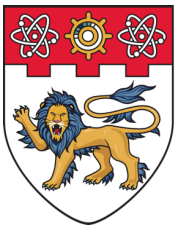


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

EVENT EXTRACTION AND BEYOND: FROM CONVENTIONAL NLP TO LARGE LANGUAGE MODELS

HANZHANG ZHOU

INTERDISCIPLINARY GRADUATE PROGRAMME

INSTITUTE OF CATASTROPHE RISK MANAGEMENT

2024

Event Extraction and Beyond: from Conventional NLP to Large Language Models

Hanzhang Zhou

Interdisciplinary Graduate Programme
Institute of Catastrophe Risk Management

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2024

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

13 August 2024
.....
Date

Hanzhang Zhou
.....
Hanzhang Zhou

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

13 August 2024

Date

Mao Kezhi

Prof. Kezhi MAO

Authorship Attribution Statement

This thesis contains material from 4 papers accepted or under review at top AI conferences in which I am listed as an author.

Chapter 3 is published as Hanzhang Zhou, and Kezhi Mao. "Document-level event argument extraction by leveraging redundant information and closed boundary loss." Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2022), pp. 3041-3052. 2022.

The contributions of the co-authors are as follows:

- I proposed the idea, constructed the model, designed and conducted the experiment, and prepared the manuscript.
- Prof. Kezhi Mao refined the proposed method, revised the manuscript.

Chapter 4 is published as Hanzhang Zhou, Junlang Qian, Zijian Feng, Hui Lu, Zixiao Zhu and Kezhi Mao. "LLMs Learn Task Heuristics from Demonstrations: A Heuristic-Driven Prompting Strategy for Document-Level Event Argument Extraction". Proceedings Of The 62ND Annual Meeting Of The Association For Computational Linguistics (ACL 2024): Long Papers. 2024.

The contributions of the co-authors are as follows:

- I proposed the idea, developed the method, designed the experiment, and prepared the manuscript.
- Junlang Qian, Hui Lu, Zixiao Zhu, and I conducted the experiments.
- Zijian Feng and I discussed the idea.
- Prof Kezhi Mao suggested the project direction and revised the manuscript.

Chapter 5 is published as Hanzhang Zhou, Zijian Feng, and Kezhi Mao. "Closed Boundary Learning for Classification Tasks with the Universum Class." Findings of the Association for Computational Linguistics: EMNLP 2023. 2023.

The contributions of the co-authors are as follows:

- I proposed the method, constructed the model, designed the experiment, and prepared the manuscript.
- Zijian Feng and I conducted the experiment.
- Prof Kezhi Mao provided the initial idea and revised the manuscript.

Chapter 6 is a papaer under review Hanzhang Zhou, Zijian Feng, Zixiao Zhu, Junlang Qian, and Kezhi Mao. "UniBias: Unveiling and Mitigating LLM Bias through Internal Attention and FFN Manipulation". Submitted to Advances in Neural Information Processing Systems (NeuIPS 2024). 2024.

The contributions of the co-authors are as follows:

- I proposed the idea, developed the method, designed the experiment, and prepared the manuscript.
- Zixiao Zhu, Junlang Qian and I conducted the experiments.
- Prof Kezhi Mao suggested the project direction and suggestions.
- Zijian Feng and Prof Kezhi Mao revised the manuscript.

13 August 2024

Date

Hanzhang Zhou

Hanzhang Zhou

Acknowledgements

As I am approaching the end of my doctoral journey, I am deeply grateful for the support and guidance I have received from many individuals who have made this achievement possible.

First and foremost, I would like to express my profound gratitude to my advisor, Professor Kezhi Mao, whose expertise, understanding, and patience added considerably to my graduate experience. His unwavering support and insightful critiques have been instrumental in shaping my research and writing. During my doctoral studies, there were times when I found myself lost in the minutiae of my research. It was during these moments that his guidance illuminates my path forward. His own relentless pursuit of knowledge and uncompromising academic integrity have profoundly shaped my research approach. He has not only been a mentor but also a role model, inspiring me to maintain the highest standards in my scholarly work.

Besides my supervisor, I would also like to thank the rest of my thesis advisory committee: Prof. Edmond Lo and Prof. Lihui Chen, for their encouragement, insightful comments, and invaluable suggestions.

I would also like to express my gratitude to my fellow lab members, Dr. Zijian Feng, Dr. Zixiao Zhu, Junlang Qian, for their collaboration and companionship throughout my doctoral studies. The stimulating discussions and the shared challenges have significantly contributed to my personal and professional growth, making my journey enjoyable I would also like to thank my seniors, Qi Li, Yuecong Xu, Jianfei Yang, who have provided me a lot of invaluable advice for research and career. The synergy within our team not only propelled our projects forward but also provided a supportive environment that made our lab a nurturing space for innovative ideas and personal growth.

Special thanks to my friends in Singapore—Zijian Feng, Zheng Liao, He Huang, Yatao Zhang, Yunpeng Xue, RongZihan Song, Zixiao Zhu, Sitan Li and to those around the globe: Haoyu Wen, Wenjie Zhou, Aichen Chen, Zhaopei Liu, Yuxuan Liu, Zhijian Weng,

Yufei Sun, Zhenjie Wang, Songyuan Geng, Dun Liu. My Ph.D. studies have been made memorable and joyful thanks to the time spent with each of them. Filled with happiness and immense strength, their camaraderie and support have been crucial to my journey, enriching my experience far beyond the academic realm.

Lastly, but most importantly, I would like to thank my family for their love and encouragement. To my parents, Quanbao Zhou, Lili Du, and my spouse, Xiyu Zhang, thank them for believing in me and standing by me through the highs and lows of academic pursuit. Their constant support and unwavering belief in my abilities have been a source of strength and motivation.

This dissertation stands as a milestone not just in my academic career but also as a testament to the collective efforts and sacrifices of all who supported me. I am truly thankful for each of you who has been part of my journey.

"Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less."

—Marie Curie

To my dear family

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

NLP	Natural Language Processing
EE	Event Extraction
IE	Information Extraction
NER	Named Entity Recognition
RE	Relation Extraction
LLMs	Large Language Models
LM	Language Model
ICL	In-Context Learning
RNNs	Recurrent Neural Networks
FFNs	Feedforward Neural Networks
SVM	Support Vector Machines
KNN	K-Nearest Neighbors
DNNs	Deep Neural Networks
CNNs	Convolutional Neural Networks
RNNs	Recurrent Neural Networks
LSTMs	Long Short-Term Memory units
GNNs	Graph Neural Networks
CoT	Chain-of-Thought
NLI	Natural Language Inference
SA	Sentiment Analysis
TSVM	Twin Support Vector Machine
OOD	Out-of-Distribution
ID	In-Distribution
KKCs	Known Known Classes
KUCs	Known Unknown Classes
UUCs	Unknown Unknown Classes

SLMs	Statistical Language Models
NLMs	Neural Language Models
PLMs	Pre-trained Language Models
MCQs	Multiple-Choice Questions
EAE	Event Argument Extraction

Abstract

The digital age has ushered in an era of unprecedented information explosion, characterized by the generation of vast volumes of text. This development has significantly heightened the demands for processing and analyzing large-scale text data. As a result, the technology of event extraction (EE) was developed to transform unstructured event information from text data into structured formats, facilitating the interpretation and application of event information in various domains.

However, extracting structured information from noisy and unstructured data presents several crucial challenges. Firstly, conventional event extraction research generally focuses on the sentence level, whereas real-world text data are typically at the document level. Thus, effectively and accurately extracting event information at the document level is both crucial and challenging. Secondly, conventional EE research necessitates a substantial amount of training data, which is particularly burdensome and costly due to the complexity inherent in EE data annotation.

Additionally, as a typical real-world task, investigating the EE task has led to identification and resolution for common problems that exist across many natural language processing (NLP) domains. For example, the Universum class, often referred to as the Other or Miscellaneous class, widely exists in the EE task and many classification-based NLP tasks. The Universum class exhibits distinct properties; however, existing works often treat it equivalently to the classes of interest. We find this treatment leads to issues such as overfitting, misclassification, and diminished model robustness. Furthermore, when applying large language models (LLMs) to EE tasks, we find that the effectiveness of LLMs is often compromised by their inherent biases, leading to issues of prompt brittleness—sensitivity to design settings such as example selection, order, and prompt formatting.

Moreover, following the revolutionary impact of LLMs since the release of ChatGPT in November 2022, our research has undergone a paradigm shift towards exploring

the capabilities of LLMs. Consequently, this thesis incorporates both traditional NLP methods based on supervised learning and the latest paradigms utilizing LLMs.

In this thesis, several works are presented to address the aforementioned challenges:

- A document-level event argument extraction method utilizing graph neural networks. This method leverages redundant event information within documents along with coreference information to enhance accuracy in document-level EE.
- A prompting strategy tailored for EE to alleviate the need for large-scale labeled data. We explore what LLMs learn from in-context learning (ICL), finding that LLMs learn task heuristics from demonstrations via ICL. Based on this insight, we propose a novel heuristic-driven prompting strategy.
- A closed boundary learning framework designed to address the unique properties of the Universum class in classification tasks. We highlight an understudied problem regarding the Universum class. Then, we propose a method that applies closed decision boundaries to classes of interest and designates the area outside all closed boundaries in the feature space as the space of the Universum class.
- An inference-only method that addresses the inherent bias of LLMs. We investigate how feedforward neural networks (FFNs) and attention heads result in the bias of LLMs. To mitigate these effects, we introduce the UniBias method, which effectively identifies and eliminates biased FFN vectors and attention heads.

Chapter 1

Introduction

1.1 Overview and Motivation

The digital age has ushered in an era of unprecedented information explosion, characterized by the generation of vast volumes of text. This development has significantly heightened the demands for processing and analyzing large-scale text data. As a result, the technology of *information extraction* (IE) was developed from the end of the last century [14]. IE aims to automatically extract structured information such as entities, relations, and events from unstructured text [15]. With the technique of information extraction, people can automatically obtain the factual information of interest from vast volumes of data that are beyond human capacity to process manually. However, the scope of the IE task is exceedingly broad. Researchers in the natural language processing (NLP) community have further divided the IE task into three sub-tasks: named entity recognition (NER), relation extraction (RE), and event extraction (EE).

EE presents the most challenging and comprehensive task of IE. EE specifically aims to transform unstructured event information from text data such as news articles, reports, and social media posts into structured formats [14]. EE facilitates the interpretation and application of event information in various domains. For example, these structured event information can be utilized to build knowledge bases to facilitate decision making [16]. In disaster response, EE can swiftly identify events like natural disasters or accidents from news and social media, enabling faster response times and more effective resource distribution [17, 18]. In the financial sector, EE helps extract critical data about corporate events such as mergers, acquisitions, and earnings announcements, providing valuable

insights for informed investment decision making and risk assessment [19, 20]. Similarly, in the supply chain domain, EE can be used to monitor news and update relevant to logistics, such as strikes, shipping delays, or other issues that could impact business operations [21].

Over the past two decades, the methodologies for EE have undergone significant evolution, transitioning from rule-based approaches to deep learning methods. Early EE systems primarily utilized rule-based systems that relied heavily on linguistic cues and pattern matching techniques to extract structured information from unstructured texts. While this approach provided a good starting point, it was inherently limited by the extensive manual effort in rule design and their inability to generalize beyond the specific designed scenarios. The advent of deep learning brought about a paradigm shift in EE, introducing models that could learn from data instead of following hard-crafted rules. This transition enabled the EE systems to be more easily adapted to different scenarios and achieve better accuracy. Techniques such as recurrent neural networks (RNNs) became popular due to their ability to capture complex patterns in text. Most recently, the emergence of large language models (LLMs) has further revolutionized NLP research, offering promising new directions for future EE systems.

However, EE research faces significant challenges:

- **Insufficient Accuracy**: Compared with other IE tasks like NER and RE, EE systems exhibit significantly lower accuracy, which is defined as the percentage of correctly identified events argument relative to the total event arguments analyzed. This reduced accuracy is attributed to the task's inherent complexity. EE requires a comprehensive understanding of the text and involves the interaction between the event trigger and multiple arguments. Thus, a crucial challenge for the EE task is to enhance accuracy.

- **Document-Level EE**: The inherent complexity of EE often confines research to the sentence level, where the focus is on extracting event information from individual sentences. However, real-world applications generally require extracting event information at the document level. This transition to document-level extraction presents significant difficulties, such as the need for broader context comprehension, management of scattered arguments, and handling multiple events within documents.

- **Dependence on Annotated Data**: EE heavily relies on supervised learning, which necessitates substantial volumes of annotated training data. Collecting and annotating such data is not only labor-intensive but also costly, compounded by the intricate requirements of EE data annotations. The dependency on large scale labeled data poses significant hurdles in the development and scalability of EE systems in real-world applications.

On the other hand, as a typical real-world task, investigation on the event extraction task is accompanied by common problems that exist across many NLP domains. For example, the Universum class, often referred to as the *Other* or *Miscellaneous* class, is defined as a collection of samples that do not belong to any class of interest. The Universum class widely exists in the EE task and many classification-based NLP tasks. The Universum class exhibits distinct properties; however, existing works often treat it equivalently to the classes of interest. We find this treatment leads to issues such as overfitting, misclassification, and diminished model robustness.
Additionally, when applying large language models to EE tasks, we find that LLMs are biased towards predicting certain answers. As a result, the effectiveness of LLMs is often compromised by this inherent bias, leading to issues of prompt brittleness—sensitivity to design settings such as example selection, order, and prompt formatting.

In order to address the aforementioned challenges, several works are presented in this thesis:

- A graph neural network based method for document-level EE. This method leverages redundant event information within documents along with coreference information to enhance accuracy in document-level EE.

- A prompting strategy tailored for EE to alleviate the need for large-scale labeled data. We explore what LLMs learn from in-context learning (ICL), finding that LLMs learn task heuristics from demonstrations via ICL. Based on this insight, we propose a novel heuristic-driven prompting strategy.

- A closed boundary learning framework designed to address the unique properties of the Universum class in classification tasks. We highlight an understudied problem regarding the Universum class. Then, we propose a method that applies closed decision boundaries to classes of interest and designates the area outside all closed boundaries in the feature space as the space of the Universum class.

- An LLM structure manipulation method that addresses the inherent bias of LLMs. We investigate how feedforward neural networks (FFNs) and attention heads result in the bias of LLMs. To mitigate these effects, we introduce the UniBias method, which effectively identifies and eliminates biased FFN vectors and attention heads.

In summary, this thesis aims to address the challenges of the EE task to achieve more accurate and practically applicable event extraction. Furthermore, by tackling challenges that are not only prevalent in EE but also common across various NLP tasks, this thesis extends its benefits to a broader context.

1.2 Scope and Objectives

To clarify the scope covered in this thesis, the following elaboration is provided:

Algorithmically, this thesis explores a range of machine learning algorithms, including both conventional NLP methods based on supervised learning and the latest paradigms involving LLMs. Specifically, we explore conventional NLP methods that employ supervised learning via algorithms such as Recurrent Neural Networks (RNNs), Transformers, Graph Neural Networks, and attention mechanisms. Additionally, we delve into the latest advancements in LLMs, particularly focusing on the in-context learning paradigm. These methodologies represent the forefront of NLP research.

Task-wise, the primary focus of this thesis is the EE task. Beyond EE, this research addresses several fundamental issues that exist not only in EE but also in diverse NLP scenarios. These include challenges like the Universum class and bias inherent in LLMs. Consequently, more diverse tasks are explored to delve into these foundational issues, encompassing Named Entity Recognition, Relation Extraction, Natural Language Inference, Common Sense Reasoning, and Sentiment Analysis.

The objectives of this thesis are twofold:

- Address the challenges of the EE task to achieve more accurate and practically applicable EE systems. Specifically, we aims to propose algorithms to enhance the performance of EE on public available EE datasets. Additionally, to enable EE systems more suitable for real-world applications, we aims to investigate

algorithms for document-level problem setting and alleviate the dependency on large-scale labeled data of the task.

- Address important issues that are not only prevalent in EE but also extend across the broader NLP context. Specifically, we explore the handling of the Universum class in classification problems and the mitigation of inherent biases in LLMs. These investigations aim to yield benefits for a broader range of NLP applications.

1.3 Main Contributions

The main contributions of this thesis are listed as follows:

- **Document-Level Event Argument Extraction Method**. Existing research works in EE are mostly restricted to sentence level. However, events are often described in the forms of document in real world. In this sense, we propose a method tailored for document-level EE. Specifically, we observe that in document-level event argument extraction, an argument is likely to appear multiple times in different expressions in the document. The redundancy of arguments underlying multiple sentences is beneficial but is often overlooked. To make use of redundant event information underlying a document, we build an entity coreference graph with the graph2token module to produce a comprehensive and coreference-aware representation for every entity and then build an entity summary graph to merge the multiple extraction results. In addition, in EE, most entities are regarded as the "*Others*" class, composed of entities existing in the document but not event argument. This class is composed of heterogeneous entities without typical common features. Classifiers trained by cross entropy loss could easily misclassify the Universum class because of their open decision boundary. To better classify the Universum class, we propose a new loss function.

- **A Pioneering Work on Prompting Strategy Tailored for EE**: Conventional supervised learning methods for EE necessitate a substantial amount of training data, which is particularly burdensome and costly given the complexity inherent to document-level EE. To alleviate the need for large-scale annotation data, we employ in-context learning (ICL) to EE as it only uses a few examples as input-output pairs of the prompt to guide LLMs in performing the task on an unseen example.

We introduce the Heuristic-Driven Link-of-Analogy (HD-LoA) prompting tailored for the EE task. Specifically, we hypothesize and validate that LLMs learn task-specific heuristics from demonstrations in ICL. Building upon this hypothesis, we introduce an explicit heuristic-driven demonstration construction approach, which transforms the haphazard example selection process into a systematic method that emphasizes task heuristics. Additionally, inspired by the analogical reasoning of human, we propose the link-of-analogy prompting, which enables LLMs to process new situations by drawing analogies to known situations, enhancing their performance on unseen classes beyond limited ICL examples. We introduce a pioneering work to leveraging large language models on EE.

- **Closed Boundary Learning for Classification Tasks with the Universum Class**. As we discovered in the EE task, the Universum class, often known as the *other* class or the *miscellaneous* class, is defined as a collection of samples that do not belong to any class of interest. It is a typical class that exists in many classification-based tasks in NLP, such as EE, relation extraction, named entity recognition, sentiment analysis, etc. The Universum class exhibits very different properties, namely heterogeneity and lack of representativeness in training data; however, existing methods often treat the Universum class equally with the classes of interest, leading to problems such as overfitting, misclassification, and diminished model robustness. In this work, we propose a closed boundary learning method that applies closed decision boundaries to classes of interest and designates the area outside all closed boundaries in the feature space as the space of the Universum class. Specifically, we formulate closed boundaries as arbitrary shapes, propose the inter-class rule-based probability estimation for the Universum class to cater to its unique properties, and propose a boundary learning loss to adjust decision boundaries based on the balance of misclassified samples inside and outside the boundary. In adherence to the natural properties of the Universum class, our method enhances both accuracy and robustness of classification models.

- **Unveiling and Mitigating LLM bias**. LLMs have demonstrated impressive capabilities in various tasks using the in-context learning paradigm. However, their effectiveness is often compromised by inherent bias, leading to prompt brittleness—sensitivity to design settings such as example selection, order, and prompt formatting. Previous studies have addressed LLM bias through external adjustment of model outputs, but the internal mechanisms that lead to such bias remain unexplored. Our work delves into these mechanisms, particularly investigating

how FFNs and attention heads result in the bias of LLMs. By Interpreting the contribution of individual FFN vectors and attention heads, we identify the biased LLM components that skew LLMs' prediction toward specific labels. To mitigate these biases, we introduce UniBias, an inference-only method that effectively identifies and eliminates biased FFN vectors and attention heads.

