between Landlord and Tenant with respect to the subject matter herein, and there are no representations, understandings, stipulations, agreements or promises not incorporated in writing herein.

Table 4.5: Sample annotated sentences for each deontic type with respect to an [Agent] and trigger annotations in **bold-face**.

tated trigger spans. This shows that LEXDEMOD covers a wide variety of linguistic expressions of deontic modality in legalese, not restricted to modal auxiliaries. Annotated samples from the dataset are provided in Table 4.5.

4.2 Proposed Benchmarking Tasks

Having established the rich variety and coverage of linguistic expressions for deontic modality in LEXDEMOD, we benchmark the corpus on the proposed two tasks defined as: **(i) Party-specific multi-label deontic modality classification.** This task aims at predicting all the deontic types expressed in a sentence with respect to an party. We pose this as a

multi-label classification task conditioned on a sentence and an party.

(ii) Party-specific deontic modality and trigger span detection. This task aims at identifying both the deontic type and corresponding triggers. We pose this as a token classification task. Every token in the corpus is assigned a BIOS tag if it belongs to a modal trigger, which is appended with a suffix indicating its deontic type. For instance, Tenant$_O$ *is*$_{B-OBL}$ *responsible*$_{I-OBL}$ *for*$_{I-OBL}$ paying$_O$ the$_O$ rent$_O$, where subscripts denote the BIOS tags.

For both the tasks, party is conditioned using special tokens added at the beginning of a sentence. This simple strategy has been successfully used previously for controlled text generation tasks (Sennrich et al., 2016; Johnson et al., 2017; Rudinger et al., 2020; Sancheti et al., 2022b).

4.3 Benchmarking Setup

We experiment with various pretrained language models (pretrained language models) (Devlin et al., 2019b; Liu et al., 2019), which have shown state-of-the-art performance on natural language understanding tasks, to study their performance on our proposed tasks. We fine-tune these models for both the tasks on binary cross-entropy loss for 20 epochs each with a batch size of 8, and maximum sequence length of 256 using HuggingFace's Transformers library (Wolf et al., 2020). We run each model on 3 seed values, use Adam optimizer with a linear scheduler for learning rate with an initial learning rate of $2e^{-5}$, and warm-up ratio set at 0.05. All the models are trained and tested on NVIDIA Tesla V100 SXM2 16GB GPU machine. We experiment with batch size $\in \{2, 4, 8\}$, number of epochs $\in \{3, 5, 10, 20, 30\}$, learning rate $\in \{1e^{-5}, 2e^{-5}, 3e^{-5}, 5e^{-5}\}$, and warm-up ratio $\in \{0.05, 0.10\}$. BERT-base (110M parameters) and Roberta-base (125M parameters) models took $\sim$ 46 mins, and RoBERTa-large (355M parameters) took $\sim$ 2 hrs to train for each of the tasks. The model(s) with the best macro-F1 score on the dev set is used to report

Type	Heuristic triggers
Obl	shall/will be required, shall be obligated, shall, must, will, have to, should, ought to have, will/shall be paid
Ent	shall/will be entitled, shall/will be paid, shall/will retain, shall/will receive, shall have the right to, shall be retained, shall be kept, shall be claimed, shall be accessible, shall be owned, shall be determined
Pro	shall/will/must/may not, cannot, shall have no right, can not, shall/will not be allowed, shall not assist, shall/will be prohibited
Per	shall be permitted, shall also be permitted, can, may, could, shall/will be allowed
Nobl	shall/will not be liable for, shall/will not be obligated to, shall/will not be obligated for, shall/will not be responsible for, shall/will not be required to
Nent	shall/will not entitled to, shall/will not have the right to, shall/will not be entitled for

Table 4.6: Heuristic triggers used by rule-based approach to identify the deontic types.

results on the test set. We also partition the data according to the party being conditioned to assess the performance of the trained models with respect to each party.

4.4 Benchmarking Multi-label Classification

Comparison models. We experiment with three kinds of approaches for the party-specific multi-label deontic modality classification task.

(1) Majority class predicted for each party.

(2) Rule-based. We implement a rule-based approach similar to the one described in Ash et al. (2020) with additional conditioning on the party. It searches for the presence of pre-defined modal triggers (Table 4.6) for a deontic type and associates it with the party using dependency tags (*e.g.*, *nsubj*, *aux* or *party*). We use spacy to tokenize each sentence and obtain a dependency parse.

(3) Fine-tuning pretrained language models. We fine-tune several pretrained language models varying in size and domain of pretraining data: (i) BERT-base-uncased (**BERT-BU**); (ii) RoBERTa-base (**RoBERTa-B**); (iii) RoBERTa-large (**RoBERTa-L**), and (iv) recently introduced Contract-BERT-base-uncased (**C-BERT-BU**) model (Chalkidis et al., 2020) which has been pretrained on US contracts from the EDGAR library.

All the above models are trained assuming trigger span information is not available and full context (*i.e.*, sentence) is used. To understand the importance of **A**gent conditioning,

Model	Accuracy	Precision	Recall	F1
Majority	39.53/28.66/34.38	6.49/5.23/11.72	14.29/14.29/21.09	8.92/7.66/15.03
Rule-based	61.32/50.54/56.22	**81.81**/75.21/**80.04**	46.66/45.54/46.64	50.13/46.16/48.76
BERT-BU	74.07/70.79/72.52	73.68/74.44/75.48	75.84/71.02/77.17	**78.81**/71.18/75.61
RoBERTa-B	75.53/71.42/73.59	73.54/72.17/74.48	78.39/72.88/78.31	74.90/71.91/75.66
C-BERT-BU	77.50/73.25/75.48	76.63/76.22/77.52	**80.47**/71.54/78.81	77.95/72.34/77.67
RoBERTa-L	**78.28**/**75.03**/**76.74**	75.05/**77.69**/77.30	79.59/**75.21**/**79.11**	76.71/**76.00**/**77.88**
RoBERTa-L-No-party	51.28/47.45/49.46	57.09/53.79/58.32	65.01/55.52/60.08	55.75/52.56/57.53
RoBERTa-L-ACT-Masked	81.52/72.02/77.02	76.39/71.31/81.46	71.22/65.74/75.90	84.25/76.29/80.42
RoBERTa-L-AT	84.72/76.22/80.70	79.84/73.60/82.29	78.00/72.96/82.47	87.58/80.58/84.20
RoBERTa-L-ACT	**91.66**/**88.62**/**90.23**	**88.44**/**85.40**/**89.48**	**88.10**/**84.43**/**89.21**	**93.38**/**91.38**/**92.42**

Table 4.7: Evaluation results for party-specific multi-label deontic modality classification task. Scores for **Tenant/Landlord/Both** are averaged over 3 different seeds. BU, B, L, A, C, and T denote base-uncased, base, large, party, context (sentence), and trigger, resp.

Deontic Type	Classification			Span Detection		
	Precision	Recall	F1	Precision	Recall	F1
Obl	92.93	85.98	89.32	80.67	76.19	78.37
Ent	77.78	86.15	81.75	70.73	78.38	74.36
Pro	75.00	54.55	63.16	41.67	38.46	40.00
Per	100.00	92.31	96.00	100.00	88.46	93.88
Nobl	100.00	85.00	91.89	29.17	29.17	29.17
Nent	58.33	63.64	60.87	40.00	33.34	36.36
None	79.41	87.80	83.40	–	–	–

Table 4.8: Deontic type-wise results for multi-label classification and modal trigger span detection (labeled) tasks on the test set from the best (out of 3 seeds) RoBERTa-L model.

Context, and **T**rigger for this task, we additionally train the following models: (i) **No-party** where special token for party is not used during training; (ii) **ACT-Masked** wherein everything in the context except the trigger span is masked using `[MASK]` token to hide the context but retain the positional information of the trigger; (iii) **AT** wherein only the tokens belonging to a trigger are used and multiple triggers are separated using a special token (*e.g.*, `[SEP]` or `</s>`), and (iv) **ACT** wherein all the triggers are appended at the end of the context separated by a special token (*e.g.*, `[SEP]` or `</s>`).

Evaluation measures. We report macro-averaged Precision, Recall, and F1 scores across all the types, calculated using Sklearn library (Pedregosa et al., 2011). We also report the Accuracy of predicting all the classes correctly for a sentence.

Results and analysis. We report the results for multi-label classification task in Table 4.7. While rule-based approach has better F1 score than majority type prediction for each party, Transformer-based models outperform these baselines indicating their ability to better capture the linguistic diversity of expressing deontic modals. As expected, Rule-based approach has the highest overall precision but low recall due to the impossibility of enumerating all the rules. While C-BERT-BU, which is pretrained on contracts, performs better than BERT-BU and RoBERTa-B, interestingly it achieves comparable F1 score to RoBERTa-L. This indicates that improvements from domain-specific pretraining may also be achieved with larger model size and more training data.

As RoBERTa-L performs the best on this task, we report the results for variants of this model to understand the importance of party conditioning, context, and trigger, in the last block of Table 4.7. The performance of RoBERTa-L-No-party, trained without party conditioning, significantly drops as compared to RoBERTa-L, indicating the importance of party conditioning during training and association of party with the modality expressed in a sentence. Using trigger information during training (RoBERTa-L-ACT) significantly improves the performance over RoBERTa-L across all the measures, showing that triggers are indicative of specific deontic type. Higher scores for RoBERTa-L-AT than RoBERTa-L-ACT-Masked show that positional information of trigger span adds noise to the representations learned by the model. Further, context is also important for identifying deontic type, as all the metric scores drop when context is masked (RoBERTa-L-ACT-Masked) or not used (RoBERTa-L-AT) during training as compared to using all the information (RoBERTa-L-ACT).

Manual inspection of deontic type-wise (Table 4.8) performance reveals that permission is the easiest, while no-entitlement and prohibition are the hardest to identify. This can be due to the use of limited variety in expressing permissions (majorly ‘may’), while use of negation within context for expressing prohibitions which makes it harder to identify. For

tenant, obligation is identified more accurately than entitlements (vice-versa for landlord) as expected due to higher frequency of obligations for tenant and entitlements for landlord.

Figure 4.5a shows the trend for RoBERTA-L's F1 score as train data size varies indicating that the rate of increase in F1 decreases with additional data.

4.5 Benchmarking Trigger Span Detection

Comparison models. We experiment with three kinds of approaches for the party-specific deontic modality and trigger span detection task.

(1) Majority. 'Shall' is the most used trigger as shown in Figure 4.4 and is used to express obligations for Tenant while entitlements for Landlord. This baseline tags each occurrence of 'shall' with S-OBL for tenant or S-ENT for landlord as party.

(2) Rule-based. We tag occurrences of predefined modal triggers in a sentence with the deontic type predicted using the rule-based approach (§4.4).

(3) Fine-tuning pretrained language models. We fine-tune the same models as described in §4.4 on a token classification task to predict the BIOS tags. Additionally, we train a 'No-party' model to verify the importance of party conditioning.

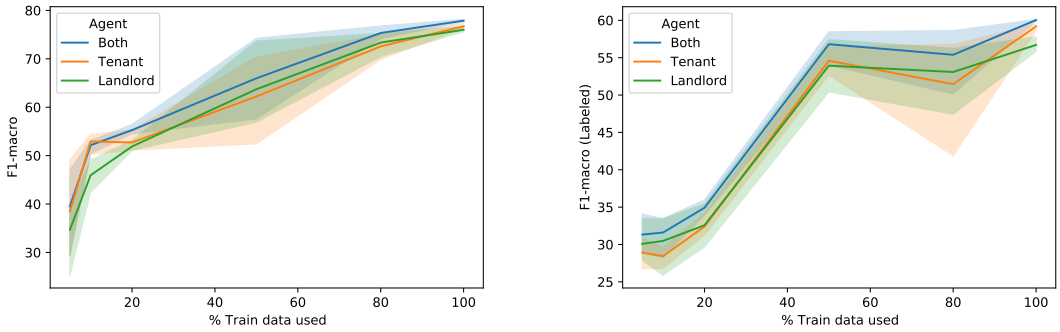
Evaluation measures. We report macro-averaged Precision, Recall, and F1 scores, calculated using seqeval library (Nakayama, 2018). We also report the Accuracy of predicting the BIOS tags for a sentence. Following (Pyatkin et al., 2021), we report these metrics in labeled (both deontic type and trigger span considered) and unlabeled (only trigger span without deontic type is considered) settings.

Results and Analysis. Labeled and unlabeled metric scores for trigger span detection task are reported in Table 4.9. RoBERTa-L has the best labeled F1 score which evaluates for both trigger detection and its correct deontic type identification. Similar to the classification

Labeled				
Model	**Accuracy**	**Precision**	**Recall**	**F1**
Majority	97.16/96.85/97.01	5.58/4.40/9.98	10.89/10.31/16.00	7.38/6.17/12.27
Rule-based	97.85/97.67/97.76	**77.42**/**76.68**/**79.66**	32.42/33.24/33.61	40.00/39.30/40.58
BERT-BU	98.45/98.36/98.41	53.04/56.11/56.48	58.49/59.05/61.97	55.11/**57.01**/58.80
RoBERTa-B	98.40/98.24/98.32	53.03/52.43/55.57	63.65/**59.63**/**64.00**	57.08/55.31/53.91
C-BERT-BU	98.44/**98.39**/**98.42**	53.46/54.70/57.08	60.76/57.37/62.42	56.45/55.68/59.31
RoBERTa-L	**98.45**/98.27/98.37	54.99/55.58/57.37	**65.55**/58.88/63.74	**59.19**/56.71/**60.04**
RoBERTa-L-NA	97.64/97.75/97.69	32.45/36.71/36.36	48.92/43.93/46.79	34.68/38.72/39.45
Unlabeled				
Model	**Accuracy**	**Precision**	**Recall**	**F1**
Majority	97.28/97.04/97.17	41.30/39.59/40.51	50.61/42.86/46.76	45.48/41.16/43.41
Rule-based	97.89/97.73/97.81	**72.59**/**73.97**/**73.22**	40.07/35.34/37.73	51.64/47.83/49.80
BERT-BU	**98.55**/**98.52**/**98.53**	68.87/69.92/69.38	76.07/75.22/75.65	72.29/72.46/72.38
RoBERTa-B	98.49/98.41/98.46	68.99/67.44/68.22	78.18/**75.36**/**76.78**	73.27/71.14/72.22
C-BERT-BU	98.52/**98.52**/98.52	69.49/70.85/70.15	76.89/74.26/75.59	72.99/**72.52**/72.76
RoBERTa-L	98.54/98.39/98.47	69.78/69.56/69.68	**79.16**/74.35/**76.78**	**74.18**/71.88/**73.06**
RoBERTa-L-NA	98.26/98.22/98.24	61.73/64.63/63.12	75.21/72.84/74.03	67.79/68.46/68.11

Table 4.9: Evaluation results for party-specific modal trigger span detection task. Macro-averaged Precision, Recall and F1 scores are presented for **Tenant/Landlord/Both**. Scores are averaged over 3 different seeds. BU, B, L, and NA denote base-uncased, base, large, and no-party respectively.

task, Rule-based approach outperforms other models on precision however, lags behind in recall for the same reason. Size of the model (RoBERTa-L) is instrumental than domain knowledge of C-BERT-BU. Consistently higher unlabeled scores, indicate that models are able to identify the trigger words. However, associating triggers with the correct deontic type is a harder task, owing to the multiple deontic types that a trigger can be used to express (*e.g.*, shall in Table 4.4) them. Similar to the classification task, importance of party conditioning is evident from the last row with significant drop in F1 scores (even lower than Rule-based approach in Labeled score). Higher accuracy scores are due to the majority tokens being labeled as 'O'. Trends with dataset size variation are shown in Figure 4.5b. Manual analysis of deontic type-wise span detection (Table 4.8) reveals that prohibition, no-entitlement, and no-obligation are hard to identify. Similar trends were observed for tenant and landlord as in §4.4.



(a) Classification (b) Trigger Span Detection

Figure 4.5: RoBERTA-L's performance with varying train dataset size for the two tasks.

Model	Accuracy	Precision	Recall	F1
Multi-label Classification (Rental/Employment)				
Majority	36.36/27.45	11.87/8.80	19.10/15.15	14.46/11.11
Rule-based	41.56/47.45	53.77/64.63	34.54/35.00	33.27/37.22
RoBERTa-L	73.16/48.72	83.08/52.87	63.42/48.90	68.90/48.32
RoBERTa-L-AR	55.19/42.55	56.87/59.29	52.38/46.48	50.66/50.30
RoBERTa-L-ARR	70.35/64.68	76.79/70.05	63.14/64.62	65.89/65.36
Trigger Span Detection (Labeled) (Rental/Employment)				
Majority	96.09/97.37	18.33/4.23	1.90/7.08	3.42/5.30
Rule-based	96.40/97.83	56.25/59.66	23.69/19.65	29.62/27.45
RoBERTa-L	97.48/97.78	49.74/36.80	45.87/37.84	45.58/34.87
RoBERTa-L-AR	97.22/98.15	49.97/48.86	44.43/42.99	44.22/43.42
RoBERTa-L-ARR	97.60/98.38	59.42/53.14	47.83/43.84	49.61/45.47
Trigger Span Detection (Unlabeled) (Rental/Employment)				
Majority	96.09/97.53	54.55/41.26	4.55/34.16	8.39/37.38
Rule-based	96.40/97.85	84.62/73.28	16.67/21.72	27.85/33.51
RoBERTa-L	97.78/98.09	69.85/54.32	68.43/60.76	69.13/57.36
RoBERTa-L-AR	97.81/98.31	68.77/59.64	69.44/57.05	69.09/58.30
RoBERTa-L-ARR	97.78/98.46	69.51/61.81	70.45/57.50	69.94/59.58

Table 4.10: Evaluation results for models with party mention anonymization on rental/employment contracts.

These results show that identification of both triggers and associating it with the deontic type is a difficult task owing to the linguistic variety of expressions used in legal language.

4.6 Beyond Lease Contracts

To investigate if the diverse linguistic expressions used for expressing deontic modality is specific to a contract type, we collect annotations via AMT using the same annotation protocol (§4.1.3) for: (1) 470 sentences from 3 employment contracts in the LEDGAR corpus, and (2) 154 sentences from 4 rental agreement templates freely available at PandaDoc.[13] We evaluate the performance of the best model (RoBERTa-L) for both the tasks on these sentences and report the results in Table 4.10. We observe a performance drop (more for employment contracts than rental agreements) when compared to model's performance on lease agreements, although it is significantly better than the rule-based approach demonstrating the non-robustness of rule-based approach towards diverse linguistic expressions. This drop is more prominent for employment contracts due to the lease-specific party conditioning (*e.g.*, tenant) used during training while commonly occurring parties in employment contracts are employee, employer, etc.

To account for this, we additionally train models with anonymizing the party mentions in the dataset in two ways: (i) **RoBERTa-L-AR**– all occurrences of an party are replaced with the same token (*e.g.*, 'a1' for tenant), and (ii) **RoBERTa-L-ARR** –party is randomly replaced with a token consistent within a sentence. Replacing party mentions leads to significant improvements for employment contracts in both the tasks, although evaluating these models on rental agreements (see Table 4.10) and lease data shows (see Table 4.11) an expected drop in the performance. These experiments show that the linguistic expressions captured by LEXDEMOD are also generalizable to other types of contracts.

[13] https://www.pandadoc.com/ We chose rental agreements as they are commonly used by layperson than lease contracts from SEC.

Model	Accuracy	Precision	Recall	F1
Multi-label Classification				
RoBERTa-L	79.37	81.00	77.16	78.52
RoBERTa-L-AR	50.73	56.51	53.96	53.65
RoBERTa-L-ARR	80.06	79.70	76.98	77.97
Trigger Span Detection (Labeled)				
RoBERTa-L	98.51	58.65	56.46	57.29
RoBERTa-L-AR	98.29	51.11	54.01	52.04
RoBERTa-L-ARR	98.48	57.93	57.21	57.33
Trigger Span Detection (Unlabeled)				
RoBERTa-L	98.56	74.46	72.97	73.69
RoBERTa-L-AR	98.41	69.95	70.91	70.37
RoBERTa-L-ARR	98.55	74.63	72.36	73.48

Table 4.11: Evaluation results of RoBERTa-L-AR and RoBERTa-L-ARR on lease test set.

Model	Precision	Recall	F1
ALeaseBERT	82.35	8.09	14.74
Ours	8.53	87.28	15.54

Table 4.12: Results from the red flag detection case study. Our (ALeaseBERT) denotes RoBERTa-L model trained on LEXDEMOD (Red flags dataset (Leivaditi et al., 2020)).

4.7 Case Study: Red flag Detection

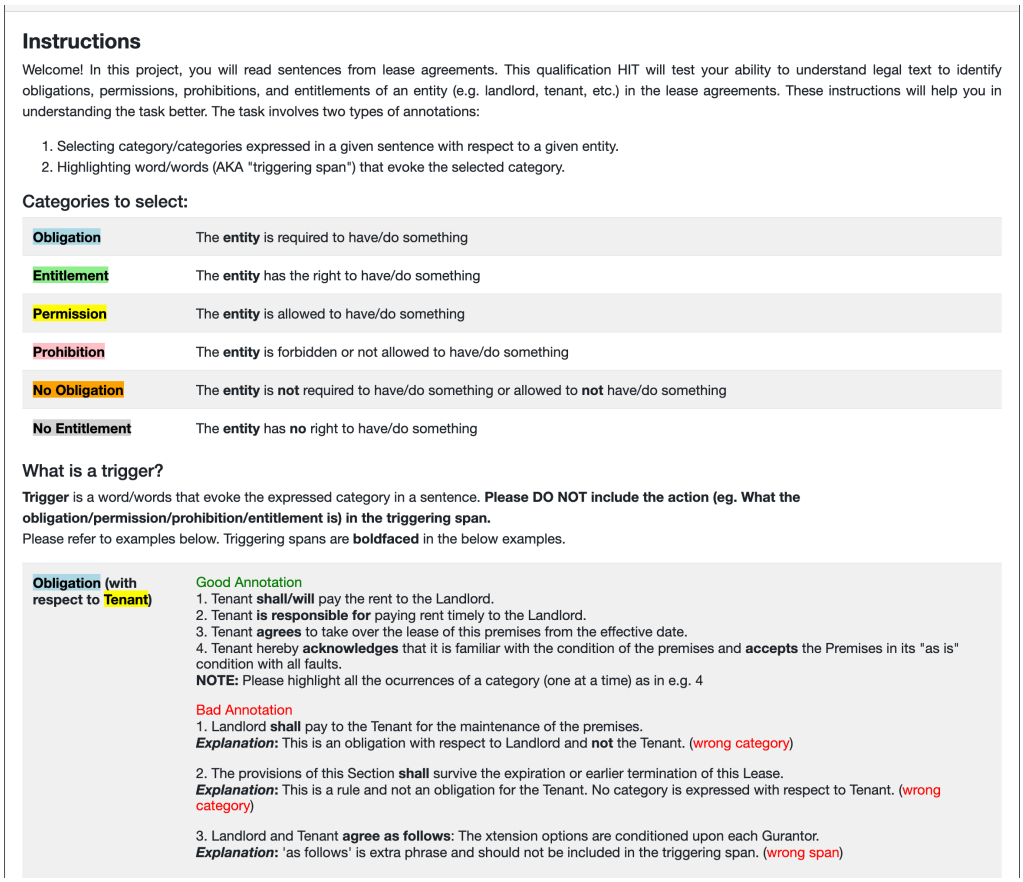
To investigate if our party-specific deontic modality classifier is capable of identifying the red flags annotated by Leivaditi et al. (2020) for lease agreements, we compare the predictions on the dev set from ALeaseBERT, proposed by Leivaditi et al. (2020), and RoBERTa-L model trained on LEXDEMOD dataset. For each sentence in the red flags dataset, we predict the deontic modality with respect to each of the party alias mentioned in that sentence. If any one of the deontic types is expressed for any of the parties then we consider the prediction as positive otherwise negative. We find that (see Table 4.12)

the model trained on LEXDEMOD has high recall and low precision while ALeaseBERT has high precision but low recall for the positive class. Our model was able to predict all the red flags predicted by ALeaseBERT and some additional red flags. This is expected as many permissions or entitlements may not be red flags but may belong to a deontic type. We also found that there were payments related obligations which were predicted as red flags by our model but were not annotated as red flags in the dataset. Therefore, our model could also be used to filter important sentences which could indicate some red flags due to high recall.

4.8 Summary

In this chapter, we introduce LEXDEMOD for deontic modality detection in the legal domain which consists of diverse linguistic expressions of deontic modality. We propose and benchmark two entity-centric tasks in the legal domain namely, party-specific multi-label deontic modality classification, and party-specific deontic modality and trigger span detection using transformer-based models. While evaluation results are promising, there is substantial room for improvement. Additionally, we demonstrate the generalizability of the diverse linguistic expressions captured in LEXDEMOD via transfer learning experiments to employment and rental lease agreements. The short case study on red flag detection using a model trained on LexDeMod datasets presents one use-case of how party-specific deontic modality detection systems could be used in the real world.

While, in this chapter, we focus on extracting information relevant to each contracting party, not each information is equally important. To formalize this aspect of importance ordering among information relevant to each contracting party and contract understanding, we introduce the task of party-specific summarization of *important* obligations, entitlements, and prohibitions in the next chapter.



Instructions

Welcome! In this project, you will read sentences from lease agreements. This qualification HIT will test your ability to understand legal text to identify obligations, permissions, prohibitions, and entitlements of an entity (e.g. landlord, tenant, etc.) in the lease agreements. These instructions will help you in understanding the task better. The task involves two types of annotations:

1. Selecting category/categories expressed in a given sentence with respect to a given entity.
2. Highlighting word/words (AKA "triggering span") that evoke the selected category.

Categories to select:

Obligation	The **entity** is required to have/do something
Entitlement	The **entity** has the right to have/do something
Permission	The **entity** is allowed to have/do something
Prohibition	The **entity** is forbidden or not allowed to have/do something
No Obligation	The **entity** is **not** required to have/do something or allowed to **not** have/do something
No Entitlement	The **entity** has **no** right to have/do something

What is a trigger?

Trigger is a word/words that evoke the expressed category in a sentence. **Please DO NOT include the action (eg. What the obligation/permission/prohibition/entitlement is) in the triggering span.**
Please refer to examples below. Triggering spans are **boldfaced** in the below examples.

Obligation (with respect to Tenant)

Good Annotation
1. Tenant **shall/will** pay the rent to the Landlord.
2. Tenant **is responsible for** paying rent timely to the Landlord.
3. Tenant **agrees** to take over the lease of this premises from the effective date.
4. Tenant hereby **acknowledges** that it is familiar with the condition of the premises and **accepts** the Premises in its "as is" condition with all faults.
NOTE: Please highlight all the ocurrences of a category (one at a time) as in e.g. 4

Bad Annotation
1. Landlord **shall** pay to the Tenant for the maintenance of the premises.
***Explanation*:** This is an obligation with respect to Landlord and **not** the Tenant. (wrong category)

2. The provisions of this Section **shall** survive the expiration or earlier termination of this Lease.
***Explanation*:** This is a rule and not an obligation for the Tenant. No category is expressed with respect to Tenant. (wrong category)

3. Landlord and Tenant **agree as follows**: The xtension options are conditioned upon each Gurantor.
***Explanation*:** 'as follows' is extra phrase and should not be included in the triggering span. (wrong span)

Figure 4.6: Part 1: Instructions and examples provided to the annotators.

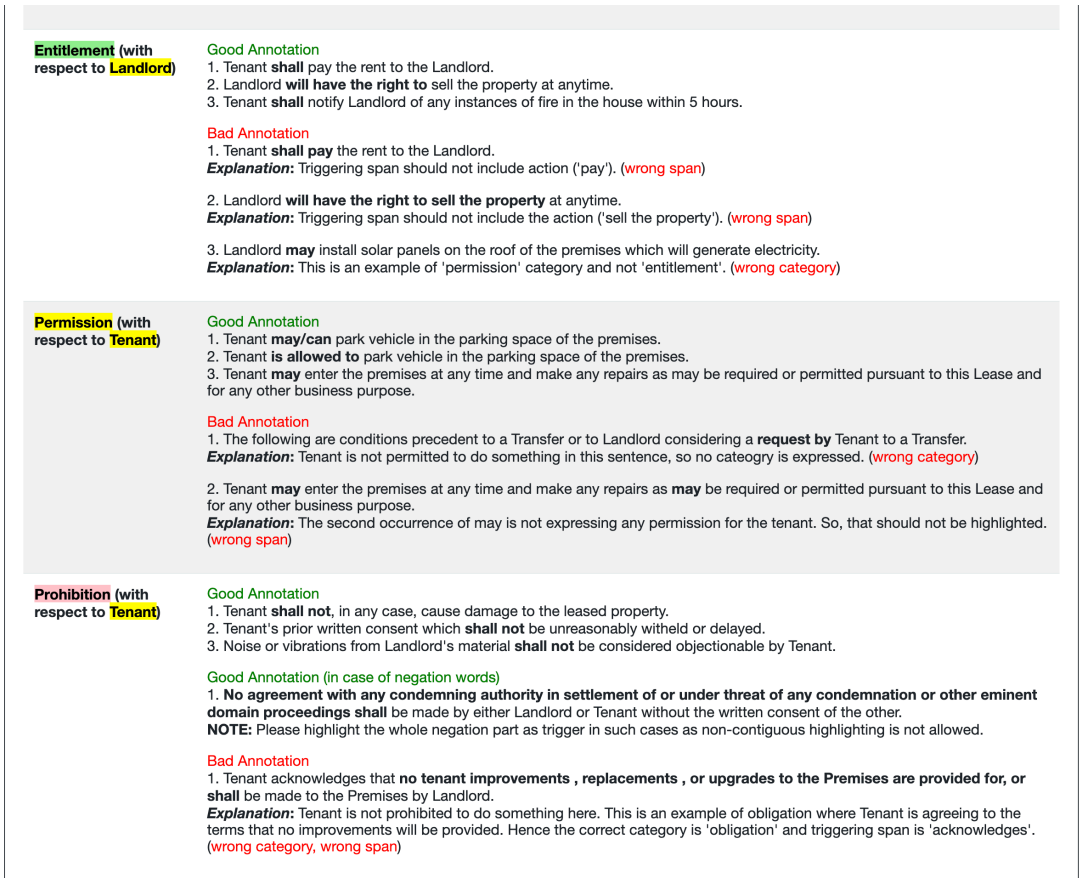


Figure 4.7: Part 2: Instructions and examples provided to the annotators.