1.4 Organization of the Thesis

This thesis is organized as follows:

Chapter 1 provides an overview of event extraction and outlines the inherent limitations of this task. Furthermore, two important issues that are not only prevalent in EE but also extend across the broader NLP context are identified. These issues collectively motivate this research work. Additionally, the scope, objectives, and key contributions of the thesis are also introduced.

Chapter 2 presents a comprehensive literature review covering event extraction (EE), the Universum class, and large language models (LLMs). Specifically, this chapter outlines key terminologies, problem settings, and dominant learning algorithms associated with event extraction. It then details the terminologies relevant to the Universum class, classification tasks that involve this class, and various Universum learning methods. Finally, the chapter reviews large language models, focusing on the evolution of language models, the in-context learning paradigm, biases inherent in LLMs, and approaches to interpreting LLMs.

The following chapters are divided into two parts. Part 1, comprising Chapter 3 and Chapter 4, introduces our contributions towards developing more accurate and practically applicable event extraction systems. Part 2, consisting of Chapter 5 and Chapter 6, discusses our contributions to addressing two important issues that are prevalent not only in EE but also widely exist across broader NLP scenarios.

Chapter 3 introduces a method for document-level event argument extraction. This method leverages redundant event information within documents along with coreference information to enhance accuracy in document-level EE. Additionally, a new loss function is proposed to properly treat the *Others* class for EE task.

Chapter 4 presents a heuristic-driven prompting strategy tailored for EE to alleviate the need for large scale annotation data of the task. Specifically, we hypothesize and validate that LLMs learn task-specific heuristics from demonstrations in ICL. Building upon this hypothesis, we introduce an explicit heuristic-driven demonstration construction approach, which transforms the haphazard example selection process into a systematic method that emphasizes task heuristics.

Chapter 5 introduces a closed boundary learning framework designed to address the unique properties of the Universum class in classification tasks. We analyze the unique properties of the Universum class and the problems resulted of current algorithms. In this work, we propose a closed boundary learning method that applies closed decision boundaries to classes of interest and designates the area outside all closed boundaries in the feature space as the space of the Universum class.

Chapter 6 presents a method to address the inherent bias of LLMs via internal structure manipulation. LLM bias has been identified as the root cause of prompt brittleness, which significantly undermines the robustness and adaptability of LLMs in diverse applications. We delve into the internal mechanisms of bias within LLMs and examine how specific components, such as FFNs and attention heads, contribute to this bias. To mitigate these effects, we introduce the UniBias method, which effectively identifies and eliminates biased FFN vectors and attention heads.

Chapter 7 summarizes the thesis and provides discussions on future research.

Chapter 2

Literature Review

2.1 Overview of Event Extraction

2.1.1 Terminologies

Event extraction (EE) aims to detect the existence of an event within the text and, if present, to discover event-related information from the text [16]. Essentially, EE converts unstructured event information into a structured format, enabling diverse downstream applications that leverage this structured information. Event structures are usually predefined, including both event types and event argument roles. The terminologies of EE are defined as follows [22]:

- **Event mention:** a phrase or sentence describing an event, including a trigger and several arguments.
- **Event trigger:** the main word that most clearly expresses an event occurrence, typically a verb or a noun.
- **Event argument:** an entity mention, temporal expression or value that serves as a participant or attribute with a specific role in an event.
- **Argument role:** the relationship between an argument and the event in which it participants.

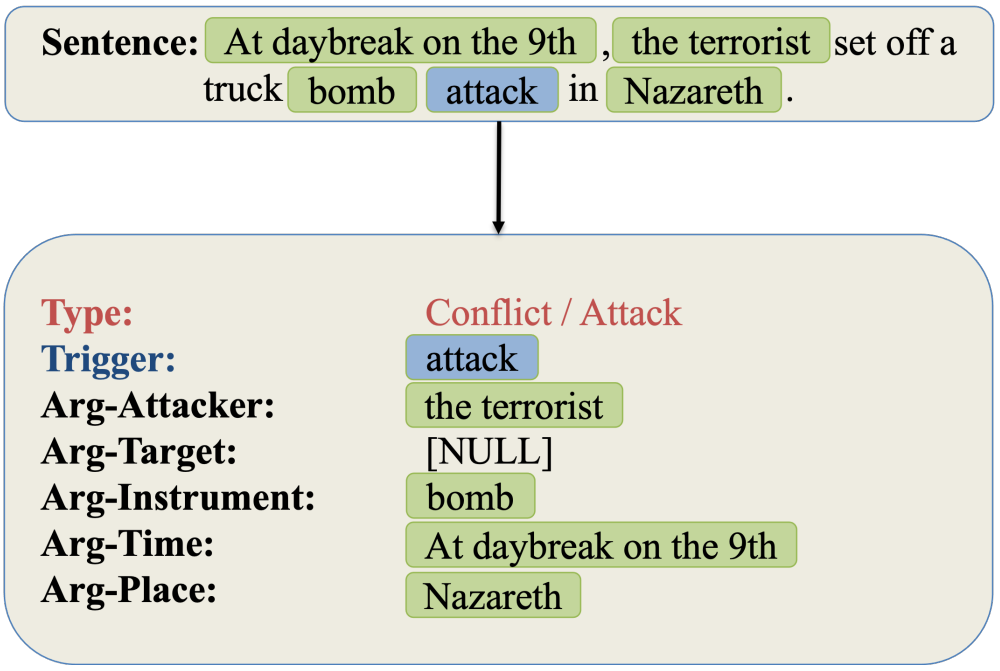


FIGURE 2.1: Illustration of event extraction task.

For example, as illustrated in Figure 2.1, given an unstructured sentence related to the *Conflict/Attack* event, the task of event extraction (EE) involves identifying and extracting the participants of the event, referred to as event arguments, from the text.

2.1.2 Problem Setting

The problem setting for EE is typically categorized into two different levels: sentence-level event extraction and document-level event extraction.

Sentence-Level Event Extraction: This level presents a simplified setting, focusing specifically on extracting event arguments from a single sentence [2, 23, 24]. It involves identifying particular event components within the sentence that collectively constitute an event. The exploration of sentence-level EE serves as a foundational step for further research into more complex, document-level tasks.

Document-Level Event Extraction: In contrast, document-level EE presents a more complex and realistic challenge that aligns closely with real-world applications. This setting involves identifying event arguments that may be spread across multiple sentences or throughout an entire document [25–28]. It requires a deep understanding of the document's global information and the ability to link details scattered across the text.

2.1.3 Learning Algorithms for Event Extraction

2.1.3.1 Pattern-Based Systems

Early methods for event extraction were based on rules crafted by domain experts. These pattern-based systems typically required extensive manual effort to develop the patterns, which were tailored to specific types of events and their linguistic expressions in text. AutoSlog is a pioneering pattern-based extraction system [29]. AutoSlog utilized a small set of linguistic patterns alongside a manually annotated corpus to efficiently derive event patterns. For example, "<subject> passive-verb" would be a linguistic pattern for extracting the *victim* "<victim> was murdered". Building on the foundation laid by AutoSlog, subsequent pattern-based systems have been developed across various domains, including biomedical event extraction [30, 31] and financial event extraction [32]. However, pattern-based systems often struggled with generalization and scalability, making them less effective for broad and real-time applications,

2.1.3.2 Traditional Machine Learning

Later approaches for event extraction (EE) utilize traditional machine learning algorithms such as Support Vector Machines (SVM) and K-Nearest Neighbors (KNN). These approaches typically involve learning classifiers based on labeled training data, which are then applied to the target text. The classifiers are developed using a variety of text features, including lexical features, syntactic features, semantic features, frequency features, unigram features, etc. Various classifiers are then learned using different algorithms, such as the nearest neighbors algorithm used in Ahn [33], the maximum entropy method employed in Chieu and Ng [34], and the SVM approach in Hong et al. [35].

These traditional machine learning methods form the backbone of many EE systems, offering strategies for feature extraction and data classification. However, they typically require extensive preprocessing and careful feature engineering to achieve optimal results. Deep learning approaches, which will be introduced in the following section, address some of these limitations by learning feature representations directly from raw data.

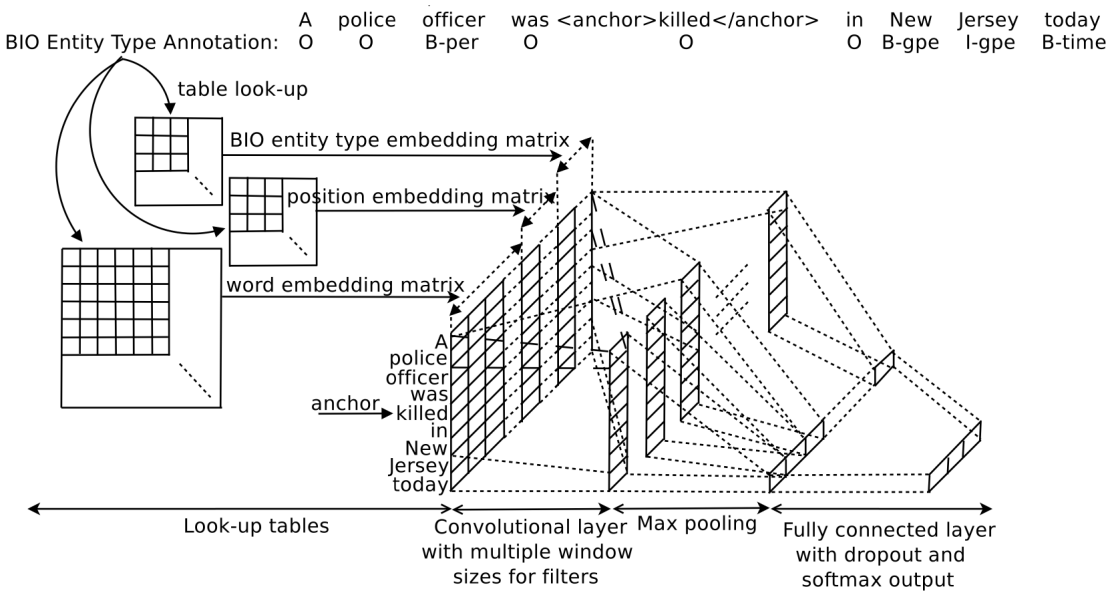


FIGURE 2.2: Illustration of the use of CNN in event extraction in [1]

2.1.3.3 Deep Learning

With the development of deep learning, deep neural networks (DNN) have been widely applied in NLP tasks due to its ability of automatically learn complex feature representations from large volumes of raw data. These models leverage architectures like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory units (LSTMs) and more recently, Transformers, which have significantly enhanced the accuracy and efficiency of EE systems. Deep learning approaches are particularly adept at capturing the complex patterns of language, such as semantics, through multiple layers of deep neural network, which allows them to generalize well across different contexts without the need for explicit feature engineering.

The general process of applying deep learning for EE is to convert text into vectors through word embeddings, and let DNN takes the word embeddings as input and output the classification results for event arguments. The main challenge in deep learning based methods is how to design an efficient DNN. Therefore, various DNN models are applied and they are summarized below.

2.1.3.4 Convolutional Neural Networks

The convolutional neural network, originally proposed by [36], consists multiple layers that include convolutional layers, pooling layers, and fully connected layers. In the

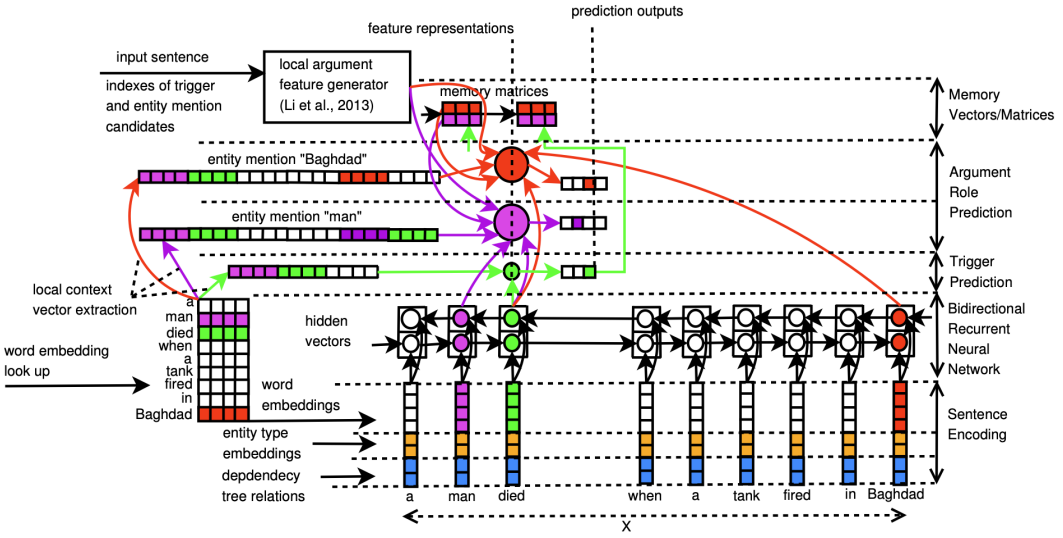


FIGURE 2.3: Illustration of the use of RNN in event extraction in [2]

convolutional layers, filters are applied to the input to extract salient features. Pooling layers then reduce the dimensionality of each feature map, preserving the most important information. Finally, the fully connected layer computes the classification results based on the features extracted by earlier layers. Due to its ability to capture both syntactic and semantic features of text, CNNs have become widely utilized in NLP tasks.

For instance, in the study by [1], CNNs are employed for the EE task to automatically extract features from input texts. As illustrated in Figure 2.2, the input to the model consists of concatenated word embeddings, position embeddings, and entity type embeddings from a sentence. The CNN learns hidden features through multiple convolutional and max pooling layers, with the fully connected layer ultimately generating the classification results for event extraction. Other researchers have employed CNNs to EE with various modifications, such as replacing the max pooling layer with a dynamic multi-pooling layer [37] and incorporating a semantic layer to better capture contextual information [38].

2.1.3.5 Recurrent Neural Networks

While CNNs excel at capturing contextual relationships among adjacent words, they are limited by their filter window size, which can restrict their ability to capture long-distance dependencies in text. In contrast, Recurrent Neural Networks [39] are particularly adept at handling such dependencies, making them ideal for processing sequential input like text. In an RNN, each neuron receives input not only from the input in the current step but

also from the hidden state in the previous step. This structure effectively creates a memory of previous inputs within the network, which is maintained through hidden layers. In the context of EE, RNNs are very useful because they can process the full context of an input text in one pass, thereby identifying relationships and dependencies that span across long stretches of text. This capability allows for more accurate identification of complex event structures.

For example, Nguyen et al. [2] adapt two RNNs that run over the input sentence in both forward and reverse directions to automatically learn representations for inputs. The learned representations are used to predict event triggers and argument roles. The overall framework of their method are illustrated in Figure 2.3. In other works, Sha et al. [23] propose a dbRNN that involve syntactic dependent information in RNN model by adding syntactic dependent connection of RNN neurons in the model, Li et al. [40] adding entity ontological knowledge into a Tree-LSTM [41] to better encode external background knowledge.

2.1.3.6 Graph Neural Networks

Graph Neural Networks (GNNs) are designed to process data that is represented as graphs, making them highly effective for applications involving relational information. The structure of a GNN includes nodes and edges. Each node within the graph encodes specific features, and edges denote the relationships between these nodes. GNNs operate by aggregating features from neighboring nodes through successive layers, allowing them to perform operations in non-Euclidean spaces. Additionally, convolutional and recurrent kernels can be applied within the graph structure to learn hidden features embedded within the graph.

In the context of EE, Xu et al. [24] propose a heterogeneous interaction graph network to capture interactions between entity mention node and sentence node, as demonstrated in Figure 2.4. This model enhances the ability to analyze contextual relationships within texts. Similarly, Liu et al. [3] employ syntactic dependencies as edges in a graph convolutional neural network, introducing syntactic shortcuts that improve information flow and enhance the model's ability to capture long-range dependencies. Huang et al. [42] use a hierarchical graph representation encoded by Graph Edge-conditioned Attention Networks. This approach incorporates domain knowledge from the Unified Medical Language System into the event extraction model. Christopoulou et al. [43] utilizes

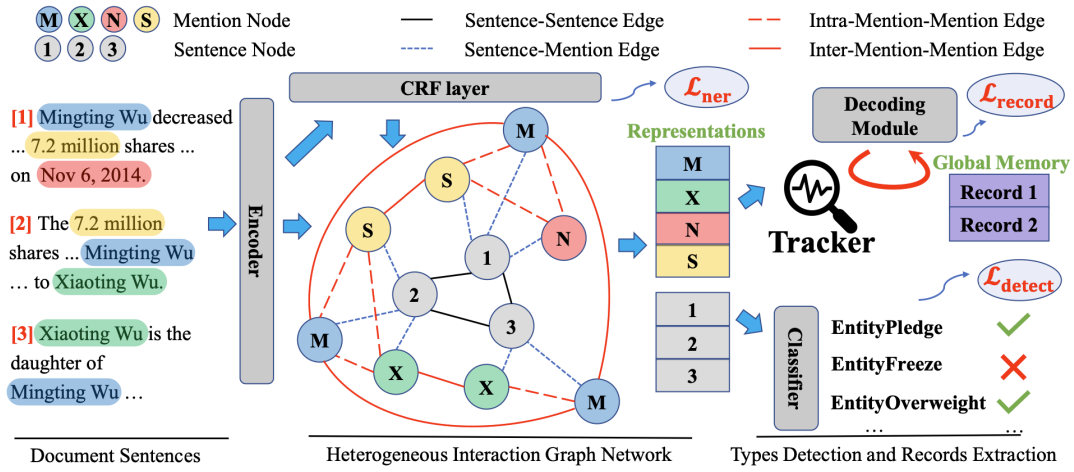


FIGURE 2.4: Illustration of the use of GNN in event extraction in [3]

various types of nodes and edges to construct a document-level graph, enabling the model to learn intra- and inter-sentence relations through internal multi-instance learning.

2.1.3.7 Transformers

Transformers, introduced by Vaswani et al. [7], feature a unique architecture that distinguishes them significantly from traditional RNNs and CNNs. A key innovation of the Transformer architecture is the self-attention mechanism, which assesses the relevance of each part of the input data to other parts, enabling the model to focus on important features without being constrained by the order of input sequence. This mechanism allows Transformers to process input data in parallel and capture complex dependencies within the data. Transformers also incorporate feed-forward neural networks, normalization steps, and residual connections in the model structure.

Transformers have been widely adopted across a range of NLP tasks. Unlike RNNs or CNNs, Transformers manage sequences in a non-sequential manner, which makes them particularly efficient at modeling long-range dependencies and parallelizing computations. Transformers are also adopted in pretrained language models like BERT, GPT, and RoBERTa, which have been applied extensively in various NLP applications.

In the domain of event extraction, several innovative applications of Transformers have been proposed. For instance, Liang et al. [25] introduce a relation-augmented attention Transformer designed to manage multi-scale and multi-amount relations in document-level event extraction. Zhang et al. [44] develop a syntax-guided graph Transformer

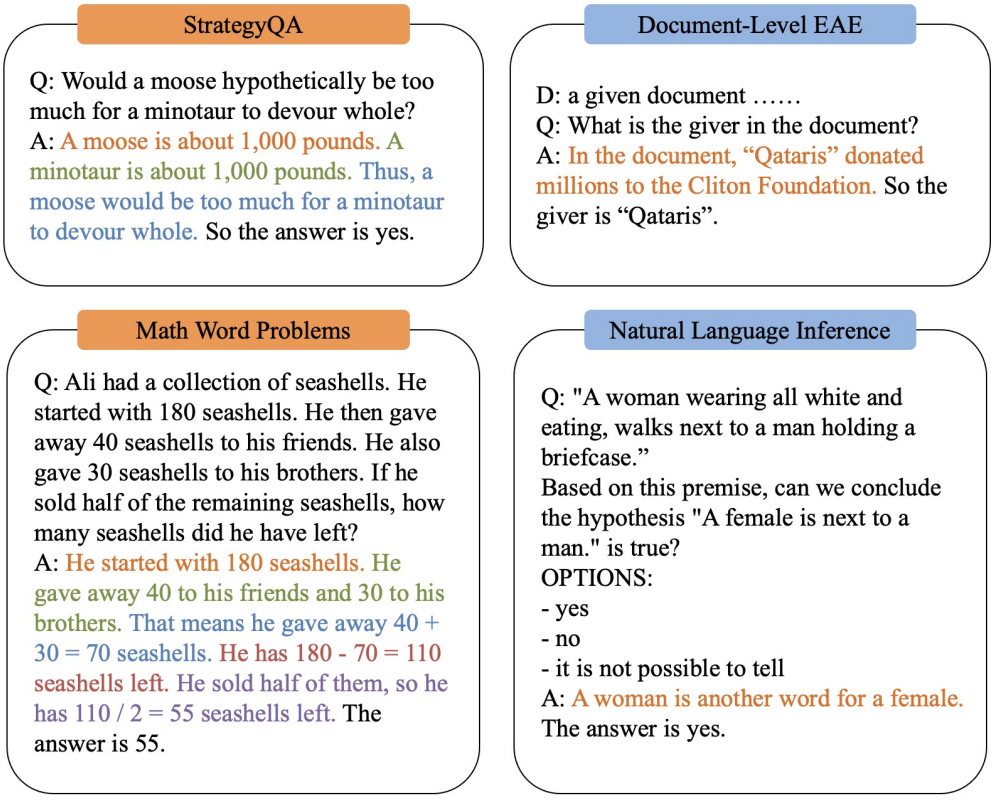


FIGURE 2.5: CoT's step-by-step reasoning degrades to a single step for non-reasoning tasks. Reasoning steps of reasoning tasks (in orange) and non-reasoning tasks (in blue) are compared. Different colors indicate distinct reasoning steps.

network that addresses the challenge of scattered events in text and use a pretrained Transformer model to generate contextual representations for input tokens. Additionally, Transformer-based pretrained language models are increasingly being utilized to generate token representations in event extraction, as demonstrated in works by Hsu et al. [45], Ma et al. [46], Xu et al. [47], Shi et al. [48].

2.1.3.8 Large Language Models

Recent advancements in LLMs have significantly revolutionized the field of natural language processing. LLMs are bringing us closer to the goal of task-agnostic machine learning [5, 49, 50]. Rather than training models on new tasks, LLMs can be directly applied to new tasks without parameter updates, using a paradigm known as in-context learning (ICL) [51, 52]. This breakthrough raises questions about the performance and potential of LLMs in the context of event extraction.

However, despite these advancements, there remains a scarcity of prompting strategies specifically tailored for EE. While the chain-of-thought (CoT) prompting is extensively used across a variety of tasks, its effectiveness is compromised in non-reasoning scenarios, such as event extraction. As shown in Figure 4.1, applying CoT to non-reasoning tasks will degrade its step-by-step reasoning into a potentially inadequate single-step. Addressing this gap, we introduce the first work that develops a prompting strategy tailored for the EE task.

2.2 The Universum Class

2.2.1 Terminologies

In the document-level event extraction task, the *other* class, or *Universum* class, is commonly observed. As illustrated in Figure 2.6, a typical document contains dozens of entities, yet only a small fraction are event arguments. The majority of these entities fall into the *other* class. This class exhibits a distinct pattern compared to targeted classes: it represents all entities that are irrelevant to the specific tasks at hand. Furthermore, the *other* class often significantly outnumbers the target classes, leading to a severe class imbalance. Beyond the observation in event extraction tasks, we further notice that such a class is prevalent across various NLP tasks, particularly in real-world applications. This widespread occurrence highlight the importance of managing the *Universum* class to improve model performance and accuracy in diverse linguistic scenarios.

In classification-based tasks, it is also common to encounter a class named as *other*, *miscellaneous*, *neutral*, or *outside (O)* class. Such a class is a collection of samples that do not belong to any class of interest, such as samples of *no relation* class in relation extraction task. We adopt the terminology in [53] to designate all such classes as the *Universum class* (U). Universum class exists not only in the task of event extraction, but also in various classification-based problems, such as relation extraction (RE) [54], named entity recognition (NER) [55], sentiment analysis (SA) [55], and natural language inference (NLI) [56]. For example, in the SemEval 2010 Task 8 dataset, a commonly used dataset for relation extraction, nearly 20% of the data belongs to the "Other" category, which accounts for the largest proportion of the whole 19 classes. To distinguish the Universum class and the rest of the classes, we call the classes of interest as *target classes* (T). The set of all classes (A) in the data can be expressed as $A = U \cup T$

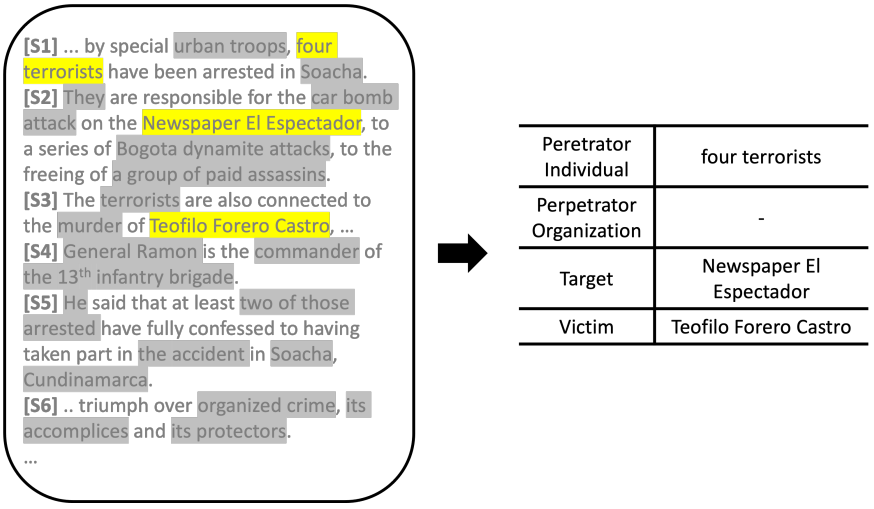


FIGURE 2.6: The presence of the Universum class in the event extraction task. Most entities in the document (highlighted in grey) belong to the *Universum* (*Other*) class, while only a few entities (highlighted in yellow) are identified as event arguments.

- *Universum class*: A collection of samples that do not belong to any class of interest.
- *Target class*: A class of interest in the task, i.e., one of the classes other than the Universum class.

The composition of the Universum class and target classes are very different, resulting distinct properties. We further highlight the differences between the Universum class and target classes in two properties, namely heterogeneity and lack of representativeness in training data, which are detailed in Chapter 5.

2.2.2 Classification Tasks with the Universum Class

The Universum class widely exists in classification based tasks in NLP, and we summarize the statistics regarding its existence in various datasets in Table 2.1. It should be noted that the span-based methods [57, 58] enumerate all possible spans for classification, which introduces an extra *other* class. Additionally, the Universum class has various names such as *other* and *miscellaneous*, etc. For example, in sentiment analysis, the *neutral* class can be considered as the Universum class because the word *neutral* is defined as "having no strongly marked or positive characteristics or features", which means the *neutral* class is a collection of all samples without strong emotions.

Task	Dataset	Label Name	Proportion
EE	MUC-4 [64]	Other	>90%
RE	SemEval 2010 Task 8 [65]	Other	17.4%
RE	TARCED [54]	No relation	79.5%
NER	CoNLL-2003 [55]	Miscellaneous	14.6%
NER (span-based)	CoNLL-2003 [55]	Other	>90%
ACSA	MAMS [66]	Neutral	43.4%

TABLE 2.1: The existence of the Universum class in classification tasks.

Despite the unique properties of the Universum class, current works [12, 13, 57, 59–63] typically approach classification problems involving the Universum class as standard multi-class classification issues, treating the Universum and target classes identically. This common treatment leads to significant problems, including overfitting, misclassification, and reduced model robustness. To address these issues, we introduce a closed boundary learning method that distinctly treats the Universum class, separating it from the target classes. Our method is detailed in Chapter 5.

2.2.3 Universum Learning Methods

Although we adopt the terminology of Universum [53], the problem setting of our work in Chapter 5 is very different from that of previous studies on Universum learning [53, 67–71].

The concept of Universum learning involves utilizing external, unlabeled Universum data to enhance the accuracy of supervised tasks. This idea has inspired various research efforts to leverage such data for classification purposes. Weston et al. [53] pioneered the use of external Universum data by maximizing the contradictions between labeled data and Universum data. Qi et al. [68] developed a Twin Support Vector Machine (TSVM) incorporating Universum to improve TSVM performance by exploiting the prior knowledge embedded in Universum data. Similarly, Shen et al. [70] introduced a boosting algorithm that extends maximal contradiction on Universum data to the realm of boosting learning, and Richhariya and Tanveer [71] utilized prior information in EEG signal classification by proposing a Universum Support Vector Machine.

However, the Universum class problem investigated in our study differs from these existing Universum learning approaches in several key aspects. In our case, the Universum class is one of the internal, labeled classes within a multi-class classification framework, rather than as a set of external, unlabeled data typically used in Universum learning studies. Methodologically, previous Universum learning methods often employ open-boundary classifiers [69] or unable to distinguish Universum samples from labeled samples [53, 67, 68], neither of which are are appropriate to the problem we presented.

2.2.4 Closed Boundary Learning Methods

We adopt the closed boundary for the classification of tasks associated with the Universum class. Therefore, we review the closed boundary learning methods in the literature in this section.

2.2.4.1 Generalized OOD Detection

Closed boundaries are commonly used in research fields such as out-of-distribution (OOD) detection [72–74], open set recognition [75, 76], anomaly detection [77], and outlier detection [78, 79]. We adopt the term "generalized OOD detection" from [80] to encapsulate these problems and to distinguish them from our proposed classification with the Universum class problem. These topics are defined as follows:

- **Anomaly Detection**: Anomaly detection involves identifying anomalous samples that deviate from what is considered normal during testing [81].
- **Novelty Detection**: Novelty detection aims to recognize test data that differs significantly from the data seen during training [82]. Both anomaly detection and novelty detection are often treated as one-class classification problems.
- **Open Set Recognition**: Open set recognition addresses the challenge of encountering new, previously unseen classes during testing. It requires classifiers to accurately identify both these new classes and the original classes seen during training [83].
- **Out-of-Distribution Detection**: This involves detecting test samples that come from a distribution different from that of the training distribution [80].

- **Outlier Detection**. Outlier detection targets to detect samples that exhibit patterns that do not conform to expected behavior [84].

Due to the similarities across these topics, we further summarize the methodologies for them together:

- **Classification-based Methods**: Initially, OOD detection relies on using the softmax confidence score as an indicator, classifying a sample as OOD if its softmax confidence score is low [85]. Recognizing the problem of overconfident posterior distributions for OOD data, Liu et al. [86] propose using the energy score for OOD detection, where samples with lower energies are considered OOD. Furthermore, Sun et al. [87] identify that OOD data could trigger unit activation patterns significantly different from in-distribution (ID) data. To address this, they introduce the ReAct approach, which utilizes activation truncation to mitigate overconfidence in OOD predictions. Additionally, Liang et al. [88] observe that temperature scaling and adding small perturbations to the input can help separate the softmax score distributions between in- and out-of-distribution samples, leading to the development of the training-based OOD detection method.

- **Density-based Methods**: Density-based methods focus on modeling in-distribution samples with probabilistic models and marking test data in low-density regions as OOD. Various approaches are used to model the probabilistic distributions, including Gaussian distribution [89], flow-based methods [90, 91], likelihood ratios [92], and likelihood regret [93].

- **Distance-based Methods**: The underlying principle of distance-based methods is that OOD samples are typically located far from the center of in-distribution samples within the feature space. Various studies leverage the distance between a test sample and the centroids of in-distribution data in the representation space as an indicator for OOD detection. This distance is assessed using several estimation techniques, including Mahalanobis distance [73], cosine similarity between representations of in-distribution data [74], radial basis function kernel [94], Euclidean distance [95], and geodesic distance [72].

2.2.4.2 Difference in Problem Setting

In this section, we highlight the distinctions between the classification problem involving the Universum class that we propose, and the existing generalized OOD detection problem.

Classification tasks can be categorized into problems based on closed-world assumption and open-world assumption [80]. The generalized OOD detection is treated under the open-world assumption, while the classification problem with the Universum class is treated under the closed-world assumption.

- *Closed-world assumption*: The test data is assumed to be drawn from the same distribution as the training data, known as in-distribution (ID).
- *Open-world assumption*: In a real-world environment, out-of-distribution (OOD) samples that are not covered in the training data are expected in the test data.

In addition, the OOD samples are not available in the training data in generalized OOD detection, whereas a considerable number of Universum samples are included in the training data in our problem setting. Leveraging information of existing Universum samples is important to generate accurate decision boundaries in our problem.

As shown in the Table 2.2, we further adopt the terminologies in [83] to illustrate the difference in problem setting.

- Known Known Classes (KKCs) are those with distinctly labeled positive training samples.
- Known Unknown Classes (KUCs) refer to labeled negative samples not grouped into meaningful categories.
- Unknown Unknown Classes (UUCs) encompass classes without any information available during training.

It is evident from Table 2.2 that the dataset composition varies significantly across these tasks. Generalized OOD detection methods primarily focus on using information within KKCs to correctly reject both KUCs and UUCs. In contrast, our approach emphasizes leveraging information from both KKCs and KUCs to accurately classify KKCs and manage KUCs effectively, as they are both integral to the classification process.

	TRAINING	TESTING	Goal
Traditional Classification	KKCs	KKCs	Classifying KKCs
Anomaly Detection and Novelty Detection	KKCs and few or none outliers from KUCs	KKCs and few or none outliers	Detecting outliers
Open Set Recognition	KKCs	KKCs and UUCs	Identifying KKCs and rejecting UUCs
Out-of-Distribution Detection	KKCs	KKCs, KUCs and UUCs	Identifying KKCs and rejecting KUCs and UUCs
Classification Problem with the Universum Class	**KKCs and extensive KUCs**	**KKCs and extensive KUCs**	**Classifying KKCs and KUCs**

TABLE 2.2: Illustration of the difference between classification problem with the Universum class and generalized OOD detection tasks.

2.2.4.3 Difference in Methodology

The methodologies used to address the classification problem with the Universum class and generalized OOD detection also differs significantly.

By definition, the OOD detection problem assumes that the training data do not contain any OOD samples. However, a branch of the OOD studies, known as outlier exposure [96–101], introduces auxiliary outlier data during training. The introduced auxiliary data makes it close to the format of our raised classification problems with the Universum class. However, outlier exposure methods are not suitable for our problem. The outlier exposure method mostly adopts a two-step approach that consists of multi-class classification and OOD identification. Such two-step approach will suffer from error propagation problem. In addition,the OOD identification step distinguishes OOD and ID samples based on a score obtained by cross entropy or energy function. However, both cross entropy and energy function are monotonically varying. As a result, the decision boundary derived from a threshold score of the monotonically varying function is an open boundary, which leaves the heterogeneity and representativeness issues we pointed out in this paper still unresolved.

From a methodological point of view, our work is also different from the works in generalized OOD using closed boundaries. In generalized OOD studies, the closed boundaries are formulated by the classic density-based method [102, 103], one-class classification method [104, 105], or distance-based method [72, 75, 106–108]. The

distance-based methods are limited to spherical boundary shapes but our method can generate arbitrary shape boundaries. The one-class classification method formulates only one closed boundary between positive and negative samples while our work generates closed boundaries for all target classes. Finally, only positive samples are used to learn decision boundaries in density-based method, while both target class samples and Universum samples are used in our work.

2.3 Overview of Large Language Models

Since the release of ChatGPT in November 2022, LLMs have garnered significant attention and have revolutionized research in NLP. Our investigation into event extraction has similarly undergone a paradigm shift, beginning to leverage the power of LLMs for enhancing EE and addressing important inherent issues in LLMs. This section reviews the development and related research topics of LLMs.

LLMs are large-scale, pre-trained, statistical language models based on neural networks [109]. Recent advancements in LLMs have transformed the NLP field by significantly enhancing performance across a broad spectrum of tasks. Increasing the scale of these models—such as model parameters and training data—consistently leads to better performance, suggesting a promising trend toward developing even larger language models. Additionally, LLMs can not only solve NLP tasks but also serve as general-purpose task solvers, positioning them as crucial components for advancing toward artificial general intelligence.

Despite their strong performance, the underlying mechanisms of LLMs remain largely under-explored. For instance, LLMs exhibit emergent abilities, which are abilities not present in smaller models but manifest in larger ones [110]. These abilities are unpredictable, leaving researchers with limited understanding of how such emergent abilities appear and how to foresee new emergent capabilities in LLMs. Furthermore, LLMs have introduced a paradigm named in-context learning (ICL). ICL allows LLMs to adapt to new tasks using just a few illustrative examples, provided in the form of demonstrations [4]. Unlike traditional machine learning algorithms, ICL does not involve updating model parameters. The mechanisms that enable ICL, as well as its broader implications for model learning and adaptability, remain largely under-explored.

In this section, we provide a systematic overview of the research on large language models, focusing on its origin, structure, ICL paradign, and bias of LLMs.

2.3.1 Language Models

LLMs is a recent breakthrough in the reseach of language models, which are designed to predict the likelihood of a generating a sequence of words. Research on language models can date back to 1990s. We review the development of language model as follow.

- **Statistical Language Models** (SLMs): SLMs are the earliest forms of language models [111, 112], primarily based on the idea that predict the next word based on the nearest previous context. SLMs calculate the probability of a word based on the frequencies of previous word sequences in a given dataset. n-gram SLMs have a fixed context length n. Although effective in capturing local context within the n-gram window, they struggle with longer dependencies and suffer from the curse of dimensionality, which makes them computationally expensive as n increases. Classic SLMs include Markov models and Hidden Markov Models that paved the way for initial explorations in automated language processing.

- **Neural Language Models** (NLMs): The introduction of NLMs [113, 114] represented a significant paradigm shift in language modeling by utilizing neural networks to address the limitations of earlier Statistical Language Models (SLMs). At the core of NLMs is the concept of distributed representations, pioneered by [115], where each word is represented by a vector. This approach allows the model to capture complex semantic and syntactic relationships between words. An important advancement in distributed representations was the development of word2vec [116], which efficiently learns distributed word representations by training a simplified neural network, proving effective across various NLP tasks. Additionally, more sophisticated models like RNNs and LSTMs were developed to maintain information across longer sequences, surpassing the capabilities of traditional n-grams in handling sentences with complex structures. NLMs have laid the groundwork for advanced text generation and understanding, significantly enhancing the scope and depth of NLP applications.