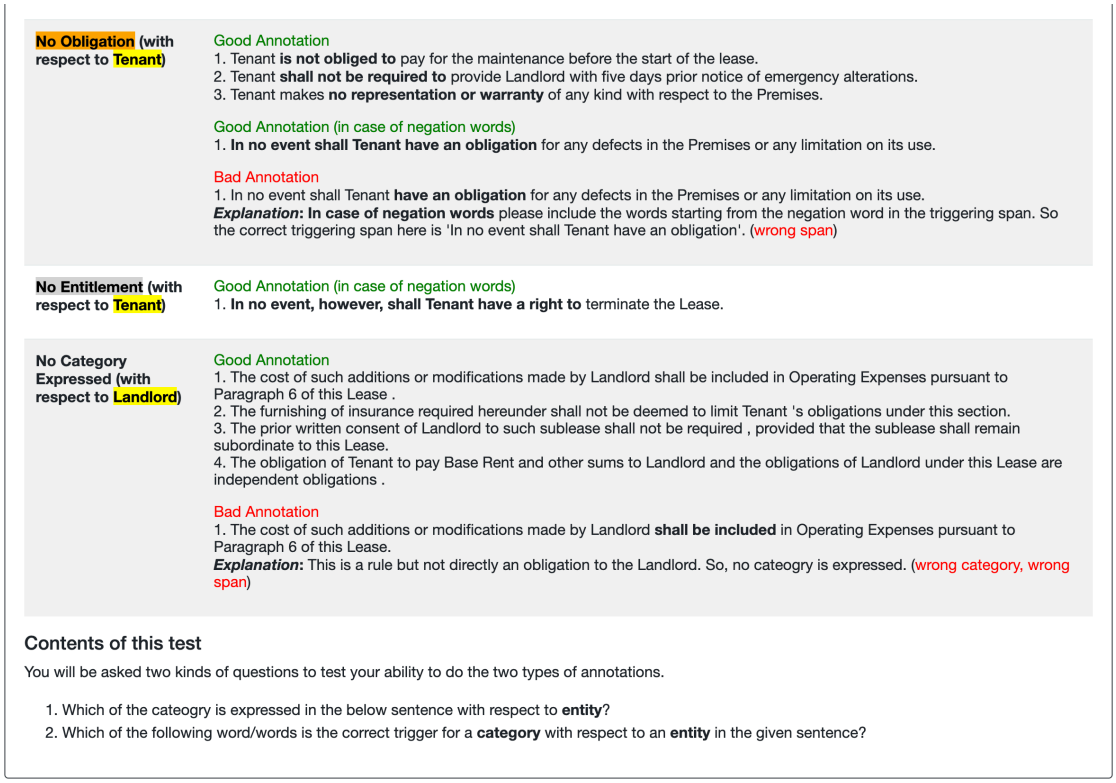


Figure 4.8: Part 3: Instructions and examples provided to the annotators.

Chapter 5: Party-specific Summarization of Legal Documents[1]

In the previous chapter, we formalize the task of identifying party-specific obligations, entitlements, prohibitions, and permissions in the legal domain, introduce a dataset, and model for the task. Although such identification solves one part of the problem (*i.e*, information overload in lengthy legal documents), parties might still have a lot of information to read through. One solution is to generate a summary of the document. However, in this chapter, we argue that a single summary may not serve all the parties as they may have different rights and duties (*e.g.,* a *tenant* is obligated to timely pay the rent to the landlord whereas, the *landlord* must ensure the safety and maintenance of rented property). Further, each right or duty may not be equally important (*e.g.*, an obligation to pay rent on time (otherwise pay a penalty) is more important for tenant than maintaining a clean and organized premise) and this importance also varies for each party.

To address this, we introduce a new task of *party-specific* extractive summarization of *important* obligations, entitlements, and prohibitions in a contract. Motivated by the different categories in the software license summaries available at tl;drLegal[2] (see Figure 5.1) describing what users *must*, *can*, and *cannot* do under a license, we define rights and duties in terms of deontic modalities (Chapter 4): *obligations* ("must"), *entitlements* ("can"), and *prohibitions* ("cannot"). In this chapter, we refer to these as (modal) categories.

[1] Abhilasha Sancheti, Aparna Garimella, Balaji Vasan Srinivasan, and Rachel Rudinger. 2023. What to read in a contract? Party-specific summarization of legal obligations, entitlements, and prohibitions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore. Association for Computational Linguistics.

[2] https://tldrlegal.com/

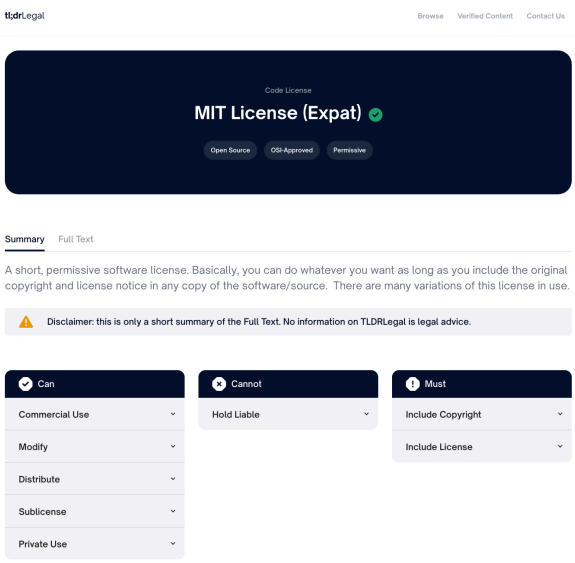


Figure 5.1: Sample summary for MIT license available on tl;drLegal website.

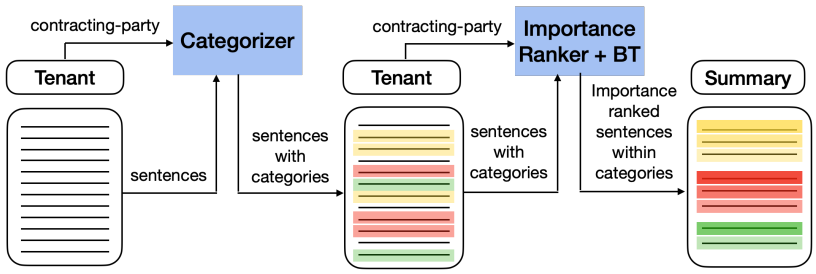


Figure 5.2: An illustration of input contract with a party (Tenant) and output from CONTRASUM. It first identifies sentences in a contract that contains any of the obligations, entitlements, and prohibitions specific to the party. Then, ranks the identified sentences based on their importance level (darker the shade, higher the importance) to produce a summary. BT: Bradley-Terry.

Building an end-to-end supervised system for this task is challenging due to: (1) the limited context length of existing transformer-based (Vaswani et al., 2017) models that are ubiquitously used for NLU tasks, and (2) the lack of annotated datasets that capture legal-domain-specific notion of *importance* for summarization. To address (1), we propose to break down the contract-level task of *party-specific* summarization into two sentence-level sub-tasks: (a) **Content Categorization** – identifying the modal categories expressed in

the sentences of a contract for a specified party, and (b) **Importance Ranking** – ranking the sentences based on their importance for a specified party. We construct a legal-expert annotated dataset of *party-specific* pairwise importance comparisons between sentences from lease agreements to handle (2). We focus on lease agreements, which have clearly defined parties with asymmetric roles (*tenant*, *landlord*), and for which modal category labels have previously been collected as described in Chapter 4.

The proposed solution has three benefits: (1) enables us to use LEXDEMOD corpus introduced in Chapter 4, (2) cognitively simplifies the annotation task for experts who only need to compare a few sentences (resulting in higher agreement) at a time rather than read and summarize a full contract (spanning $10-100$ pages), and (3) reduces the cost of annotation as a contract-level end-to-end summarization system requires more data to train than does a sentence-level categorizer and a ranker. We demonstrate the need for incorporating domain-specific notions of importance during summarization by comparing our system against various summarization baselines using both automatic and human evaluation.

5.1 Task Definition

We formally define the new task as: given a contract C consisting of a sequence of sentences (c_1, c_2, ..., c_n) and a party P, the task is to generate an extractive summary S consisting of the most important m_o obligations, m_e entitlements, and m_p prohibitions (where $m_i < n \; \forall i \in \{o, e, p\}$) for the specified party. As previously mentioned, we break this task into two sub-tasks: (1) **Content Categorization** to identify the categories expressed in a sentence of a contract for a given party, and (2) **Importance Ranking** to rank the sentences based on their *importance* to a given party.

5.2 Dataset Curation

The LEXDEMOD dataset (Chapter 4) contains sentence-level party-specific annotations for *obligations*, *entitlements*, *prohibitions*, *permissions*, *non-obligations*, and *non-entitlements* in lease agreements. However, the data does not contain any importance annotations. To facilitate importance ranking, we collect *party-specific* pairwise importance comparison annotations for sentences in this dataset. We only consider the three major categories: *obligations*, *entitlements*,[3] and *prohibitions*.

Annotation Task. Rating the *party-specific* importance of a sentence in a contract on an absolute scale requires well-defined importance levels to obtain reliable annotations. However, defining each importance level can be subjective and restrictive. Moreover, rating scales are also prone to difficulty in maintaining inter- and intra-annotator consistency (Kiritchenko and Mohammad, 2017), which we observed in pilot annotation experiments. We ran pilots for obtaining importance annotations for each sentence in a contract, as well as a pair of sentences on a scale of 0-5 (0 denoting 'not at all important', 1 least important, and 5 most important) taking inspiration from prior works (Sakaguchi et al., 2014; Sakaguchi and Van Durme, 2018) but found they had a poor agreement. Thus, following Abdalla et al. (2023), we use Best–Worst Scaling (BWS), a comparative annotation schema, which builds on pairwise comparisons and does not require $\binom{N}{2}$ labels. Annotators are presented with n=4 sentences from a contract and a party, and are instructed to choose the best (*i.e.*, most important) and worst (*i.e.*, least important) sentence. Given N sentences, reliable scores are obtainable from $(1.5N$–$2N)$ 4-tuples (Louviere and Woodworth, 1991; Kiritchenko and Mohammad, 2017). To our knowledge, this is the first work to apply BWS for importance

[3] We merge *entitlement* and *permission* categories as there were a small number of permission annotations per contract. Entitlement is defined as a right to have/do something while permission refers to being allowed to have/do something.

annotation in legal contracts.

From a list of $N = 3,300$ sentences spanning 11 lease agreements from LEXDEMOD, we generate[4] $4,950$ $(1.5N)$ unique 4-tuples (each consisting of 4 distinct sentences from an agreement) such that each sentence occurs in at least six 4-tuples. We hire 3 legal experts (lawyers; 1 female, 2 males based in India) from Upwork with >2 years of experience in US-based contract drafting and reviewing to perform the task. We do not provide a detailed technical definition for importance but brief them about the task of summarization from the perspective of review and compliance, and encourage them to rely on their intuition, experience, and expertise (see below section for annotation reliability). Each tuple is annotated by 2 experts. We observed an increase in annotation reliability with the increase in the number of annotations done by the annotators. As the task of importance comparison is subjective, we also observed that the perception of importance changes with years of experience. This observation was confirmed by the legal experts involved in our data annotation. This is because legal professionals develop a more holistic view of cases over time. Experienced lawyers are better at assessing potential risks associated with different legal obligations, leading them to prioritize those that could have significant negative consequences even if they seem less urgent in the short term. Therefore, for more experienced experts, the relative importance of different obligations is based on a broader understanding of legal complexities and potential outcomes.

Annotation Aggregation. Annotation for each 4-tuple provides us 5 pairwise inequalities. *E.g.*, if a is marked as the most important and d as the least important, then we know that $a \succ b$, $a \succ c$, $a \succ d$, $b \succ d$, and $c \succ d$. From all the annotations and their inequalities, we calculate real-valued importance scores for all the sentences using Bradley-Terry (Bradley-Terry) model (Bradley and Terry, 1952) and use these scores to obtain $\binom{N}{2}$ pair-

[4]The tuples are generated using the BWS scripts provided in Kiritchenko and Mohammad (2017) and available here.

Party	%Obl.≻Ent.	%Ent.≻Pro.	%Pro.≻Obl.
Tenant	**52.25**	39.31	**54.49**
Landlord	46.56	**55.98**	46.82

Table 5.1: Importance comparison statistics (per party) computed from a subset of pairwise comparisons where the category of the sentences in a pair is different. Bold win %s are significant ($p < 0.05$). ≻: more important.

wise importance comparisons. Bradley-Terry is a statistical model used to convert a set of paired comparisons (need not be a full or consistent set) into a full ranking. This model is being widely used in aligning language models from human feedback (Rafailov et al., 2024) as it helps in ranking which response is considered better between two options, thus guiding the model towards generating more human-aligned outputs.

Annotation Reliability. A commonly used measure of quality and reliability for annotations producing real-valued scores is split-half reliability (SHR) (Kuder and Richardson, 1937; Cronbach, 1951). It measures the degree to which repeating the annotations would lead to similar relative rankings of the sentences. To measure SHR,[5] annotations for each 4-tuple are split into two bins. These bins are used to produce two different independent importance scores using a simple counting mechanism (Orme, 2009; Flynn and Marley, 2014): the fraction of times a sentence was chosen as the most important minus the fraction of times the sentence was chosen as the least important. Next, the Spearman correlation is computed between the two sets of scores to measure the closeness of the two rankings. If the annotations are reliable, then there should be a high correlation. This process is repeated 100 times and the correlation scores are averaged. Our dataset obtains a SHR of 0.66 ± 0.01 indicating moderate-high reliability.[6]

[5]We use scripts available at https://www.saifmohammad.com/WebDocs/Best-Worst-Scaling-Scripts.zip.

[6]As per the interpretation mentioned here https://www.statstutor.ac.uk/resources/uploaded/spearmans.pdf

Dataset Analysis. We obtain a total of $293,368$ paired comparisons after applying the Bradley-Terry model. Table 5.1 shows the percentage of sentences containing a category annotated as more important than those from another.

For tenants, prohibitions are annotated as more important than both obligations and entitlements, perhaps as prohibitions might involve penalties. Similarly, obligations are more important than entitlements, as knowing them might be more important for tenants (*e.g.,* failure to perform duties such as "paying rent on time" may lead to late fees). For landlords, on the other hand, knowing entitlements is more important than obligations or prohibitions, possibly due to the nature of lease agreements where landlords face fewer prohibitions and obligations than tenants.

As we do not explicitly define the notion of "importance" during annotation. We learned from feedback and interaction with the annotators about several factors they considered when determining importance. These factors included the degree of liability involved (*(e.g.,* 'blanket indemnifications' were scored higher than 'costs incurred in alterations' by a tenant since blanket rights can have an uncapped amount of liability), and liabilities incurred by a more probable event were rated as more important than those caused by less probable events, among others. As is evident from these examples, the factors that can influence importance may be complex, multi-faceted, and difficult to exhaustively identify. Therefore, our approach in this chapter was to allow our annotators to use their legal knowledge to inform a holistic judgment.

5.3 CONTRASUM: Methodology

We build a pipeline-based extractive summarization system, CONTRASUM (Figure 5.3), for the proposed task. It consists of two modules for the two sub-tasks: **Content Categorizer** (§5.3.1) and **Importance Ranker** (§5.3.2). The Content Categorizer first identifies

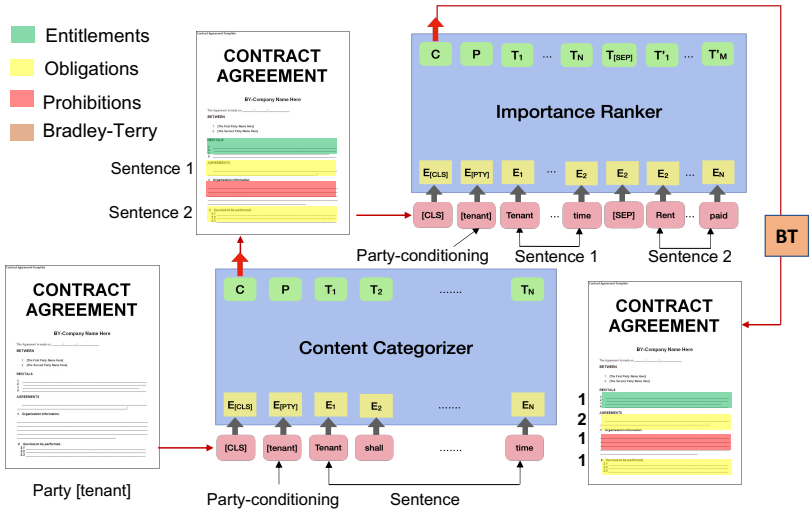


Figure 5.3: CONTRASUM takes a contract and a party to first identify all the sentences containing party-specific obligations, entitlements, and prohibitions using a content categorizer. Then, the identified sentences for each category are pairwise-ranked using an importance ranker. A ranked list of sentences is obtained using the Bradley-Terry model to obtain the final summary.

sentences from a contract that mention any of the obligations, entitlements, or prohibitions for a given party. Then, the identified sentences per category for a given party are pairwise ranked for importance using the Importance Ranker. From all the pairwise comparisons, a ranked list of sentences is obtained using the Bradley-Terry model (Bradley and Terry, 1952) to produce a final party-specific summary consisting of most important m_o obligations, m_e entitlements, and m_p prohibitions.

5.3.1 Content Categorizer

The Content Categorizer takes in a sentence c_i from C and a party to output all the categories (such as, *obligations, entitlements, prohibitions*) mentioned in c_i. Such categorization helps in partitioning the final summary as per the categories of interest. We use the multi-label classifier introduced in the previous chapter (Section 4.4) as the categorizer. Since a sentence can contain multiple categories, the same sentence can be a part of

different categories.

5.3.2 Importance Ranker

A contract may contain a large number of obligations, entitlements, and prohibitions for each party; however, not all instances of each category may be equally important. Also, the importance of sentences within each category may vary for each of the parties. Therefore, this module aims to rank the sentences belonging to each category based on their level of *importance* for the specified party. As indicated in §5.2, we do not define the notion of "importance"; instead, we rely on the annotations from legal experts based on their understanding from contract review and compliance perspective to build the ranker. This module (Figure 5.3) takes in a pair of sentences (c_i, c_j) from C and a party P to predict if c_i is more important to be a part of the summary than c_j for P. To realize this, we train a binary classifier that orders the input pair of sentences based on their importance for the specified party.

5.3.3 End-to-end Summarization

Recall that the Content Categorizer and Importance Ranker work at the sentence- and sentence-pair-level, respectively. Therefore, to produce the desired *party-specific* category-based summary for a contract, we obtain (1) categories for each sentence in the contract (using the Content Categorizer), and (2) a ranking of the sentence pairs within each category according to their relative importance (using the Importance Ranker) with respect to the party. We do not explicitly account for the diversity within each category (although organization by category helps ensure some degree of diversity across categories). As the ranker provides ranking at a sentence pair level, to obtain an importance-ranked list of sentences from all the pairwise predictions for each category, we use the Bradley-Terry

(Bradley-Terry) model as described in §5.2. We produce the final summary by selecting the m_o, m_e, and m_p most important sentences predicted as obligations, entitlements, and prohibitions, respectively.

5.4 Experimental Setup

CONTRASUM is a pipeline-based system consisting of two modules. The two modules are trained (and evaluated) separately and pipelined for generating end-to-end summaries as described in §5.3.3.

Datasets. We use the category annotations in LEXDEMOD dataset to train and evaluate the **Content Categorizer**. Dataset statistics are shown in Table 4.2.

For training the **Importance Ranker**, we need sentence pairs from contracts ordered by relative importance to a particular party. We use the pairwise importance comparison annotations collected in §5.2 to create a training dataset. If, for a pair of sentences (a,b), $a \succ b$ (a is more important than b), then the label for binary classification is positive, and negative otherwise. We retain the same train/dev/test splits as LEXDEMOD to avoid any data leakage. Table 5.2 present the data statistics.

As mentioned earlier, the two modules are separately trained and evaluated at a sentence- or sentence pair-level. However, **CONTRASUM** generates *party-specific* summaries at a contract-level. Therefore, for evaluating the system for the end-to-end summarization task, we need reference summaries. To obtain reference summaries, we need ground-truth category and importance ranking between sentences in a contract. Since we collected importance comparison annotations for the same sentences for which LEXDEMOD provides category annotations, we group the sentences belonging to a category (ground-truth) and then derive the ranking among sentences within a category using the gold importance anno-

Label/Split	Train	Dev	Test
Positive	85529	2538	71576
Negative	62857	2137	68731

Table 5.2: Statistics of pairwise importance annotations.

tations and Bradley-Terry model (as described earlier) for each party. We obtain reference summaries at different compression ratios (CR) ($5\%, 10\%, 15\%$) with a maximum number of sentences capped at 10 per category to evaluate the output summaries against different reference summaries. CR is defined as the % of total sentences included in the summary for a contract. Please note that the reference summaries are extractive.

Training Details. We use the RoBERTa-large (Liu et al., 2019) model with a binary (multi-label) classification head and fine-tune it on the collected importance dataset (LEXDEMOD) for the **Importance Ranker** (**Content Categorizer**) using binary-cross entropy loss.

Implementation Details. We use HuggingFace's Transformers library (Wolf et al., 2019) to fine-tune PLM-based classifiers. The content categorizer is fine-tuned for 20 epochs, and the importance ranker for 1 epoch with gradient accumulation steps of size 4. We report the test set results for model(s) with the best F1-macro score on the development set. Both the category and importance classifiers are trained with maximum sequence length of 256 and batch size of 8. For both the classifiers, party conditioning is done with the help of special tokens prepended to the sentence, as this has been successfully used previously for controlled text generation tasks (Sennrich et al., 2016; Johnson et al., 2017). The input format for the Content Categorizer is - [PARTY] SENTENCE and for the Importance Ranker is - [PARTY] SENTENCE1 $\langle/s\rangle$ SENTENCE2. We use Adam optimizer with a linear scheduler for learning rate having an initial learning rate of $2e^{-5}$, and warm-up

ratio set at 0.05. All the models are trained and tested on NVIDIA Tesla V100 SXM2 16GB 904 GPU machine. We experiment with batch size $\in \{2, 4, 8\}$ ($\in \{8, 16, 32\}$), number of epochs $\in \{3, 5, 10, 20, 30\}$ ($\in \{1, 3\}$), learning rate $\in \{1e^{-5}, 2e^{-5}, 3e^{-5}, 5e^{-5}\}$ ($\in \{1e^{-5}, 2e^{-5}, 3e^{-5}, 5e^{-5}\}$), and warm-up ratio $\in \{0.05, 0.10\}$ (0.05) for the content selector (importance predictor). BERT-base ($110M$ parameters) and Roberta-base ($125M$ parameters) models took 46 (120) mins, and RoBERTa-large ($355M$ parameters) took 2 (6) hrs to train for content selector (importance predictor). We use choix python package for Bradley-Terry model's implementation. We use official implementation and released models of PACSUM.[7]

Dataset Preprocessing. We use lexnlp (Bommarito II et al., 2021) to get sentences from a contract and filter the sentences which define any terms (using regular expressions such as, "means", "mean", and "shall mean") or does not mention any of the parties (mentioned in the form of alias such as, "Tenant", "Landlord", "Lessor", and "Lessee").

5.5 Evaluation

We perform an extrinsic evaluation of CONTRASUM to assess the quality of the generated summaries as well as an intrinsic evaluation of the two modules separately to assess the quality of categorization and ranking. Intrinsic evaluation of the Categorizer and the Ranker is done using the test sets of the datasets (§5.4) used to train the models. Although the size of datasets used to evaluate the two modules at a sentence- ($\sim 1.5K$ for Categorizer) or sentence pair-level ($\sim 130K$ for Ranker) is large, it amounts to 5 contracts for the extrinsic evaluation of CONTRASUM for the end-to-end summarization task at a contract-level. Since the number of contracts is small, we perform 3-fold validation of CONTRASUM. Each fold contains 5 contracts (in the test set) sampled from 11 contracts

[7] https://github.com/mswellhao/PacSum

(remaining contracts are used for training the modules for each fold) for which both category and importance annotations are available to ensure the availability of reference summaries. CONTRASUM uses the best Categorizer and the best Ranker (RoBERTa-L models) trained on each fold.

Evaluation Measures. We report macro-averaged Precision, Recall, and F1 scores for predicting the correct categories or importance order for the **Content Categorizer** and **Importance Ranker**. We also report the accuracy of the predicted labels. As both the reference and predicted summaries are extractive in nature, we use metrics from information retrieval and ranking literature for the end-to-end evaluation of **CONTRASUM**. We report Precision@k, Recall@k, F1@k (following Keymanesh et al., 2020), Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG) (as defined below) computed at a category-level for each party.

$$Precision@k = \frac{\text{true positives@k}}{k}$$

$$Recall@k = \frac{\text{true positives@k}}{\text{true positives@k} + \text{false negatives@k}}$$

$$F1@k = \frac{2 * \text{Precision@k} * \text{Recall@k}}{(\text{Precision@k} + \text{Recall@k})}$$

$$MAP = \frac{1}{Q}\sum_{q=1}^{Q}\sum_{k=1}^{n} Prec.@k * (Recall@k - Recall@(k-1))$$

$$DCG = \sum_{i=1}^{n} \frac{rel_i}{log_2(i+1)}$$

$$IDCG = \sum_{i=1}^{n} \frac{rel_i^{true}}{log_2(i+1)}$$

$$nDCG = \frac{DCG}{IDCG}$$

where rel_i is the relevance (1 or 0) for the predicted sentences and rel_i^{true} is the relevance of ideal importance ordering of sentences. We take $n = min(1,$ number of predicted sentences for NDCG. Q is the number of parties or contracts on which the measure is averaged over. The average score (averaged across categories for each party and then across parties for a contract) is computed by using the following formula for each metric m, N contracts in the test set with number of parties as N_{party}, and m_k metric value for the kth category.

$$Avg(m) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_{party}}\sum_{j=1}^{N_{party}}\frac{1}{N_{category}}\sum_{k=1}^{N_{category}} m_k$$

We believe these metrics give a more direct measure of whether any sentence is correctly selected and ranked in the final summary. For completeness, we also report ROUGE-1/2/L scores.

Importance Ranker Baselines. We compare the Ranker against various pre-trained language models; **BERT-BU** (Devlin et al., 2019a), contract BERT (**C-BERT-BU**) (Chalkidis et al., 2020), and **RoBERTa-B** (Liu et al., 2019), and majority.

Content Categorizer Baselines. We do not compare the categorizer against other baselines, instead directly report the sores in Table 4.7, as we use the best-performing model based on F1-score on the development set as described in the previous chapter.

Extractive Summarization Baselines. We compare CONTRASUM against several unsupervised baselines which use the same content categorizer but the following rankers.

1. **Random** baseline picks random sentences from the set of predicted sentences belonging to each category. We report average scores over 5 seeds.
2. **KL-Sum** (Kullback and Leibler, 1951) aims to minimize the KL-divergence between the input contract and the produced summary by greedily selecting the sentences.

3. **LSA** (Latent Semantic Analysis) (Ozsoy et al., 2011) analyzes the relationship between document sentences by constructing a document-term matrix and performing singular value decomposition to reduce the number of sentences while capturing the structure of the document.
4. **TextRank** (Mihalcea and Tarau, 2004) uses the PageRank algorithm to compute an importance score for each sentence. High-scoring sentences are then extracted to build a summary.
5. **LexRank** (Erkan and Radev, 2004) is similar to TextRank as it models the document as a graph using sentences as its nodes. Unlike TextRank, where all weights are assumed as unit weights, LexRank utilizes the degrees of similarities between words and phrases, then calculates an importance score for each sentence.
6. **PACSUM** (Zheng and Lapata, 2019) is another graph-based ranking algorithm that employs BERT (Devlin et al., 2019a) to better capture the sentential meaning and builds graphs with directed edges as the contribution of any two nodes to their respective centrality is influenced by their relative position in a document.
7. **Upper-bound** picks sentences from a contract for a category based on the ground-truth importance ranking derived from human annotators. It indicates the performance upper-bound of an extractive method that uses our categorizer.

We limit the summaries to contain $m_i = 10$ (chosen subjectively) sentences per category for experimental simplicity and that it is short enough to be processed quickly by a user while capturing the most critical points but this number can be adjusted. Note that if less than m_i sentences are predicted to belong to a category then the summary will have less than m_i sentences. We also compare the performance of summarization systems that utilize the ground-truth categories (GC) followed by the above ranking methods to produce the summary. These systems help us investigate the effect of error propagation from the

categorizer to the ranker and the final summaries. We use the Sumy[8] python package for implementing these ranking methods.