- **Pre-trained Language Models** (PLMs): PLMs mark a significant evolution in language modeling by leveraging vast amounts of unlabeled text to learn universal

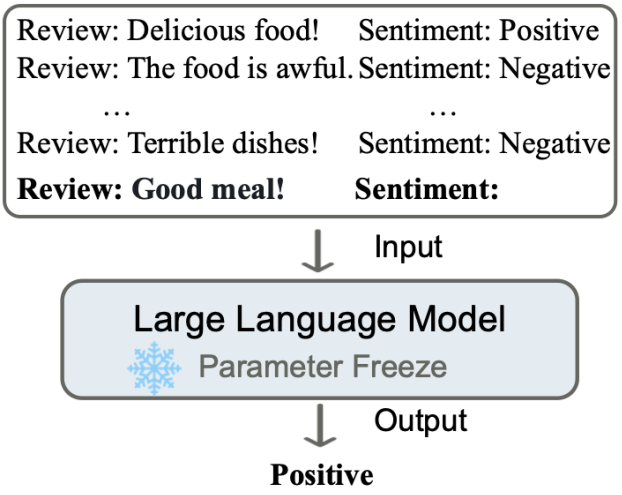


FIGURE 2.7: An illustration of the ICL paradigm from [4].

language representations before being fine-tuned for specific tasks. Unlike earlier models, PLMs are inherently task-agnostic, designed to be adaptable across a wide range of applications. This adaptability significantly enhances the efficiency and effectiveness of language models in targeted settings. Pioneering efforts such as ELMo [117] introduced the concept of deep contextualized word representations. Subsequently, Transformer-based models like BERT [118], T5 [119], and RoBERTa [120] have significantly improved performance across a broad range of NLP benchmarks by leveraging extensive pretraining on large corpora.

- **Large Language Models** (LLMs): LLMs mainly refer to transformer-based pre-trained language models that contain tens to hundreds of billions of parameters [109]. LLMs represent the cutting edge in language model research, encompassing models such as GPT-3 [49], Llama [121], PaLM [122], and GPT-4 [123]. These models not only continue the trend of pre-trained models but also scale it up significantly. LLMs exhibit different behaviors that are not observed in smaller PLMs, including multi-step reasoning—solving complex problems by breaking them into multiple steps, in-context learning—adapting to new tasks with a few examples of the task, and instruction following—executing tasks follow the instructions.

2.3.2 In-context Learning

ICL enables LLMs to perform a target task by feeding a few prompted examples as part of the input [49]. As demonstrated in Figure 2.7, given a prompt containing a few examples

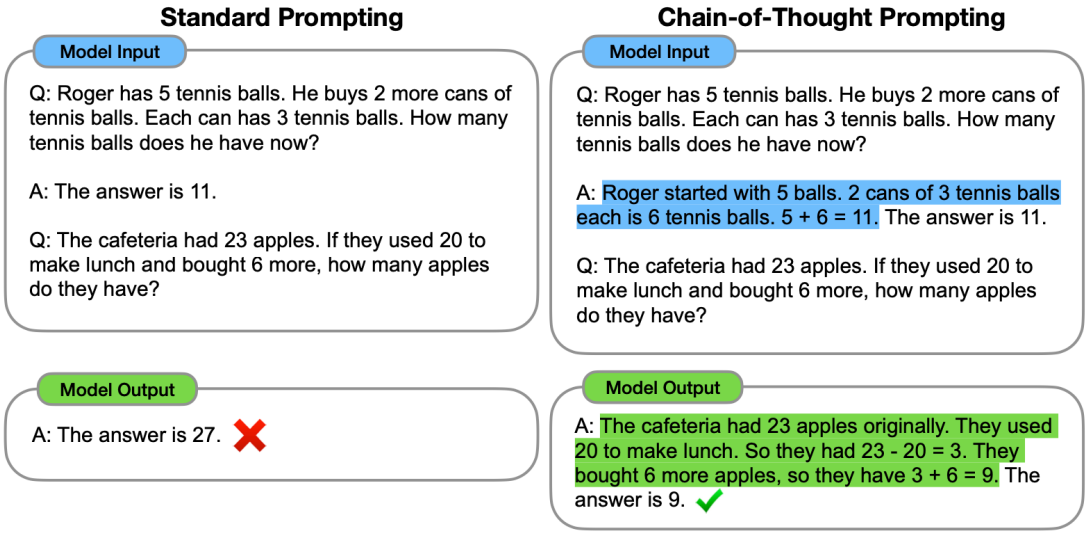


FIGURE 2.8: An illustration of the CoT prompting in [5].

of the task, LLMs will make predictions on the target query with all model parameters frozen. Specifically, given the question $\mathbf{Q}$, a set of examples $\mathbf{C}_k = \{(\mathbf{Q}_i, \mathbf{R}_i, \mathbf{A}_i)\}_{i=1}^{k}$, a large language model f_{LLM}, where $(\mathbf{Q}_i, \mathbf{R}_i, \mathbf{A}_i)$ represents an example comprising the question, reasoning steps, and its answer, and k denotes the number of examples. The jth token o_j of the LLM's output is generated as:

$$o_j = f_{LLM}(\mathbf{C}_k, \mathbf{Q}, o_{<j}) \tag{2.1}$$

where $o_{<j}$ denotes output tokens generated by LLM before o_j

The performance of ICL is highly sensitive to several factors, including the selection of examples, the formatting of answers, and the order in which examples are presented [124, 125]. This sensitivity has spurred extensive research aimed at optimizing the design of prompts for ICL. Numerous studies propose methods to select ICL examples based on various criteria such as complexity [126], mutual information [127], diversity [128], availability of labeled datasets [8], and feedback from LLMs [129]. Additionally, considerable effort has been devoted to the formatting of demonstrations. One of the most impactful prompting strategies is the chain-of-thought (CoT) prompting [5]. As demonstrated in Figure 2.8, CoT prompting guides LLMs to generate a series of intermediate reasoning steps, which has been shown to significantly enhance LLM performance on tasks involving complex reasoning. Furthermore, how to optimize the order of demonstration examples is also investigated in [130].

The working mechanism of ICL remains largely under-explored [4]. The mechanism of ICL is fundamentally different from supervised machine learning: Unlike traditional supervised machine learning, which learns and updates model parameters during training, ICL operates with all parameters frozen. This fundamental difference presents unique challenges for understanding the mechanism of ICL. Min et al. [125] have identified that the label space, the distribution of input text, and the overall format significantly contribute to ICL effectiveness. Further, Liu et al. [51] demonstrate that examples which are semantically closer to the test sample tend to yield better results, suggesting that the relevance of training examples plays a crucial role in ICL. Moreover, Akyürek et al. [131] discover that transformer-based ICL can implicitly emulate the effects of standard fine-tuning, indicating that transformers can adjust their behavior based on the context provided during the ICL process.

Additionally, despite the amazing performance of ICL, a crucial concern of LLMs is their tendency to generate hallucinations—seemingly plausible yet factually unsupported content. Hallucination occurs when models generate content that is not based on factual or accurate information [132]. This phenomenon not only degrades the performance of LLMs but also raises safety concerns for their real-world applications. For example, in healthcare, hallucinations can pose serious risks to patient safety [133], and in privacy-sensitive scenarios, they may lead to potential violations [134]. Specifically, in the task of event extraction, hallucination can result in the generation of non-existent event arguments, i.e., information not mentioned in the provided text, thereby reducing accuracy and challenging the reliability of the results. Consequently, it is essential to carefully design the prompting method for ICL in event extraction tasks to mitigate hallucinations.

2.3.3 LLM Bias

LLMs have demonstrated remarkable capabilities across a variety of NLP tasks through the use of the ICL paradigm. However, despite the impressive performance of ICL, LLMs are also found to be prone to prompt brittleness, characterized by a high sensitivity to the choice [6], order [135], and formatting [125] of in-context examples. As depicted in Figure 2.9, variations in the examples and their sequence within prompts can lead to significant changes in ICL performance on the SST-2 dataset. This prompt brittleness arises from the inherent biases within LLMs towards favoring certain answers [6]. For

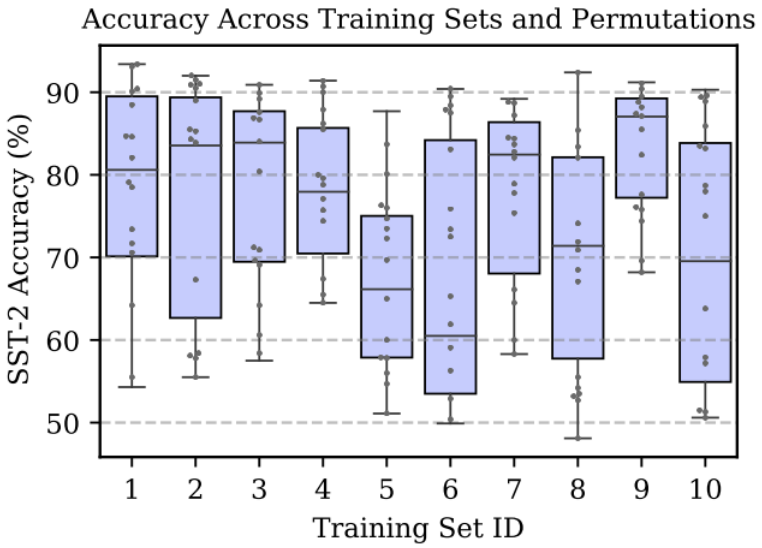


FIGURE 2.9: An Illustration of prompt brittleness in [6]

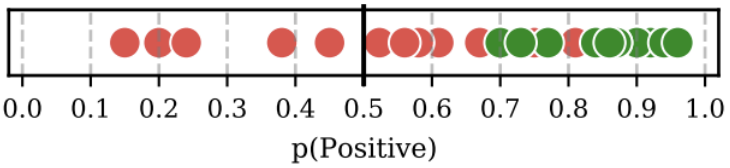


FIGURE 2.10: An Illustration of Bias in LLMs Toward Certain Labels

instance, as illustrated in Figure 2.10, when provided with random samples from a sentiment analysis task-where red dots represent negative sentiments and green dots represent positive sentiments—LLMs exhibit a bias towards predicting positive outcomes. Similarly, in the task of event extraction, we also observe that LLMs are biased towards predicting 'not specified' as answer for event arguments. The existence of LLM bias significantly undermines the robustness and adaptability of LLMs in diverse applications.

To understand the bias of LLMs, extensive studies have identifyied various bias factors, including including recency bias, majority label bias, common token bias [6], vanilla label bias [136], and domain label bias [136]. For instance, vanilla label bias [136] and recency bias [6] demonstrate the LLM's inherent non-contextual preference for certain labels and contextual preference for specific positions, respectively. More recently, due to the wide adaption of multiple-choice questions (MCQs) in evaluating LLMs' performance, selection bias, which consistently favors specific options in MCQs, has also been identified [137, 138].

Additionally, several calibration methods have been proposed to mitigate the bias in LLMs. Specifically, Zhao et al. [6] pioneered work revealing the instability of in-context

learning (ICL) demonstrations and identified several factors that contribute to LLM bias. They introduced contextual calibration as a method to address this bias by calibrating predictions based on LLMs' outputs from content-free samples. Meanwhile, Fei et al. [136] focused on identifying domain label bias and utilized random tokens from the target domain as input samples to assess and calibrate this bias. Han et al. [139] examined the effects on decision boundaries in bias calibration methods, suggesting the use of Gaussian mixture models to establish a more robust decision boundary. Zhou et al. [140] analyzed existing calibration methods and proposed a batch calibration method that mitigates bias by adjusting LLM outputs based on a batch of samples.

However, all existing calibration methods adjust the decision boundaries of model output probabilities to mitigate LLM bias and they largely rely on external observations or adjustments of LLM outputs. Consequently, the internal mechanisms within LLMs that lead to such biases remain poorly understood. This gap underscores a crucial area for further research—delving into the internal dynamics of LLMs to develop a deeper understanding of how biases are internally structured and potentially devising more intrinsic solutions to mitigate them.

2.3.4 Interpreting LLMs

Understanding the inner workings of LLMs is crucial for advancing their applications and addressing their limitations. This section delves into the structure of LLMs and reviews current research on mechanistic interpretability of LLMs.

2.3.4.1 Transformer

LLMs are predominantly composed of a sequence of transformer layers. The transformer architecture, as illustrated in Figure 2.11, fundamentally consists of two main components: multi-head self-attention mechanisms and feed-forward networks.

- **Multi-Head Attention Mechanism**: The core of the transformer is the multi-head attention mechanism. This component allows the model to focus on different parts of the input sequence simultaneously, providing a dynamic way of understanding and representing the input data. It achieves this by projecting the queries, keys, and values with different, learned linear projections to a higher dimension. By doing

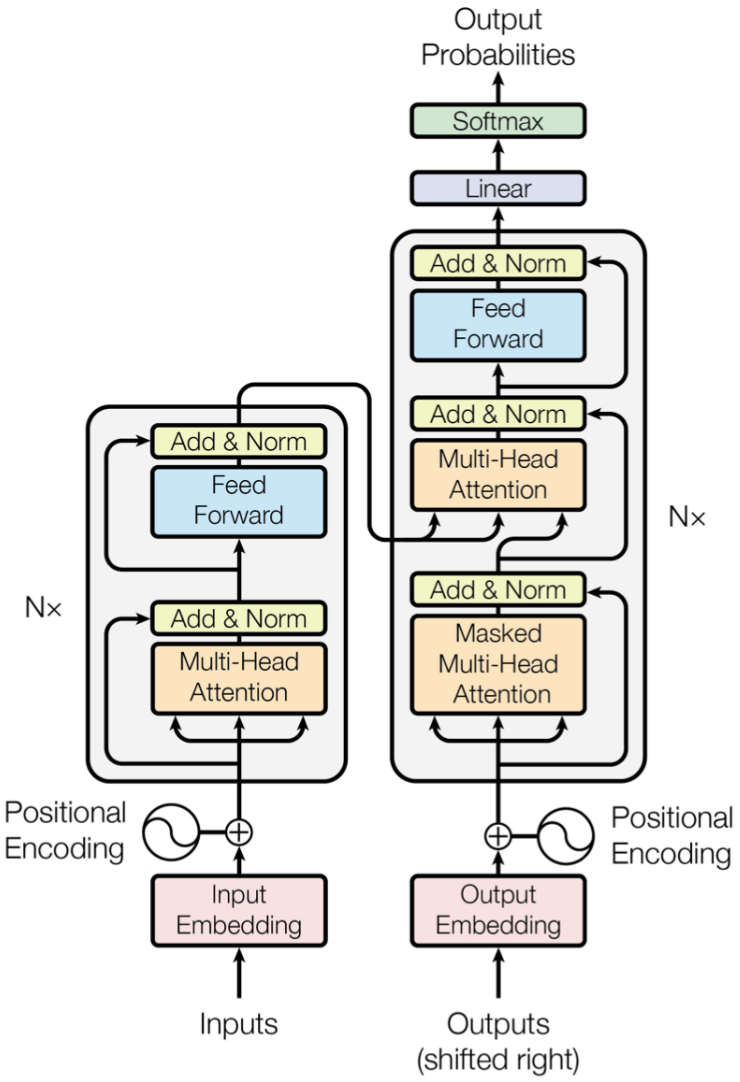


FIGURE 2.11: An Illustration of the transformer architecture in [7].

so, it can attend to information from different representation subspaces at different positions, aggregating these diverse perspectives to form a richer representation of the input.

- **Position-wise Feed-Forward Networks**: Following the attention mechanism, each layer includes a position-wise feed-forward network, which applies the same fully connected layer separately to each position. These networks consist of two linear transformations with a ReLU activation in between. Despite their simplicity, these networks play a crucial role in refining the output from the attention mechanism by integrating non-linear transformations into the model.

- **Residual Connection and Layer Norm**: Each sub-layer in a transformer (multi-head attention or feed-forward) is equipped with a residual connection, followed by layer normalization. The residual connections can mitigate the vanishing gradient problem by allowing gradients to flow through the network directly. The layer normalization standardizes the activations across the features, stabilizing the training process.
- **Positional Encoding**: Since the model itself contains no recurrence or convolution, positional encodings are added to the input embeddings at the bottom of the encoder and decoder stacks. This addition provides the model with information about the relative or absolute positioning of the tokens in the sequence.

The structure of the transformer allows for significantly parallelized processing, unlike RNNs that must process data sequentially. This parallelization reduces training times and enables the handling of long data sequences more effectively, which is particularly beneficial for the training of LLMs. Notably, LLMs are predominantly implemented using decoder-only architectures, meaning they only utilize the decoder part of the transformer architecture.

2.3.4.2 Mechanistic Interpretability of LLMs

Mechanistic interpretability [141–143] aims to explain the internal processes in language models (LMs), facilitating the interpretation of the contributions of individual model components to the final prediction. Research findings in Mechanistic interpretability interpret the contribution of each FFN vector and attention head to the models' prediction as follows.

The Residual Stream We interpret Transformers following the view of residual stream [141, 144]. Due to the residdual connection of Transformers, each layer takes a hidden state as input, and adds information obtained by its attention layer and FFN layer to the hidden state through residual connection. In this sence, the hidden state is a residual stream passed along layers, and each attention layer and FFN layer contribute to the final prediction by adding information to the residual stream.

Attention Heads Following Elhage et al. [141], Dar et al. [144], the output of each attention layer of LM can be computed as the sum of all its attention heads. Specifically,

for l-th layer, the input is $X^l \in \mathbb{R}^{N\times d}$, and the attention layer is parameterized by four matrices W_Q^l, W_K^l, W_V^l, $W_O^l \in \mathbb{R}^{d\times d}$. The columns of each projection matrix and the rows of the output matrix can be split into H parts: $W_Q^{\ell,j}, W_K^{\ell,j}, W_V^{\ell,j} \in \mathbb{R}^{d\times \frac{d}{H}}$ and $W_O^{\ell,j} \in \mathbb{R}^{\frac{d}{H}\times d}$, where H is the number of attention heads. Then, it can be derived that:

$$\begin{aligned}\text{Att}^\ell(X^\ell) &= \textbf{Concat}\left[A^{\ell,1}X^\ell W_V^{\ell,1}, A^{\ell,2}X^\ell W_V^{\ell,2}, \ldots, A^{\ell,H}X^\ell W_V^{\ell,H}\right] W_O^\ell \\ &= \sum_{j=1}^{H} A^{\ell,j}(X^\ell W_V^{\ell,j})W_O^{\ell,j}\end{aligned}$$

where $A^{\ell,j} = \text{softmax}\left(\frac{(X^\ell W_Q^{\ell,j})(X^\ell W_K^{\ell,j})^T}{\sqrt{d/H}} + M^{\ell,j}\right)$, $M^{\ell,j}$ is the attention mask. Therefore, the output of an attention layer is equivalent to computing attention heads independently, multiplying each by its own output matrix, and adding them into the residual stream of the LM.

FFN In line with Geva et al. [143, 145], transformer FFN layers can be cast as linear combination of vectors. Specifically, for an input vector $\mathbf{x}^\ell \in \mathbb{R}^d$, FFN parameter matrices $\mathbf{K}^\ell, \mathbf{V}^\ell \in \mathbb{R}^{d_m\times d}$, the FFN output can be derived as:

$$\text{FFN}^\ell(\mathbf{x}^\ell) = f(\mathbf{x}^\ell \mathbf{K}^{\ell^T})\mathbf{V}^\ell = \sum_{i=1}^{d_m} f(\mathbf{x}^\ell \cdot \mathbf{k}_i^\ell)\mathbf{v}_i^\ell = \sum_{i=1}^{d_m} m_i^\ell \mathbf{v}_i^\ell$$

where f is the activation function, i is the index of the vector. Then, the FFN layer can be viewed as a linear combination of vectors: the multiplication of $\mathbf{x}^\ell$ and the key vector $\mathbf{k}_i$ produces the coefficient m_i^ℓ that weights the corresponding value vector $\mathbf{v}_i$.

Logit Lens The logit lens [146] is a technique that directly decode hidden states into the vocabulary space using the unembedding matrix of the LLM for interpretation. This approach has been validated in various studies as an efficient method for interpreting the weight matrix or hidden states of LLMs [143, 144, 147, 148].