Model	Accuracy	Precision	Recall	F1
Majority	51.58/50.35/51.01	25.79/25.17/25.51	50.00/50.00/50.00	34.03/33.49/33.78
BERT-BU	69.80/67.72/68.83	69.77/67.73/68.81	69.75/67.71/68.80	69.76/67.71/68.80
RoBERTa-B	70.50/68.11/**69.75**	70.47/68.15/68.52	70.46/68.09/68.43	70.46/68.08/68.41
C-BERT-BU	70.32/68.24/69.29	70.30/68.25/68.94	70.28/68.23/69.06	70.29/68.22/68.97
RoBERTa-L	**70.55**/**68.54**/69.60	**70.53**/**68.61**/**69.62**	**70.47**/**68.51**/**69.54**	**70.48**/**68.49**/**69.54**

Table 5.3: Evaluation results for the **Importance Ranker**. Scores for **Tenant/Landlord/Both** are averaged over 3 different seeds. BU, B, and L denote base-uncased, base, and large, respectively.

Module	Accuracy	Precision	Recall	F1
Content Categorizer-2	76.92	77.80	77.19	77.15
Content Categorizer-3	76.29	78.89	79.50	79.12
Importance Ranker-2	68.74	68.34	68.39	68.36
Importance Ranker-3	69.42	69.23	69.31	69.26

Table 5.4: Evaluation results of content categorizer (-fold) and importance ranker (-fold) for remaining two folds. RoBERTa-L is fine-tuned for both the modules as it was the best performing model for fold-1.

5.6 Results and Analysis

Automatic Evaluation of the Categorizer and the Ranker. We report the evaluation results (on fold 1) for the Ranker in Table 5.3. Fine-tuned PLMs outperform the majority baseline on F1 score as expected. While C-BERT-BU, which is pre-trained on contracts, performs better than BERT-BU and RoBERTa-B, the overall F1 score is the highest for RoBERTa-L, suggesting increased model size and training compensate for lack of pretraining on contracts. For the Categorizer, results are reported in Table 4.7. Table 5.4 presents the results for the models on the other folds.

[8] https://pypi.org/project/sumy/

	Model	CR=0.05		CR=0.10		CR=0.15	
		MAP	**NDCG**	**MAP**	**NDCG**	**MAP**	**NDCG**
PC	Random	0.113	0.169	0.123	0.199	0.141	0.228
	KL-Sum	0.126	0.173	0.132	0.194	0.138	0.218
	LSA	0.093	0.154	0.103	0.177	0.142	0.225
	TextRank	0.121	0.182	0.133	0.219	0.152	0.244
	PACSUM (BERT)	0.135	0.194	0.148	0.232	0.150	0.249
	LexRank	0.130	0.186	0.151	0.226	0.165	0.253
	CONTRASUM	**0.223**	**0.319**	**0.234**	**0.351**	**0.262**	**0.392**
PC	Upper-bound	0.579	0.628	0.607	0.680	0.614	0.698
GC	Random	0.206	0.298	0.209	0.316	0.228	0.346
	KL-Sum	0.174	0.252	0.191	0.293	0.204	0.313
	LSA	0.234	0.309	0.239	0.332	0.287	0.389
	TextRank	0.144	0.237	0.151	0.264	0.178	0.303
	PACSUM (BERT)	0.162	0.254	0.178	0.291	0.210	0.330
	LexRank	0.198	0.278	0.219	0.317	0.236	0.353
	CONTRASUM-CC	0.398	0.525	0.393	0.535	0.431	0.575

Table 5.5: Evaluation results of end-to-end summarization for different compression ratios (CR) averaged across 3 folds. GC (PC) denotes use of ground-truth (predicted from CC) categories. CC: Content Categorizer.

Automatic Evaluation of Summaries. We report the automatic evaluation results for the end-to-end summarization of contracts in Table 5.5. CONTRASUM achieves the best scores for both MAP and NDCG computed against the gold-references at different compression ratios establishing the need for domain-specific notion of importance which is not captured in the other baselines. Surprisingly, Random baseline performs better than (LSA) or is comparable to other baselines (KL-Sum) when predicted categories are used. While PACSUM achieves better scores than LexRank at low compression ratios, using centrality does not help at CR=0.15. As expected, we observe a consistent increase in NDCG with the increase in the compression ratio as the number of sentences per category increases in the gold-references. While CONTRASUM outperforms all the baselines, it is still far away from the upper-bound performance which uses the gold importance ranking. This calls

	Model	CR=0.05	CR=0.10	CR=0.15
		ROUGE-1/2/L	**ROUGE-1/2/L**	**ROUGE-1/2/L**
PC	Random	32.51/18.21/21.06	41.22/23.75/25.03	47.26/28.23/28.39
	KL-Sum	32.32/17.53/20.71	40.26/21.77/23.84	46.37/26.21/27.16
	LSA	32.75/18.05/20.78	41.40/23.58/24.86	47.16/27.84/27.33
	TextRank	32.39/18.76/21.55	41.73/25.02/25.98	48.03/29.33/29.16
	PACSUM (BERT)	**32.85**/18.66/21.35	41.75/24.74/25.72	48.03/29.63/29.27
	LexRank	32.78/18.31/21.38	42.38/24.67/25.85	48.19/28.90/29.32
	CONTRASUM	32.61/**22.92**/**23.99**	**43.76**/**30.91**/**29.35**	**51.37**/**37.31**/**33.19**
PC	Upper-bound	37.04/30.75/32.04	51.39/43.99/44.04	59.91/51.50/51.55
GC	Random	36.40/24.91/26.61	45.19/30.54/30.55	52.28/36.62/34.38
	KL-Sum	34.82/21.89/24.27	43.86/28.31/38.66	50.87/33.76/32.13
	LSA	36.43/24.24/25.78	46.04/30.45/30.51	52.77/36.25/33.30
	TextRank	34.88/23.44/25.32	44.56/29.61/29.67	51.90/35.93/34.05
	PACSUM (BERT)	36.19/24.33/26.21	45.04/30.10/29.73	51.92/36.19/33.64
	LexRank	35.18/23.13/24.76	44.67/29.32/29.40	52.24/36.42/35.15
	CONTRASUM-CC	37.55/32.28/30.84	49.16/40.67/36.49	57.68/48.37/40.91

Table 5.6: Rouge scores for end-to-end summarization for different compression ratios (CR) averaged across 3 folds. GC (PC) denotes use of ground-truth (predicted from the content categorizer) categories. CC: content categorizer.

for a more sophisticated and knowledge-driven approach to learning the importance of different sentences. We observe similar trends in rouge scores (Table 5.6). CONTRASUM outperforms all other baselines on ROUGE-2 and ROUGE-L for all compression ratios. A minor difference in ROUGE-1 scores among different baselines might be because of the limited vocabulary used in legal documents resulting in similar 1-grams.

As CONTRASUM is a pipeline-based system where the sentences predicted as containing each of the categories are input to the importance ranker, erroneous category predictions may affect the final summaries. Thus, we present scores (last block in Table 5.5) from different systems which use ground-truth (GC) categories and various ranking methods. While the performance of all the systems improves over using predicted categories, the random baseline with GC outperforms all other baselines with GC, except for LSA, suggesting off-the-shelf summarizers are not well-suited for this task. Nevertheless, CON-

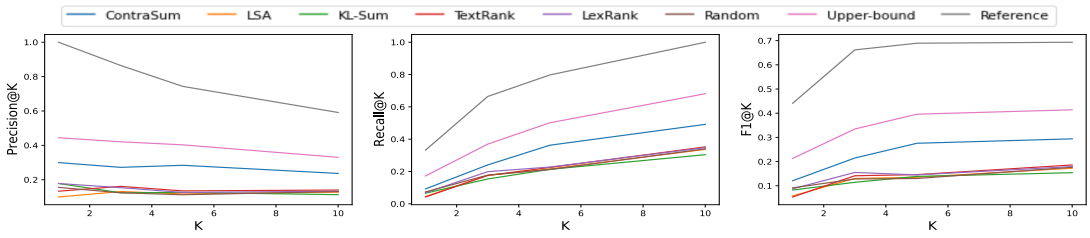


Figure 5.4: Precision@K, Recall@K, and F1@K ($K \in \{1, 3, 5, 10\}$) scores averaged across 3 folds for the end-to-end summaries at CR=0.15. All the models (except Reference) use predicted categories for the final summary.

TRASUM beats all the systems without the Categorizer indicating the effectiveness of the Importance Ranker in comparing the importance of sentences. Similar trends are observed in party-wise results in Table 5.7). An output summary is provided in Figure 5.5 and 5.6.

Figure 5.4 shows the Precision@k, Recall@k, and F1@k $k \in \{1, 3, 5, 10\}$ trends. As

		Tenant						Landlord					
		CR=0.05		CR=0.10		CR=0.15		CR=0.05		CR=0.10		CR=0.15	
	Model	**MAP**	**NDCG**	**MAP**	**NDCG**	**MAP**	**NDCG**	**MAP**	**NDCG**	**MAP**	**NDCG**	**MAP**	**NDCG**
PC	Random	0.102	0.162	0.116	0.195	0.139	0.228	0.124	0.176	0.130	0.203	0.144	0.228
	KL-Sum	0.076	0.137	0.089	0.163	0.11	0.200	0.176	0.210	0.175	0.226	0.167	0.236
	LSA	0.089	0.150	0.102	0.176	0.140	0.221	0.097	0.157	0.103	0.178	0.145	0.229
	TextRank	0.095	0.156	0.111	0.199	0.125	0.224	0.148	0.208	0.155	0.238	0.180	0.265
	PACSUM (BERT)	0.151	0.208	0.167	0.257	0.168	0.268	0.119	0.180	0.129	0.206	0.133	0.229
	LexRank	0.144	0.200	0.179	0.252	0.183	0.270	0.117	0.172	0.129	0.206	0.148	0.236
	CONTRASUM	**0.207**	**0.317**	**0.209**	**0.340**	**0.236**	**0.378**	**0.240**	**0.321**	**0.258**	**0.361**	**0.289**	**0.406**
PC	Upper-bound	0.579	0.630	0.614	0.688	0.613	0.694	0.579	0.626	0.600	0.671	0.615	0.702
GC	Random	0.195	0.298	0.196	0.313	0.221	0.349	0.217	0.298	0.221	0.319	0.236	0.343
	KL-Sum	0.119	0.196	0.119	0.238	0.131	0.366	0.229	0.309	0.263	0.368	0.276	0.389
	LSA	0.159	0.241	0.173	0.267	0.267	0.366	0.310	0.377	0.305	0.397	0.307	0.413
	TextRank	0.111	0.209	0.123	0.242	0.146	0.275	0.176	0.265	0.180	0.285	0.210	0.330
	PACSUM (BERT)	0.142	0.256	0.161	0.289	0.192	0.321	0.182	0.253	0.194	0.293	0.228	0.338
	LexRank	0.217	0.319	0.238	0.347	0.232	0.354	0.181	0.238	0.200	0.288	0.241	0.352
	CONTRASUM-CC	0.355	0.499	0.342	0.495	0.395	0.549	0.441	0.551	0.444	0.574	0.467	0.601

Table 5.7: Evaluation results of end-to-end summarization with respect to Tenant and Landlord for different compression ratios (CR) averaged across 3 folds. GC (PC) denotes use of ground-truth (predicted from the content categorizer) categories; CC: content categorizer.

expected, precision decreases with k while recall and F1 increase. Similar to Table 5.5, CONTRASUM outperforms the baselines in each plot while there is a huge gap between our model's performance and the upper-bound leaving scope for improvement in the future.

Owing to the recent advancements and power of LLMs, we prompt ChatGPT to asses

Model	Info.↑	Use↑	AoC↑	Red.↓	AoIR↑	O↑
Rand	2.96	2.87	2.92	1.92	2.50	2.94
LR	3.00	2.79	3.04	2.04	2.46	3.06
CS	**3.25**	**3.37**	**3.25**	2.04	2.54	**3.50**
Ref	3.17	3.00	3.17	**1.75**	**2.87**	3.06

Table 5.8: Results for human evaluation of model summaries. Mean scores of the 2 annotators are averaged across categories and parties. Overall (O) rating is scaled down to half to make it comparable. Info., Use, AoC, Red., AoIR denote informativeness, usefulness, accuracy of categorization, redundancy, and accuracy of importance ranking, resp. Rand: Random (PC), LR: LexRank (PC), CS: CONTRASUM, Ref: References.

its performance on this task. Since contracts are much longer than the token limit of ChatGPT (3.5-turbo), we cannot prompt it to generate the summary of all the obligations for a tenant in one go. Instead, we segment the contract at a page-level and first prompt (*Can you extract sentences that mention any obligations for the Tenant from this text. Ensure that sentences are present in the provided contract.*) ChatGPT to extract all the obligations mentioned for the tenant in the provided text from this contract.[9] Since the output of ChatGPT is not deterministic, after running the same prompt for 5 times, we observed that the output sentences were extractive only 50% of the times. Also, generated obligations were sometimes hallucinated; generic but not present in the text of the page provided as context. For *e.g.*, it generated *"Section 13.1 of the lease mentions that it is the Tenant's responsibility to maintain the Leased Premises in good condition and repair."* as an obligation for the first page in the contract. Therefore, it is not straightforward for ChatGPT to perform the task with simple prompting due to hallucinations in the generated output and token limit. Further work is needed to look into how to best use LLMs for such tasks in legal domain.

Human Evaluation of Summaries. In addition to automatic evaluation, we also give contracts along with their *party-specific* summaries from CONTRASUM, gold-references

[9]We experiment for one contract https://www.sec.gov/Archives/edgar/data/1677576/000114420419013746/tv516010_ex10-1.htm

(CR=0.15), the best baseline-LexRank (PC), and Random (PC) baseline to legal experts from India (1 female, 1 male) for human evaluation. 16 summaries, each consisting of a maximum of 10 sentences per category, for 2 contracts (along with the complete contract) are provided to 2 experts. They are asked to rate the summaries for each category on a 5-point scale (1 least; 5 most) as per: (1) informativeness–How informative is the summary for the given party from the review or compliance perspective?; (2) usefulness–How useful is the summary for the given party from the review or compliance perspective?; (3) accuracy of categorization–How correct is the partitioning of the sentences in each category?; (4) redundancy–How much of the content of the summary is repetitive or about the same topic?, and (5) accuracy of importance ranking–How correctly are the sentences within a category ranked? In addition, we ask them to rate the overall quality of a summary on a 10-point scale. The average scores are presented in Table 5.8. CONTRASUM produces the best summaries overall but lacks in the diversity of sentences within each category as we do not explicitly account for diversity during modeling. Interestingly, experts found summaries from CONTRASUM to be more informative, useful, and correctly categorized than the gold-references. This may happen as both predicted and reference summaries are capped at 10 sentences per category however, because the category labels themselves were predicted, the system and reference summaries did not always contain the same number of sentences per category if one was below the allowed limit (and in principle it is not possible to enforce a minimum number of sentences per category). Surprisingly, possible miscategorizations led to higher human ratings of overall summary outputs; this is an unexpected finding that highlights the potential challenges of evaluating complex, structured summarization outputs such as in this task. Experts rated usefulness taking into consideration a summary that they have in mind after reading the contract, whereas informativeness is scored only on the basis of the produced summary. Furthermore, experts also considered the importance of a category expressed in a sentence to a party. Hence, they penalize for

the accuracy of categorization if the category is indirectly (although correctly) expressed in a sentence for a party. For *e.g., "Tenant shall pay the rent to the landlord"*, represents a direct obligation for the Tenant and an indirect entitlement for the Landlord. So, in case this sentence is present in the summary for both tenant and landlord, then categorization score with respect to tenant will be more than for landlord.

5.7 Real-world Utility of Legal Document Understanding Systems

In Chapter 4 we introduced the task of identifying deontic modalities in legal agreements and showed that fine-tuned transformer-based models on LexDeMod dataset not only achieve high precision but also high recall as compared to a rule-based approach (see Table 4.7). We established the importance of associating a contracting party with their rights and duties as evidenced by significant improvements over a system trained without party-specific conditioning (see Table 4.7). In this chapter, we present an example of how such an identification system could enable the development of a practical application such as a party-specific summarization system (*i.e.*, ContraSum). However, we acknowledge that improvement in these systems as indicated by increased evaluation metric scores over baseline systems may result in differing real-world value or utility. This is because the real-world value is dependent on the use case and stakeholders. For instance, if the summarization system is used by the contracting party then even small improvements in the evaluation scores could be useful if that helps the party in identifying a couple more obligations (similarly other modalities) correctly or better rank the sentences based on their importance. However, if such a system is used by a legal professional for making decisions then the stakes are higher and thus small improvements may not add the same value as for the case of contracting parties. Therefore, to understand the translation of improvements in the metric scores to the real world, a system needs to be aligned to the specific requirements

of the stakeholders, and then deployed to measure the impact in terms of key performance indicators. One such indicator could be gain in productivity *i.e.* how fast can legal professionals make a decision or review when such a system is used vs in its absence? Since this is out of the scope of the current work, we leave further investigation of the real-world utility of such systems to future work.

5.8 Summary

We introduced a task to extract *party-specific* summaries of *important obligations*, *entitlements*, and *prohibitions* in legal contracts. Obtaining absolute importance scores for contract sentences can be particularly challenging, as we noted in pilot studies, thus indicating the difficulty of this task. Instead, we collected a novel dataset of legal expert-annotated *pairwise importance comparisons* for $293K$ sentence pairs from lease agreements to guide the Importance Ranker of CONTRASUM built for the task. Automatic and human evaluations showed that our system that models domain-specific notion of "importance" produces good-quality summaries as compared to several baselines. We leave the generation of simplified abstractive summaries for future work.

In this and the previous chapter, we define new tasks, introduce datasets, and design models to gather contracting party-specific information in the legal domain. As the legal documents are multi-page long, we handle the long context problem by breaking the tasks at a sentence-level considering local context either independent of each other in case of deontic modality detection or compare sentences in local context for importance ordering. The information extracted in these tasks is static and the entities are involved in one event. However, in settings where multiple entities interact in a sequence of events, information such as the relationship between these entities could evolve over time. To realize this aspect of entity information, we shift to the domain of narratives, in the next chapter, to model this

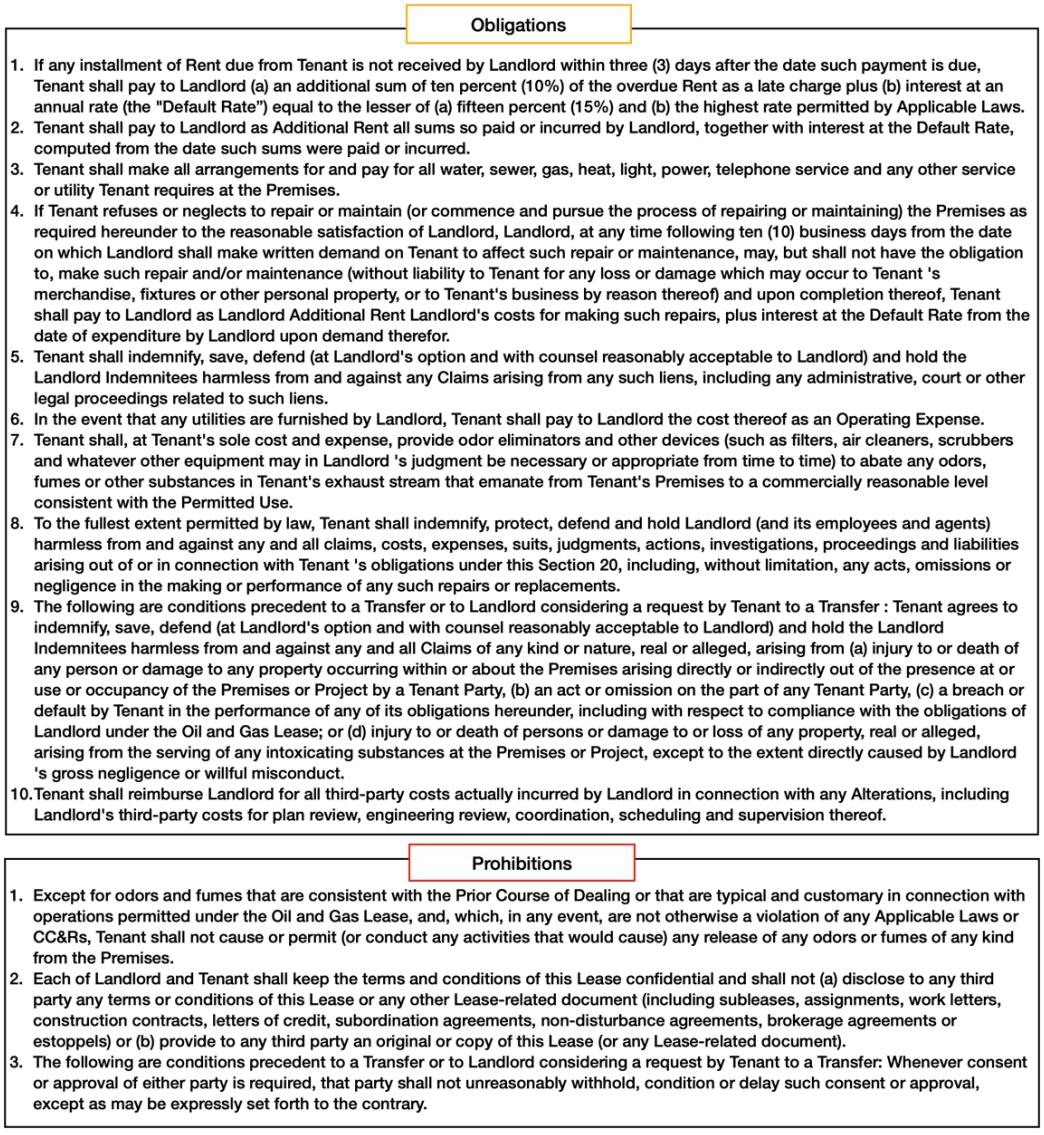


Figure 5.5: Obligations and permissions of full sample summary produced by CONTRASUM with respect to Tenant for a contact.[10] The sentences appear in decreasing order of importance in each category. Entitlements in Figure 5.6

aspect of the evolving nature of relationships between characters of a novel.

[10] The contract is available at https://www.sec.gov/Archives/edgar/data/1677576/000114420419013746/tv516010_ex10-1.htm

[11]The contract is available at https://www.sec.gov/Archives/edgar/data/1677576/000114420419013746/tv516010_ex10-1.htm

Entitlements

1. Notwithstanding anything to the contrary set forth in this Lease, in the event that (a) a Notice of Demand is filed with the AAA in connection with a Disbursement Claim (as such terms are defined in the Development Agreement), (b) the Final Arbitration Decision (as defined in the Development Agreement) requires Landlord to pay any Required Arbitration Construction Payments or any other monetary damages to Tenant, and (c) Landlord fails to pay such monetary damages to Tenant within thirty (30) days of the issuance of such decision, then Tenant shall be entitled to offset any payments of Rent due under this Lease required to be made by Tenant under from Tenant under this Lease until such time as Tenant has been reimbursed in full for the unpaid amount of such award.
2. The following are conditions precedent to a Transfer or to Landlord considering a request by Tenant to a Transfer: Notwithstanding anything to the contrary contained herein, Tenant may, at any time and from time to time, without the consent of Landlord, assign this Lease or any interest hereunder to, or sublease or license the Premises or any part thereof to (each of the following is referred to herein as a "Permitted Transfer"): (a) any successor entity of Tenant resulting from a merger, reorganization, or consolidation with Tenant (provided that such merger, reorganization or consolidation is undertaken primarily for independent business purposes, and not primarily for purposes of transferring this Lease or any interest in the Premises); (b) any initial public offering by Tenant or any or its affiliates, (c) any entity succeeding to all or substantially all of the business and assets of Tenant (provided that such transaction is undertaken primarily for independent business purposes, and not primarily for purposes of transferring this Lease or any interest in the Premises); (d) any entity that, as of the date of determination, is an Affiliate of Tenant; or (e) any entity that, concurrently with such Transfer, is acquiring all or substantially all of the business being conducted at the Premises by Tenant or its affiliates, provided that (i) Tenant shall notify Landlord in writing at least twenty (20) days prior to the effectiveness of such Permitted Transfer, (ii) Tenant shall provide Landlord with evidence reasonably satisfactory to Landlord that the Transfer qualifies as a Permitted Transfer and shall otherwise comply with the requirements of this Lease regarding such Transfer, (iii) the transferee has a net worth that is equal to or greater than the net worth of the transferring Tenant, and (iv) Tenant and each Guarantor (in accordance with such Guarantor's Guaranty) shall remain fully liable for the payment of all Rent and other sums due or to become due hereunder, and for the full performance of all other terms, conditions and covenants to be kept and performed by Tenant.
3. If (i) the estimated cost of restoration is less than One Million Dollars ($1,000,000.00), (ii) prior to commencement of restoration, no Default or event which, with the passage of time, would give rise to a Default shall exist and no mechanics' or materialmen's liens shall have been filed and remain undischarged, (iii) the architects, contracts, contractors, plans and specifications for the restoration shall have been approved by Landlord (which approval shall not be unreasonably withheld or delayed), (iv) Landlord shall be provided with reasonable assurance against mechanics' liens, accrued or incurred, as Landlord or its lenders may reasonably require and such other documents and instruments as Landlord or its lenders may reasonably require, and (v) Tenant shall have procured acceptable performance and payment bonds reasonably acceptable to Landlord in an amount and form, and from a surety, reasonably acceptable to Landlord, and naming Landlord as an additional obligee; then Landlord shall make available that portion of the Insurance Proceeds to Tenant for application to pay the costs of restoration incurred by Tenant and Tenant shall promptly complete such restoration.
4. If the estimated cost of restoration is equal to or exceeds One Million Dollars ($1,000,000.00), and if Tenant provides evidence satisfactory to Landlord that sufficient funds are available to restore the Premises, Landlord shall make disbursements from the available Insurance Proceeds from time to time in an amount not exceeding the cost of the work completed since the date covered by the last disbursement, upon receipt of (i) satisfactory evidence, including architect's certificates, of the stage of completion, of the estimated cost of completion and of performance of the work to date in a good and workmanlike manner in accordance with the contracts, plans and specifications, (ii) reasonable assurance against mechanics ' or materialmen's liens, accrued or incurred, as Landlord or its lenders may reasonably require, (iii) contractors' and subcontractors' sworn statements, (iv) a satisfactory bring-to-date of title insurance, (v) performance and payment bonds reasonably acceptable to Landlord in an amount and form, and from a surety, reasonably acceptable to Landlord, and naming Landlord as an additional obligee, (vi) such other documents and instruments as Landlord or its lenders may reasonably require, and (vii) other evidence of cost and payment so that Landlord can verify that the amounts disbursed from time to time are represented by work that is completed, in place and free and clear of mechanics' lien claims
5. Tenant shall keep the Premises free from any liens arising out of work or services performed, materials furnished to or obligations incurred by Tenant, or, in the alternative, Tenant may bond over any liens to the reasonable satisfaction of Landlord.
6. The following are conditions precedent to a Transfer or to Landlord considering a request by Tenant to a Transfer: If Tenant sublets the Premises or any portion thereof, Tenant hereby immediately and irrevocably assigns to Landlord, as security for Tenant's obligations under this Lease, all rent from any such subletting, and appoints Landlord as assignee and attorney-in-fact for Tenant, and Landlord (or a receiver for Tenant appointed on Landlord's application) may collect such rent and apply it toward Tenant's obligations under this Lease; provided that, until the occurrence of a Default (as defined below) by Tenant, Tenant shall have the right to collect such rent.
7. Subject to the terms of the Oil and Gas Lease and except for emergency maintenance or repairs, the right of entry contained in this paragraph shall be exercisable during business hours with not less than 24 hours prior notice, and subject to Tenant's authorized personnel accompanying Landlord's agents in all areas of the Premises.
8. Any Insurance Proceeds remaining upon completion of restoration shall be refunded to Tenant up to the amount of Tenant's payments pursuant to the immediately preceding sentence.
9. The following are conditions precedent to a Transfer or to Landlord considering a request by Tenant to a Transfer: Tenant shall be entitled to any award that is specifically awarded as compensation for (a) the taking of Tenant's personal property or trade fixtures that were purchased or installed at Tenant's expense, including without limitation, any crops or inventory at the Property and (b) the costs of Tenant moving to a new location.
10. The following are conditions precedent to a Transfer or to Landlord considering a request by Tenant to a Transfer: In the event of any default by Landlord, Tenant shall give notice by registered or certified mail to any (a) beneficiary of a deed of trust or (b) mortgagee under a mortgage covering the Premises or any portion thereof and to any landlord of any lease of land upon or within which the Premises are located, and shall offer such beneficiary, mortgagee or landlord a reasonable opportunity to cure the default, including time to obtain possession of the Premises by power of sale or a judicial action if such should prove necessary to effect a cure; provided that Landlord shall furnish to Tenant in writing, upon written request by Tenant, the names and addresses of all such persons who are to receive such notices.

Figure 5.6: Entitlements of full sample summary produced by CONTRASUM with respect to Tenant for a contact.[11] The sentences appear in decreasing order of importance in each category. Obligations and Permissions in Figure 5.5

Chapter 6: Tracking Evolving Relationship Between Characters in Books[1]

In Chapter 3, we focus on scripts where events happen with respect to one entity (*i.e.,* the protagonist). However, in the broader world of narratives such as that of novels, which contain character-centric narratives, multiple characters interact with each other in a sequence of interrelated events throughout the novel. As novels are substantially longer and more complicated than legal documents, relationships between characters may develop chapter-by-chapter in response to various events as the story progresses. Such evolving relationships (*e.g.*, Ron and Hermoine's relationship in the *Harry Potter* series transitions from initial tension and friendship to a deep romantic connection) play a critical role in making a narrative interesting. These relationships are often expressed implicitly through actions and need to be inferred, as opposed to static information that we extracted about each contracting party in chapter 4 and 5. Consider, for instance, *As the moon cast a silvery glow on their faces, James leaned in, his warm breath mingling with Emma's. Their lips met in a tender, lingering kiss, a silent affirmation that had grown over countless stolen moments like this.* From the context, we can infer that *Emma*, and *James* share a romantic relationship although it is not explicitly stated.

Relationship prediction is a challenging task due to its multi-faceted, complex, and uncertain nature in real-world scenarios (Hovy and Yang, 2021; Zhao et al., 2024). Since relationships are often expressed indirectly through actions, similar actions and conversations

[1] Abhilasha Sancheti, and Rachel Rudinger. 2024. Tracking Evolving Relationship Between Characters in Books in the Era of Large Language Models. *Under Submission*.

may have different interpretations depending upon the context, and appropriateness (Jurgens et al., 2023). Therefore, a lot of manual hours are spent in understanding the evolution in the relationships between characters. Having an automated system that can discover such insights has many practical applications (useful to the readers of the books) that include book recommendation systems based on similar or diverse relation-archetype narratives, question-answering systems that can aid readers in recalling the relation-archetypes until any point during the novel, and systems to predict character's personality or next action based on the trajectory of the relationship. Such an automated system could also be used as a tool by media researchers who spend a lot of manual hours in identifying such insights.

While prior works have considered the task of predicting static relationships from movie dialogues (Jia et al., 2021), TV series (Tigunova et al., 2021; Jurgens et al., 2023), or book summaries (Srivastava et al., 2016) and dynamic relationships from book summaries (Chaturvedi et al., 2016, 2017) or a sequence of passages from books (Iyyer et al., 2016a), limited efforts had been made due to the finite context window of models available at that time, and unavailability of annotated datasets. With the advent of large language models, known for their zero/few-shot reasoning capabilities (Brown et al., 2020; Touvron et al., 2023b; Jiang et al., 2023) and increased context window size (Team et al., 2023; Dubey et al., 2024), in this chapter, we consider the following research questions: (1) how can we characterize evolution in the relationship between characters in book-length text? and (2) how can we use large language models' zero-shot reasoning capabilities (in the absence of gold labels) to predict evolution in the relationship between characters?

We first characterize evolution in the relationship in terms of predefined relationship types and different ways (such as positive, negative, or stable) in which a relationship can evolve in §6.1. Then, we formally define the task of tracking evolution in the relationship (see Figure 6.1) between two characters in §6.2, and propose several strategies (§6.3) ranging from providing the full context from the book, a running summary of the previous

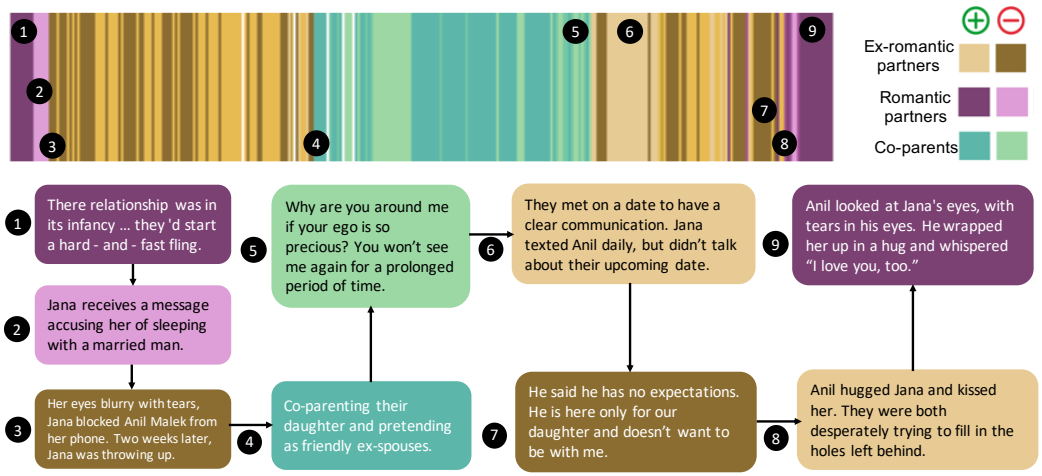


Figure 6.1: Sample trajectory of evolution in relationship between *Jana* and *Anil* in the book *Jana goes Wild*. Jana and Anil start as romantic partners followed by a tumultuous break-up but years later, co-parenting responsibilities of their daughter force them to confront lingering feelings, reevaluate their past, and rediscover love through shared growth and proximity. $\oplus$ ($\ominus$) denote a positive (negative) evolution in the relationship.

story to only providing the current passage. To address the issue of the unavailability of gold annotations for this task, we evaluate the predictions from the proposed strategies by thoroughly analyzing the agreement between predictions from different families of LLMs (§6.6). Low-to-moderate agreement (Krippendorff's α between $0.1 - 0.5$) between predictions from different LLMs suggests the difficulty of the task even for LLMs with increased context window size. To provide an upper bound on the performance achievable from the proposed strategies on this task, we manually examine the consensus predictions at both local and trajectory-level (§6.7). Low-to-moderate agreement between humans reinforces the difficulty of the task. Finally, we shed light on the challenging nature of this task by performing a quantitative (§6.8.1) and qualitative analysis (§6.8.2) of the predictions and disagreement between humans.

Category	Relationship Types
Romantic	Engaged, Married, Romantic interest, Dating, One-sided romantic interest, Separated, Ex-romantic interest, Ex-engaged
Social	Stranger, Acquaintance, Friend, Best friend
Anti-Social	Competitor or Enemy

Table 6.1: The ontology of relationships used following prior work (Jurgens et al., 2023).

6.1 Characterizing Evolution in Interpersonal Relationships

Prior works that model the evolution in the relationship between characters use an ontology of relationships that is either coarse-grained (cooperative vs non-cooperative) (Srivastava et al., 2016; Chaturvedi et al., 2016) or unsupervised (Chaturvedi et al., 2017; Iyyer et al., 2016a) (such as topics from a topic model). However, relationships can be of various types such as familial (*e.g.*, parent and siblings), social (*e.g.*, friends and acquaintance), romantic (*e.g.*, married and engaged), and professional (*e.g.*, boss and colleague) (Rashid and Blanco, 2018; Tigunova et al., 2021; Jurgens et al., 2023). Following Jurgens et al. (2023), we use a subset of relationship types (see Table 6.1), that are observed in the most frequently used tropes[2] (*e.g.*, enemies-to-lovers, and friends-to-lovers) in romance novels where relationships evolve with time (Lissauer, 2014). Furthermore, relationships are defined by multiple interrelated *interactions* (Blumstein and Kollock, 1988), and the fine-grained characteristics of interactions are not necessarily the same as those of a relationship (*e.g.*, two friends can have a heated argument during an interaction but that does not affect the long-term friendship). Such fine-grained characteristics of interactions and relationships are called *dimensions* in social science (Wish et al., 1976; Deri et al., 2018; Qamar et al., 2021). Inspired by this, we define the interactions between characters using a set of dimensions (such as similarity, trust, romance, social support, identity, respect, knowledge

[2]Trope refers to a recurring plot device, character archetype, or theme that is commonly used in books.

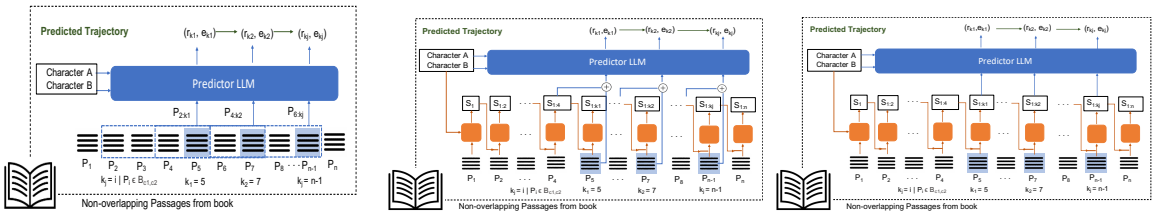
exchange, power, fun, and conflict) proposed by Deri et al. (2018). We believe that change in the intensity of such dimensions determines the *fine-grained* evolution in relationships which can of three types: **positive**, **negative**, and **stable**. A positive evolution signifies deepening connection, increasing trust, support or respect, spending more time together, and sharing similar goals. Any tension in a relationship due to conflicts, arguments, distrust, disrespect, lack of support, or misunderstandings denotes negative evolution. A stable relationship neither evolves positively nor negatively.

6.2 Task of Tracking Evolution in Relationship

We consider tracking evolution in the relationship between characters as a classification task formally defined as follows. Consider a book $B = \{P_1, P_2, \ldots P_n\}$ consisting of n chronologically ordered[3] (in book's passages) non-overlapping passages of a fixed length, c_1 and c_2 as the two characters, and $B_{c_1,c_2} = \{P_i \in B \mid \text{both } c_1 \text{ and } c_2 \text{ are mentioned in } P_i\}$. Note that B_{c_1,c_2} is a non-contiguous sub-sequence of $P_1, \ldots P_n$. The task is to predict a tuple (r_i, e_i) where $r_i \in R$ and $e_i \in E$, respectively, denote the type of relationship and evolution (from a predefined set as described in §6.1) between the two characters by the end of the passage $P_i \in B_{c_1,c_2}$ given the passages $P_{1:i}$. We define evolution in the relationship between c_1 and c_2 in a book B as a trajectory $T_{c_1,c_2} = \{(r_1, e_1), (r_2, e_2), \ldots, (r_j, e_j)\}$ of relationship[4] and evolution types at each passage P_{k_j} where $k_j = \{i \mid P_i \in B_{c_1,c_2}\}$.

[3]Please note that we assume a temporally linear plot structure, and leave the modeling of nonlinear timelines (or other complex structures, like worlds within worlds, etc.) for future work (Pustejovsky et al., 2003; Vashishtha et al., 2019).

[4]We consider the presence of one relationship type at one point in this work however, we acknowledge that multiple relationship types may relate two characters at the same time.



(a) Complete Passages (b) Summary with Passage (c) Complete Summary

Figure 6.2: Proposed strategies varying in the granularity of passages provided to an LLM for predicting the evolution in the relationship between given characters. ≡: Passage where both characters are mentioned ■: Summarizer LLM.

6.3 Proposed Strategies for Tracking Evolution in Relationship

Automatic tracking of fine-grained evolution in the relationship between characters in a book-length text poses two main challenges: (1) handling long context, and (2) lack of annotated datasets. To address these challenges, we aim to assess the zero-shot social reasoning capabilities of recent large language models (Jiang et al., 2023; Team et al., 2024; Dubey et al., 2024) with increased context window size by proposing various strategies (Figure 6.2) based on the granularity of information (*i.e.*, passages $P_{1:i}$) provided to an LLM to predict a relationship and evolution type by the end of each $P_i \in B_{c_1,c_2}$ (see §6.2).

6.3.1 Proposed Strategies

Complete Passages. This strategy uses the large context window of LLMs to provide passages until $P_i \in B_{c_1,c_2}$ (that can fit in the window) as-is in its highest granularity to predict the status of relationship and evolution type until P_i.

Summary with Passage. As books can be arbitrarily long, $P_{1:i} \in B_{c_1,c_2}$ may not always fit in the context window of LLMs. Further, a reader may know the relationship either because the text in passage P_i reveals information about it directly; or because they recall

it from prior passages, and no new information is introduced to change or contradict it; or relevant information is introduced in the passage P_i that is best understood in the context of information presented in prior passages. Thus, we hypothesize that providing a "memory" of prior passages is sufficient for relationship type prediction. However, evolution type changes are defined for each interaction between the two characters making it a more granular and local characteristic of relationships. Hence, instead of providing $P_{1:i}$ as-is, this strategy uses a summary of the type and nature of evolution in the relationship between two characters for passages $P_{1:i-1}$[5] (see §6.3.2) along with the passage P_i to predict the status of the relationship and evolution type by the end of P_i.

Complete Summary. To study if this task can be performed solely with a summary, in this strategy, we provide the complete context until passage P_i as a summary of the type and nature of evolution in the relationship between the two characters.

[5]Note that P_{i-1} denote the previous passage as per the chronology in B and not B_{c_1,c_2}.

Prompt 6.3.1: Iterative Summary Generation: First Passage

```
System Prompt:  You are a helpful assistant who follows the
instructions.  No preambles and postambles.  Avoid explanations if
not asked explicitly.

Prompt:  Below is the beginning part of a story from a book:

----
{story}
----

We are going over segments of a story sequentially to gradually
update one comprehensive summary depicting the evolution in the
relationship between {char_a} and {char_b}.  Write a summary of the
evolution in the relationship between {char_a} and {char_b} as the
story progresses.  Make sure to include vital information related
to key events that shape the relationship between {char_a} and
{char_b}, their objectives, and motivations.  The story may feature
non-linear narratives, flashbacks, switches between alternate worlds
or viewpoints, etc.  Therefore, you should organize the summary so
it presents a consistent and chronological narrative.  Despite this
step-by-step process of updating the summary, you need to create a
summary that seems as though it is written in one go.  The summary
should roughly contain 300 words.

Constraint:  If the provided segment does not mention both {char_a}
and {char_b}, then do not make up or predict anything regarding the
relationship between the characters.  Just provide a general summary
of the provided segment or keep it empty.

Summary:
```

Prompt 6.3.2: Iterative Summary Generation: Updating Previous Summary

```
System Prompt:  You are a helpful assistant who follows the
instructions.  No preambles and postambles.  Avoid explanations if
not asked explicitly.

Prompt:  Below is a segment of a story from a book:

----
{story}
----

Below is a summary of the evolution in the relationship between
{char_a} and {char_b} up until this point in the story.

----
{summary}
----

We are going over segments of a story sequentially to gradually
update one comprehensive summary depicting the evolution in the
relationship between {char_a} and {char_b}.  You are required to
update the provided summary to incorporate any new vital information
related to the relationship between {char_a} and {char_b} that is
present in the current segment of the story.  This information may
relate to key events, turning points in the relationship between the
characters, their objectives, and motivations.  The story may feature
non-linear narratives, flashbacks, switches between alternate worlds
or viewpoints, etc.  Therefore, you should organize the summary so
it presents a consistent and chronological narrative.  Despite this
step-by-step process of updating the summary, you need to create a
summary that seems as though it is written in one go.  The updated
summary should roughly contain 300 words.

Constraint:  If the provided segment does not mention both {char_a}
and {char_b}, then avoid making up or predicting anything regarding
the relationship between the characters.  Just copy the provided
summary as-is or update it with the general aspects of the story or
keep it empty.

Updated summary:
```

Prompt 6.3.3: Compress Summary

```
System Prompt:  You are a helpful assistant who follows the
instructions.  No preambles and postambles.  Avoid explanations if
not asked explicitly.

Prompt:  Below is a summary of the relationship between {char_a} and
{char_b} from a part of a story:

--
{Summary}
--

Currently, this summary contains {summary_length} words.  Your
task is to condense it to less than 300 words while maintaining the
chronological order.  The condensed summary should remain clear,
overarching, and fluid while being brief.  Whenever feasible,
maintain details about key events that shape the relationship between
{char_a} and {char_b}, how does the relationship evolve over time,
character's objectives, and motivations - but express these elements
more succinctly.  Remove insignificant details that do not add much
to the overall evolution in the relationship between {char_a} and
{char_b} and phrases like "in this ..  segment", "in this part ...
story", etc.  The story may feature non-linear narratives, flashbacks,
switches between alternate worlds or viewpoints, etc.  Therefore,
you should organize the summary so it presents a consistent and
chronological narrative.

Condensed summary (to be within 300 words):
```

6.3.2 Iterative Summary Generation

To obtain the required summary in the above strategies, following Chang et al. and Stiennon et al. (2020), we prompt an LLM (in a zero-shot setting) to iteratively generate a summary and update it with every new passage. Formally, $\mathcal{S}(P_{1:i}) = \mathcal{S}(\mathcal{S}(P_{1:i-1}), P_i)$ where, $\mathcal{S}$ is the summarizer LLM, $\mathcal{S}(P_{1:i-1})$ is the previous summary until passage P_{i-1} and $\mathcal{S}(P_{1:i})$ denotes the updated summary until passage P_i. As summaries may exceed a word limit, following Chang et al., we repeatedly prompt LLM to compress (Prompt 6.3.3) the summary until it is within the word limit. Generating a summary iteratively allows for the use of LLMs with smaller context windows, making the process faster, less expensive,

and more efficient in terms of inference time and number of generated tokens. We use different prompts to obtain the summary of the first passage (Prompt 6.3.1) and to update the previous summary with a new passage (Prompt 6.3.2).

Prompt 6.3.4: Relationship and Evolution Type Prediction Prompt for Complete Passages

```
System Prompt:  You are a helpful assistant who follows the
instructions.  No preambles and postambles.  Avoid explanations if
not asked explicitly.

Prompt:  Based on the provided context and the following segment,
answer the below questions about the type of relationship between
{char_a} and {char_b} and its evolution by the end of the provided
segment.

Context:
---
{previous passages}
---

Segment:
---
{segment}
---

Are {char_a} and {char_b} mentioned in the provided segment?  Answer
in one word <ANS> ["yes", "no", "unsure"] </ANS>.
{Model's output}

Can you infer any type of relationship between {char_a} and {char_b}
from the segment?  Answer in one word <ANS> ["yes", "no", "unsure"]
</ANS>
{Model's output}

Choose the type of relationship between {char_a} and {char_b} from
these options:  acquaintances, strangers, friends, best friends,
romantic interest, dating, engaged, married, separated, divorced,
enemies, spouse, ex-spouse, one-sided romantic interest, ex-romantic
interest, others or cannot be determined.  Answer only from the
provided options.  Relationship type:
{Model's output}

Is the chosen relationship between {char_a} and {char_b} evolving
"positively", "negatively", is "stable" or "nothing can be determined"
by the end of the segment?  A "positive" evolution can result from
deepening connection, increasing trust, support or respect, spending
more time together etc.  A "negative" evolution means any tension or
straining relationship that can result from conflicts, arguments,
distrust, disrespect, lack of support, or misunderstandings.  A
"stable" relationship means there is neither positive nor negative
evolution.  Do not provide any explanation.  Evolution type:
```

Prompt 6.3.5: Relationship and Evolution Type Prediction Prompt for Complete Summary

```
System Prompt:  You are a helpful assistant who follows the
instructions.  No preambles and postambles.  Avoid explanations if
not asked explicitly.

Prompt:  Based on the given summary that depicts the evolution in the
relationship between char_a and char_b until a point in a book answer
the below questions about the type of relationship between {char_a}
and {char_b} and its evolution at the end of the summary.

Summary:
---
{summary}
---

Are {char_a} and {char_b} mentioned in the provided summary?  Answer
in one word <ANS> ["yes", "no", "unsure"] </ANS>.
{Model's output}

Can you infer any type of relationship between {char_a} and {char_b}
from the summary?  Answer in one word <ANS> ["yes", "no", "unsure"]
</ANS>
{Model's output}

Choose the type of relationship between {char_a} and {char_b} from
these options:  acquaintances, strangers, friends, best friends,
romantic interest, dating, engaged, married, separated, divorced,
enemies, spouse, ex-spouse, one-sided romantic interest, ex-romantic
interest, others or cannot be determined.  Answer only from the
provided options.  Relationship type:
{Model's output}

Is the chosen relationship between {char_a} and {char_b} evolving
"positively", "negatively", is "stable" or "nothing can be determined"
by the end of the summary?  A "positive" evolution can result from
deepening connection, increasing trust, support or respect, spending
more time together etc.  A "negative" evolution means any tension or
straining relationship that can result from conflicts, arguments,
distrust, disrespect, lack of support, or misunderstandings.  A
"stable" relationship means there is neither positive nor negative
evolution.  Do not provide any explanation.  Evolution type:
```

Prompt 6.3.6: Relationship and Evolution Type Prediction for Summary with Passage

```
System Prompt:  You are a helpful assistant who follows the
instructions.  No preambles and postambles.  Avoid explanations if
not asked explicitly.

Prompt:  Based on the summary of the evolution in type and nature
of the relationship between {char_a} and {char_b} until a point in a
book and the following segment answer the below questions about the
type of relationship between {char_a} and {char_b} and its evolution
by the end of the provided segment.

Summary:
---
{summary}
---

Segment:
---
{segment}
---

Are {char_a} and {char_b} mentioned in the provided segment?  Answer
in one word <ANS> ["yes", "no", "unsure"] </ANS>.
{Model's output}

Can you infer any type of relationship between {char_a} and {char_b}
from the segment?  Answer in one word <ANS> ["yes", "no", "unsure"]
</ANS>
{Model's output}

Choose the type of relationship between {char_a} and {char_b} from
these options:  acquaintances, strangers, friends, best friends,
romantic interest, dating, engaged, married, separated, divorced,
enemies, spouse, ex-spouse, one-sided romantic interest, ex-romantic
interest, others or cannot be determined.  Answer only from the
provided options.  Relationship type:
{Model's output}

Is the chosen relationship between {char_a} and {char_b} evolving
"positively", "negatively", is "stable" or "nothing can be determined"
from the segment?  A "positive" evolution can result from deepening
connection, increasing trust, support or respect, spending more time
together etc.  A "negative" evolution means any tension or straining
relationship that can result from conflicts, arguments, distrust,
disrespect, lack of support, or misunderstandings.  A "stable"
relationship means there is neither positive nor negative evolution.
Do not provide any explanation.  Evolution type:
```

6.3.3 Relationship and Evolution Prediction

Given the input for each of the described strategies, we iteratively prompt an LLM to first determine the relationship type and then the evolution type for the chosen relationship in a zero-shot setting. In addition to the predefined set of relationship and evolution types in §6.1, we also allow LLMs to predict *cannot be determined* for both the tasks and *others* for the relationship type to cover instances when a relationship type may be determined but is not provided in the predefined set. We use the Prompt 6.3.4, Prompt 6.3.5, and Prompt 6.3.6 to get the relationship and evolution type predictions from an LLM for the *Complete Passages*, *Complete Summary*, and *Summary with Passage* strategies, respectively. Iteratively asking for relationship type and then evolution type helps in presenting only the required question and information to an LLM and makes it easier to parse the output.

6.4 Experimental Setup

We provide details on the source of dataset, preprocessing steps, predictor and summarizer LLMs.

6.4.1 Dataset Source

While many books are available on resources like Project Gutenberg[6] (Stroube, 2003), LLMs have memorized them along with their summaries available on online sources as study guides[7] (Chang et al., 2023). Using these books might result in data contamination therefore, we use 11 books (published in 2023, and novel text is not freely available online) collected by Chang et al. that are less likely to be memorized by LLMs used in this work.[8]

[6] https://www.gutenberg.org/

[7] https://www.sparknotes.com/lit/

[8] We manually prompted LLMs for details on metadata such as main characters given the name of the novel and the author however, the generated answers were incorrect. This indicates that even if the model

We select books from the romance genre as they frequently use tropes (*e.g.*, enemies-to-lovers, friends-to-lovers, second chance, and forbidden love) with evolving relationships between the main characters to make the story interesting. We manually refer to online reading forums such as Goodreads[9] to obtain the specific trope depicted in the selected books and their main characters. We use this information for global-level evaluation of the predicted trajectory for a book (§6.7). We perform experiments for a pair of main characters per book however, the proposed strategies are agnostic to the pair of characters and can be used for any two characters in theory.

6.4.2 Preprocessing Books

We preprocess books using BookNLP (Bamman et al., 2014b) library[10] to get coreferences for characters in a book. We first divide the book text into non-overlapping passages of human-readable length ($100 - 200$ words). Then, replace the first occurrence of any third-person pronouns used as subject with a representative alias for a character. The most frequently used proper noun for a character is considered the representative alias for that character. We do such a replacement to ensure the comprehensibility of a standalone passage. We refer to the above process as **coreference substitution**. We obtain 644 ± 104 passages per book, of which 98 ± 128 passages have both main characters mentioned in them. Huge variation is due to differing author writing styles. We do not perform any coreference substitution for the complete passages strategy since prior passages are provided as-is and as per centering theory (see Section 2.1) coreferences are used to maintain local coherence (Grosz et al., 1995). Therefore, when all the passages are provided as-is to a language model, we believe that the model has access to salient characters and coreferences used for it and leave any coreference resolutions to the implicit understanding of

would have seen the novels it does not remember the details.

[9] https://www.goodreads.com/

[10] https://github.com/booknlp/booknlp

these models. This allows us to assess their true human-like reasoning abilities.

6.4.3 Summarizer and Predictor Models

We use open-sourced LLMs from three familiesnd it should be able to resolve pronouns by itself., namely, Llama3.1-8B-chat (Dubey et al., 2024), Mistral-7B-Instruct (Jiang et al., 2023), and Gemma2-9B (Team et al., 2024), to obtain the iterative summaries and predict relationship and evolution type for the *Summary with Passage* and *Complete Summary* strategies. However, for the *Complete Passages* strategy, we use Llama3.1-8B-chat with a maximum of $30K$ context window size. We generate summaries of a maximum 300 words given passages of a maximum 200 words. Recall that the summaries are generated iteratively. Therefore, the summary length is more than the passage length to capture any new relationship relevant information in the new passage for updating the previous summary. We use nucleus sampling (Holtzman et al.) with radius of $p = 0.9$ and top $k = 50$ tokens for generating summaries and greedy decoding for the prediction tasks.

6.5 Evaluation Without Gold Labels

One of the major challenges of tracking evolution in the relationship between characters is the unavailability of gold labels and the difficulty in collecting crowd-sourced annotations due to the length of the books; making it extremely expensive, and cognitively challenging. We make a novel contribution by providing insights into the feasibility of this task without gold labels by analyzing the agreement between predictions from multiple LLMs. Additionally, we conduct a quantitative (§6.8.1) and qualitative (§6.8.2) manual analysis of the predictions.

Owing to the increasing use of LLM-as-evaluators (Chan et al.; Gu et al., 2024) and LLM-as-annotators (Chiang and Lee, 2023; Tan et al., 2024), we hypothesize that if mul-

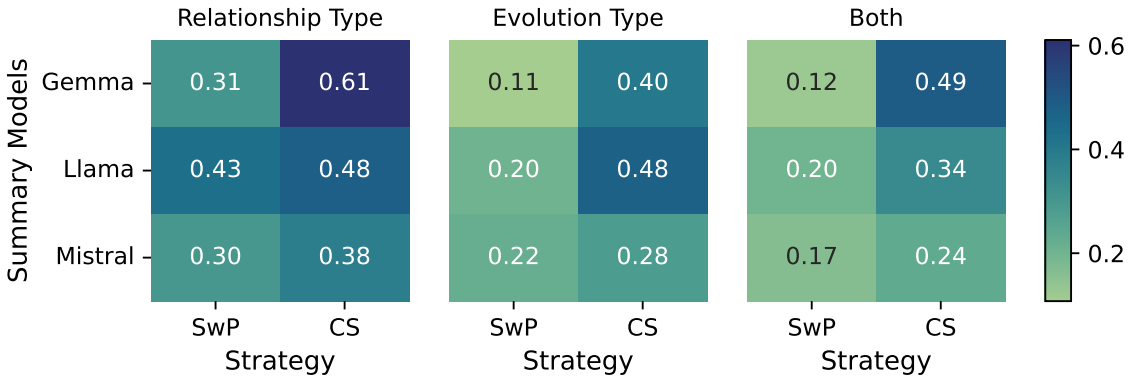


Figure 6.3: Krippendorff's alpha between different predictions for *Summary with Passage* (SwP) and *Complete Summary* (CS) strategies using summaries generated from various summary models.

tiple LLMs agree on a label then it is more likely to be the gold label. Therefore, we thoroughly analyze the quality of the consensus predictions from multiple LLMs (conditioned on the input) via Krippendorff's alpha (α), and compare it against the agreement between human annotators. We also use α scores to compare agreement between different strategies (Table 6.3 and Figure 6.4), and preprocessing methods (Figure 6.5 and Figure 6.6). We manually examine the consensus predictions at a global- and local-level to provide an upper bound on the performance of LLMs for this task. **Global-level evaluation** measures the accuracy of correctly predicting the trope given the predicted trajectory of evolution in the relationship between two characters for a book. For **local-level evaluation**, a subset of examples per book per trope is manually annotated and compared against the consensus predictions. We report scores separately for **relationship** and **evolution type** as well as when **both** are considered together.

6.6 Findings

Agreement between prediction models. We report the agreement (α scores) between predictions from different LLMs for *Summary with Passage* and *Complete Summary* strate-

Strategy	Relationship Type	Evolution Type	Both
Summary with Passage	0.24	0.20	0.13
Complete Summary	0.22	0.08	0.11

Table 6.2: Agreement between predictions across summary and prediction models.