In summary, each attention layer and FFN layer contribute to the final prediction by adding their output hidden states to the residual stream. These outputs can be viewed as the sum of their respective attention heads and FFN vectors. Each attention head or FFN vector's output can be interpreted through the logit lens.

2.4 Conclusion

This chapter presents a comprehensive review of event extraction, the Universum class, and large language models. It begins by defining key terminologies and outlining the problem settings of event extraction, then delves into a variety of learning algorithms. These range from rule-based systems to traditional machine learning methods, and further to advanced deep learning approaches such as CNNs, RNNs, GGNNs, Transformers, and LLMs. The review of the Universum Class introduces its terminologies and target classes, illustrates its widespread presence in classification problems, and highlights the distinctions between traditional closed boundary learning methods and the proposed problem setting. Additionally, this chapter reviews LLMs, discussing the evolution of language models, emerging paradigms with LLMs, issues of bias and hallucination in LLMs, and mechanistic interpretability aimed at understanding the workings of LLMs. This chapter covers methodological aspects of the thesis and lays the foundation for subsequent chapters.

Part I

Towards More Accurate and Practically Applicable Event Extraction Systems

Chapter 3

Document-Level Event Argument Extraction by Leveraging Redundant Information and Closed Boundary Loss

3.1 Introduction

The inherent complexity of EE often confines research to the sentence level, where the focus is on extracting event information from individual sentences. However, real-world applications generally require extracting event information at the document level. This transition to document-level extraction presents significant difficulties. In this chapter, we present a effective method for document-level event extraction by leveraging redundant information within documents and closed boundary loss.

Event argument extraction (EAE) is the main task of event extraction, which aims to identify the arguments of a given event and recognize the specific roles they play. Previous works are mostly focused on sentence-level EE [149–156]. However, events are often described in the form of documents in the real world. Document-level event extraction has received consideration in recent years.

Research on document-level event extraction has been focused on tackling challenges such as arguments-scattering and multiple-events [27, 157–165]. The benefit of redundant event information in a document is largely neglected. We believe that the redundant event information in a document can be used to improve event extraction, as illustrated in

No.	Sentence	Entity label	Difficulty
s1	The killers, approximately 30 men in uniform, arrived before 0230.	1	★
s2	Soldiers with their faces painted black arrived in Cayara last Saturday. They broke down doors, looted stores, and burned several houses.	1	★★★
s3	The murder was carried out by soldiers.	1	★
s4	The house was surrounded by soldiers.	0	-
s5	The house was searched by the soldiers 2 days before the crime.	0	-
s6	How can men in uniform be in a militarized area?	0	-
s7	He said that the library was burned.	-	-
...	.....	...	...

Argument role	Entity	Entity label	Summative label
Perpetrator individual	men in uniform	1	1
	soldiers with their faces painted black	1	0
	soldiers	1	0
Physical target	houses	1	1
	library	1	1
...	...	...	...

FIGURE 3.1: An example of redundant event information in the document-level event argument extraction. Sentence s2 is most challenging primarily due to its longer length and the requirement to understand coreference information ("They") for accurate event information extraction.

the example in Figure 3.1. The upper part of Figure 3.1 shows seven simplified sentences selected from a document in the MUC-4 dataset. All entities marked in blue are the same entity "soldiers", which appears in different expressions in different sentences. For ease of description, we call it entity S. We can observe from Figure 3.1 that: 1) The argument information in the document is redundant since entity S appears in the article multiple times as an argument and we can successfully extract the argument by correctly recognizing any of these occurrences. This property can be potentially used to improve the robustness of the model. 2) The difficulty of extracting the entity S as an argument in its different occurrences is different. Extracting entity S in sentence 1 and sentence 3 is much easier than extracting it from sentence 2. Hence, by utilizing the redundant

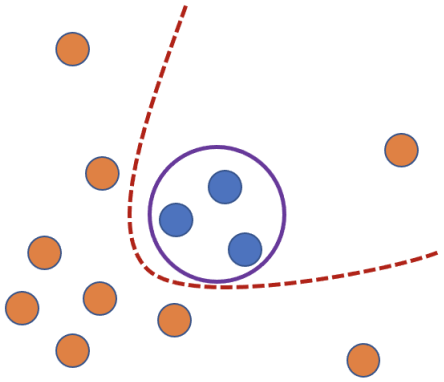


FIGURE 3.2: A simplified illustration of closed boundary loss. Blue dots represent target samples, orange dots represent Universum samples. The red dotted line represents cross entropy loss, the purple solid line represents proposed closed boundary loss.

event information of the document, we can extract arguments from relatively simple positions and reduce the difficulty of the task. 3) An entity may appear multiple times in the document, directly averaging them as the entity's feature representation [27] may introduce noise. For example, although entity S is an event argument in the document, its occurrences in s4, s5 and s6 should not be recognized as a correct pattern to identify the event argument. 4) The redundant argument information can result in redundant extraction results, as shown in the bottom table in Figure 3.1. The three entities extracted as perpetrator individual need to be merged into one. However, the extracted physical target "houses" and "library" are different entities and should not be merged. Therefore, the use of redundant event information underlying a document is not straightforward, a sophisticated algorithm for merging multiple extraction results is needed.

Extraction of arguments can be solved as an entity classification problem by treating entities as argument candidates [27, 157, 163]. In document-level event argument extraction, only a subset of the entities in a document are arguments, while the majority of entities are regarded as class "others" or "neither"(neither of the target classes). This kind of data was first studied by Weston et al. [53] under the name Universum. The Universum data are usually very diverse and do not have typical common features. In addition, Universum data is much more than the target class data, exhibiting a severe class imbalance problem. Figure 3.2 demonstrates a simplified distribution of data samples in document-level event argument extraction. The blue dots represent argument entities, the orange dots represent a large number of Universum class entities. Since the samples in the Universum class do not have typical common features, they tend to scatter in the feature space. This characteristic of the Universum data is largely overlooked in the information

extraction community. Universum data is simply considered as another class "others", without any special consideration in the classifier design. Cross entropy loss is usually employed in classifier training [27, 157, 162, 163]. However, classifiers trained by cross entropy loss have open decision boundaries, and hence some Universum samples, such as the orange dot on the upper right of the figure, could be easily misclassified. We think a classifier with a closed decision boundary could better deal with the Universum class in document-level event argument extraction, as illustrated by the purple line in Figure 3.2.

The contribution of this work is three-fold.

- Firstly, it is the pioneering work to leverage redundant event information in documents for event extraction. We propose the entity coreference graph with graph2token module and entity summary graph to leverage the redundant event information. Experimental results show that redundant information helps improve recall significantly.
- Secondly, we analyze the issue of Universum data in document-level event argument extraction and the problem of classifiers trained by cross entropy loss, and propose a closed boundary loss to address the problems.
- Finally, our model consistently outperforms the latest baseline models in F1-score and achieves state-of-the-art performance. Compared to the three baseline models, our proposed model improves the absolute F1-score by 3.35%, 5.27%, and 6.45%, respectively.

3.2 Related Work

3.2.1 Event Argument Extraction

Most previous event argument extraction models make predictions at sentence-level [150–155, 166]. Considering that the real world events are often distributed across sentences, document-level event extraction has attracted more attention recently. Zheng et al. [157] propose the Doc2EDAG model to overcome the argument scattering problem. Du and Cardie [158] first argue the importance of document-level extraction and adopt a sequence model for document-level event extraction. Lou et al. [160] investigate a novel bidirectional decoder to overcome the long-range forgetting problem. Li et al. [161]

formulate the document-level event extraction model as conditional generation based on templates. Huang and Peng [162] attach importance to event coreference and entity coreference in document-level event extraction tasks. Xu et al. [27] build a heterogeneous graph with the Tracker module to deal with problems of event scattering and multiple events. Yang et al. [163] adopt parallel prediction networks to extract events parallelly from document-level representations. However, none of these works pay attention to the characteristic of information redundancy in the document, which we believe is a unique and beneficial property for document-level event argument extraction. In addition, to our knowledge, closed boundary classification has never been adopted in event extraction. Classification-based event argument extraction models [27, 162, 163] all employ cross entropy loss for classifier training, without considering the characteristics of Universum class: scattered distribution in the feature space due to heterogeneity and diversity of the samples in this class.

3.2.2 Closed Boundary Loss

We found that a classifier trained by cross entropy could easily misclassify entities in the class "others", i.e. Universum class, as demonstrated in Figure 3.2. The root cause of the issue is the open decision boundary of the classifier. For target classes, the model effectively learns predictive patterns from their training data. However, for the Universum class, samples are identified by an extrinsic pattern that they do not belong to any class of interest. Consequently, the extrinsic predictive pattern for this class cannot be learned from simply increasing the prediction probabilities of Universum samples. This leads to their failure in forming a compact space for test samples, causing them to scatter across the logit space. As a result, these samples easily fall within the open boundaries of the target classes, leading to frequent misclassification. To address this problem, we propose a novel closed boundary loss for classifier training.

Research works in Universum usually employ additional unlabeled Universum data to provide prior knowledge for the task, such as Universum support vector machine (SVM) [53, 68, 167], and semi-supervised learning [168, 169]. However, the SVM-based methods above are developed for structured data and are hard to integrate with deep neural network-based representation learning to form an end-to-end training procedure for natural language processing tasks. One possible solution is to use a deep neural network to learn representations first, and then feed the representations learned to the

Universum SVM classifiers. But the disadvantage of this two-step procedure is that the classification result cannot be back-propagated to representation learning. It is desired that the closed boundary classifier could be integrated with deep neural network-based representation learning to form end-to-end training for optimal performance.

Closed boundary classification methods are also developed in anomaly detection and open set recognition, such as deep one-class learning [170, 171], auto-encoder based anomaly detection [172], OpenMax layer for open set recognition [173]. However, these methods cannot use the information in outlier samples due to task setting.

A closed boundary classifier works best in feature space with compact class distribution. In the literature, some loss functions have been proposed to generate such feature space such as Deep SVDD [170], contrastive loss [174], and ii-loss [175]. However, Deep SVDD only minimizes the intra-class distance and cannot maximize the inter-class distance. Contrastive loss and ii-loss need to be combined with cross entropy loss to classify samples. But cross entropy loss generates open decision boundaries for the classifier.

In this paper, we propose a new loss function that could train a classifier with a closed decision boundary. In addition, it can be directly integrated with representation learning layers in a neural network to form an end-to-end training procedure to produce a feature space with minimum intra-class difference and maximum inter-class difference, which in turn leads to improved performance.

3.3 Method

As shown in Figure 3.3, our model consists of four main components: context encoding module, entity coreference graph, closed boundary loss, and entity summary graph, which are illustrated in this section.

3.3.1 Context Encoding

Given the input document, we apply a Bi-LSTM to obtain token representations of the document: $\mathrm{D} = \{\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{n-1}\} \in \mathbb{R}^{n \times l}$ where n is the document length, and

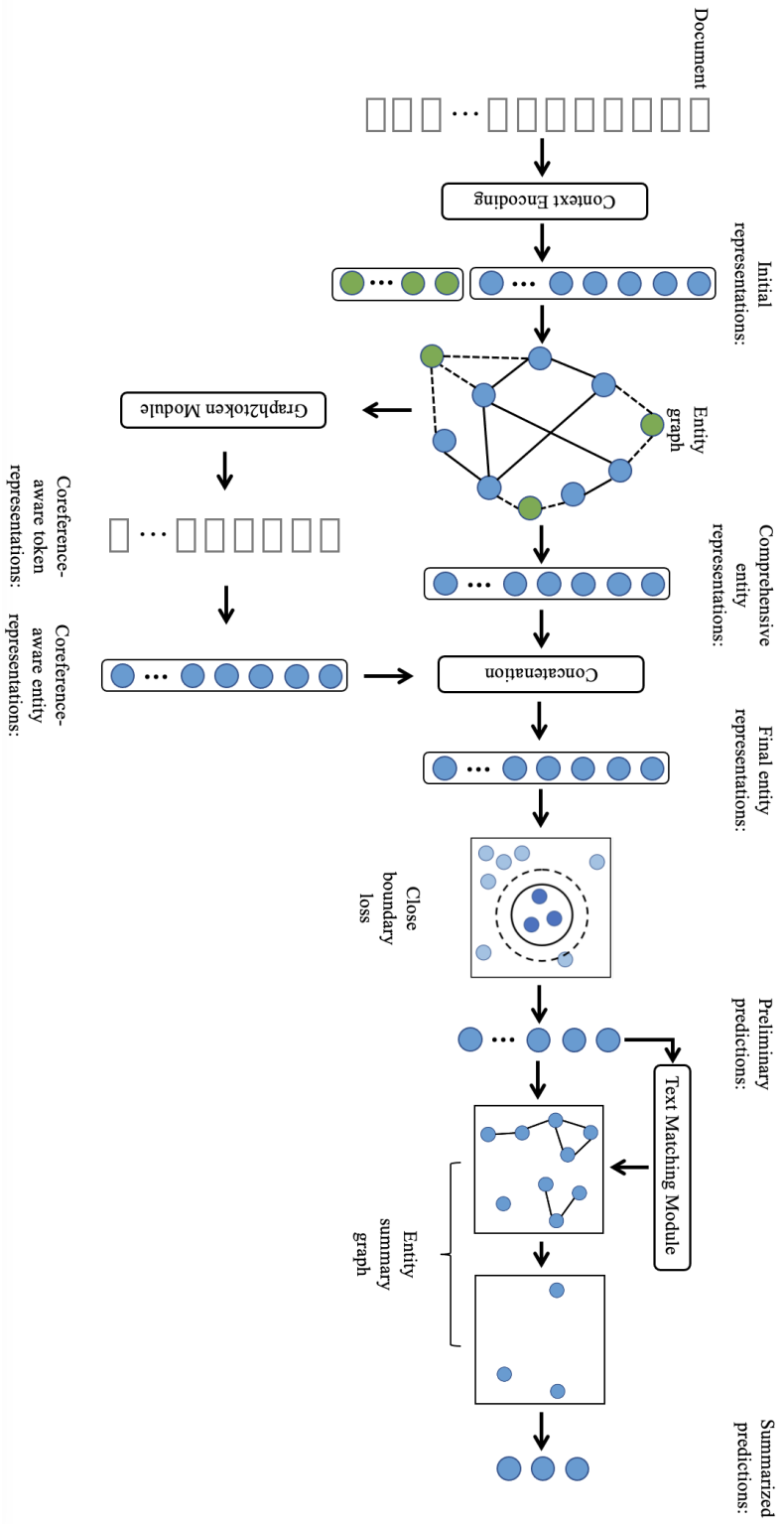


FIGURE 3.3: The overall model structure. Blue dots represent entity nodes, green dots represent sentence nodes.

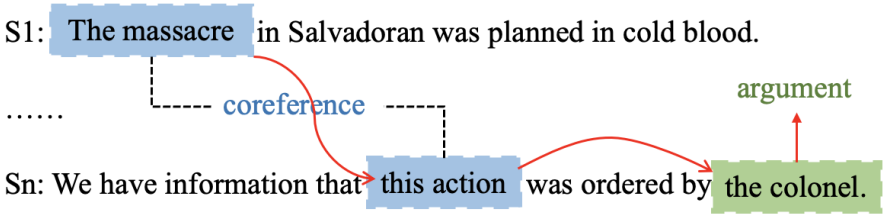


FIGURE 3.4: An example of coreference in a document and its impact on entity understanding and document-level event argument extraction

l is the the hidden state dimension. We construct entity representation and sentence representation from the start and end tokens in an entity or sentence:

$$\mathbf{e}_i = \left(\mathbf{e}^{(i)}_{\text{memory}}; \mathbf{e}^{(i)}_{\text{rule}}\right) \tag{3.1}$$

$$\mathbf{s}_i = \left(\mathbf{s}^{(i)}_{\text{memory}}; \mathbf{s}^{(i)}_{\text{rule}}\right) \tag{3.2}$$

$$\begin{aligned}
\mathbf{e}^{(i)}_{\text{memory}} &= \left(\mathrm{D}\left[ent^{(i)}_{\text{start}}[l:]\right]; \mathrm{D}\left[ent^{(i)}_{\text{end}}[:l]\right]\right) \\
\mathbf{e}^{(i)}_{\text{rule}} &= \left(\mathrm{D}\left[ent^{(i)}_{\text{start}}[:l]\right]; \mathrm{D}\left[ent^{(i)}_{\text{end}}[l:]\right]\right) \\
\mathbf{s}^{(i)}_{\text{memory}} &= \left(\mathrm{D}\left[sent^{(i)}_{\text{start}}[l:]\right]; \mathrm{D}\left[sent^{(i)}_{\text{end}}[:l]\right]\right) \\
\mathbf{s}^{(i)}_{\text{rule}} &= \left(\mathrm{D}\left[sent^{(i)}_{\text{start}}[:l]\right]; \mathrm{D}\left[sent^{(i)}_{\text{end}}[l:]\right]\right)
\end{aligned}$$

where D is the output of the Bi-LSTM encoding layer, $ent^{(i)}_{\text{start}}$, $ent^{(i)}_{\text{end}}$, $sent^{(i)}_{\text{start}}$ and $sent^{(i)}_{\text{end}}$ are the start and end position of the i-th entity and the i-th sentence, respectively, and [;] denotes the concatenation operation. $\mathbf{e}^{(i)}_{\text{memory}}$ and $\mathbf{s}^{(i)}_{\text{memory}}$ mainly contain the information inside the entity and sentence. $\mathbf{e}^{(i)}_{\text{rule}}$ and $\mathbf{s}^{(i)}_{\text{rule}}$ mainly contain the context information outside the entity and sentence. The model predicts memory representations mainly based on remembering entity names and predicts rule representations mainly based on recognizing the contextual patterns. Therefore, we separate the memory representation and rule representation as they correspond to the memory-based and the rule-based learning process of humans [176, 177].

3.3.2 Entity Coreference Graph

Leveraging redundant event information in a document is not straightforward to classify every entity in the document. On the one hand, better entity representation is needed.

Therefore, we construct an entity coreference graph with graph2token module to produce a comprehensive and coreference-aware representation for every entity.

The entity coreference graph is inspired by the observation of coreference's role in document understanding. Firstly, for the repeatedly referred entity (coreference entity), the understanding to this entity itself is constantly enriched or enhanced by each reference. For the example illustrated in Figure 3.4, "the massacre" and "this action" are two different mentions of the same entity. The understanding of this entity is enriched by combining the location of the massacre mentioned in the first sentence and the commander of the massacre mentioned in the second sentence. Secondly, for other entities located in the context of the coreference entity, their meanings are clearer by recognizing the connotation of the coreference entity. For example, "the colonel" cannot be recognized as an argument unless the model understands that "this action" refers to "the massacre". Research works in event extraction [27, 178, 179] consider the first observation but neglect the second one. Specifically, previous works in event extraction use graph structure to merge information in different mentions of the same entity. However, such a graph structure cannot feed back the fused information to the context of coreference entities because the representations of the context tokens are fixed from the initial encoding process. In this sense, for the representation of "the colonel", its context information still excludes "the massacre". Therefore, we adopt a graph2token module to feed back the comprehensive entity information obtained through graph structure to tokens, and then rebuild entity representations that are both comprehensive and coreference-aware.

Graph Construction. There are two types of nodes in the entity graph: entity nodes and sentence nodes. Entities are recognized from document following Fisher and Vlachos [180]. Entity nodes and sentence nodes are denoted as $\mathrm{E} = \{\mathbf{e}_0, \mathbf{e}_1, \ldots, \mathbf{e}_p\}$, and $\mathrm{S} = \{\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_q\}$, respectively.

There are two types of edges in the entity graph: 1) entity-entity edge is created according to the coreference relationship. We use SpanBERT [181] to implement coreference resolution on documents during preprocessing. 2) entity-sentence edge is the connection between the entity node and the sentence node where it is located.