Strategy	Relationship Type	Evolution Type	Both
Summary with Passage	0.50	0.44	0.38
Complete Summary	0.47	0.12	0.17

Table 6.3: Agreement between predictions from *Complete Passages* and consensus predictions for *Summary with Passage* and *Complete Summary* strategies.

gies that use summaries generated from various summary models in Figure 6.3. While this analysis depends on the correctness of the summaries generated from LLMs, we do not explicitly evaluate their correctness as LLM-generated summaries have been shown to be on par with human-written summaries (Goyal et al., 2022; Zhang et al., 2024). Low-to-moderate agreement suggests that tracking evolution in the relationship is a challenging task. Lower agreement for evolution type than relationship type emphasizes the difficulty of predicting fine-grained evolution. We observe higher agreement between predictions for *Complete Summary* than *Summary with Passage* as summaries contain more direct evidence of the type and nature of the relationship whereas in the presence of a more granular passage, the evidence needs to be inferred. Interestingly, agreement between predictions across all the summary models is higher for *Summary with Passage* than the *Complete Summary* strategy (see Table 6.2). This indicates that using a passage acts as a regularizer for mitigating the effect of differences in summaries generated from different summarizers.

***Summary with Passage* is a better approximation of *Complete Passages* than the *Complete Summary* strategy.** To analyze which strategy – *Summary with Passage* or *Com-*

Strategy	Relationship Type (α)	Evolution Type (α)	Both (α)
Summary with Passage	86.04 (0.70)	44.18 (0.26)	39.54 (0.31)
Complete Summary	88.37 (0.79)	70.93 (0.58)	67.44 (0.61)

Table 6.4: **Local-level evaluation:** Accuracy of consensus predictions averaged across annotations from humans. Scores in the parenthesis denote the agreement (α) between human annotators.

plete Summary – using a short context window best approximates *Complete Passages* that uses a long context window, we report agreement between the predictions from *Complete Passages* strategy and the consensus predictions across all the summarizers and predictors for the two shorter-window strategies in Table 6.3. Combining a passage with a prior summary strikes a good balance in providing the information necessary for the task, compared to using the complete summary. This supports the regularization aspect of using a passage, as seen in Table 6.2.

6.7 Manual Evaluation

How accurate are the consensus predictions from LLMs? To study the upper bound performance achievable from the proposed strategies, two annotators label the relationship and evolution type conditioned on the input for different strategies. For local-level evaluation, we first select one book per trope that attains the highest prediction agreement over all the predictors and summarizers across different strategies. Then, we randomly sample a maximum of 20 passages (where both the characters are mentioned) per selected book and consensus predictions from the summarizer with the highest agreement[11] for each strategy. Passages are selected such that the consensus predictions from different strategies are different as it has a two-fold benefit: (1) the accuracy of prediction for the sampled passages

[11]Summaries from Llama resulted in higher agreement between predictions as compared to that from Gemma or Mistral.

Strategy	Summarizer	Accuracy (%)
Majority	N/A	54.54
Complete Passages	N/A	63.64
Summary with Passage	Llama	27.27
	Mistral	18.18
	Gemma	18.18
	Overall	21.21
Complete Summary	Llama	9.09
	Mistral	9.09
	Gemma	0.00
	Overall	6.06

Table 6.5: **Global-level evaluation**: Percentage accuracy for predicting the trope of a book from the predicted trajectory from various strategies.

acts as a good approximation of overall accuracy as the two strategies will have the same accuracy for the same predictions, and (2) it makes it easier to qualitatively compare the two strategies (as shown in Table 6.7). We report the accuracy of prediction averaged over the two annotators and α (in parenthesis) between them in Table 6.4. We observe that agreement entails accuracy for both strategies. A similar agreement between humans, as observed for LLMs (see Figure 6.3), indicates that while relationship type can be determined with high agreement, evolution type prediction is challenging for humans as well.

How accurately can trope be identified from the LLM predicted trajectory? For global-level evaluation, we present a visualization (similar to Figure 6.1) of the trajectory of evolution in the relationship between two characters to annotators[12] and ask them to select the best applicable trope out of enemies-to-lovers, friends-to-lovers, second-chance, and forbidden love.[13] We keep the book and characters' names anonymous to the annotators to

[12]Manual examination is done internally by people who frequently read novels.
[13]We also provide "cannot be determined" and "others" as options.

ensure no use of online resources. We provide visuals obtained from the predictor having the highest agreement with the consensus predictions for each strategy and summarizer.[14] As mentioned in §6.4.1, we have gold trope labels for each book to compute the accuracy of trope prediction.

Table 6.5 shows that trope can be predicted with the highest accuracy when all the passages are provided to an LLM with a large context window (*i.e. Complete Passages*). Higher accuracy for summaries from Llama indicates that it has a better understanding of social relationships than Gemma or Mistral. While we see a higher local-level accuracy for *Complete Summary* than *Summary with Passage* strategy (Table 6.4), we observe a reverse trend for trope prediction. This shows that an overall summary is unable to capture the fine-grained details; aligning with our previous finding that *Summary with Passage* is a better approximation of *Complete Passages* than *Complete Summary* (see Table 6.2). Lower accuracy than a majority baseline emphasizes the difficulty of this task when information is not provided at the highest granularity.

6.8 Analysis and Discussion

We present an ablation study of *Summary with Passage* strategy, need for coreference substitutions, and intermediate passages in §6.8.1 and shed light on the challenges with this task in §6.8.2.

6.8.1 Quantitative Analysis

Evolution type is determined (mostly) based on the passage whereas relationship type is (majorly) influenced by the previous summary. To analyze the source of information used to predict the evolution and relationship types in the *Summary with Passage* strategy,

[14] 11 visualizations per strategy per summarizer.

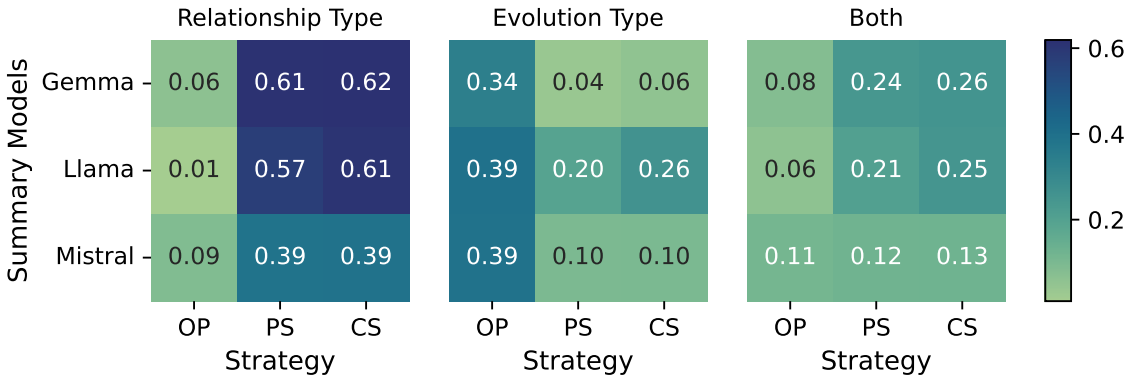


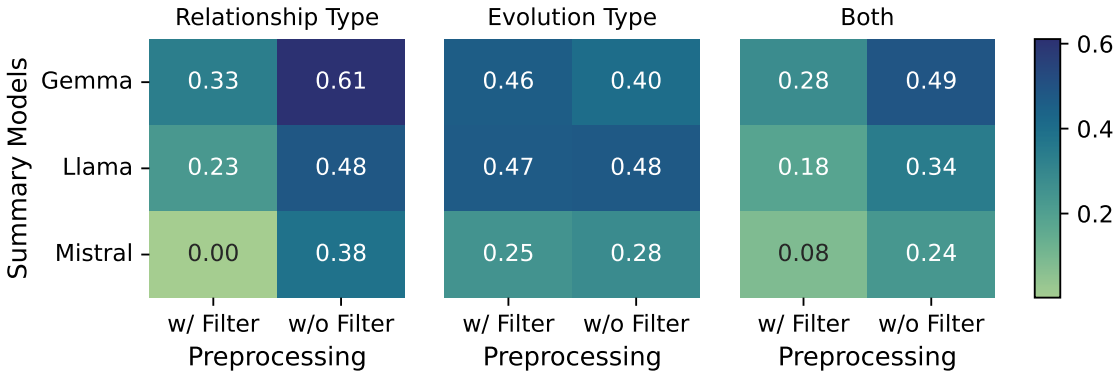
Figure 6.4: Krippendorff's alpha between consensus predictions from *Summary with Passage* and *Only Passage* (OP) or *Previous Summary* (PS) ablations. Agreement with *Complete Summary* (CS) is shown to emphasize its difference as opposed to *Previous Summary*.

Strategy	**Relationship Type** (α)	**Evolution Type** (α)	**Both** (α)
Only Passage	59.43 (0.34)	49.05 (0.33)	39.62 (0.23)
Previous Summary	82.95 (0.59)	53.41 (0.40)	45.45 (0.32)
Summary with Passage	72.64 (0.72)	41.51 (0.24)	33.02 (0.38)

Table 6.6: Accuracy (%) of consensus predictions averaged across annotations from humans annotators. Scores in the parenthesis denote the agreement (α) between human annotators.

we perform an ablation study by using only the passage or the previous summary and compare the consensus predictions with that from using both the passage and the previous summary. Higher agreement (Figure 6.4) between the *Previous Summary* and *Summary with Passage* strategy than *Only Passage* for relationship type shows that LLMs rely on information in the summary for relationship type prediction. However, the evolution type predictions are determined based on the provided passage. As expected, agreement between predictions from the complete summary is higher than that from the previous summary since it contains more information. Additionally, we ask two annotators to label a subset of examples selected in the same way as in §6.7 for local-level evaluation and report the results in Table 6.6. Low agreement between annotators (in parenthesis) except for relationship predictions from *Summary with Passage* strategy shows that this task is difficult

(a) Summary with Passage



(b) Complete Summary

Figure 6.5: Krippendorff's alpha between consensus predictions from *Summary with Passage* and *Complete Summary* which use summaries generated from passages with (w/ Filter) and without filtering (w/o Filter) the passages that do not mention both the characters.

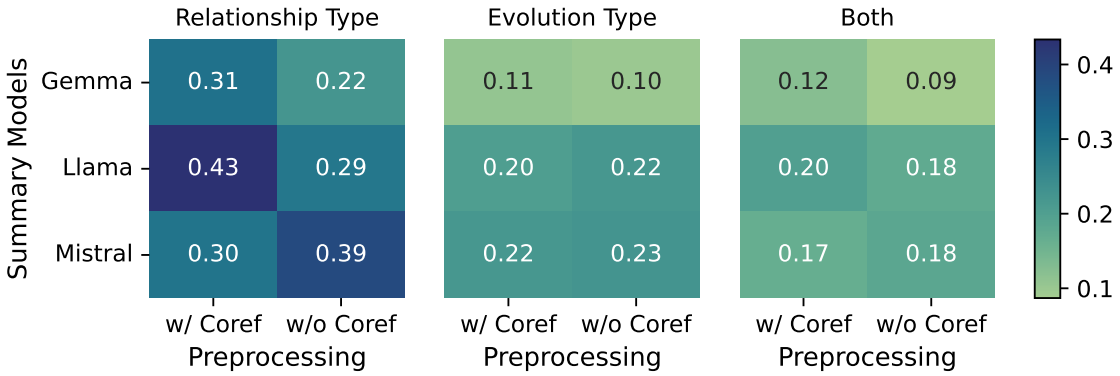
even for humans. This is due to the involved subjectivity leading to different annotations (see Table 6.8 for examples). Accuracy for relationship predictions for *Only Passage* is much lower than that for *Summary with Passage* due to the possibility of multiple interpretations when a passage is provided out-of-context. However, evolution prediction is less accurate when a previous summary is provided with a passage. This may happen when relationships are in a transition phase, or characters may have different emotional states toward each other. Since a summary captures all this information, it may be difficult to infer an evolution type with certainty. We discuss such examples in §6.8.2.

Intermediate context is a useful source of information. Prior studies (Chaturvedi et al., 2016; Iyyer et al., 2016b; Chaturvedi et al., 2017) that model evolution in relationships have focused solely on passages where both characters are mentioned. In contrast, we hypothesize that interactions between other characters or one of the main characters with others is a useful source of information for this task. To test this, we employ the same strategies as described in §6.3 but only use the passages where both the characters are mentioned and compare the obtained agreements with those when passages without both character mentions are not filtered. Figure 6.5 indeed shows that intermediate context results in higher agreement for relationship predictions when passages are not filtered than when filtered. However, we do not observe significant improvement for evolution prediction.

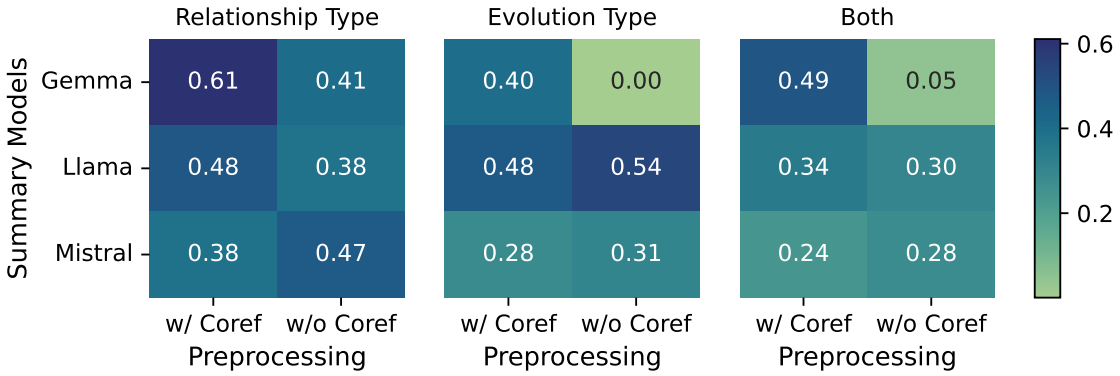
Coreference resolution results in (mostly) higher or comparable agreement between predictions than without it. We apply the *Summary with Passage* and *Complete Summary* strategy on passages from a book without coreference substitutions (described in §6.4.2) to analyze its impact on the agreement between predictions as shown in Figure 6.6. While we mostly see a higher agreement between predictions when coreference substitution is done during data preprocessing (w/ Coref) than in its absence (w/o Coref), we also see instances of lower or comparable agreement. Manual analysis reveals that in the absence of a specific character mention, LLMs tend to assume that the pronouns refer to the characters under study both during summary generation, and relationship and evolution prediction. This is a result of the widely acknowledged issue of hallucination (Ye et al., 2023; Huang et al., 2023) and context sensitivity (Min et al., 2022) in LLMs.

6.8.2 Qualitative Analysis

Maintaining a running summary helps resolve the ambiguity between different relationship types. We provide qualitative examples in Table 6.7 for cases when the current



(a) Summary with Passage



(b) Complete Summary

Figure 6.6: Krippendorff's alpha between consensus predictions from *Summary with Passage* and *Complete Summary* with (w/ Coref) and without substituting the character coreferences (w/o Coref) in the passages.

passage does not contain enough information to make a prediction or the prediction is correct given the passage but changes when the previous story is considered. Providing a previous summary helps propagate the prior knowledge about the relationship that can make a models' predictions more certain, and resolve any misinterpretations.

Uncertainty in relationship or evolution type prediction results in disagreements between humans. Table 6.8 presents a few qualitative examples showcasing uncertainty of the task and different interpretations of the context resulting in disagreement between humans. The first example depicts a scenario when the relationship is in a "transition/devel-

Summary: *Mina stands at a crossroads, torn between her village and the allure of the open sea, driven by her love for Shin. Her memories reveal the early stages of their romance, suggesting her love for him may be a choice rather than a predetermined fate. As Mina's devotion to Shin reaches a boiling point, she breaks the Sea God's three rules, and Joon's concern for her safety demonstrates the strong bond between them. . . . She encounters Shin, who looks at her with longing in his eyes, breaking her heart. Shin's words of encouragement, " "Don't chase fate, Mina. Let fate chase you," remind her to find her own path and destiny. Mina confesses her feelings to Shin, and he reciprocates, stating that he doesn't need the Red String of Fate to know that he loves her. They share a passionate kiss, and Shin says, "Lord Crane was mistaken. He said once the Red String of Fate was formed, you would know how to break the curse."* **Passage:** *Namgi says. He leans back, and I get a good look at his face. There's joy there, and wonder. "We know everything, about the emperor, about the Sea God. Shin is the Sea God! Can you believe it?" " Where is he?" I ask. "In the hall. We arrived right before you." Kirin approaches from behind Namgi, his always astute eyes watching me carefully. "What were you saying, Mina? That you wouldn't see us before...?" I release Namgi, stepping back.* **Predictions:** cannot be determined, romantic interest
Summary: *Jana's relationship with Anil began with a transformative experience, marked by a week and a half of intense physical connection. . . . As they traveled together, Jana and Anil grew closer, visiting Tajikistan . . . In London, they transitioned from traveling companions to intimate partners, engaging in a hard-and-fast fling amidst their days of attending meetings and nights in a tiny hotel room. . . . Their connection deepened, and they found themselves lost in intimate moments, discussing their work in the development field and goals of bringing grassroots-style microdevelopment to a larger scale. dots Jana's trust in Anil was shaken when she received anonymous messages accusing her of sleeping with a married man, claiming Anil was still married to the sender's sister. . . .* **Passage:** *Jana knew she was falling in love with him. And maybe love was a little blind. She picked up her phone again, not to call Anil, but to message Rasheed, the manager of the project in Tajikistan. The one who Anil had been visiting when this relationship started. Jana: Rasheed, is Anil married? It was the middle of the night in Tajikistan. Jana wasn't expecting an answer. But he responded. Rasheed: Have you asked him that question?* **Predictions:** one-sided romantic interest, romantic interest

Table 6.7: Sample relationship predictions depicting importance of using a previous summary which helps resolve uncertain predictions, and propagate prior context to avoid misinterpretation from just the passage. Color denotes the predictions and evidences from *Only passage* and *Summary with Passage* strategy.

oping" phase. The second example presents another scenario where evolution is different from each character's perspective due to unresolved issues in one's mind. The remaining examples showcase different interpretations of annotators in the absence of proper context leading to a disagreement.

Failure cases We provide examples (see Table 6.9) to depict some failure cases where LLMs potentially rely on surface-level cues or may adopt heuristics, tend to resolve pro-

Summary: *Jana's relationship with Anil began with a transformative experience, marked by a week and a half of intense physical connection. . . . However, Jana's trust in Anil was shaken when she received anonymous messages accusing her of sleeping with a married man, claiming Anil was still married. Anil's revelation that he was indeed married marked the end of their relationship, and Jana became pregnant with his child. Years later, . . . Anil moved to Toronto to be close to Imani, and as co-parents, Jana and Anil discussed their complicated family dynamics. Jana expressed concerns about becoming overprotective, while Anil demonstrated a willingness to be involved in Imani's life. Jana struggled with her past and her growing closeness to Anil, confronting him about his past behavior and accusing him of trying to stroke her ego. . . . As they reconnect, Jana questions whether she is letting other people's opinions get in the way of her own happiness, particularly in regards to her relationship with Anil. Jana is starting to inch out of her comfort zone, and now considers a date with Anil, suggesting a possible rekindling of their relationship.*
Passage: *Did you forget what happened at Hatari? Jana asked. Anil chuckled low, sending a shiver down Jana's spine. . . . "I've honestly thought of little else," he said. And there it was. He'd been as preoccupied with thoughts of that night as she'd been. Jana could feel a heat burning inside her. He still wanted her.*
Discussion: Jana and Anil are ex-romantic partners who are co-parenting their daughter and are considering to give another chance (romantic interest) to their relationship.

Summary: *Jana's relationship with Anil began with a transformative experience, marked by a week and a half of intense physical connection. . . . However, Jana's trust in Anil was shaken when she received anonymous messages accusing her of sleeping with a married man, claiming Anil was still married. Anil's revelation that he was indeed married marked the end of their relationship. Jana became pregnant with his child, and they navigated a co-parenting agreement with the help of a family lawyer. Years later, Anil surprised Jana . . . revealing he wanted to surprise their daughter Imani . . . This gesture suggested a desire to reconnect with Jana and their daughter. . . . Their recent interactions highlighted the challenges they still face, including Anil's condescending behavior and Jana's lingering discomfort. . . . Jana's hesitation stems from her need for self-care, as being around Anil triggers memories and makes it difficult for her to think straight.*
Passage: *Jana couldn't travel for two weeks alone with just him and Imani. His betrayal still hurt too much. She could finally take a trip with Imani, and this would taint it. "Come on, Jana," Anil said. "I don't want to break your mother's heart, either. She's so great for Imani."*
Discussion: Jana and Anil are ex-romantic partners as well as co-parents. While Anil is putting in efforts to reconnect with Jana and his daughter, Jana has unresolved emotions and is hesitant resulting in a positive evolution in the relationship from Anil's side however, still negative from Jana's perspective.

Passage: *Kirin strides in, bowing low. His keen eyes glance at Shin's hand, still holding my own. "You called for me?" "Mina's been hurt." "Ah, I see." I frown at the two of them, the unspoken words thick in the air. Why had Shin asked for Kirin and not a physician? As Shin releases my hand, Kirin reaches inside his robes and pulls out a small silver dagger. . . . I only have a moment to gape before he grabs my wrist, placing his now bloodied hand over my burned one.*
Dicussion: Here, "holding hands" was interpreted as showing care as *romantic interests* by one annotator while from another's perspective it was considered as a gesture any *friend* would do if someone is injured.

Passage: *It was the cutest thing Jana had ever seen. She lifted Imani in her arms to see it better. Everyone in the Land Cruiser was in awe, giddy with excitement. As the drive continued, they saw gazelles, the most vibrant striped zebras yet, and some giraffes. But Jana understood why this was called the elephant park. There were so many elephants. On their own, in herds, at the watering hole , everywhere. Anil kept pointing out new ones, and Imani eventually stopped counting (she really couldn't get past forty, anyway).*
Discussion: As Jana and Anil are spending time together one annotator considered it as a positive evolution however, for another, there was not enough information to determine evolution type from the passage.

Summary: *Jana's relationship with Anil began with a transformative experience, marked by a week and a half of intense physical connection. . . . Their connection deepened in London, . . . However, Jana's trust in Anil was shaken when she received anonymous messages accusing her of sleeping with a married man, claiming Anil was still married. Anil's revelation that he was indeed married marked the end of their relationship. Jana became pregnant with his child, and they navigated a co-parenting agreement . . . This gesture suggested a desire to reconnect with Jana and their daughter. . . . Now, Anil is considering Jana for a director of research and programs position, creating a delicate situation for Jana. She must navigate her past anger and work with Anil to co-parent their daughter effectively.*
Passage: *And now she had two weeks with Anil, this was the perfect time to put it all behind her for Imani's well-being and her own. It was time to show everyone, including Anil Malek, that the last five years hadn't broken Jana. Most of all, it was time for Jana to show herself that. . . . It was a revised wedding schedule. Jana assumed Elsie had rearranged everything so Dr. Lopez wasn't in the same activities as Mom, Jana, Anil, or Imani.*
Discussion: While the evolution is negative as per one annotator due to 'creating a delicate situation for Jana', nothing 'can be determined' from another's perspective as not there is no direct interaction between Anil and Jana in the passage but it mostly mentions Jana's feelings.

Table 6.8: Examples where relationship and evolution prediction may be uncertain and open to interpretation leading to disagreement between annotators.

nouns to the character in question in the absence of an explicit mention, and are sensitive to subtle changes in the context (such as substituting pronouns for other characters in the context). Such behavior raises questions on the *true* understanding of evolving social relationships in LLMs and if they are right for the wrong reasons.

6.9 Summary

In this chapter, we formally define the task of tracking evolution in relationships in terms of predicting a trajectory of predefined relationship and evolution types at different points in a book. We investigate LLM's social reasoning capabilities by proposing strategies that differ in the granularity of information provided to the LLM. In the absence of gold annotations for this task, we thoroughly analyze the agreement between predictions from different families of LLMs for the proposed strategies. We find that maintaining a running summary of the type and nature of evolution in the relationship between the characters along with a passage is a better approximation of a strategy that uses all the passages until a point as opposed to providing a complete summary. Ablation analysis reveals that the relationship type predictions get propagated from the previous summary while evolution type predictions are more local and are based on the provided passage. Through quantitative and qualitative analysis of the predictions from LLMs and annotations from human annotators, we show that tracking evolution in the relationship is challenging for both LLMs and humans as reflected by low-to-moderate agreement between their predictions. Human disagreement can be attributed to their differing interpretations due to the multi-faceted nature of relationships. Further analysis reveals that LLMs make conservative predictions, lack a proper understanding of the context required for this task, and are sensitive to subtle changes in the provided context.

As LLMs may adopt heuristics and surface-level cues for predicting the evolution in

the relationship between characters, it is unclear if they have a *true* understanding of social relationships. Furthermore, relationships may be predicted based on several causal factors such as the age and gender of the characters, tone of the speakers, location or setting of the story, and contextual appropriateness of a statement. To investigate if LLMs resort to such spurious correlations for predicting relationships, in the next chapter, we study the impact of changing the gender and race associated with the characters involved in a conversation on the relationship predictions from LLMs.

Passage	Discussion
Heuristics/Surface-level cues	
In all sincerity, the lesson here is for me to never doubt Fizzy, Connor says, and the audience Awwwwws. "But listen," Lanelle says. "The two of you really had an amazing connection on-screen." Unease thrums beneath my skin. I don't want her to put Connor on the spot like this. "A corpse would have chemistry with this man, Lanelle. Be serious. "The Connor fangirls in the audience scream." No, no, this is something special.	LLM predicts romantic interest between Connor and Fizzy may be due to an incorrect understanding about on-screen vs real connection.
Connor was trying to talk it out with you, Jess says over the steaming top of her mug. I don't need reminding. Every regrettable, overreactive moment of my meltdown is imprinted in my brain like a bad, drunken tattoo. . . . "I know he was. And I know this all happened like eight years ago, and he was upset, and he's older and wiser, but the fact that he decided to not just end his marriage but explode it..." "Fizzy, we are all dumb when we're young. . . .	LLM predicts Fizzy and Connor as ex-spouses potentially due to surface-level cues and improper understanding of who is speaking to/about whom.
Nothing can console him. "My heart is breaking." Why are you telling me this? "Because, as you suggested, I've taken on the role of the Goddess of Women and Children. Do you know what that means?" I shake my head. "It means that everyone who once feared me now loves me. Even Shin, my greatest enemy, loves me. He knows me now as a goddess of motherhood and children. He knows me as a goddess who is loving and kind and giving. Tell me, Mina, how could I be cruel to someone who loves me?" "I do n't know. Can you?" "It's... strange. When I was feared, I hated everything and everyone."	LLM predicts that Mina and Shin are enemies due to misinterpretation of 'me' as 'Mina' instead of the Goddess which shows that it ignores the context and resorts to surface-level cues.
Incorrect (assumed) pronoun resolution	
It's a barely restrained Uh, okay, buddy. It's a laugh held in. Connor's smile remains, but it doesn't look totally natural anymore. "Do you read her books? " Ashley shakes her head. " Oh, I don't read books with just romance in them; I need there to be some plot, too. "Fizzy goes quietly stony." There's plenty of plot. And Fizzy's are the gold standard. "I stare up at him with fondness. This liar, still pretending he's read my books.	LLM assumes "I" to refer to Fizzy resulting in romantic interest prediction between Fizzy and Connor.
So, he says, and smiles shyly over at me in a way that acknowledges how heavy things just got, how there is something hot and tangible in the air between us but maybe if we talk over it, it will dissipate. "You ready for tomorrow?" Inhaling sharply, I sit up straighter. Right. Get yourself together, Fizzy. "I am. I hope I can sleep tonight. I really don't want to show up all puffy and shadowed tomorrow." "I was going to say," he says, smiling, "you've appeared very calm for someone who's about to be on television."	LLM predicts *romantic interest* between Fizzy and Connor even though Connor is not mentioned in the passage.
Sensitivity to irrelevant changes	
"Where is she?" Mom frowned. "Where did you sleep?" Jana rummaged through her bag to get clothes. "Imani's with Anil. They're fine. Everything is fine. I'm just going to take a shower." "But where did you sleep?" Mom asked again. Jana did not want to answer the question. She did not want to say she slept with Anil Malek's arm around her. Or that he wasn't wearing a shirt. Or that they weren't sleeping at all early in the morning and were instead watching the most beautiful sunrise Jana had ever seen and maybe thinking about kissing.	Evolution prediction between Jana and Anil changes from *positive* to *cannot be determined* when *she* is substituted with *Imani*.

Table 6.9: Examples where LLM's predictions are incorrect due to potential reliance on surface-level heuristics, the tendency to resolve pronouns to the character in question in the absence of an explicit mention, and sensitivity to irrelevant changes in the context.

Chapter 7: Investigating the Influence of Implicit Character Attributes on Relationship Predictions from Large Language Models[1]

As observed in the previous chapter, identifying relationships between characters from a given story presents a challenging task in natural language understanding Jia et al. (2021); Tigunova et al. (2021). This is because relationships are often expressed indirectly through actions and inferred emotional connections. Conversations between characters can have multiple interpretations depending upon the context (*e.g.* a sarcastic remark may imply a friendship or animosity), and contextual appropriateness (*e.g.* it is appropriate to say *you look beautiful* to a friend but may not to a colleague) (Jurgens et al., 2023) which requires an understanding beyond the literal meaning of the text.

Relationship predictions may also be influenced by several causal factors such as the perceived gender, age or race of the characters, tone of the characters in a conversation, and the location of the story. Further, language models are sensitive to subtle changes (Min et al., 2022) and the presence of multiple characters in the provided context. Therefore, they may adopt some heuristics for performing the task while ignoring the context (McCoy et al., 2019). In this chapter, we systematically investigate the influence of such causal factors on relationship predictions to study if language models have a *true* understanding of social relationships or if they are right for the *wrong* reasons (McCoy et al., 2019; Zhou et al.,

[1]Abhilasha Sancheti, Haozhe An, and Rachel Rudinger. 2024. On the influence of race and gender in romantic relationship prediction from large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Miami, Florida, USA. Association for Computational Linguistics.

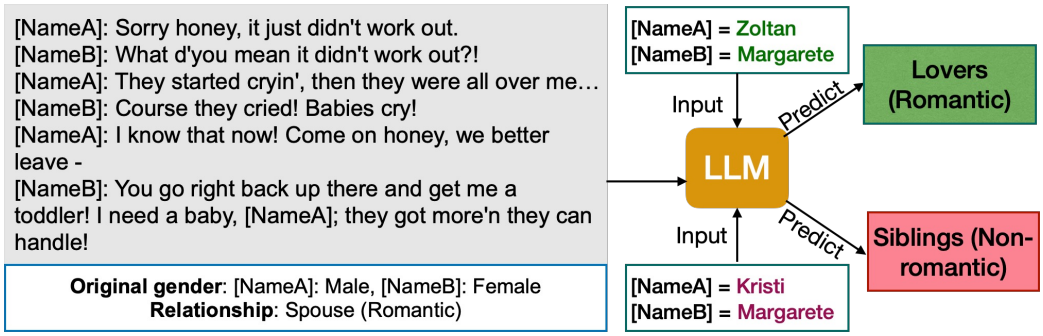


Figure 7.1: Sample conversation from DDRel (Jia et al., 2021) dataset and relationships predicted by Llama2-7B when characters are replaced by names with different-gender and same-gender. LLM tends to predict differently despite the same conversation.[2]

2024). To this end, we examine the impact of changing the gender and race of characters involved in a dialogue on the relationship predictions from LLMs. We specifically consider gender and race as these attributes are often inferred from the names and are easy to change in a controlled setting as opposed to other factors (such as tone of speakers and location of the story) where such changes are non-trivial.

Formally, we consider the task of predicting romantic relationships from dialogues in movie scripts to study the impact of gender and race associated with a pair of character names on the accuracy of predictions from LLMs. For instance, Figure 7.1 shows a conversation between a female and a male spouse pair, for which Llama2-7B predicts a romantic relationship when the names in the conversation are replaced with a pair of different-gender names, but predicts a non-romantic relationship when replaced by same-gender names. Ideally, name-replacement should not significantly alter the predictions of a model, as the utterance content plays a more substantial role in language understanding, despite the potential interdependence between utterances and original names. We hypothesize that predictions from language models are a consequence of heteronormative biases and prejudice against interracial romantic relationships. Such spurious correlations get ingrained in these models due to the under-representation of such relationships in the train-

Relationship Labels	Frequency	Romantic	#Gender Neutral
Lovers	182	✓	155
Courtship	15	✓	12
Spouse	57	✓	46
Siblings	15	✗	13
Child-Other Family Elder	13	✗	7
Child-Parent	39	✗	11
Colleague/Partners	70	✗	59
Workplace Superior-Subordinate	48	✗	24
Professional Contact	27	✗	10
Opponents	20	✗	11
Friends	95	✗	83
Roommates	21	✗	21
Neighbours	8	✗	7
Total	610	-	459

Table 7.1: Frequency of relationships in the test split of DDRel dataset (Jia et al., 2021).

ing corpus. Through controlled character name-replacement experiments (Figure 7.1), we find that relationships between (a) same-gender character pairs; and (b) intra/inter-racial character pairs involving Asian names are less likely to be predicted as romantic. These findings reveal how some LLMs may resort to spurious correlations ignoring the context. Further analysis of contextualized embeddings corresponding to the character names in the dialogues suggests that gender for Asian names is less discernible than non-Asian names. This suggests how LLMs stereotypically interpret interactions between people which can potentially be harmful and raises ethical concerns.

7.1 Experimental Setup

We define the following task. Given a conversation C which consists of a sequence of turns $((S_1, u_1), (S_2, u_2), \ldots, (S_n, u_n))$ between characters A and B, where $S_i \in \{S_A, S_B\}$ indicates that the speaker of an utterance (u_i, $i \in \{1 : n\}$) is either A or B, the task is to identify the relationship represented as a categorical label from a predefined set.

We carry out controlled name-replacement experiments by prompting LLMs (zero-

shot) to predict the relationship type between A and B given C.

Models We study Llama2 ({7B, 13B}-chat) (Touvron et al., 2023a) with its official implementation,[3] and Mistral-7B-Instruct (Jiang et al., 2023) using its huggingface library.

Dataset We use the test set of DDRel Jia et al. (2021) which consists of movie scripts from IMSDb, with annotations for relationship labels between the characters according to 13 predefined types Table 7.1 presents the frequency of each relationship type along with romantic and non-romantic categories used for the purpose of this study. We consider Lovers, Spouse, or Courtship predictions as romantic and the rest as non-romantic. For our experiments, we use 327 instances of the test set in which characters originally have different genders (manually annotated) because the test set has no dialogues between same-gender characters with the romantic label. As a consequence, we lack conversations that effectively depict interactions between same-gender partners. However, in cases where same-gender partners exhibit behavior similar to different-gender couples, our results (see Figure 7.2) indicate that LLMs tend to demonstrate heteronormative biases in the intersection of these interaction styles.

[3] https://github.com/facebookresearch/llama

Prompt 7.1.1: Relationship Prediction

```
System Prompt:  You are an avid novel reader and a code generator.
Please output in JSON format.  No preambles

Prompt:  Your task is to read a conversation between two people and
infer the type of relationship between the two people from the given
list of relationship types.

Input:  Following is the conversation between {char_a} and {char_b}.
{dialogue} What is the type of the relationship between {char_a}
and {char_b} according to the below list of type of relationships:
[Child-Parent, Child-Other Family Elder, Siblings, Spouse, Lovers,
Courtship, Friends, Neighbors, Roommates, Workplace Superior -
Subordinate, Colleague/Partners, Opponents, Professional Contact]

Constraint:  Please answer in JSON format with the type of
relationship and explanation for the inferred relationship.  Type
of relationship can only be from the provided list.
Output in JSON format:
```

Prompt Selection As LLMs are sensitive to prompts (Min et al., 2022), we experimented with several prompt formulations on the original data (test set) for accuracy, and selected the prompt (as shown in Prompt 7.1.1) resulting in the highest accuracy which was closest to scores reported by others Jia et al. (2021); Ou et al. (2024). We note that our prompt selection is done prior to running the name-replacement experiments.

Evaluation We compare the average recall of predicting romantic relationships across different gender assignments and races/ethnicities. We study recall as we hypothesize that any heteronormative and interracial relationship biases would manifest as low (romantic) recall for same-gender and interracial groups. For completeness, we also report the mean precision, F1, and accuracy scores.

Implementation Details Each model uses nucleus sampling (Holtzman et al.) with default parameters, a temperature of 0, and a maximum generation length of 512. We observe inconsistencies in the outputs predicted by LLMs despite clear instructions regarding for-

matting. We use regular expressions to extract the JSON outputs and the predictions from them. We consider invalid outputs (*i.e.*, non-pre-defined class) from LLMs as a separate class (invalid) for evaluation purposes across all experiments.

Each experiment over 327 test instances takes $\sim$ 30mins for Llama2-7B, $\sim$ 1hr for Llama2-13B, and $\sim$ 25mins for Mistral-7B. We ran 870 experiments per race (560 for Hispanic) for studying gender bias and 1600 experiments (400 per race-pair) for racial bias. For presenting the results, we first compute precision, recall, F1, and accuracy scores for each name-pair-replacement and report the average scores for each name-pair bin, and each race-pair for studying the influence of gender, and race associated with names, respectively.

7.1.1 Studying the Influence of Gender Pairings

We ask whether the models are equally likely to recognize romantic relationships for character pairs of varying gender assignments and if this behavior is the same across different races. We hypothesize that models are prone to heteronormative bias and are more likely to predict romantic relationships for contrastive gender assignments. To test this, we collect 30 names per race,[4] dividing them into 10 non-linearly segmented bins that cover gender-neutral names (shown in Figure 7.2) based on the percentage of population that has been assigned as female at birth. Detailed name inclusion criteria and data sources are elaborated in §7.5.1. We replace the original name-pair in each conversation with all pairs of distinct names per race.

As dialogues may reveal gender identities (*e.g.*, "sir", "ma'am", "father", etc.), we manually identify a subset (271 instances) where such explicit cues are absent (to the best of our judgment) to minimize gender information leakage and avoid explicit gender inconsistency between the dialogue and the gender associated with the replaced name. In these dialogues, gendered pronouns typically refer to a third person who is not part of the conversation. As

[4]Except for Hispanic wherein we did not get any names in $5-10\%$ bin and only 1 name in $25-50\%$ bin.

a result, they do not reveal the speakers' gender identity. However, pronouns can indicate the sexual orientation of a speaker (*e.g.*, "Betty: *You do love him, don't you?*"). Such cues, along with other implicit cues about gender identity that are harder to detect, may confound our analysis. However, our findings as discussed in §7.2 reveal that implicit cues are not a major confounding factor.

7.1.2 Studying the Influence of Intra/Inter-Racial Pairings

We examine whether the models exhibit prejudice against interracial romantic relationships when making predictions. We collect another set of 80 first names that are both strongly race- and gender-indicative, evenly distributed among four races/ethnicities and two genders (details described in §7.5.1).

We perform pairwise name-replacements using these 80 names for the 327 test samples to analyze the relationship predictions among different intra/inter-racial name pairs.

7.1.3 Anonymous Name-replacements

In addition to the above experiments, we perform two types of anonymous name-replacement experiments differing in whether both names are anonymized or only one.

Both Names Are Anonymized We substitute names with generic placeholders ("X" and "Y") to get a baseline where a model has no access to character names to test the hypothesis that models' tendency to associate gender with the names influences their relationship predictions. This experiment also reflects what can be inferred purely from the conversation without resorting to any heuristics or spurious correlations.

One Name Is Anonymized We substitute one name and keep the other anonymized to analyze the impact of one character's gender on romantic relationship predictions indepen-

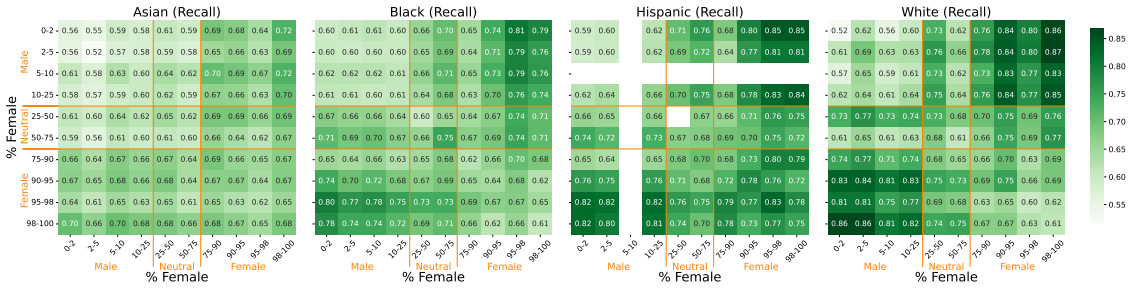


Figure 7.2: Recall of predicting romantic relationships from Llama2-7B for a subset of the dataset where characters originally have different genders. Horizontal and vertical axes denote % female of the name replacing an originally female and male character name from the dialogue. The upper-triangle (lower-triangle) shows the scores when names are replaced preserving (swapping) the genders of characters' names as-is in the original conversation. We consider the names with lesser % female as male names for determining gender preservation for name-replacement.

dent of the second. We replace one name with a male, female or a neutral name either preserving or swapping the original gender of the non-anonymized name while keeping the other name anonymized. Male, neutral, and female names belong to $0-25$, $25-75$, and $75-100\%$ bins, respectively.

7.2 Findings

Same-gender relationships are less likely to be predicted as romantic than different-gender ones. We observe a significant variation in recall of romantic relationship predictions from Llama2-7B (see Figure 7.2) for name-replacements involving different (top-right, and bottom-left)- versus same-gender pairs. This reveals that the model conservatively predicts romantic relationships when both the characters have names associated with the same gender (top-left – both male; bottom-right – both female). However, the precision across all races ranges between $0.78-0.84$ (see Figure 7.4 in §7.5.2). Such (relatively) low difference indicates that, while the model makes precise romantic predictions across all gender assignments and races, romantic predictions are more likely for contrastive gender

Model	Race	Male	Neutral	Female
Llama2-7B	Asian	0.6049/0.6128	0.6085/0.6203	0.6663/0.6517
	Black	0.6069/0.6230	0.6454/0.6392	0.6572/0.6458
	Hispanic	0.6292/0.6284	0.6486/0.6541	0.7093/0.6897
	White	0.6387/0.6372	0.6328/0.6297	0.6887/0.6761
Llama2-13B	Asian	0.2991/0.2940	0.2806/0.2798	0.3090/0.3043
	Black	0.3066/0.2854	0.3004/0.2909	0.3054/0.3105
	Hispanic	0.3021/0.2801	0.2956/0.2980	0.3206/0.3190
	White	0.3149/0.2952	0.2924/0.2878	0.3121/0.3121
Mistral	Asian	0.1789/0.1694	0.1808/0.1840	0.1895/0.1906
	Black	0.1855/0.1828	0.1902/0.1871	0.1922/0.1859
	Hispanic	0.1986/0.1955	0.1848/0.1776	0.2048/0.1973
	White	0.1895/0.1836	0.1887/0.1871	0.1942/0.1922

Table 7.2: Recall scores (same/swapped) for romantic relationship predictions when one name is anonymous while another is either a male, neutral, or female name as per bins marked in Figure 7.2. The results show that models are more likely to predict a romantic relationship when one of the names is a female name.

assignments.

Higher recall (Figure 7.2) for both female (bottom-right) replacements than both male (top-left) across all races indicates a potential **stronger heteronormative bias against both male than both female pairs**. This could potentially be an effect of associating female names with romantic relationships as indicated by higher recall for female-neutral than male-neutral pairs. We test this hypothesis using our one-name anonymized experiments. Recall scores in Table 7.2 indeed reflect that name pairs containing one female name tend to have higher recall than those containing one male name. This could either be due to a stronger association of female names with romantic relationships in general, or stronger heteronormative bias against male-male romantic relationships if models are (effectively) marginalizing probabilities over the anonymous character. A possible explanation for the former is that women tend to be portrayed only as objects of romance in fictional works, *e.g.*, as popularly evidenced by the failure of many movies to pass the Bechdel test (Agar-

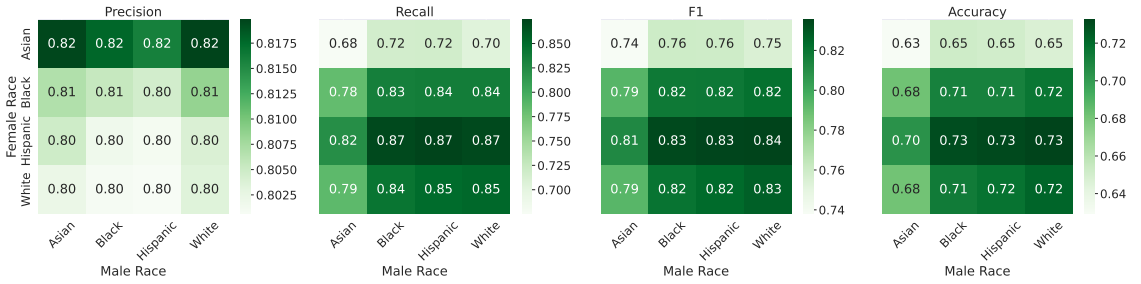


Figure 7.3: Precision, Recall, F1, and Accuracy of predicting romantic relationships from Llama2-7B for subset of the dataset where characters have different genders and are replaced with names associated with different races/ethnicities.

wal et al., 2015).

The smaller gap in the recall between both female (bottom-right) name-replacements and different-gender (top-right and bottom-left) ones for Asian and Hispanic as compared to White and Black may result from model's inability to discern gender from Asian and Hispanic names as accurately as for White and Black names. Figures 7.5 and 7.6 (see additional results in §7.5.2) show similar trends for Llama2-13B and Mistral-7B, respectively.

The unnaturalness of movie scripts with name and gender substitutions could, in theory, provide an alternative explanation for the observed biases, but the evidence shows this is not the cause. As female characters may speak differently from male characters, our name-replacements can introduce statistical inconsistency between the gender associated with a character name and the style or content of the lines they speak, potentially confounding our observations. However, comparable recall between name-replacements that preserve the gender (upper-triangle; specifically top-right) associated with the original speakers and the swapped variants (lower-triangle; specifically bottom-left) in Figure 7.2, indicates that swapping both characters' genders has minimal impact on model's performance in the conversations we used. Hence, we conclude the potential inconsistency between gender and linguistic content is not a major confounding factor.

Race/Ethnicity		Asian	Black	Hispanic	White
Gender	Logistic regression	53.3±12.7	**96.4±2.9**	**80.5±13.0**	**99.9±0.2**
	Majority baseline	54.2±0.0	54.2±0.0	54.2±0.0	53.9±0.3
Race	Logistic regression	**97.6±1.9**	**70.5±6.3**	**89.5±4.1**	**94.2±3.8**
	Majority baseline	50.6±0.2	50.6±0.4	50.9±0.4	50.9±0.3

Table 7.3: Logistic regression classification accuracy (%) of predicting the demographic attributes associated with a name from Llama2-7B contextualized embeddings.

Character pairs involving Asian names have lower romantic recall; however, we do not find strong evidence against interracial pairings. While Llama2-7B has similar precision of predicting a romantic relationship across all racial pairs (0.80 – 0.82), Figure 7.3 shows name pairs involving at least one Asian name have significantly lower recall. Noticeably, the recall is the lowest (0.68) when both character names are associated with Asian. Although there are variations in recall values among different racial setups, we do not observe disparate differences between interracial and intraracial name pairs for non-Asian names. Results for Llama2-13B and Mistral-7B, shown respectively in Figure 7.7 and 7.8 in the §7.5.2, demonstrate a similar trend that Asian names lead to substantially lower recall values. Such systematically worse performance on Asian names potentially perpetuates known algorithmic biases Chander (2016); Akter et al. (2021); Papakyriakopoulos and Mboya (2023).

7.3 Analysis and Discussion

We perform additional experiments to understand the observed model behavior.

Why does a model tend to predict fewer romantic relationships for racial pairings that involve Asian names? Although we select names for each race that have strong real-world statistical associations with one gender, we hypothesize that low recall on pairs

Model	Precision	Recall	F1	Accuracy
Gender Pairings				
Llama2-7B	0.7978	0.6887	0.7392	0.6125
Llama2-13B	0.8649	0.3019	0.4476	0.4170
Mistral-7B	0.8269	0.2028	0.3258	0.3432
Racial Pairings				
Llama2-7B	0.8063	0.7131	0.7569	0.6422
Llama2-13B	0.8696	0.3287	0.4665	0.4404
Mistral-7B	0.8406	0.2311	0.3625	0.3761

Table 7.4: Evaluation scores for anonymous name-replacements (character replaced with "X" or "Y") for different models under study. These results depict the model's performance solely based on the context.

with one or more Asian names may be due to model's inability to discern gender from Asian names. To test this hypothesis, we retrieve the contextualized embeddings from Llama2-7B for each name (collected in §7.1.2) occurrence in 15 romantic and 15 non-romantic random dialogues. We obtain $209,800$ embeddings, which are used to train logistic regression models (separately for each race) to classify the gender and race (one-vs-all) associated with a name, using embeddings of 70% of names in a race as the training set and the remaining as the test set. We control the train and test set in the racial setups to have a balanced number of positive and negative samples by down-sampling the instances from other races ($1/3$ from each other race).

As we compare the average classification accuracy (across 5 different train-test splits) against a majority baseline, we observe, in Table 7.3, that gender could be effectively predicted for non-Asian name embeddings, and the embeddings are distinguishable by race for all races/ethnicities in a One-vs-All setting. However, Asian name embeddings encode minimal gender information, decreasing the likelihood of a model leveraging the inferred gender identity when making relationship predictions that reflect heteronormative biases.

Does gender association have a stronger influence on model's prediction than race/ethnicity? We believe that after name-replacements, any deviation from results obtained via anonymous name-replacements (see Table 7.4) would indicate that a model exploits the implicit information from first names. In Figure 7.2, multiple settings have recall values that significantly differ from those in the anonymized setting (0.6887). This disparity suggests name-replacements introduce gender information that significantly influences the model behavior. Such trends are less prominent for Asian names due to the model's apparent inability to distinguish gender information in Asian names (Table 7.3). By contrast, racial information encoded in first names exerts a lesser impact. Non-Asian heterosexual intra/inter-racial pairs give rise to similar recall in Figure 7.3. We thus do not observe strong prejudice against interracial romantic relationships here.

Does potential data contamination impact our findings? As large language models are pretrained on massive web-scraped data from a variety of sources, it is likely that they have either explicitly or implicitly (*i.e*, summary of movies available online) seen the test set used in this work (Magar and Schwartz, 2022). However, we believe that the potential data contamination could only lead to underestimation of our results but not overestimation. As the focus of this work is not on evaluating the accuracy of these models on the task of relationship prediction but on examining the change in predictions as the names of the characters are replaced while keeping the conversation same. We believe that if the models would have memorized the relationship between characters of the movie scripts used in this work then the predictions would not have changed with the change in the character's name and gender associated with the names. The fact that recall of romantic relationship predictions (see Figure 7.2 and Figure 7.3) significantly increases when names are replaced with that of different gender characters pairs in the conversations as compared to when names were anonymized (see Table 7.4) and decreases with the same gender ones, confirms that

relationship predictions from these models are influenced by the gender and race associated with the names. We would not have observed such behavior if these predictions were a consequence of memorized information by the language models.

7.4 Summary

Through controlled name-replacement experiments, we show that LLMs resort to spurious correlations while making relationship predictions between two characters involved in a conversation. LLMs predict romantic relationships based on the demographic identities (gender and race) associated with their names. Specifically, LLMs are less likely to predict a romantic relationship between same-gender and intra/inter-racial character pairs involving Asian names. Our analysis of contextualized name embeddings reveals that one of the potential causes of this behavior is that gender is less discernible from Asian names than from other races. These findings highlight the heuristics that LLMs often rely on, which could result in incorrect or correct predictions but for the wrong reasons, thereby adding another layer of complexity to the task of relationship prediction.

7.5 Name Selection Details and Additional Results

7.5.1 First Names

We detail the name selection criteria in our experiments. We also list all first names we have used in our experiments to study the influence of different gender and racial/ethnic name pairing.

First Names Used to Study the Influence of Gender Pairing

We first collect names that have frequency over 200 and have more than 80% of the population having that name identify themselves as a particular race (Asian, Black, Hispanic, and White) from Rosenman et al. 2023. Then, we partition these names into 10 non-linearly segmented bins (shown in Figure 7.2) based on the percentage of population that has been assigned as female at birth using statistics from the Social Security Application dataset (SSA[5]). We randomly sample 3 names per bin totaling to 30 names per race[6] for performing the replacements. We consider names belonging to a spectrum of female gender associations to ensure coverage of gender-neutral names.

We list all the names used in this set of experiments. We include the percentage of the population assigned female gender at birth in parentheses.

Asian Seung (0.00%), Quoc (0.00%), Dat (0.00%), Nghia (2.30%), Thuan (2.40%), Thien (2.70%), Hoang (6.40%), Sang (6.60%), Jun (9.60%), Sung (13.50%), Jie (17.30%), Wei (21.80%), Hyun (39.00%), Khanh (41.90%), Wen (44.60%), Hien (51.70%), An (54.80%), Ji (61.40%), In (80.80%), Diem (88.60%), Quyen (88.90%), Ling (91.30%), Xiao (91.50%), Ngoc (92.40%), Su (95.40%), Hanh (95.60%), Vy (97.00%), Eun (98.30%), Trinh (100.00%), Huong (100.00%)

Black Deontae (0.00%), Antwon (0.10%), Javonte (1.00%), Dejon (2.90%), Jamell (3.40%), Dijon (4.60%), Dashawn (5.80%), Deshon (6.20%), Pernell (8.30%), Rashawn (10.10%), Torrance (13.20%), Semaj (22.60%), Demetris (25.60%), Kamari (33.60%), Amari (42.00%), Shamari (56.10%), Kenyatta (57.10%), Ivory (59.30%), Chaka (76.20%), Ashante (89.40%), Unique (89.90%), Kenya (92.20%), Nikia (93.80%), Akia

[5] https://www.ssa.gov/oact/babynames/

[6] Except for Hispanic wherein we did not get any names in $5-10\%$ bin and only 1 name in $25-50\%$ bin.

(94.30%), Kenyetta (95.50%), Shante (96.40%), Shaunta (97.00%), Laquandra (100.00%), Lakesia (100.00%), Daija (100.00%)

Hispanic Nestor (0.00%), Fidel (0.00%), Raul (0.60%), Leonides (2.70%), Yamil (4.50%), Reyes (10.80%), Cruz (13.10%), Neftali (14.90%), Noris (38.10%), Nieves (62.40%), Guadalupe (72.60%), Ivis (75.00%), Monserrate (78.20%), Ibis (82.60%), Johanny (89.40%), Elba (91.50%), Matilde (93.40%), Rocio (96.90%), Lucero (97.30%), Cielo (97.50%), Lucila (100.00%), Zuleyka (100.00%), Yaquelin (100.00%)

White Zoltan (0.00%), Leif (0.10%), Jack (0.40%), Ryder (3.30%), Carmine (3.40%), Haden (4.10%), Tate (5.30%), Dickie (5.50%), Logan (7.40%), Parker (17.50%), Sawyer (20.90%), Hayden (22.50%), Dakota (29.70%), Britt (38.30%), Harley (41.70%), Campbell (53.90%), Barrie (56.10%), Peyton (61.90%), Kelley (88.00%), Jodie (88.20%), Leigh (88.70%), Clare (90.90%), Rylee (92.20%), Meredith (94.70%), Baylee (97.00%), Lacey (97.30%), Ardith (97.70%), Kristi (99.80%), Galina (100.00%), Margarete (100.00%)

First Names Used to Study the Influence of Intra/Inter-racial Pairing

By referencing Rosenman et al. 2023 and the SSA dataset again, we collect another set of both race- and gender-indicative first names with a minimum frequency of 200, applying a threshold of 90% for the percentage of the population assigned either female or male at birth. For race threshold, we set it to be 90% for Asian, Black, and Hispanic, and 70% for White. Although we choose a lower threshold for White to account for the phenomenon of name Anglicization Zhao and Biernat (2019), we still obtain empirical results that strongly indicate these names are represented differently from names associated with other races/ethnicities. In total, we obtain 80 names that are evenly distributed among four races/ethnicities and two genders. We replace name-pairs while preserving the gender as-

sociated with the names in the original dialogue.

Asian Female Thuy, Thu, Huong, Trang, Ngoc, Hanh, Hang, Xuan, Trinh, Eun

Asian Male Tuan, Hai, Sang, Hoang, Nam, Huy, Quang, Duc, Trung, Hieu

Black Female Latoya, Ebony, Latasha, Latonya, Tamika, Kenya, Tameka, Lakeisha, Tanisha, Precious

Black Male Tyrone, Cedric, Darius, Jermaine, Demetrius, Malik, Jalen, Roosevelt, Marquis, Deandre

Hispanic Female Luz, Mayra, Marisol, Maribel, Alejandra, Yesenia, Migdalia, Xiomara, Mariela, Yadira

Hispanic Male Luis, Jesus, Lazaro, Osvaldo, Heriberto, Jairo, Rigoberto, Adalberto, Ezequiel, Ulises

White Female Mary, Patricia, Jennifer, Linda, Elizabeth, Barbara, Susan, Jessica, Kimberly, Sandra

White Male James, Michael, John, Robert, William, David, Christopher, Richard, Joseph, Charles

7.5.2 Additional Results

We report the results for Llama2-13B (Figures 7.5 and 7.7) and Mistral-7B (Figures 7.6 and 7.8). We observe similar trends as Llama2-7B discussed previously in this chapter. We also report the F1 and accuracy scores for Llama2-7B, for completeness, in Figure 7.4.