Graph Propagation. After the graph is constructed, Graph Attention Network [182] is applied to propagate information between connected nodes. Assuming that graph nodes are denoted by $\mathrm{H} = \{\mathrm{E}, \mathrm{S}\} = \{\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{p+q}\} \in \mathbb{R}^{(p+q)\times 2l}$, the information that a

node receives from its neighbors is formulated as:

$$\mathbf{h}_i' = \text{RELU}\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}\mathbf{W}\mathbf{h}_j\right) \tag{3.3}$$

$$\alpha_{ij} = \frac{\exp\left(L\left(\mathbf{W}_{e_{ij}}[\mathbf{W}\mathbf{h}_i;\mathbf{W}\mathbf{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(L\left(\mathbf{W}_{e_i}[\mathbf{W}\mathbf{h}_i;\mathbf{W}\mathbf{h}_k]\right)\right)} \tag{3.4}$$

where $\mathbf{h}_i'$ is the neighbor information of the i-th node, $\mathbf{h}_j$ is the representation of the j-th node, $\mathbf{W}$, $\mathbf{W}_{e_i}$ are weight matrixes, $\mathcal{N}_i$ denotes the set of neighbors of node i, and $L(\cdot)$ is the LeakyReLU function.

The representation of the i-th node $\mathbf{h}_i$ and its neighbor information $\mathbf{h}_i'$ is fused by the gated mechanism:

$$\beta_i = \sigma\left(f\left(\mathbf{h}_i;\mathbf{h}_i'\right)\right) \tag{3.5}$$

where $\sigma(\cdot)$ is the sigmoid function, $f(\cdot)$ denotes the linear transformation. The fused representation of the i-th node $\mathbf{h}_i''$ is obtained as:

$$\mathbf{h}_i'' = \beta_i \odot \mathbf{h}_i + (1-\beta_i)\odot\mathbf{h}_i' \tag{3.6}$$

where $\odot$ stands for element-wise multiplication. Through propagating and fusing information of coreference entities and the corresponding sentence, a comprehensive representation of the entity is obtained.

Graph2token. To address the second insight we put forward in this section, we adopt the graph2token module to feed back the information behind coreference entities to their neighboring tokens.

We concatenate the original token representation $\mathbf{d}_i$ with the entity representation $\mathbf{h}_j''$ in which it is located, and feed it to an LSTM layer. In this way, the comprehensive entity representation $\mathbf{h}_j''$ is propagated to context tokens outside the entity and a coreference-aware token representation $\mathbf{d}_i'$ is generated:

$$\mathbf{d}_i' = \text{LSTM}(\mathbf{d}_i;\mathbf{h}_j'') \tag{3.7}$$

Then, we build coreference-aware entity representations from updated token representations.

$$\mathbf{e}_{\text{rule}}^{(i)\,\prime} = \left(D'\left[ent_{\text{start}}^{(i)}[:l]\right];D'\left[ent_{\text{end}}^{(i)}[l:]\right]\right)$$

where $\mathrm{D}' = \left\{\mathbf{d}'_0, \mathbf{d}'_1, \ldots, \mathbf{d}'_{n-1}\right\}$. Finally, a comprehensive and coreference-aware entity representation $\mathrm{E}\prime = \{\mathbf{e}_0\prime, \mathbf{e}_1\prime, \ldots, \mathbf{e}_p\prime\}$ is obtained by concatenation:

$$\mathbf{e}_i\prime = \left(\mathbf{h}''_i; \mathbf{e}^{(i)\,\prime}_{\text{rule}}\right) \tag{3.8}$$

3.3.3 Closed Boundary Loss

As detailed in Section 3.1, we have analyzed that classifiers trained by cross entropy loss have open decision boundaries and could easily misclassify the Universum class. To address this problem, we propose a novel loss function that could be used to train classifiers with closed decision boundaries.

Comprehensive and coreference-aware entity representations $\mathrm{E}' = \left\{\mathbf{e}'_0, \mathbf{e}'_1, \ldots, \mathbf{e}'_p\right\}$ are obtained in the last section. We treat entities as argument candidates and classify entities by classifiers trained by our proposed closed boundary loss:

$$\begin{aligned}\mathrm{L}_{\text{CB}} = \lambda R^2 + \frac{1}{n}\sum_{i=1}^{n} \max\left(0, \left\|\mathbf{e}'_i - \mathbf{c}\right\|^2 - R^2\right) \\ + \frac{1}{m}\sum_{i=1}^{m} \max\left(0, (1+\mu)R^2 - \left\|\mathbf{e}'_i - \mathbf{c}\right\|^2\right)\end{aligned}$$

where n is the number of target class samples, m is the number of Universum class samples, the center $\mathbf{c}$ is initialized as the mean of target samples in the feature space, and the radius $\mathbf{R}$ is initialized as ν-quantile of the distance of target samples to the center $\mathbf{c}$ in the feature space. $\mathbf{R}$ and $\mathbf{c}$ are initialized after a few warm-up epochs. The closed boundary loss intends to include samples of each target class using a hypersphere characterized by center $\mathbf{c}$ and radius R in the feature space and locate Universum samples outside the hypersphere. Due to the heterogeneous nature of Universum samples, we allow them to scatter outside the hypersphere and do not require them to be aggregated like cross entropy loss.

The goal of the first term λR^2 is to minimize the volume of the hypersphere, and λ is a hyperparameter between 0 and 1. The second term aims to enclose target class samples by the hypersphere. If the Euclidean distance between the sample $\mathbf{h}''_i$ and the center $\mathbf{c}$ exceeds the radius, it will lead to a penalty in the loss function. The third term aims to keep the universe samples outside the hypersphere. Parameter μ is introduced to adjust the gap between the closed boundary hypersphere and Universum samples.

Unlike contrastive loss and ii-loss which cannot be directly used for classifying samples in the test set and need to be combined with cross entropy loss, our proposed closed boundary loss can be easily adopted for classification by the following calculation:

$$g(\mathbf{e}_i\prime) = \begin{cases} 1 & \|\mathbf{e}'_i - \mathbf{c}\|^2 - R^2 < 0 \\ 0 & \|\mathbf{e}'_i - \mathbf{c}\|^2 - R^2 > 0 \end{cases}$$

3.3.4 Entity Summary Graph

To make full use of the redundant argument information, we classify every entity in the document. For the same argument, we may obtain multiple preliminary extraction results. The advantage is the robustness because the correct argument is more likely to be extracted from relatively simple positions. The challenge is how to merge the multiple extraction results. To address the challenge, we propose an entity summary graph.

Text Matching Module. We notice that most redundant expressions of the same entity are either character-level spelling similar or word-level semantics similar. In some cases, special domain knowledge is needed to determine if two expressions are the same. For example, "Army of National Liberation" and "ELN" are referred to the same entity. Therefore, we adopt a text matching model with both character embedding and word embedding to evaluate the spelling similarity and semantics similarity of extracted arguments. We also construct a text matching dataset from ground truth labels of the training set of our event extraction dataset to make the model learn necessary domain knowledge.

We build the text matching module (TMM) by adopting the structure of RE2 [183] and adding character embedding to the RE2 model to enhance the model's capability of recognizing spelling similarity. We denote the initially predicted arguments as $\mathrm{A} = \{\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{k-1}\}$. Then, we feed these entities into the text matching module to produce the matching score for each pair of arguments.

$$\mathbf{M}_{ij} = \mathrm{TMM}\left(\mathbf{a}_i, \mathbf{a}_j\right) \tag{3.9}$$

where $\mathbf{M}$ is the matching score matrix, which contains text matching score of every pair of entities from A. $\mathbf{M} = [\mathbf{M}_{ij}], i, j = 1, 2, \ldots, k$.

Entity Summary Graph. The graph node is composed of preliminary predicted entities A. The i-th node and j-th node are connected if $\mathbf{M}_{ij} > s$, where s is a boundary score.

The weight of each edge is the text matching score M_{ij} of two entity nodes at the ends of the edge.

The constructed entity summary graph is mostly disconnected because there usually exist multiple argument clusters in a document. The argument cluster refers to a set of different expressions of the same argument, for example "the armed forces" and "military" refer to the same argument, thus forming an argument cluster. The entity summary graph consists of several connected subgraphs as shown in Figure 3.3. Each subgraph corresponds to an argument cluster. We denote the entity summary graph G and its subgraphs as $\mathrm{G} = \left\{\mathrm{G}_{sub}^{(1)}, \mathrm{G}_{sub}^{(2)}, \ldots, \mathrm{G}_{sub}^{(u)}\right\}$. The final predicted arguments are summarized by selecting an entity node with the largest sum of weights (LSW) from each subgraph.

$$\mathrm{A}' = \left\{\mathbf{a}'_0, \mathbf{a}'_1, \ldots, \mathbf{a}'_{v-1}\right\}, \quad \mathbf{a}'_i = \mathrm{LSW}\left(\mathrm{G}_{sub}^{(i)}\right)$$

3.4 Experiments

3.4.1 Dataset

Our model is evaluated on the MUC-4 dataset [64]. The dataset is composed of 1,700 documents, each containing an average of 400 tokens and 7 paragraphs. We use 1300 documents for training, 200 documents for testing, and 200 documents for the development set following [158]. Five argument roles are extracted in the dataset: perpetrator individual, perpetrator organization, target, victim, and weapon.

3.4.2 Baselines and Evaluation Metric

In this work, we propose a document-level EAE model leveraging **R**edundant **I**nformation and **C**losed **B**oundary Loss (**RICB**). We compare our model with the following baseline models: **DYGIE++** [184] incorporates local and global contexts to build a multi-task framework for named entity recognition, relation extraction, and event extraction. **NST** [158] aggregates sentence representation and paragraph representation via a gate mechanism and treats document-level EAE as a sequence tagging problem. **GTT** [159] proposes a generative transformer based framework for document-level EAE. **DEEDP**

Hyper-parameter	Value
Embedding size	300
Hidden size	150
Bidirectional	True
Layers of encoder	2
Layers of graph2token module	1
Layers of graph	1
Heads of graph	2
Optimizer	Adam
Learning rate	$5e^{-4}$
Batch size	4
Dropout	0.3
Training epoch	120
Boundary score	0.65

TABLE 3.1: Hyper-parameter setting in the experiment.

[185] introduces a multi-granularity encoder framework for event extraction. **KDR** [186] propose key-value memory networks to enhance document-level contextual information. Both **DEEDP** and **KDR** were published subsequent to our work; we have incorporated their results to highlight the rapid progress in the field.

We evaluate the performance of our model by the **CEAF-TF** metric following [159]. The metric finds the best alignment of predicted arguments and gold arguments. It penalizes the system that does not merge multiple extraction results by setting a constraint that a gold argument can be aligned with at most one predicted argument. Precision (P), recall (R), and F1-score (F1) are used to evaluate the model's performance.

3.4.3 Implementation Details

Spacy 3.0.3 is used in data preprocessing. Experiments are conducted on NVIDIA GTX 1080Ti, and the training time is about four hours. The hyper-parameters are given in the Table 3.1.

3.4.4 Overall Results

The per-role EAE results on the MUC-4 dataset of our RICB model and baseline models are summarized in Table 3.2, and the micro-averaged performance is shown in Table 3.3.

	PerpInd	PerpOrg	Target	Victim	Weapon
GTT [159]	65.48/39.86/49.55	66.04/42.68/51.85	55.05/44.12/48.98	76.32/61.05/**67.84**	61.82/56.67/59.13
NST [158]	48.39/32.61/38.96	60.00/43.90/50.70	54.96/52.94/53.93	62.50/63.16/62.83	61.67/61.67/61.67
DYGIE++ [184]	59.49/34.06/43.32	56.00/34.15/42.42	53.49/50.74/52.08	60.00/66.32/63.00	57.14/53.33/55.17
RICB	50.76/49.62/**50.18**	50.00/63.75/**56.04**	65.63/63.64/**64.62**	64.86/50.52/56.80	63.49/65.57/**64.51**

TABLE 3.2: Performance comparison with baseline models for each argument role on MUC-4 dataset. Results for each column are displayed in the order of precision, recall, and F1 score.

Models	P	R	F1
GTT [159]	**64.19**	47.36	54.50
NST [158]	56.82	48.92	52.58
DYGIE++ [184]	57.04	46.77	51.40
KDR [186]	52.91	57.63	55.16
RICB	57.68	**58.03**	**57.85**

TABLE 3.3: Averaged EAE result on the MUC-4 dataset. Precision (P), recall (R), and F1-score are used for evaluation.

	PerpInd	PerpOrg	Target	Victim	Weapon
Without graph2token	50.39/49.24/49.80	50.02/58.83/54.07	63.87/57.58/60.56	62.54/49.53/55.28	58.72/69.47/63.64
Cross entropy loss	50.00/50.34/50.17	48.57/63.75/55.14	62.04/64.39/63.19	49.55/58.95/53.85	55.13/70.49/61.87
String matching	48.80/45.86/47.28	45.30/66.25/53.81	65.71/63.44/64.56	59.49/49.47/54.02	58.57/67.21/62.60
RICB	50.76/49.62/**50.18**	50.00/63.75/**56.04**	65.63/63.64/**64.62**	64.86/50.52/**56.80**	63.49/65.57/**64.51**

TABLE 3.4: Ablation studies on graph2token module, closed boundary loss, and entity summary graph, respectively. The results in each column are displayed in the order of precision, recall, and F1 score.

Table 3.3 shows that our model consistently outperforms the latest baselines in F1-score and achieves the state-of-the-art (SOTA) performance. Specifically, the proposed model improves the absolute F1-score by 3.35%, 5.27%, and 6.45% compared with baseline models. Noticeably, our model achieves an over 9% improvement in recall. In terms of the per-role extraction performance of our model, it achieves the highest F1-score in four of five argument roles: perpetrator individual, perpetrator organization, target, and weapon. Specifically, the absolute F1-score is improved by 0.63%, 4.19%, 10.69%, and 2.84% in these argument roles. Notably, the DEEDP baseline, which was published subsequent to our work, achieved an F1-score of 58.71. Our method achieves comparable performance to this more recent work.

3.4.5 Effect of Graph2token Module

Graph structure is used in EAE to produce a comprehensive representation of coreference entities [27, 178, 179]. In this work, we further adopt a graph2token module to feed back the comprehensive representation of coreference entities to their context tokens. The updated token representations can generate additional coreference-aware representations for entities near the coreference entity. For the ablation study, we experiment on without applying the graph2token module. We compare per-role extraction results with and without the graph2token module in Table 3.4. We find that the experiment without the graph2token module results in a performance drop on every argument role. In addition, the recall is decreased by 0.38%, 4.92%, 6.06%, and 0.99% in four argument roles. This indicates that the model can recognize more arguments by providing argument candidates with additional coreference-aware representations.

3.4.6 Effect of Closed Boundary Loss

We find that classifier trained by cross entropy loss could easily misclassify entities in the Universum class. We propose a closed boundary loss to address this issue. For the ablation study, we conduct experiments of applying cross entropy loss for argument classification, and compare the performance with our model. The comparison of two loss functions is summarized in Table 3.4, which shows that in all argument roles, closed boundary loss consistently outperforms cross entropy in the F1 score. We further notice that the precision of the model is improved in all argument roles at 0.76%, 1.43%, 3.59%, 15.31%, and 8.36% by using closed boundary loss. The variations in precision enhancement primarily stem from the differing sample compositions across classes. When using cross-entropy loss, many Universum class samples are incorrectly classified into the victim class. This leads to a notable improvement in precision after switching to closed boundary loss. The improvement in precision indicates that the use of closed boundary results in a smaller number of Universum samples that are misclassified as target samples.

… [2] The massacre against the Salvadoran Workers National Union Federation (FENASTRAS) was planned in cold blood. … [4] We have trustworthy information from our intelligence organs that this action was ordered by Colonel Ponce, that Cristiani knew about it and approved it. … [6] Terrorism is an old practice of the Nationalist Republican Alliance (ARENA). [7] Only a few days ago, ARENA assassins tried to kill the president of the Mortgage bank, Mr Mason, for not following their orders. [8] The people demand the resignation of chief of staff Col Emilio Ponce because his responsibility in this criminal action is real. … [18] The war of the armed forces, government, and ARENA is aimed against the people. …

	Peretrator Individual	Perpetrator Organization
GTT	ARENA assassins	-
RICB	Colonel Ponce, ARENA assassins	ARENA

FIGURE 3.5: An example of the differences in event argument extraction between GTT and our proposed RICB. The differences in extracting perpetrator individual and perpetrator organization are used for illustration. RICB successfully extracts *Colonel Ponce* and *ARENA*, while GTT fails. In the example, sentence numbers are marked in green, and identical entities are marked with the same color.

3.4.7 Effect of Entity Summary Graph

To fully leverage the redundant argument information, we classify every entity in the document. For the same argument, we may obtain multiple preliminary extraction results. We propose the entity summary graph to merge the results. For the ablation study, we conduct experiments on merging multiple extraction results based on string matching following Xu et al. [27], Zheng et al. [157]. We compare the string matching performance with our proposed entity summary graph in Table 3.4. It shows that the entity summary graph outperforms the string matching method significantly in the F1-score. Furthermore, the precision of the model is improved in four of five argument roles by 1.96%, 4.70%, 5.37%, and 4.92% by using the entity summary graph, and this verifies the effect of our proposed entity summary graph, i.e. merging multiple extraction results and reducing false positives accordingly.

3.4.8 Case Study

Figure 3.5 demonstrates an example of the differences in predicting event arguments between GTT [159] and our proposed RICB method. To avoid involving excessive sentences in the document, only roles of perpetrator individual and perpetrator organization are used for illustration. RICB successfully extracts "Colonel Ponce" and "ARENA", while GTT fails. Both event arguments "Colonel Ponce" and "ARENA" appear multiple times in the document, which shows the redundant event information in the document.

Specifically, among all their occurrences in the document, it is easier to recognize "Colonel Ponce" from sentence 8 and recognize "ARENA" from sentence 7. This is an illustration of our idea that by utilizing redundant event information in the document, we can extract arguments from relatively simple positions. In addition, to recognize "Colonel Ponce" from sentence 4, it is necessary to understand that "this action" refers to "the massacre". Our model can recognize it because the graph2token module can feed back the coreference information to "this action".

3.4.9 Further Analysis

Firstly, it is effective to leverage redundant event information in documents for document-level EAE, which is not only reflected in the F1 score, but also in the significant improvement in recall. The micro-averaged recalls of baseline models are distributed between 46% to 49%, but our model reaches 58%. As we analyzed in the introduction, leveraging redundant argument information of a document allows the model to extract the argument from any of its occurrences and relatively simple positions. Therefore, the difficulty of argument extraction is reduced and the recall is improved accordingly. We also notice a drop in precision rate in our model compared to baseline models. It is because baseline methods adopt sequence-to-sequence models and we classify a few arguments from a great number of entities in the document, which will naturally result in a decrease in precision. However, the precision and recall of our model are very close, which is more balanced compared to baseline models.

Secondly, leveraging redundant event information in a document is not simply classifying every entity in the document. On the one hand, better entity representations need to be produced, on the other hand, multiple extraction results need to be merged. Therefore, we add the graph2token module to the entity coreference graph, which improves the recall significantly. We also propose the entity summary graph to merge multiple extraction results, which successfully improves the precision.

Additionally, we propose a novel closed boundary loss to better deal with the Universum class in our task. Its effectiveness is verified in ablation studies. We highlight two other potential benefits of closed boundary loss here. Firstly, since it generates a closed decision boundary for classifiers, it may also be valid for dealing with unseen samples in the test set. This property is not evaluated in this work. In addition, our dataset is highly imbalanced because only a small number of entities are arguments. Weighted

cross entropy loss is cumbersome to adjust the appropriate weights, however, the closed boundary loss does not need to adjust weights and works well with the imbalanced dataset.

Finally, the redundancy in event arguments can vary significantly across different domains. In domains like news, which is extensively examined in this work, as well as in clinical or financial documents, there is typically a high repetition of event arguments, making our proposed redundancy-based methods particularly effective. However, in domains where event information is presented more succinctly, our method may contribute less to performance improvements. Our method is particularly targeted for scenarios like news, where there is usually a high repetition of event arguments.

3.5 Conclusion

In this work, we emphasize that the redundant event information in documents is beneficial but is often overlooked in document-level EAE. In addition, we find that classifiers trained by cross entropy loss are problematic in classifying the Universum class. Specifically, we generate a comprehensive and coreference-aware representation for every entity through the entity coreference graph with the graph2token module. In addition, we propose an entity summary graph to merge the multiple extraction results of the same argument. Furthermore, we propose a novel closed boundary loss to deal with the Universum class in classification. As a limitation, our proposed closed boundary loss is used for binary classification because we extract arguments in a role-by-role manner to make full use of the property of each argument role. In the future, we will extend it for multiclass classification and apply it to other tasks in natural language processing that face the problem of classifying Universum class. Experimental results show that our RICB model achieves the SOTA performance and outperforms prior approaches on the MUC-4 dataset.

Chapter 4

LLMs Learn Task Heuristics from Demonstrations: A Heuristic-Driven Prompting Strategy for Document-Level Event Argument Extraction

4.1 Introduction

In the previous chapter, we introduced a method for document-level EE to align with real-world scenarios. However, another critical demand exists for the practical application of EE systems: the reliance on labeled data significantly limits their use in real-world settings due to the substantial cost of human annotations and the challenge of adapting systems to new event types. In this chapter, we aim to address this dependency on annotated data by leveraging the in-context learning (ICL) paradigm of large language models (LLMs). Additionally, we delve deeper into the workings of ICL by investigating what LLMs learn from this paradigm.

Document-level Event Argument Extraction (EAE) aims to transform unstructured event information from documents into structured formats encapsulating event arguments, facilitating their interpretation and application in various domains [14]. The prevalent

approach for this task relies on the collection of labeled data and the subsequent model training via supervised learning [11, 158, 187–189]. While effective, this approach comes with a significant drawback: it necessitates a substantial amount of training data, which is particularly burdensome and costly given the complexity inherent to document-level EAE.

In this context, ICL [49, 51, 52], an emergent ability of LLMs, offers a promising alternative to supervised learning. ICL alleviates the need for large-scale data as it only uses a few examples as input-output pairs of the prompt to guide LLMs in performing the task on an unseen example.

However, applying ICL to document-level EAE presents numerous challenges. The ICL performance is highly sensitive to the design of in-context demonstrations, such as the selection of examples and the formatting of reasoning steps [126, 128, 190]. Consequently, several crucial challenges emerge concerning the prompting strategy:

- **Example Selection Challenge.** Selecting optimal in-context examples for ICL is pivotal, yet the understanding of what LLMs learn from these examples remains largely under-explored [4, 191]. This gap leads to a lack of systematic guidelines, resulting in a disorganized and inefficient example selection process.
- **Context Length Limit**. In document-level EAE, selecting multiple documents as ICL examples could significantly extend the context length, potentially surpassing the token limit of LLMs.
- **Abundance of Event Types.** The EAE task can involve more than a hundred distinct event types and argument roles. Yet, ICL examples can only capture a narrow subset, leaving the majority of argument roles unseen. Handling unseen classes beyond limited ICL examples is a common problem in classification tasks with diverse class types.
- **Prompting Strategy for Non-reasoning Task.** While the chain-of-thought (CoT) prompting is extensively used across a variety of tasks, its effectiveness is compromised in non-reasoning scenarios. As shown in Figure 4.1, applying CoT to non-reasoning tasks will degrade its step-by-step reasoning into an potentially inadequate single-step. Consequently, there is a need for prompting strategy tailored for non-reasoning tasks.

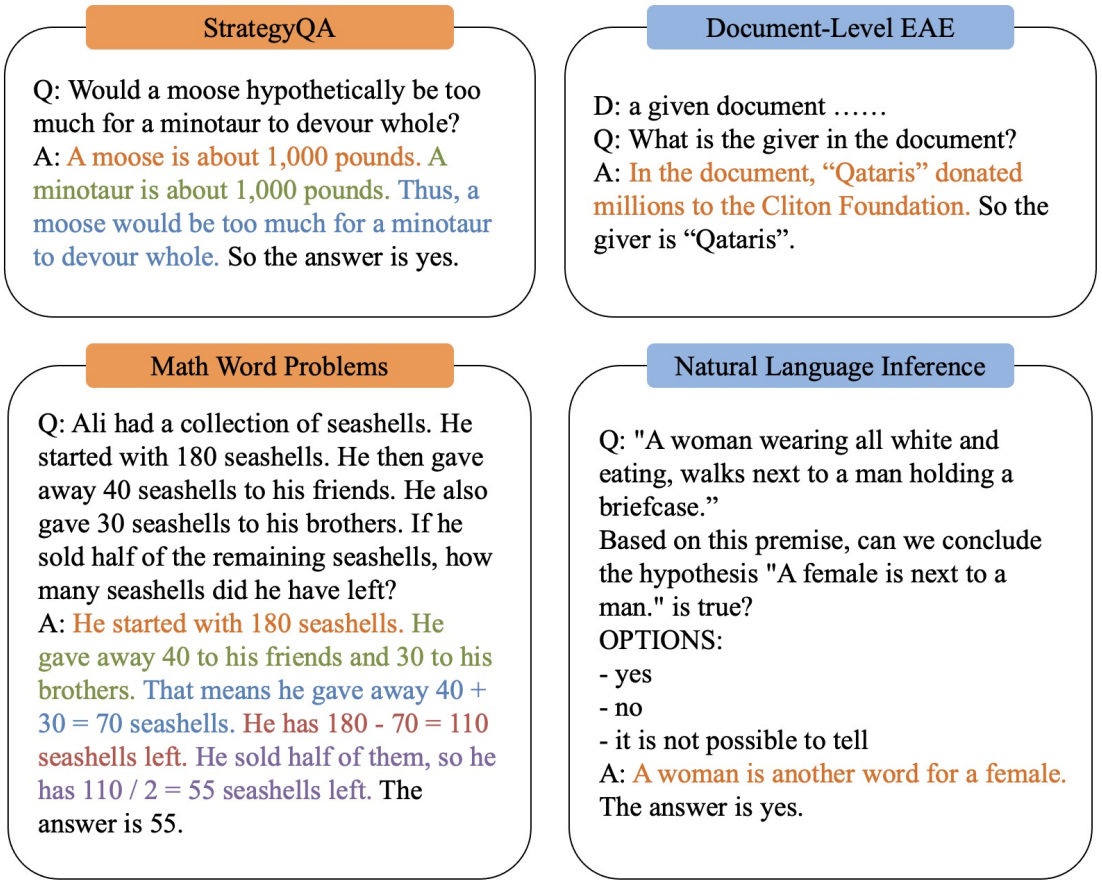


FIGURE 4.1: CoT's step-by-step reasoning degrades to a single step for non-reasoning tasks. Reasoning steps of reasoning tasks (in orange) and non-reasoning tasks (in blue) are compared. Different colors indicate distinct reasoning steps. Prompts are from [8].

In this work, we put forward a novel hypothesis that LLMs learn task-specific heuristics from examples and validate it through experiments. Building upon this hypothesis, we propose heuristic-driven link-of-analogy prompting to address the aforementioned challenges. To elaborate:

We propose and empirically validate the hypothesis that LLMs learn task specific heuristics from examples in ICL. Heuristics, defined as *'a high-level rule or strategy for inferring answers to a specific task'*, play a crucial role in human cognition. Humans use heuristics as efficient cognitive pathways, which often lead to more accurate inferences than complex methods [192, 193]. Similarly, in supervised machine learning (ML) systems, models also learn task-specific patterns through training [194, 195]. Drawing a parallel, we hypothesize that LLMs learn task-specific heuristics from explanations of in-context examples to aid inference. We qualitatively illustrate how heuristics are

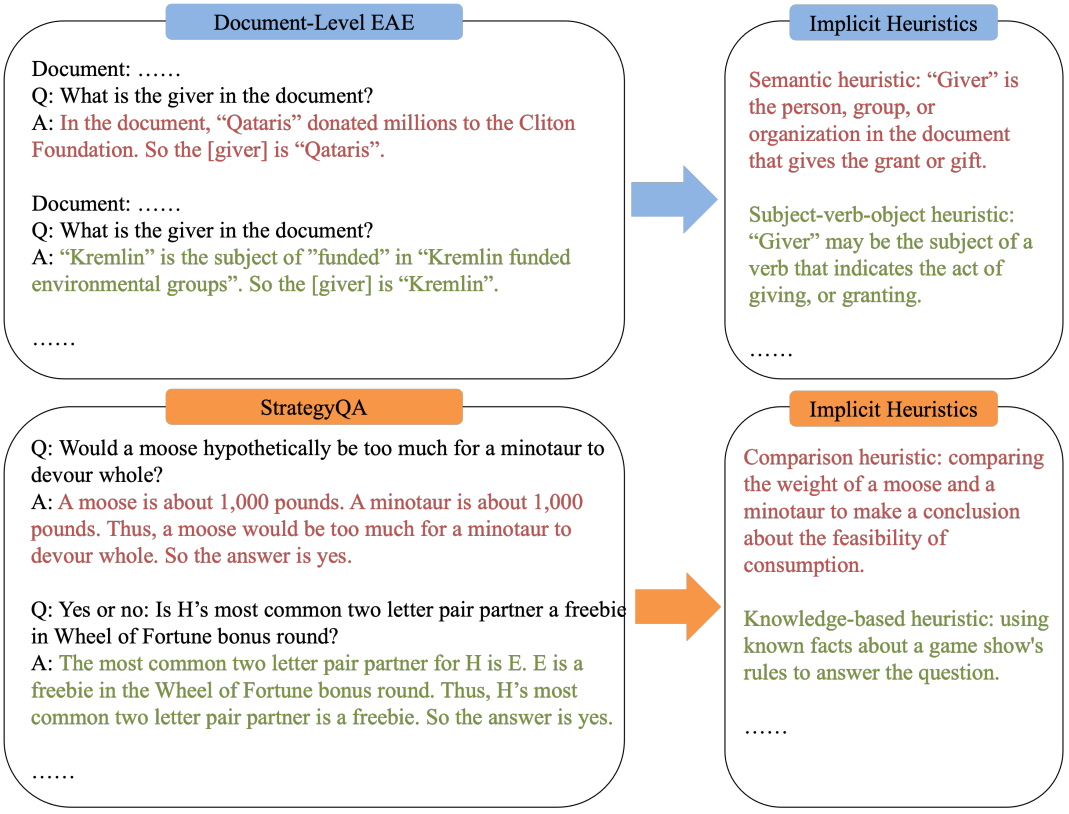


FIGURE 4.2: Heuristics are implicitly embedded within explanations of in-context examples.

implicitly embedded in explanations of in-context examples in Figure 4.2, and quantitatively validates our hypothesis with experiments detailed in Section 4.2.

Notably, while drawing parallels to supervised ML, ICL is fundamentally different from supervised ML in its mechanism: supervised ML learns and updates model parameters during training, whereas LLMs do ICL with all parameters frozen. Therefore, understandings of supervised ML systems (e.g. pattern learning) are not applicable for ICL [125, 131], which necessitates distinct explorations on the mechanism of ICL.

We propose a heuristic-driven demonstration construction method. Based on our hypothesis, task heuristics are crucial for the ICL performance of LLMs, yet they are often *implicitly* conveyed through examples. This implicitness complicates the examination of whether demonstrations contain diverse heuristics and leads to uncertainty about whether LLMs have recognized these heuristics. Furthermore, the selection of in-context examples remains an underexplored challenge for ICL. To address these issues, in parallel with human's exploitation of explicit heuristics, our method *explicitly* incorporates task heuristics into demonstrations, transforming the haphazard example selection process into a systematic method that emphasizes task heuristics.

We propose the link-of-analogy prompting method that is suitable for non-reasoning tasks. To address the aforementioned challenges of abundance of event types in EAE and the limitations of CoT prompting on non-reasoning tasks, we present the link-of-analogy prompting. Inspired by the analogical reasoning—a core mechanism of human cognition, this approach enables LLMs process new situations (new classes) through drawing an analogy to known situations (known classes). Empirical results demonstrate its effectiveness in enhancing the ICL performance for classes not seen in ICL examples. Our contributions are as follows:

- We introduce a pioneering work to prompting strategies for the document-level EAE, showcasing significant accuracy improvements on two document-level EAE datasets compared to prompting methods and few-shot supervised learning methods.
- We investigate what LLMs learn from ICL, and unveil a new insight that LLMs learn task-specific heuristics from ICL examples.
- We propose an heuristic-driven demonstration construction approach, tackling the example selection issue with a fresh perspective on task heuristics, facilitating explicit heuristic learning in ICL. Furthermore, we propose the link-of-analogy prompting, which allows LLMs to process new situations by drawing analogies to known situations.
- To further evaluate the adaptability of our method, we validate it on the sentiment analysis and natural language inference tasks, achieving notable accuracy enhancements.

4.2 What do LLMs learn from the demonstration?

The understanding of what LLMs learn from the demonstration of ICL remains an open question. In this work, we hypothesize that **LLMs learn task-specific heuristics from examples in ICL**. We validate this hypothesis with carefully designed experiments in three aspects.

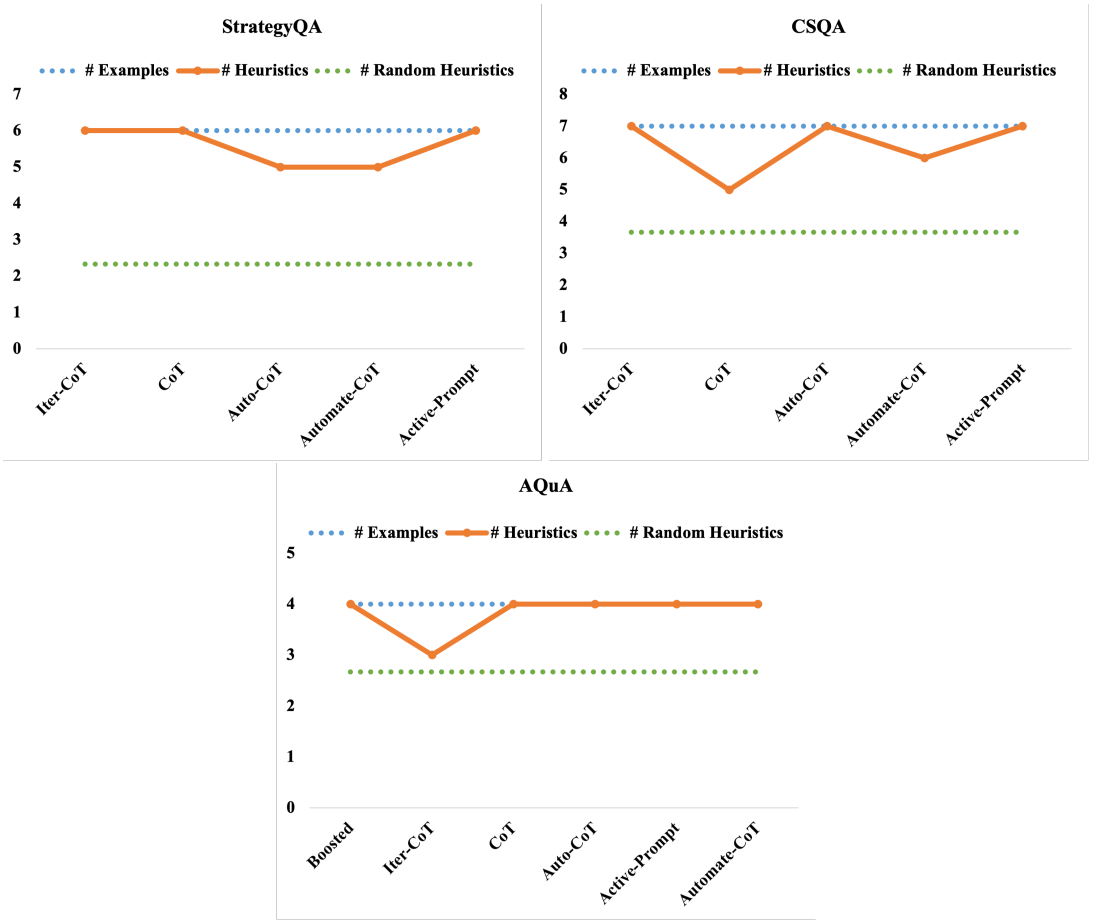


FIGURE 4.3: An illustration of the correlation between example quantity and heuristic diversity in well-designed prompts. # Examples: the number of examples used in each prompt of the corresponding paper. # Heuristics: the number of heuristics identified in each prompt of the corresponding paper. # Heuristics in Rand.: the average number of heuristics in the randomly constructed prompt.

4.2.1 Correlation between Example Quantity and Heuristic Diversity in Well-Designed Prompts

Our first experiment operates on the assumption that *if LLMs indeed learn task-specific heuristics from demonstrations, then successful prompts should inherently incorporate a diverse range of heuristics in their examples*, as these heuristics are learnable for LLMs. To examine this proposition, we assess both the quantity of examples and the quantity of different embedded heuristics within prompts from published papers, which are categorized as well-designed prompts. The prompts constructed from randomly selected samples are utilized as baselines in this experiment.

To objectively identify the number of implicit heuristics embedded in prompts, we

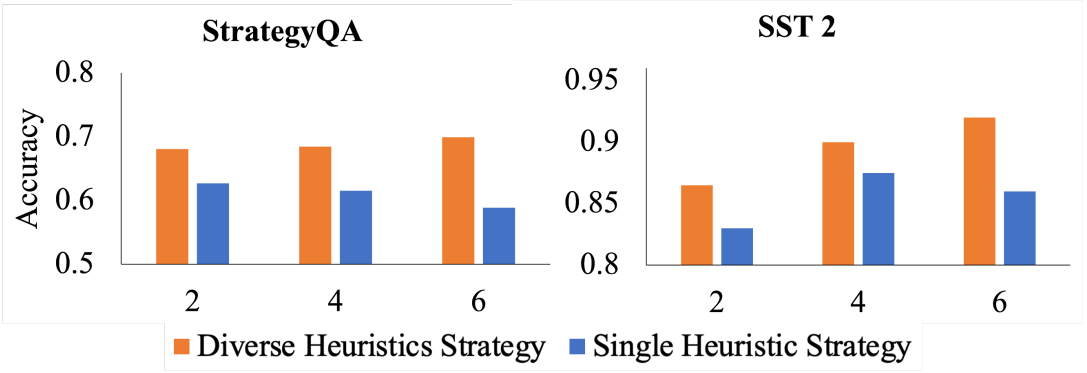


FIGURE 4.4: Comparison of ICL performance using single-heuristic strategy versus diverse-heuristics strategy across different number of example on the StrategyQA and SST-2 Dataset.

employ `GPT-4` to recognize the embedded heuristic for each example and to determine if it is a shared heuristic across multiple examples. An detailed example of the prompt we used and the heuristics identified by `GPT-4` can be found in Appendix A.1.