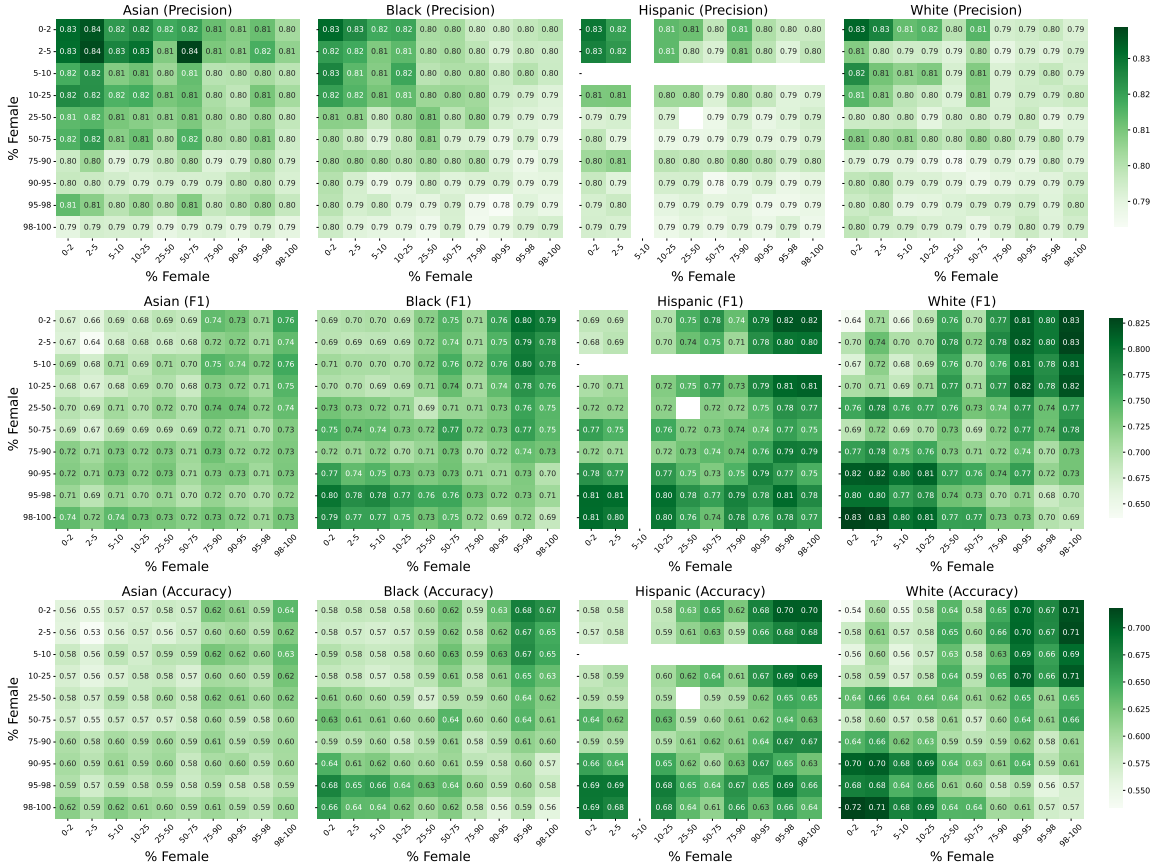


Figure 7.4: Precision, F1-score and Accuracy plots for romantic predictions from Llama2-7B model.

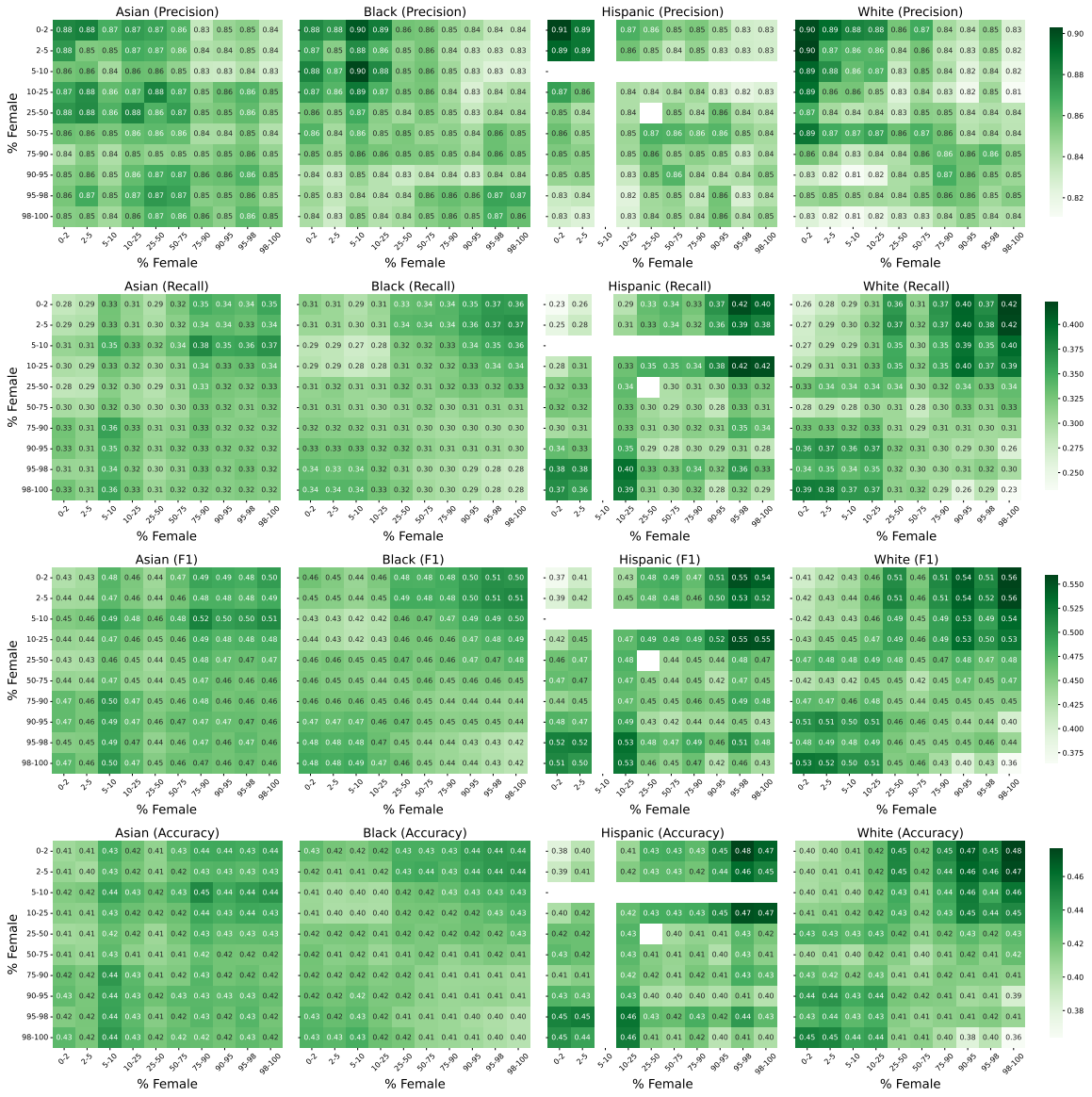


Figure 7.5: Precision, Recall, F1-score and Accuracy plots for romantic predictions from Llama2-13B model.

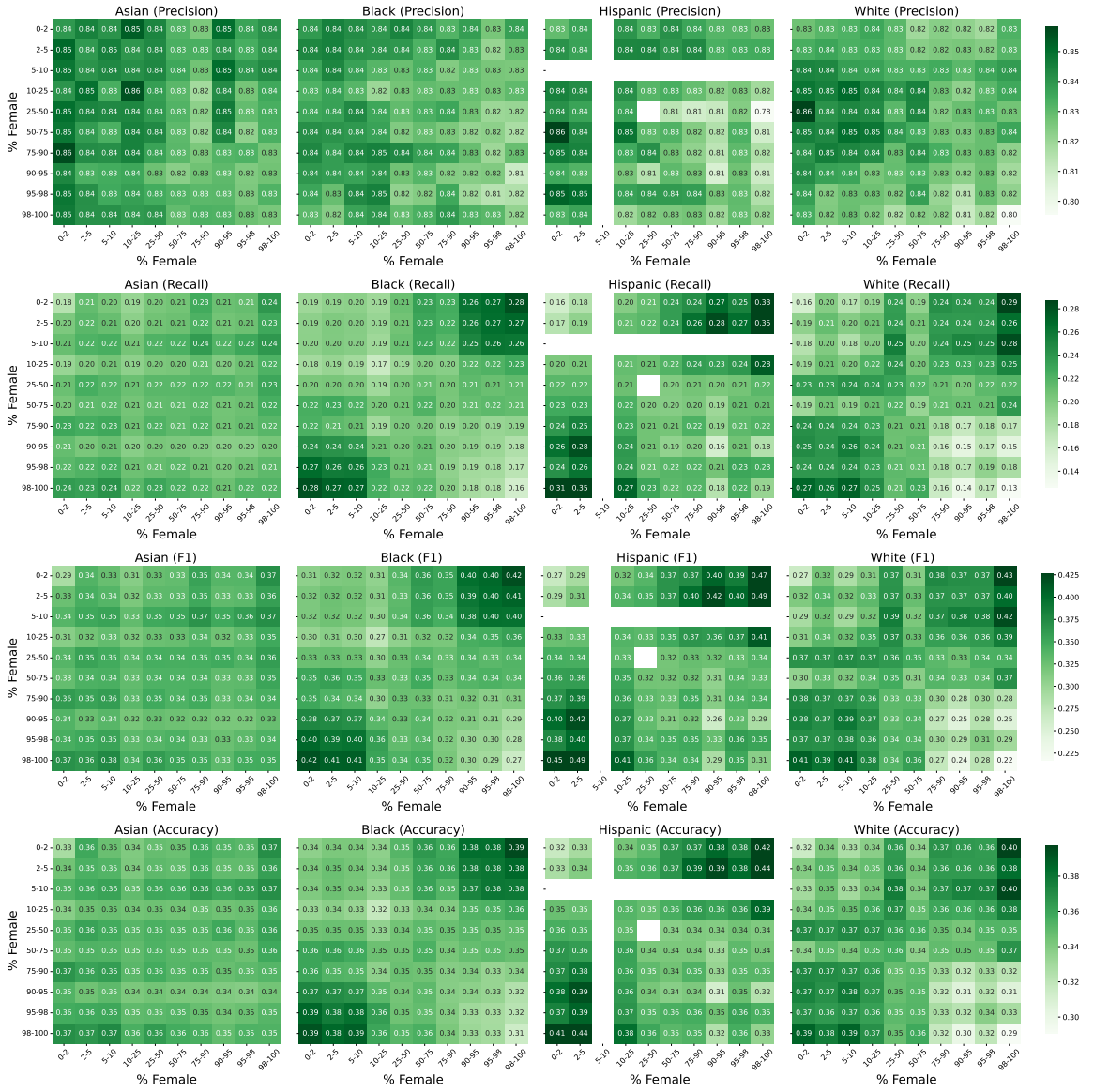


Figure 7.6: Precision, Recall, F1-score and Accuracy plots for romantic predictions from Mistral-7B model.

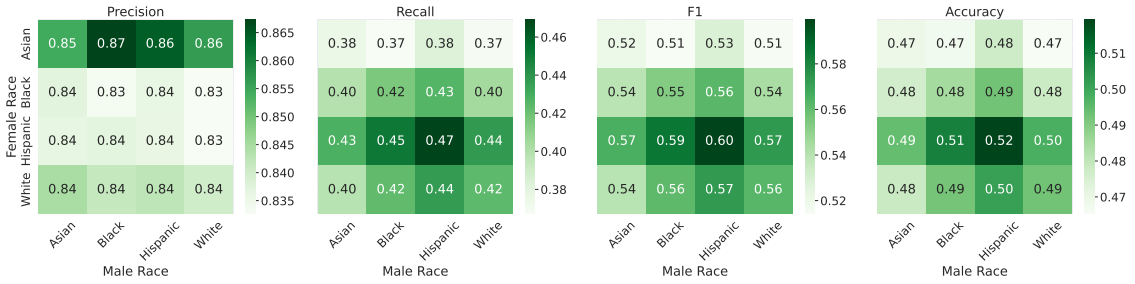


Figure 7.7: Precision, Recall, F1, and Accuracy of predicting romantic relationships from Llama2-13B for subset of the dataset where characters have different genders and are replaced with names associated with different races/ethnicities.

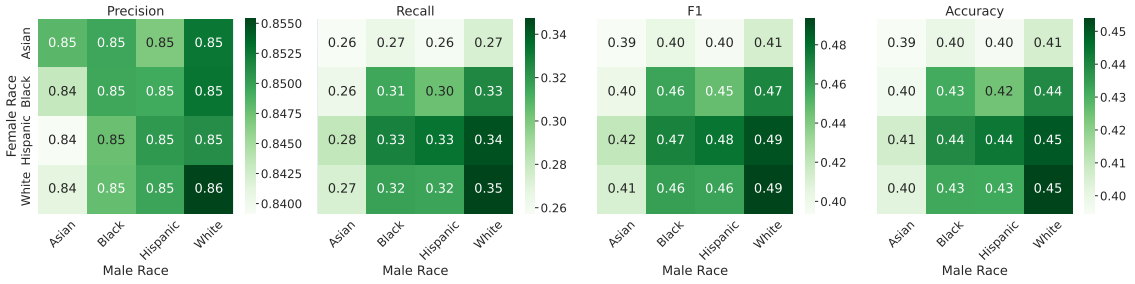


Figure 7.8: Precision, Recall, F1, and Accuracy of predicting romantic relationships from Mistral-7B for subset of the dataset where characters have different genders and are replaced with names associated with different races/ethnicities.

Chapter 8: Conclusion and Future Work

8.1 Summary

In this dissertation, we study static or evolving entity-related information mentioned in narratives and legal documents either explicitly or implicitly. We identify and extract entity-related information from unstructured text in a structured form to enable easier comprehension of long documents and serve the information needs of readers across narrative and legal domains. We contribute by designing new tasks, collecting datasets, and evaluating and proposing models or frameworks covering these aspects of information to improve and emphasize the importance of entity-centric understanding of language specifically for long documents.

We first assess the knowledge of scripts present in existing pretrained generative language models by introducing a task of generating full event sequence descriptions (ESDs) from a protagonist's perspective given a scenario in the form of a natural language prompt (Sancheti and Rudinger, 2022). Through zero-shot prompting, we find that language models generate poor ESDs with mostly omitted, irrelevant, repeated, or misordered events due to the inability to track the state of entities participating in the events. To address this, we propose a pipeline-based protagonist-oriented script induction framework that is shown to generate good-quality ESDs for unseen scenarios (*e.g.,* bake a cake). This work contributes towards modeling knowledge that humans implicitly use in the form of protagonist-oriented sequences of events for a scenario, required for narrative comprehension.

Next, we introduce new tasks, datasets, and models in the legal domain wherein the legally binding contracts act as a good example of written communication where multiple contracting parties are involved in events that are necessary or possible to happen. As contracts are often signed without being read, to facilitate compliance monitoring and improve understanding of rights and duties for each of the contracting parties, we introduce the task of identifying party-specific deontic modalities (Sancheti et al., 2022a) to extract static and explicitly mentioned information (such as obligations, entitlements, prohibitions, and permissions) specific to each of the contracting parties. We present a linguistically informed taxonomy for annotating deontic types in the legal domain, and use the taxonomy to build a corpus (LexDeMod) of English contracts consisting of deontic type and their trigger (*i.e*, words or phrases that evoke the deontic type) annotations. We introduce two new tasks: (1) party-specific multi-label deontic modality classification, and (2) (party-specific deontic modality and trigger span detection to extract explicitly mentioned information related to the parties. We benchmark this dataset on the two tasks using transformer-based models that can accurately perform the task demonstrating that the linguistic diversity of expressing such modalities is learnable using our dataset. Transfer learning experiments from lease to employment agreements further indicate the generalizability of these expressions.

Recognizing the issue of information overload due to the presence of several sentences per deontic type, and different levels of implicit importance associated with each sentence for each of the parties, we introduce the task of party-specific extractive summarization of important obligations, entitlements, and prohibitions (Sancheti et al., 2023) to make long contracts easier to read and understand. To this end, we collect an expert-annotated dataset consisting of party-specific pair-wise importance annotations for a subset of the obligations, prohibitions, and entitlements from LexDeMod (Sancheti et al., 2022a) dataset. We propose, ContraSum, a party-specific extractive summarization system that first identifies the category of each sentence in a contract for a party and then ranks sentences in each

category based on their importance score for a given party. Experts found the extracted summaries from our proposed system to be more informative, useful, and correctly categorized as compared to text-ranking-based and a random baseline.

Having designed new tasks, datasets, and party-specific legal document understanding systems where entity-related information is static, we then consider the task of tracking evolving relationships between characters in narratives such as books (Chapter 6). We define the evolution in the relationship between characters as a task of predicting a trajectory of predefined relationship and evolution type at different points in the book. To address the challenge of providing book-length context for the prediction task, we propose different strategies that vary based on the granularity (ranging from complete passages to previous summary along with a passage to complete summary) of information provided to a language model. Owing to the high cost and difficulty of obtaining an annotated dataset for evaluation purposes, we provide a thorough analysis of the agreement between predictions from different language models and a manual evaluation of the consensus predictions. Low-to-moderate agreement shows that tracking evolution in the relationship is a hard task even for language models with a large context window and humans. This happens due to the multi-faceted nature of relationships leading to different interpretations, the sensitivity of language models to subtle changes in the context, and the use of surface-level cues; raising questions on their *true* understanding of social relationships.

To investigate the observations from the previous work that language models are sensitive to subtle (irrelevant) changes in the provided context and may adopt some heuristics, in our final work, we study the impact of changing different attributes associated with the names (such as gender, and race) of the characters involved in a conversation on the predictions from language models (Sancheti et al., 2024). Through controlled name-replacement experiments, we find that large language models resort to spurious correlations between relationship type and causal factors (such as gender and race) which could result in incor-

rect or correct predictions but for the wrong reasons. Specifically, large language models are less likely to predict a romantic relationship between same-gender and intra/inter-racial character pairs involving Asian names, adding another layer of complexity to the task of relationship prediction.

To conclude, through our work in the legal domain, we establish the importance of associating parties to their rights and duties (Chapter 4) and modeling each party's level of importance for these rights and duties to enable the development of a party-specific extractive summarization system (Chapter 5) which is shown to generate useful, informative, and correctly categorized summary of what must, can, and cannot be done under a legal agreement to serve the information needs of contracting parties, and legal professionals. Large language models are ubiquitously used as they have been shown to do well on several NLU tasks. Through our work in the narrative domain, we evaluate the entity-centric understanding of these models. To do this, we assess the extent of script knowledge accessible through these models and their social reasoning capabilities; both of which are implicit to humans. We shed light on their performance on several tasks (such as generating event sequences and tracking evolving relationships) that are recognized to be important for narrative comprehension. Through these works, we contribute by characterizing these tasks in the era of large language models and throwing light on their shortcomings. We build systems to improve the generated protagonist-oriented event sequences (Chapter 3), share insights on the challenges in tracking relationships (Chapter 6), and how different attributes of entities can influence relationship predictions from large language models (Chapter 7). While progress from these models is impressive, it remains unclear if increasing context window size *truly* enables long document understanding and improves their reasoning abilities.

While this dissertation takes one step towards entity-centric understanding of long documents, our work opens up several research directions detailed in the next section.

8.2 Limitations and Future Research Directions

In this section, we explore the limitations of our work and highlight new questions and opportunities for future research.

8.2.1 Generating context and event-granularity aware event sequences

In Chapter 3 we generate multiple event sequences per given scenario however, we acknowledge that there can be diverse sequences of events for the same scenario depending upon the context and the granularity with which an event is described. For instance, '*eating at a restaurant*' script could be different based on the type of the restaurant (such as fine-dining or fast-casual). While the event of payment happens at the end of having food in the context of fine-dining, it occurs at the beginning in the context of a fast-casual restaurant. Consider another scenario of '*baking a cake*', without specific constraints (such as for diabetic people), a model may generate a standard sequence of events. Such context-aware event sequence generation in the form of single or multiple constraints (Brahman et al., 2023; Yuan et al., 2023) can help in disambiguation and context-guided evaluation of the generated scripts.

When such event sequence generation systems are used for developing planning agents (Kannan et al., 2023), the granularity of the event description becomes critical. For instance, the event of going to a restaurant (in a car) could be further broken down into – sit in a car, start the car, setup the directions to the restaurant, and follow the directions. Future work may consider building granularity-based event sequence generation systems for practical use-cases.

8.2.2 Extending the scope of deontic modality detection to different types of agreements and languages

Although we demonstrate reasonable generalization of models trained on LexDeMod dataset (introduced in Chapter 4) to employment agreements, the dataset is limited to lease agreements which may not cover all the linguistic expressions for deontic modality occurring in the legal domain. There is a huge value in extending our data annotation framework to other agreement types such as privacy policies, sales contracts, real estate purchase agreements, and loan agreements. Since agreements in different countries and languages may differ considerably, there is a scope for research in studying any commonalities and differences across languages.

8.2.3 Party-specific summarization beyond explicitly mentioned parties

The party-specific summarization system (CONTRASUM), proposed in Chapter 5, may not be able to include sentences (containing obligations, entitlements, or prohibitions) that do not explicitly mention either of the parties (*e.g.*, (a)“On termination of the lease, the apartment has to be handed over freshly renovated." and (b)“The tenant can vacate the place at any point in time without any consequences.") in the extracted summaries. While these sentences contain important obligations for tenant and information that is important for landlord to know, our work is constrained by the cases that the prior work (Sancheti et al., 2022a) covers as our dataset is built atop that dataset. A more sophisticated system could handle example (a) by identifying the tenant as an implicit participant in the “handing over” event, (and the landlord in example (b) even arguably plays an oblique role as the owner of “the place"); the omission of these cases is arguably due to our method rather than fundamental limitations of the task formulation. Follow-up work could explore

these aspects either through the identification of implicit participants, or the expansion of categories.

8.2.4 Intent-focused party-specific summarization

Our proposed method in Chapter 5 generates summaries that contain the 10 most important obligations, entitlements, and prohibitions from the complete agreement. However, in certain situations, leaseholders may only be interested in a summary of a particular aspect such as *subleasing the apartment* or *smoking in the apartment*. We leave the extension of the proposed summarization system to an intent-focused summarization system (Hayashi et al., 2021; Yang et al., 2023) for future research. Building such a system would require another module to filter the sentences based on the desired aspect. Topic modeling is a potential way of filter such sentences (Nagwani, 2015; Akhtar et al., 2019; Rani and Lobiyal, 2021; Belwal et al., 2023).

8.2.5 Modeling asymmetric relationships and identifying change points for evolving relationships

While relationship prediction is a multi-faceted task, we consider it as a single-label prediction task for tracking the evolution in relationships in Chapter 6. However, we acknowledge that two characters may be related with multiple relationship types such as former romantic interests as well as co-parents at the same time. Further, relationships are not always symmetric from each character's perspective, they may be nice to each other but may have negative/positive emotions in the back of their minds. To capture such situations, our work can be extended to cover evolution in relationship from each character's perspective or track the emotional states of each character with respect to a relationship as the novel progresses. Another research direction is to analyze the different implicit ways in

which a particular relationship type could be expressed in text to identify appropriateness and normative expectations (*e.g.*, interactions between strangers tend to be more formal) in social interactions (Baxter and Wilmot, 1985; Argyle et al., 1985; Snyder and Stukas Jr, 1999). This could involve finding the supporting evidences in the provided context for each relationship prediction, and then using deductive reasoning to identify the common implications. These evidences could be further used to discover the significant turning points resulting in an evolution in the relationship.

8.2.6 Extending the coverage of names associated with different identities and conversations to represent same-gender relationships

We recognize that the set of names used for name-replacement experiments in Chapter 7 does not represent all the gender and race identities. This is due to the unavailability of data sources to compile a sufficiently large number of names strongly associated with a wide range of underrepresented race and gender identities. Future work may extend our work by considering race and gender identity statistics from countries other than the US for obtaining the names. Another limitation of our work is that the dataset Jia et al. (2021) used lacks conversations that effectively depict interactions between same-gender partners. Since such datasets are not available, a promising future direction is to leverage large language models to synthetically generate them (Veselovsky et al., 2023; Li et al., 2023; Ou et al., 2024).

Bibliography

1991. *MUC3 '91: Proceedings of the 3rd Conference on Message Understanding*. Association for Computational Linguistics, USA.

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 782–796.

Abhishek Agarwal, Shanshan Xu, and Matthias Grabmair. 2022. Extractive summarization of legal decisions using multi-task learning and maximal marginal relevance. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1857–1872, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Apoorv Agarwal, Sriramkumar Balasubramanian, Anup Kotalwar, Jiehan Zheng, and Owen Rambow. 2014. Frame semantic tree kernels for social network extraction from text. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 211–219, Gothenburg, Sweden. Association for Computational Linguistics.

Apoorv Agarwal, Jiehan Zheng, Shruti Kamath, Sriramkumar Balasubramanian, and Shirin Ann Dey. 2015. Key female characters in film have more to talk about besides men: Automating the Bechdel test. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 830–840, Denver, Colorado. Association for Computational Linguistics.

Md Shad Akhtar, Abhishek Kumar, Asif Ekbal, Chris Biemann, and Pushpak Bhattacharyya. 2019. Language-agnostic model for aspect-based sentiment analysis. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 154–164, Gothenburg, Sweden. Association for Computational Linguistics.

Shahriar Akter, Grace McCarthy, Shahriar Sajib, Katina Michael, Yogesh K. Dwivedi, John D'Ambra, and K.N. Shen. 2021. Algorithmic bias in data-driven innovation in the age of ai. *International Journal of Information Management*, 60:102387.

Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro, and Vasileios Lampos. 2016. Predicting judicial decisions of the european court of human rights: A natural language processing perspective. *PeerJ Computer Science*, 2:e93.

Layman E Allen and Charles S Saxon. 1994. Controlling inadvertent ambiguity in the logical structure of legal drafting by means of the prescribed definitions of the a-hohfeld structural language. *Theoria: An International Journal for Theory, History and Foundations of Science*, pages 135–172.

Iosif Angelidis, Ilias Chalkidis, and Manolis Koubarakis. 2018. Named entity recognition, linking and generation for greek legislation. In *JURIX*, pages 1–10.

Michael Argyle, Monika Henderson, and Adrian Furnham. 1985. The rules of social relationships. *British Journal of Social Psychology*, 24(2):125–139.

Elliott Ash, Jeff Jacobs, Bentley MacLeod, Suresh Naidu, and Dominik Stammbach. 2020. Unsupervised extraction of workplace rights and duties from collective bargaining agreements. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pages 766–774. IEEE.

Tara Athan, Harold Boley, Guido Governatori, Monica Palmirani, Adrian Paschke, and Adam Wyner. 2013. Oasis legalruleml. In *proceedings of the fourteenth international conference on artificial intelligence and law*, pages 3–12.

Mahmoud Azab, Noriyuki Kojima, Jia Deng, and Rada Mihalcea. 2019. Representing movie characters in dialogues. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 99–109, Hong Kong, China. Association for Computational Linguistics.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 79–85, Montreal, Quebec, Canada. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Keshav Balachandar, Anam Saatvik Reddy, A Shahina, and Nayeemulla Khan. 2021. Summarization of commercial contracts.

Niranjan Balasubramanian, Stephen Soderland, Mausam, and Oren Etzioni. 2013. Generating coherent event schemas at scale. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1721–1731, Seattle, Washington, USA. Association for Computational Linguistics.

Miguel Ballesteros, Rishita Anubhai, Shuai Wang, Nima Pourdamghani, Yogarshi Vyas, Jie Ma, Parminder Bhatia, Kathleen McKeown, and Yaser Al-Onaizan. 2020. Severing the edge between before and after: Neural architectures for temporal ordering of events. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5412–5417, Online. Association for Computational Linguistics.

Rachelle Ballesteros-Lintao, Maria Regina P Arriero, Judith Ma Angelica S Claustro, Kristina Isabelle U Dichoso, Selenne Anne S Leynes, Maria Rosario R Aranda, and Jean Reintegrado-Celino. 2016. Deontic meanings in philippine contracts. *International Journal of Legal Discourse*, 1(2):421–454.

David Bamman. 2015. People-centric natural language processing. *PhD diss., PhD thesis, Carnegie Mellon University*.

David Bamman, Brendan O'Connor, and Noah A. Smith. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 352–361, Sofia, Bulgaria. Association for Computational Linguistics.

David Bamman, Ted Underwood, and Noah A. Smith. 2014a. A Bayesian mixed effects model of literary character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 370–379, Baltimore, Maryland. Association for Computational Linguistics.

David Bamman, Ted Underwood, and Noah A Smith. 2014b. A bayesian mixed effects model of literary character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 370–379.

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Leslie A Baxter and William W Wilmot. 1985. Taboo topics in close relationships. *Journal of Social and Personal Relationships*, 2(3):253–269.

Ramesh Chandra Belwal, Sawan Rai, and Atul Gupta. 2023. Extractive text summarization using clustering-based topic modeling. *Soft Computing*, 27(7):3965–3982.

B Douglas Bernheim and Michael D Whinston. 1998. Incomplete contracts and strategic ambiguity. *American Economic Review*, pages 902–932.

Paheli Bhattacharya, Kaustubh Hiware, Subham Rajgaria, Nilay Pochhi, Kripabandhu Ghosh, and Saptarshi Ghosh. 2019. A comparative study of summarization algorithms applied to legal case judgments. In *European Conference on Information Retrieval*, pages 413–428. Springer.

Paheli Bhattacharya, Soham Poddar, Koustav Rudra, Kripabandhu Ghosh, and Saptarshi Ghosh. 2021. Incorporating domain knowledge for extractive summarization of legal case documents. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, pages 22–31.

Philip Blumstein and Peter Kollock. 1988. Personal relationships. *Annual Review of Sociology*, 14(1):467–490.

Michael J Bommarito II, Daniel Martin Katz, and Eric M Detterman. 2021. Lexnlp: Natural language processing and information extraction for legal and regulatory texts. In *Research Handbook on Big Data Law*. Edward Elgar Publishing.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from BERT. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7456–7463.

Gordon H Bower and Daniel G Morrow. 1990. Mental models in narrative comprehension. *Science*, 247(4938):44–48.

David Bracewell, David Hinote, and Sean Monahan. 2014. The author perspective model for classifying deontic modality in events. In *The Twenty-Seventh International Flairs Conference*.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Faeze Brahman, Chandra Bhagavatula, Valentina Pyatkin, Jena D. Hwang, Xiang Lorraine Li, Hirona Jacqueline Arai, Soumya Sanyal, Keisuke Sakaguchi, Xiang Ren, and Yejin Choi. 2023. Plasma: Making small language models better procedural knowledge models for (counterfactual) planning. *CoRR*, abs/2305.19472.

Faeze Brahman and Snigdha Chaturvedi. 2020. Modeling protagonist emotions for emotion-aware storytelling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5277–5294, Online. Association for Computational Linguistics.

Faeze Brahman, Meng Huang, Oyvind Tafjord, Chao Zhao, Mrinmaya Sachan, and Snigdha Chaturvedi. 2021. "let your characters tell their story": A dataset for character-centric narrative understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1734–1752, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Razvan Bunescu and Marius Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

Orson Scott Card. 1999. Characters & viewpoint: Elements of fiction writing. *Cincinnati, OH: Writer's Digest Books*.

Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany, and Serena Villata. 2017. Legal NERC with ontologies, Wikipedia and curriculum learning. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 254–259, Valencia, Spain. Association for Computational Linguistics.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019a. Neural legal judgment prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323, Florence, Italy. Association for Computational Linguistics.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019b. Neural legal judgment prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323, Florence, Italy. Association for Computational Linguistics.

Ilias Chalkidis, Ion Androutsopoulos, and Achilleas Michos. 2017. Extracting contract elements. In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*, pages 19–28.

Ilias Chalkidis, Ion Androutsopoulos, and Achilleas Michos. 2018a. Obligation and prohibition extraction using hierarchical RNNs. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 254–259, Melbourne, Australia. Association for Computational Linguistics.

Ilias Chalkidis, Ion Androutsopoulos, and Achilleas Michos. 2018b. Obligation and prohibition extraction using hierarchical RNNs. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 254–259, Melbourne, Australia. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Nathanael Chambers. 2013. Event schema induction with a probabilistic entity-driven model. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Seattle, Washington, USA. Association for Computational Linguistics.

Nathanael Chambers. 2017. Behind the scenes of an evolving event cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 41–45.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797, Columbus, Ohio. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610, Suntec, Singapore. Association for Computational Linguistics.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations*.

Anupam Chander. 2016. The racist algorithm. *Mich. L. Rev.*, 115:1023.

Kent Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. 2023. Speak, memory: An archaeology of books known to ChatGPT/GPT-4. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7312–7327, Singapore. Association for Computational Linguistics.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. Booookscore: A systematic exploration of book-length summarization in the era of llms. In *The Twelfth International Conference on Learning Representations*.

Snigdha Chaturvedi, Mohit Iyyer, and Hal Daume III. 2017. Unsupervised learning of evolving relationships between literary characters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Snigdha Chaturvedi, Shashank Srivastava, Hal Daume III, and Chris Dyer. 2016. Modeling evolving relationships between characters in literary novels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Huajie Chen, Deng Cai, Wei Dai, Zehui Dai, and Yadong Ding. 2019. Charge-based prison term prediction with deep gating network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6362–6367, Hong Kong, China. Association for Computational Linguistics.

Yi-Ting Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2020. MPDD: A multi-party dialogue dataset for analysis of emotions and interpersonal relationships. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 610–614, Marseille, France. European Language Resources Association.

Yu-Hsin Chen and Jinho D. Choi. 2016. Character identification on multiparty conversation: Identifying mentions of characters in TV shows. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 90–100, Los Angeles. Association for Computational Linguistics.

Cheng-Han Chiang and Hung-Yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631.

Eunsol Choi. 2019. *Learning To Understand Entities In Text*. Ph.D. thesis.

Sandra Chung. 1985. Tense, aspect and mood. *Language typology and syntactic description*, pages 202–258.

Kenneth Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

G Marcus Cole. 2015. Rational consumer ignorance: When and why consumers should agree to form contracts without even reading them. *JL Econ. & Pol'y*, 11:413.

Lee J Cronbach. 1951. Coefficient alpha and the internal structure of tests. *psychometrika*, 16(3):297–334.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, Prague, Czech Republic. Association for Computational Linguistics.

Gregory Currie. 2009. Narrative and the psychology of character. *The journal of aesthetics and art criticism*, 67(1):61–71.

Sebastian Deri, Jeremie Rappaz, Luca Maria Aiello, and Daniele Quercia. 2018. Coloring in the links: Capturing social ties as they are perceived. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–18.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. 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. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. 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. Association for Computational Linguistics.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Mauro Dragoni, Serena Villata, Williams Rizzi, and Guido Governatori. 2016. Combining nlp approaches for rule extraction from legal documents. In *1st Workshop on MIning and REasoning with Legal texts (MIREL 2016)*.

Xingyi Duan, Baoxin Wang, Ziyue Wang, Wentao Ma, Yiming Cui, Dayong Wu, Shijin Wang, Ting Liu, Tianxiang Huo, Zhen Hu, et al. 2019. Cjrc: A reliable human-annotated benchmark dataset for chinese judicial reading comprehension. In *China National Conference on Chinese Computational Linguistics*, pages 439–451. Springer.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Mohamed Elaraby and Diane Litman. 2022. ArgLegalSumm: Improving abstractive summarization of legal documents with argument mining. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6187–6194, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Micha Elsner. 2012. Character-based kernels for novelistic plot structure. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 634–644, Avignon, France. Association for Computational Linguistics.

David Elson, Nicholas Dames, and Kathleen McKeown. 2010. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147, Uppsala, Sweden. Association for Computational Linguistics.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Angela Fan, David Grangier, and Michael Auli. 2018a. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018b. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Susan T Fiske, Shelley E Taylor, Nancy L Etcoff, and Jessica K Laufer. 1979. Imaging, empathy, and causal attribution. *Journal of Experimental Social Psychology*, 15(4):356–377.

Lucie Flekova and Iryna Gurevych. 2015. Personality profiling of fictional characters using sense-level links between lexical resources. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1805–1816, Lisbon, Portugal. Association for Computational Linguistics.

Terry N Flynn and Anthony AJ Marley. 2014. Best-worst scaling: theory and methods. In *Handbook of choice modelling*, pages 178–201. Edward Elgar Publishing.

Ruka Funaki, Yusuke Nagata, Kohei Suenaga, and Shinsuke Mori. 2020. A contract corpus for recognizing rights and obligations. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2045–2053, Marseille, France. European Language Resources Association.

Filippo Galgani, Paul Compton, and Achim Hoffmann. 2012. Combining different summarization techniques for legal text. In *Proceedings of the workshop on innovative hybrid approaches to the processing of textual data*, pages 115–123.

Filippo Galgani and Achim Hoffmann. 2010. Lexa: Towards automatic legal citation classification. In *Australasian Joint Conference on Artificial Intelligence*, pages 445–454. Springer.

Morton Ann Gernsbacher, Brenda M Hallada, and Rachel RW Robertson. 1998. How automatically do readers infer fictional characters' emotional states? *Scientific studies of reading*, 2(3):271–300.

René Girard and Robert Hamerton-Kelly. 1987. Generative scapegoating. *Violent origins*, pages 73–145.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 25–30.

Amit Goyal, Ellen Riloff, and Hal Daumé III. 2010. Automatically producing plot unit representations for narrative text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 77–86, Cambridge, MA. Association for Computational Linguistics.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Ralph Grishman and Beth Sundheim. 1995. Design of the MUC-6 evaluation. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Barbara Grosz, Aravind Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Ben Hachey and Claire Grover. 2006. Extractive summarisation of legal texts. *Artificial Intelligence and Law*, 14(4):305–345.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. 2021. Pre-trained models: Past, present and future. *AI Open*, 2:225–250.

Hiroaki Hayashi, Prashant Budania, Peng Wang, Chris Ackerson, Raj Neervannan, and Graham Neubig. 2021. WikiAsp: A dataset for multi-domain aspect-based summarization. *Transactions of the Association for Computational Linguistics*, 9.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. Cuad: An expert-annotated nlp dataset for legal contract review. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Jerry R Hobbs. 1987. World knowledge and word meaning. In *Theoretical Issues in Natural Language Processing 3*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Nils Holzenberger and Benjamin Van Durme. 2021. Factoring statutory reasoning as language understanding challenges. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2742–2758, Online. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Dirk Hovy and Diyi Yang. 2021. The importance of modeling social factors of language: Theory and practice. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 588–602.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016a. Feuding families and former Friends: Unsupervised learning for dynamic fictional relationships. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1534–1544, San Diego, California. Association for Computational Linguistics.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016b. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1534–1544.

Bram Jans, Steven Bethard, Ivan Vulić, and Marie Francine Moens. 2012. Skip n-grams and ranking functions for predicting script events. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 336–344, Avignon, France. Association for Computational Linguistics.

Otto Jespersen. 2013. *The philosophy of grammar*. Routledge.

Qi Jia, Hongru Huang, and Kenny Q Zhu. 2021. Ddrel: A new dataset for interpersonal relation classification in dyadic dialogues. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13125–13133.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

David Jurgens, Agrima Seth, Jackson Sargent, Athena Aghighi, and Michael Geraci. 2023. Your spouse needs professional help: Determining the contextual appropriateness of messages through modeling social relationships. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10994–11013, Toronto, Canada. Association for Computational Linguistics.

Prathamesh Kalamkar, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn, and Vivek Raghavan. 2022. Named entity recognition in Indian court judgments. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 184–193, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

H Kamp. 1981a. A theory of truth and semantic represenation. *Formal Methods in the Study of Language, Amsterdam Mathematical Centre Tracts.*, 136:277–322.

Hans Kamp. 1981b. Evénements, représentations discursives et référence temporelle. *langages*, (64):39–64.

Hans Kamp, Josef Van Genabith, and Uwe Reyle. 2010. Discourse representation theory. In *Handbook of Philosophical Logic: Volume 15*, pages 125–394. Springer.

Shyam Sundar Kannan, Vishnunandan LN Venkatesh, and Byung-Cheol Min. 2023. Smart-llm: Smart multi-agent robot task planning using large language models. *arXiv preprint arXiv:2309.10062*.

XJ Kennedy and Dana Gioia. 1983. Literature: An introduction to fiction. *Poetry, Drama, and writing*.

XJ Gioia Kennedy. 2015. *LITERATURE: An Introduction to Fiction, Poetry, Drama, and Writing with Myliteraturelab,... Compact Edition, Global Edition*. PEARSON EDUCATION Limited.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Moniba Keymanesh, Micha Elsner, and Srinivasan Sarthasarathy. 2020. Toward domain-guided controllable summarization of privacy policies. In *NLLP@ KDD*, pages 18–24.

Evgeny Kim and Roman Klinger. 2019a. An analysis of emotion communication channels in fan-fiction: Towards emotional storytelling. In *Proceedings of the Second Workshop on Storytelling*, pages 56–64, Florence, Italy. Association for Computational Linguistics.

Evgeny Kim and Roman Klinger. 2019b. Frowning Frodo, wincing Leia, and a seriously great friendship: Learning to classify emotional relationships of fictional characters. 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 647–653, Minneapolis, Minnesota. Association for Computational Linguistics.

Svetlana Kiritchenko and Saif Mohammad. 2017. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 465–470, Vancouver, Canada. Association for Computational Linguistics.

Nadzeya Kiyavitskaya, Nicola Zeni, Travis D. Breaux, Annie I. Antón, James R. Cordy, Luisa Mich, and John Mylopoulos. 2008. Automating the extraction of rights and obligations for regulatory compliance. In *Conceptual Modeling - ER 2008*, pages 154–168, Berlin, Heidelberg. Springer Berlin Heidelberg.

Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology*. Sage publications.

Vinodh Krishnan and Jacob Eisenstein. 2015. "you're mr. lebowski, I'm the dude": Inducing address term formality in signed social networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1616–1626, Denver, Colorado. Association for Computational Linguistics.

G Frederic Kuder and Marion W Richardson. 1937. The theory of the estimation of test reliability. *Psychometrika*, 2(3):151–160.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Vincent Labatut and Xavier Bost. 2019. Extraction and analysis of fictional character networks: A survey. *ACM Comput. Surv.*, 52(5).

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

Wendy G Lehnert. 1981. Plot units and narrative summarization. *Cognitive science*, 5(4):293–331.

Elena Leitner, Georg Rehm, and Julian Moreno-Schneider. 2019. Fine-grained named entity recognition in legal documents. In *International Conference on Semantic Systems*, pages 272–287. Springer.

Spyretta Leivaditi, Julien Rossi, and Evangelos Kanoulas. 2020. A benchmark for lease contract review. *arXiv preprint arXiv:2010.10386*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiwei Li and Dan Jurafsky. 2017. Neural net models of open-domain discourse coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 198–209.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore. Association for Computational Linguistics.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360.

Gabrielle Lissauer. 2014. *The Tropes of Fantasy Fiction*. McFarland.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jordan J Louviere and George G Woodworth. 1991. Best-worst scaling: A model for the largest difference judgments. Technical report, Working paper.

Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. 2017. Learning to predict charges for criminal cases with legal basis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2727–2736, Copenhagen, Denmark. Association for Computational Linguistics.

Mounica Maddela, Mayank Kulkarni, and Daniel Preotiuc-Pietro. 2022. EntSUM: A data set for entity-centric extractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3355–3366, Dublin, Ireland. Association for Computational Linguistics.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 157–165, Dublin, Ireland. Association for Computational Linguistics.

Aibek Makazhanov, Denilson Barbosa, and Grzegorz Kondrak. 2014. Extracting family relationship networks from novels. *arXiv preprint arXiv:1405.0603*.

Gideon Mann and David Yarowsky. 2003. Unsupervised personal name disambiguation. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 33–40.

Laura Manor and Junyi Jessy Li. 2019a. Plain English summarization of contracts. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, pages 1–11, Minneapolis, Minnesota. Association for Computational Linguistics.

Laura Manor and Junyi Jessy Li. 2019b. Plain English summarization of contracts. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, pages 1–11, Minneapolis, Minnesota. Association for Computational Linguistics.

Mehdi Manshadi, Reid Swanson, and Andrew S Gordon. 2008. Learning a probabilistic model of event sequences from internet weblog stories. In *FLAIRS Conference*, pages 159–164.

Philip Massey, Patrick Xia, David Bamman, and Noah A Smith. 2015. Annotating character relationships in literary texts. *arXiv preprint arXiv:1512.00728*.

Aleksandra Matulewska. 2010. Deontic modality and modals in the language of contracts.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Neil McIntyre and Mirella Lapata. 2010. Plot induction and evolutionary search for story generation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1562–1572, Uppsala, Sweden. Association for Computational Linguistics.

Robert McKee. 1997. *Story: style, structure, substance, and the principles of screenwriting*. Harper Collins.

Gerald Mead. 1990. The representation of fictional character. *Style*, pages 440–452.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Risto Miikkulainen. 1995. Script-based inference and memory retrieval in subsymbolic story processing. *Applied Intelligence*, 5(2):137–163.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, pages 641–648.

Ashutosh Modi, Tatjana Anikina, Simon Ostermann, and Manfred Pinkal. 2016. InScript: Narrative texts annotated with script information. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3485–3493, Portorož, Slovenia. European Language Resources Association (ELRA).

Ashutosh Modi and Ivan Titov. 2014. Inducing neural models of script knowledge. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 49–57, Ann Arbor, Michigan. Association for Computational Linguistics.

Ashutosh Modi, Ivan Titov, Vera Demberg, Asad Sayeed, and Manfred Pinkal. 2017. Modeling semantic expectation: Using script knowledge for referent prediction. *Transactions of the Association for Computational Linguistics*, 5:31–44.

Franco Moretti. 2013. " operationalizing": or, the function of measurement in modern literary theory.

Erik T Mueller. 2004. Understanding script-based stories using commonsense reasoning. *Cognitive Systems Research*, 5(4):307–340.

Naresh Kumar Nagwani. 2015. Summarizing large text collection using topic modeling and clustering based on mapreduce framework. *Journal of Big Data*, 2(1):6.

Hiroki Nakayama. 2018. seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Eric T. Nalisnick and Henry S. Baird. 2013. Character-to-character sentiment analysis in shakespeare's plays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 479–483, Sofia, Bulgaria. Association for Computational Linguistics.

James O' Neill, Paul Buitelaar, Cecile Robin, and Leona O' Brien. 2017. Classifying sentential modality in legal language: a use case in financial regulations, acts and directives. In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*, pages 159–168.

Jonathan A Obar and Anne Oeldorf-Hirsch. 2020. The biggest lie on the internet: Ignoring the privacy policies and terms of service policies of social networking services. *Information, Communication & Society*, 23(1):128–147.

Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2230–2235, Austin, Texas. Association for Computational Linguistics.

OpenAI. 2022. Openai. 2022. introducing chatgpt.

Bryan Orme. 2009. Maxdiff analysis: Simple counting, individual-level logit, and hb. *Sawtooth Software*.

Simon Ostermann. 2020. Script knowledge for natural language understanding.

Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, and Manfred Pinkal. 2018. MCScript: A novel dataset for assessing machine comprehension using script knowledge. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Simon Ostermann, Michael Roth, and Manfred Pinkal. 2019. MCScript2.0: A machine comprehension corpus focused on script events and participants. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 103–117, Minneapolis, Minnesota. Association for Computational Linguistics.

Jiao Ou, Junda Lu, Che Liu, Yihong Tang, Fuzheng Zhang, Di Zhang, and Kun Gai. 2024. DialogBench: Evaluating LLMs as human-like dialogue systems. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Mexico City, Mexico. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Makbule Gulcin Ozsoy, Ferda Nur Alpaslan, and Ilyas Cicekli. 2011. Text summarization using latent semantic analysis. *Journal of Information Science*, 37(4):405–417.

Frank Robert Palmer. 2001. *Mood and modality*. Cambridge university press.

Orestis Papakyriakopoulos and Arwa M. Mboya. 2023. Beyond algorithmic bias: A socio-computational interrogation of the google search by image algorithm. *Social Science Computer Review*, 41(4):1100–1125.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Alison H Paris and Scott G Paris. 2003. Assessing narrative comprehension in young children. *Reading Research Quarterly*, 38(1):36–76.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Wim Peters and Adam Wyner. 2016a. Legal text interpretation: Identifying hohfeldian relations from text. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 379–384.

Wim Peters and Adam Wyner. 2016b. Legal text interpretation: Identifying hohfeldian relations from text. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 379–384, Portorož, Slovenia. European Language Resources Association (ELRA).

Karl Pichotta and Raymond Mooney. 2014. Statistical script learning with multi-argument events. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 220–229, Gothenburg, Sweden. Association for Computational Linguistics.

Karl Pichotta and Raymond Mooney. 2016a. Statistical script learning with recurrent neural networks. In *Proceedings of the Workshop on Uphill Battles in Language Processing: Scaling Early Achievements to Robust Methods*, pages 11–16, Austin, TX. Association for Computational Linguistics.

Karl Pichotta and Raymond J. Mooney. 2016b. Using sentence-level LSTM language models for script inference. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 279–289, Berlin, Germany. Association for Computational Linguistics.

Massimo Poesio, Rosemary Stevenson, Barbara Di Eugenio, and Janet Hitzeman. 2004. Centering: A parametric theory and its instantiations. *Computational Linguistics*, 30(3):309–363.

Seth Polsley, Pooja Jhunjhunwala, and Ruihong Huang. 2016. Casesummarizer: a system for automated summarization of legal texts. In *Proceedings of COLING 2016, the 26th international conference on Computational Linguistics: System Demonstrations*, pages 258–262.

Vladimir Propp. 1968. *Morphology of the Folktale*. University of texas Press.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003. Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

Valentina Pyatkin, Shoval Sadde, Aynat Rubinstein, Paul Portner, and Reut Tsarfaty. 2021. The possible, the plausible, and the desirable: Event-based modality detection for language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 953–965, Online. Association for Computational Linguistics.

Saira Qamar, Hasan Mujtaba, Hammad Majeed, and Mirza Omer Beg. 2021. Relationship identification between conversational agents using emotion analysis. *Cognitive Computation*, 13:673–687.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Altaf Rahman and Vincent Ng. 2012. Resolving complex cases of definite pronouns: The Winograd schema challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 777–789, Jeju Island, Korea. Association for Computational Linguistics.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Ruby Rani and DK Lobiyal. 2021. An extractive text summarization approach using tagged-lda based topic modeling. *Multimedia tools and applications*, 80:3275–3305.

Farzana Rashid and Eduardo Blanco. 2018. Characterizing interactions and relationships between people. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4395–4404, Brussels, Belgium. Association for Computational Linguistics.

Michaela Regneri, Alexander Koller, and Manfred Pinkal. 2010. Learning script knowledge with web experiments. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 979–988, Uppsala, Sweden. Association for Computational Linguistics.

Evan TR Rosenman, Santiago Olivella, and Kosuke Imai. 2023. Race and ethnicity data for first, middle, and surnames. *Scientific Data*.

Rachel Rudinger, Pushpendre Rastogi, Francis Ferraro, and Benjamin Van Durme. 2015. Script induction as language modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1681–1686, Lisbon, Portugal. Association for Computational Linguistics.

Rachel Rudinger, Vered Shwartz, Jena D. Hwang, Chandra Bhagavatula, Maxwell Forbes, Ronan Le Bras, Noah A. Smith, and Yejin Choi. 2020. Thinking like a skeptic: Defeasible inference in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4661–4675, Online. Association for Computational Linguistics.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2014. Efficient elicitation of annotations for human evaluation of machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 1–11.

Keisuke Sakaguchi and Benjamin Van Durme. 2018. Efficient online scalar annotation with bounded support. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 208–218, Melbourne, Australia. Association for Computational Linguistics.

Abhilasha Sancheti, Haozhe An, and Rachel Rudinger. 2024. On the influence of gender and race in romantic relationship prediction from large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Miami, Florida, USA. Association for Computational Linguistics.

Abhilasha Sancheti, Aparna Garimella, Balaji Vasan Srinivasan, and Rachel Rudinger. 2022a. Agent-specific deontic modality detection in legal language. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Abhilasha Sancheti, Aparna Garimella, Balaji Vasan Srinivasan, and Rachel Rudinger. 2023. What to read in a contract? Party-specific summarization of legal obligations, entitlements, and prohibitions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14708–14725, Singapore. Association for Computational Linguistics.

Abhilasha Sancheti and Rachel Rudinger. 2022. What do large language models learn about scripts? In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, Seattle, Washington. Association for Computational Linguistics.

Abhilasha Sancheti, Balaji Vasan Srinivasan, and Rachel Rudinger. 2022b. Entailment relation aware paraphrase generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10).

Yisi Sang, Xiangyang Mou, Mo Yu, Dakuo Wang, Jing Li, and Jeffrey Stanton. 2022a. MBTI personality prediction for fictional characters using movie scripts. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6715–6724, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yisi Sang, Xiangyang Mou, Mo Yu, Shunyu Yao, Jing Li, and Jeffrey Stanton. 2022b. TVShowGuess: Character comprehension in stories as speaker guessing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4267–4287, Seattle, United States. Association for Computational Linguistics.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.

Roger C Schank and Robert P Abelson. 1975. Scripts, plans, and knowledge. In *IJCAI*, volume 75, pages 151–157.

Roger C Schank and Robert P Abelson. 1977. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40.

Zejiang Shen, Kyle Lo, Lauren Yu, Nathan Dahlberg, Margo Schlanger, and Doug Downey. 2022. Multi-lexsum: Real-world summaries of civil rights lawsuits at multiple granularities. *Advances in Neural Information Processing Systems*, 35:13158–13173.

Matthew Sims and David Bamman. 2020. Measuring information propagation in literary social networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 642–652, Online. Association for Computational Linguistics.

Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 1223–1237. Springer.

Eliot R Smith, Diane M Mackie, and Heather M Claypool. 2014. *Social psychology*. Psychology Press.

Mark Snyder and Arthur A Stukas Jr. 1999. Interpersonal processes: The interplay of cognitive, motivational, and behavioral activities in social interaction. *Annual review of psychology*, 50(1):273–303.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Shashank Srivastava, Snigdha Chaturvedi, and Tom Mitchell. 2016. Inferring interpersonal relations in narrative summaries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Bryan Stroube. 2003. Literary freedom: Project gutenberg. *XRDS: Crossroads, The ACM Magazine for Students*, 10(1):3–3.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation and synthesis: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957, Miami, Florida, USA. Association for Computational Linguistics.

Gemini Team. 2023. Gemini: A family of highly capable multimodal models technical report, google.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Welderufael B Tesfay, Peter Hofmann, Toru Nakamura, Shinsaku Kiyomoto, and Jetzabel Serna. 2018. Privacyguide: towards an implementation of the eu gdpr on internet privacy policy evaluation. In *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics*, pages 15–21.

Peter M Tiersma. 1999. *Legal language*. University of Chicago Press.

Anna Tigunova, Paramita Mirza, Andrew Yates, and Gerhard Weikum. 2021. PRIDE: Predicting Relationships in Conversations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4636–4650, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al.

2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Don Tuggener, Pius von Däniken, Thomas Peetz, and Mark Cieliebak. 2020. LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1235–1241, Marseille, France. European Language Resources Association.

Hardik Vala, David Jurgens, Andrew Piper, and Derek Ruths. 2015. Mr. bennet, his coachman, and the archbishop walk into a bar but only one of them gets recognized: On the difficulty of detecting characters in literary texts. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 769–774, Lisbon, Portugal. Association for Computational Linguistics.

An Van Linden. 2012. *Modal adjectives: English deontic and evaluative constructions in diachrony and synchrony*, volume 75. Walter de Gruyter.

Siddharth Vashishtha, Benjamin Van Durme, and Aaron Steven White. 2019. Fine-grained temporal relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2906–2919, Florence, Italy. Association for Computational Linguistics.

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

Veniamin Veselovsky, Manoel Horta Ribeiro, Akhil Arora, Martin Josifoski, Ashton Anderson, and Robert West. 2023. Generating faithful synthetic data with large language models: A case study in computational social science.

Georg Henrik Von Wright. 1951. Deontic logic. *Mind*, 60(237):1–15.

Lilian Wanzare, Alessandra Zarcone, Stefan Thater, and Manfred Pinkal. 2017. Inducing script structure from crowdsourced event descriptions via semi-supervised clustering.

Lilian DA Wanzare, Alessandra Zarcone, Stefan Thater, and Manfred Pinkal. 2016. A crowdsourced database of event sequence descriptions for the acquisition of high-quality script knowledge.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Myron Wish, Morton Deutsch, and Susan J Kaplan. 1976. Perceived dimensions of interpersonal relations. *Journal of Personality and social Psychology*, 33(4):409.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Alex Woloch. 2009. *The One vs. the Many: Minor Characters and the Space of the Protagonist in the Novel*. Princeton University Press.

Adam Wyner and Wim Peters. 2011. On rule extraction from regulations. In *Legal Knowledge and Information Systems*, pages 113–122. IOS Press.

Zhiyang Xu, Ying Shen, and Lifu Huang. 2023. MultiInstruct: Improving multi-modal zero-shot learning via instruction tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11445–11465, Toronto, Canada. Association for Computational Linguistics.

Xianjun Yang, Kaiqiang Song, Sangwoo Cho, Xiaoyang Wang, Xiaoman Pan, Linda Petzold, and Dong Yu. 2023. OASum: Large-scale open domain aspect-based summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada. Association for Computational Linguistics.

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. 2023. Cognitive mirage: A review of hallucinations in large language models. *arXiv preprint arXiv:2309.06794*.

Mo Yu, Jiangnan Li, Shunyu Yao, Wenjie Pang, Xiaochen Zhou, Zhou Xiao, Fandong Meng, and Jie Zhou. 2023. Personality understanding of fictional characters during book reading. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14784–14802, Toronto, Canada. Association for Computational Linguistics.

Siyu Yuan, Jiangjie Chen, Ziquan Fu, Xuyang Ge, Soham Shah, Charles Jankowski, Yanghua Xiao, and Deqing Yang. 2023. Distilling script knowledge from large language models for constrained language planning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada. Association for Computational Linguistics.

Razieh Nokhbeh Zaeem, Rachel L German, and K Suzanne Barber. 2018. Privacycheck: Automatic summarization of privacy policies using data mining. *ACM Transactions on Internet Technology (TOIT)*, 18(4):1–18.

Albin Zehe, Julia Arns, Lena Hettinger, and Andreas Hotho. 2020. Harrymotions-classifying relationships in harry potter based on emotion analysis. In *SwissText/KONVENS*.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2024. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

Zhexin Zhang, Jiaxin Wen, Jian Guan, and Minlie Huang. 2022. Persona-guided planning for controlling the protagonist's persona in story generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3346–3361, Seattle, United States. Association for Computational Linguistics.

Runcong Zhao, Qinglin Zhu, Hainiu Xu, Jiazheng Li, Yuxiang Zhou, Yulan He, and Lin Gui. 2024. Large language models fall short: Understanding complex relationships in detective narratives. In *Findings of the Association for Computational Linguistics ACL 2024*, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Xian Zhao and Monica Biernat. 2019. Your name is your lifesaver: Anglicization of names and moral dilemmas in a trilogy of transportation accidents. *Social Psychological and Personality Science*, 10(8):1011–1018.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247, Florence, Italy. Association for Computational Linguistics.

Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal judgment prediction via topological learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3540–3549.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020a. How does NLP benefit legal system: A summary of legal artificial intelligence. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5218–5230, Online. Association for Computational Linguistics.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020b. Jec-qa: A legal-domain question answering dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9701–9708.

Shijia Zhou, Leonie Weissweiler, Taiqi He, Hinrich Schütze, David R. Mortensen, and Lori Levin. 2024. Constructions are so difficult that Even large language models get them right for the wrong reasons. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3804–3811, Torino, Italia. ELRA and ICCL.