We investigate the correlation between the number of examples in a prompt and the number of embedded heuristics within the same prompt, analyzing six SOTA prompting methods applied across three distinct datasets. Specifically, prompting methods including CoT [5], Automate-CoT [8], Auto-CoT [128], Iter-CoT [196], Boosted [197], Active-CoT [198] are investigated and datasets of commonsense reasoning and arithmetic reasoning are evaluated. Our findings in Figure 4.3 reveal that: in well-designed prompts, the number of heuristics closely matches the number of examples. Furthermore, the number of heuristics in carefully constructed prompts significantly exceeds that in randomly constructed prompts. This observation substantiates our statement that successful prompts indeed embed a wide array of heuristics in examples.

4.2.2 Comparing Diverse-Heuristics and Single-Heuristic Strategies

The second experiment empirically evaluates how the diversity of heuristics within examples impacts ICL performance of the LLM. This experiment is premised under the assumption that *if LLMs **cannot** learn heuristics from demonstrations, then demonstrations featuring multiple heuristics should yield similar performance with those incorporating a single heuristic*, as heuristics cannot be utilized. To explore this, we compare two distinct example selection strategies. The single-heuristic strategy formulates prompts where all explanations of examples follow a same heuristic. Conversely,

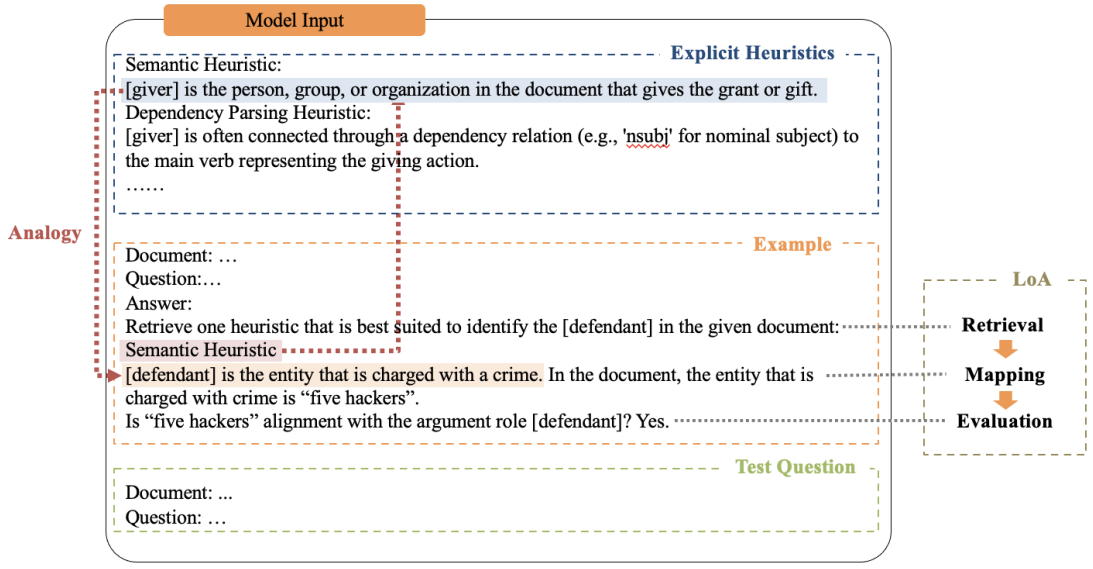


FIGURE 4.5: An illustration of HD-LoA prompting.

the diverse-heuristic strategy constructs prompts where all explanations of examples exhibit different heuristics. We construct prompts that follow these two strategies based on prompts in Shum et al. [8], Diao et al. [198].

The performance comparison of prompts constructed by the two different strategy on the StrategyQA [199] and SST-2 [9] datasets is illustrated in Figure 4.4. The results indicates that, given an equal number of examples, the diverse-heuristics strategy significantly outperforms the single-heuristic approach, which contradicts the assumption. This finding not only validates our hypothesis that LLMs can learn heuristics from in-context examples but also underscores the value of incorporating a variety of heuristics in enhancing ICL performance.

	ER	Comp	KB	Def	Chron	Others
Count	14	55	125	47	91	168

TABLE 4.1: Distribution of samples by heuristic type. "*Others*" includes samples with heuristics not categorized in the predefined types.

	ER	Comp	KB	Def	Chron	Others
Original Demonstration	78.5	72.7	87.2	85.1	74.7	65.5
Heuristic Deduction	71.4 (**-7.1**)	65.4 (**-7.3**)	81.6 (**-5.6**)	82.9 (**-2.2**)	70.3 (**-4.4**)	-

TABLE 4.2: Performance comparison between original demonstration and a demonstration with heuristic deduction (replacing the example of a distinct heuristic type with another example containing a repeated heuristic type).

4.2.3 Impact of Heuristic Deduction Towards ICL Performance

To validate our hypothesis, we further investigate the impact of reducing an implicit heuristic embedded in demonstration examples. If the classification accuracy of test samples corresponding to this heuristic decreases accordingly, we can validate that LLMs learn heuristics from demonstrations.

We use 500 test samples from the StrategyQA [199] dataset and the prompt from Shum et al. [8] for evaluation. As discussed in Section 4.2.2, we use `GPT-4` to identify all implicit heuristics embedded in examples of the demonstration: empathetic reasoning (ER), comparison (Comp), knowledge-based (KB), definition-based (Def), and chronological (Chron) heuristics. The prompt for implicit heuristic identification and LLM output are detailed in Appendix A.1. We then employ `GPT-4` to label each test sample with the corresponding heuristic that could be used to guide the prediction of the sample. Next, we group the test samples by heuristic type, and the statistics are illustrated in Table 4.1. Finally, given the prompt embedded with five different heuristic types, we eliminate the demonstration of a specific heuristic type by replacing its example with another example containing a repeated heuristic type, and monitor the performance change in the corresponding test group. `gpt-4-1106-preview` is used for evaluation.
Experimental results are demonstrated in Table 4.2. These results indicate that eliminating the demonstration of a certain heuristic type indeed results in a significant performance drop in the test samples associated with that heuristic, further substantiating our hypothesis that LLMs learn task-specific heuristics from examples. Interestingly, we also observe that samples with heuristics not represented in the demonstration examples (*Others* samples) show significantly lower accuracy, which not only support our hypothesis, but also shed light on example selection, suggesting that selecting examples with their implicit heuristics that cover a wider range of test samples is likely to enhance the ICL performance.

4.3 Heuristic-Driven Demonstration Construction

Building on our understanding of heuristic learning during ICL, we aim to address the challenge of example selection for ICL. Experiments in Section 4.2 indicates that heuristics are crucial for ICL performance of LLMs, yet they are *implicitly* conveyed through explanations of examples. This implicitness complicates the examination of

whether ICL demonstrations contain diverse heuristics and leads to uncertainty about whether LLMs have recognized these heuristics. Additionally, when solving a task, humans possess the ability to not only learn from examples but also learn from heuristics for efficient and accurate inference [192]. This leads us to question whether LLMs can similarly leverage *explicit* heuristics to improve ICL performance. Therefore, we are motivated to *explicitly* providing LLMs with task-specific heuristics. Our approach is illustrated below:

Replacing examples with explicit heuristics: Diverging from traditional prompting strategies that construct prompt with examples where heuristics are implicitly embedded, we propose to replace most examples in the prompt with distinct task-specific heuristics, as demonstrated by the heuristics in Figure 4.5.

Retaining minimum examples: A minimal number of examples are preserved to (1) illustrate the formatting of target task and reasoning steps, such as one example is required to illustrate the format of our link-of-analogy prompting, and (2) ensure a balanced coverage of labels in prompt to avoid introducing label bias. Specifically, for document-level EAE task, a single example is maintained to demonstrate the reasoning format.

Heuristic generation: A remaining question is how to create the explicit heuristics in the prompt. Both human crafted heuristics and LLM-generated heuristics can be adopted as explicit heuristics. To automate this process, we utilize `GPT-4` to generate a set of distinct heuristics $S = \{s_1, s_2, ..., s_n\}$ for the document-level EAE task. We adopt $n = 10$ in this work. The prompt for heuristic generation and its output are provided in the Appendix A.2.

Heuristic selection: Given that not every generated heuristic may suit the target task, we introduce a heuristic selection step. Each heuristic in the generated heuristic set S is individually adopt into a prompt, the ICL performance of each heuristic is evaluated using a subset of the training dataset. Specifically, we employ 1% of the training dataset, identical to the sample size used in the few-shot supervised learning baseline. Through this evaluation, the top-performing heuristics, determined by accuracy, are selected to constitute the explicit heuristic list $\mathbf{H}$ in our prompt. We adopt the top 3 heuristics in this work.

Through this heuristic selection step, low-quality heuristics are excluded. For example, the semantic role labeling heuristic generated in the heuristic generation step (in Appendix A.2) is too specific and of lower quality, thus it demonstrates a significantly low evaluation accuracy (26.52%) compared to a high-quality syntactic heuristic (33.69%).

There are three advantages of our approach. Firstly, it provides a guidance on the example selection process. The example selection process of ICL is often an indiscriminate, manual process [5, 52, 200]. However, our method converts the directionless and indiscriminate process into a methodical approach that emphasizes task-specific heuristics. Secondly, by emulating human cognitive strategies that leverage explicit heuristics for improved inference—a technique supported by cognitive studies [192]—our method enables LLMs to also benefit from heuristic learning during ICL. Finally, it condenses lengthy examples that consists of input-output pairs into compact heuristics, reducing the context length of prompts.

4.4 Link-of-Analogy Prompting

We propose the link-of-analogy prompting to address the challenges below: First, the EAE task is characterized by its extensive variety of argument roles and event types, often exceeding a hundred, yet ICL examples can only cover a very limited subset. This discrepancy raises a critical challenge: designing a prompting strategy that effectively addresses unseen event types. Notably, the issue of handling unseen classes beyond limited ICL examples is a prevalent problem in various NLP tasks. Additionally, to concretize heuristic generation process, we provide heuristics for a specific argument, *giver*, within the prompt. This leads to the question of how to extend *giver* heuristics to other argument roles. Finally, as highlighted in the Introduction, applying CoT prompting to non-reasoning tasks tends to degrade the step-by-step analysis into a one-step rationale [8, 198], necessitating more proper prompting strategies for such tasks.

Inspired by the analogical reasoning [201], a core mechanism of human cognition, we seek to resolve the challenges presented. Humans often understand a new situation by drawing an analogy to a familiar situation. For example, students often solve new problems by mapping solutions from known problems [202]. Similarly, we anticipate that LLMs will be able to extract information of unseen events or generate heuristics for unseen argument roles by drawing an analogy to events and heuristics provided in in-context examples. Empirically, we find that LLMs are indeed capable of doing analogical reasoning when prompted appropriately. For example, when provided with the heuristic for *giver* in the prompt: "*[giver] is the person, group, or organization in the document that gives the grant or gift*", LLMs can make an analogy and generate

the heuristic for the argument *vehicle* in the target question: "*[vehicle] is the means of transport used to move the person or object*".

To further enhance the analogical reasoning capabilities of LLMs, we introduce our link-of-analogy (LoA) prompting strategy, which emulates the analogical reasoning process of human. Cognitive science studies reveals that humans perform analogical reasoning through a sequence of *retrieval*, *mapping*, and *evaluation* [203, 204]. In alignment with this process, our method involves the same steps. Specifically, in the retrieve step, given the base argument role r_b, a set of heuristics $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_k\}$ for identifying r_b, a target question and a target argument role r_t, the LLM will select the most suitable heuristic $\mathbf{h}_b$ from $\mathbf{H}$ for identifying r_t. In the mapping step, the LLM employs analogy mapping $r_b : \mathbf{h}_b :: r_t : \mathbf{h}_t$ to deduce the heuristic $\mathbf{h}_t$ for r_t. The LLM then infers the argument $\mathbf{a}_t$ of the target role based on the heuristic $\mathbf{h}_t$. Finally, in the evaluation step, the LLM will reassess the identified argument $\mathbf{a}_t$. This methodology is exemplified in the in-context example presented in Figure 4.5.

4.5 Experiments

In this section, we aim to explore the following research questions (RQs) regarding our **H**euristic-**D**riven **L**ink-**o**f-**A**nalogy (HD-LoA) prompting. **RQ1** Does HD-LoA prompting improve in-context learning performance in document-level EAE task? **RQ2** Can HD-LoA prompting effectively mitigate the dependency on extensive labeled data while enhancing accuracy for EAE task? **RQ3** Is the HD-LoA prompting effective when applied to tasks beyond EAE? **RQ4** Do each components of the HD-LoA prompting effectively contributing to its performance?

4.5.1 Experimental Setup

Dataset: For the evaluation of the document-level EAE task, we adopt RAMS [165] and DocEE [209] datasets. The WIKIEVENTS dataset [161] is excluded from our study because it relies on preprocessed entity candidates for annotating event arguments the annotation, which diverges from the direct argument identification of LLMs. For evaluation, we follow the metrics in [205], namely the argument identification F1 score (Arg-I), and the argument classification F1 score (Arg-C).

Method		RAMS		DocEE-Normal	DocEE-Cross
		Arg-I	**Arg-C**	**Arg-C**	**Arg-C**
Supervised learning (few-shot)	EEQA [153]	-	19.54	-	-
	PAIE [205]		29.86		
	TSAR [206]		26.67		
	CRP [11]		30.09		
	FewDocAE [207]		-	12.07	10.51
`text-davinci-003`	Standard [208]	39.96	31.6	25.55	25.41
	CoT [5]	43.03	34.94	27.68	28.64
	HD-LoA (ours)	**46.17**	**39.59**	**30.22**	**31.03**
`gpt-3.5 -turbo-instruct`	Standard [208]	42.44	32.46	25.67	24.48
	CoT [5]	40.63	33.64	26.77	25.99
	HD-LoA (ours)	**43.34**	**37.05**	**27.98**	**27.34**
`gpt-4`	Standard [208]	44.73	37.08	29.53	27.36
	CoT [5]	44.93	38.09	30.32	30.95
	HD-LoA (ours)	**52.41**	**44.12**	**31.53**	**33.48**

TABLE 4.3: Overall Performance: Evaluated using the F1-score for Argument Identification (Arg-I) and Argument Classification (Arg-C). In few-shot setting, the scores of supervised learning methods on RAMS dataset are based on results reported in Liu et al. [11], where 1% of the training data is used.

Dataset	Task Type	# Example	# Eval.	Eval. Split
RAMS [165]	Doc-Level EAE	1	871	Test
DocEE [209]	Doc-Level EAE	1	800	Test
SST-2 [9]	Sentiment Analysis	2	872	Validation
SNLI [210]	Natural Language Inference	3	500	Test

TABLE 4.4: The overall statistics of the dataset. # Example: The number of examples used in the HD-LoA prompting. # EVAL.: the number of samples used for evaluation of different prompting methods. EVAL. Split: evaluation split.

Additionally, we utilize the SST-2 [9] and SNLI [210] datasets to assess the effectiveness of our HD-LoA prompting strategy on other non-reasoning tasks: sentiment analysis and natural language inference.

The statistics of the dataset are provided in Table 4.4. We use the test split of RAMS dataset and the validation split of SST-2 for evaluation, following the setting in Wang et al. [211]. Considering the extensive size of the DocEE and SNLI datasets, which makes a full-scale evaluation using LLMs impractical, we follow Shum et al. [8], Wang

et al. [211] and evaluate a subset of these datasets. Owing to the substantial costs associated with deploying GPT-4, we restrict its evaluation on the RAMS dataset and DocEE dataset to 200 samples. In addition, regarding the DocEE dataset, it presents two distinct settings. In the conventional configuration, the training and testing data share an identical distribution. Conversely, the cross-domain setup features training and testing data composed of non-overlapping event types.

Baselines Our HD-LoA approach is compared against several state-of-the-art prompting methods, including the standard prompting [208] used in clinical EAE, and the Chain-of-Thought (CoT) prompting [5]. Agrawal et al. [208] presents the only existing method that prompts LLMs in the context of EAE task. Given to its direct question-and-answer format, we refer to it as 'Standard Prompting' in accordance with terminology prevalent in ICL research [5]. Notably, as there is no existing prompting strategies tailored for EAE, neither the standard prompting nor CoT prompting has been applied to document-level EAE datasets in the literature. Thus, we report the reproduced results here.

Additionally, we compare our method with various supervised learning methods in EAE, such as FewDocAE [207], CRP [11], PAIE [205], TSAR [206], EEQA [153], etc. The few-shot comparison results are based on the few-shot performance reported in Liu et al. [11].

LLMs: The experiments are carried out using three large language models: the publicly available GPT-3 [49] in its `text-davinci-003` and `gpt-3.5-turbo-instruct` versions [212], as well as `GPT-4` [213]. Notably, due to the high cost associated with GPT-4, its evaluation is limited to part of the dataset. The pricing for running these models ranges from USD 0.0015 per 1,000 tokens to USD 0.03 per 1,000 tokens. The `gpt-3.5-turbo-instruct` model is of the lowest cost but exhibits limited reasoning capabilities. We employ these LLM models from the OpenAI API. During the all experiments, the temperature is fixed as 0. Setting the temperature to a low value can make the output of the LLM more deterministic and repetitive, adhering closely to the most likely outcomes and reducing the occurrence of hallucinations.

Furthermore, our heuristic-driven demonstration construction method necessitates far fewer examples than traditional prompting methods, only keeping the minimum number of examples to avoid bias in example answers. Specifically, for the EAE task, we use only one example, and for sentiment analysis and natural language inference tasks, two and three examples are employed respectively.

4.5.2 Overall Experimental Results

Addressing **RQ1**, the experimental results presented in Table 4.3 indicate that our HD-LoA prompting significantly enhances in-context learning for document-level EAE task. The HD-LoA method consistently surpasses CoT prompting [5] across all three LLMs and both datasets, achieving the largest F1 score improvements of 4.65%, 3.41%, and 6.03% in Arg-C on each LLM, respectively. In addition, the improvement over the standard prompting [208] reaches 7.99% on the `text-davinci-003` model.

In response to **RQ2**, our HD-LoA method, augmented with external knowledge in heuristics, significantly enhances performance in few-shot settings compared to supervised learning approaches. With only one example adopted in the prompt, our HD-LoA achieves a 9.50% F1 score improvement over the CRP method [11] on the RAMS dataset using the `text-davinci-003` model. Similarly, on the DocEE dataset, our method achieves a substancial 20.52% improvement against FewDocAE [207]. Experimental findings indicate that our method can successfully mitigate the document-level EAE task's reliance on extensive labeled data while enhancing accuracy.

	SST-2	SNLI
CoT	91.39	77.97
HD-LoA (ours)	**94.26**	**80.60**

TABLE 4.5: Evaluation of the HD-LoA prompting on sentiment analysis and natural language inference tasks, measured by accuracy.

4.5.3 Adaptability of HD-LoA Prompting for Other Tasks

In addressing **RQ3**, we have extended our HD-LoA prompting method to sentiment analysis (SA) and natural language inference (NLI) tasks, utilizing the SST-2 [9] and SNLI [210] datasets for evaluation. We adopt the CoT style prompts on these two datasets from Shum et al. [8]. Experimental results are presented in Table 4.5. Compared to CoT prompting, our method gets accuracy enhancements of 2.87% and 2.63% on SST-2 and SNLI datasets, respectively. These findings indicate that our HD-LoA prompting can be effectively adapted to a diverse array of non-reasoning NLP tasks.

Method		RAMS		DocEE-Normal	DocEE-Cross
		Arg-I	**Arg-C**	**Arg-C**	**Arg-C**
Supervised learning	EEQA [153]	48.70	46.70	33.50	24.00
	MG-Reader [158]	-	-	32.90	21.40
	BART-Gen [161]	51.20	47.10	-	-
	OntologyQA [209]	-	-	41.00	29.80
	PAIE [205]	56.80	52.20	-	-
`text-davinci-003`	HD-LoA (ours)	46.17	39.59	30.22	**31.03**

TABLE 4.6: Comparison with Fully Trained Supervised Models.

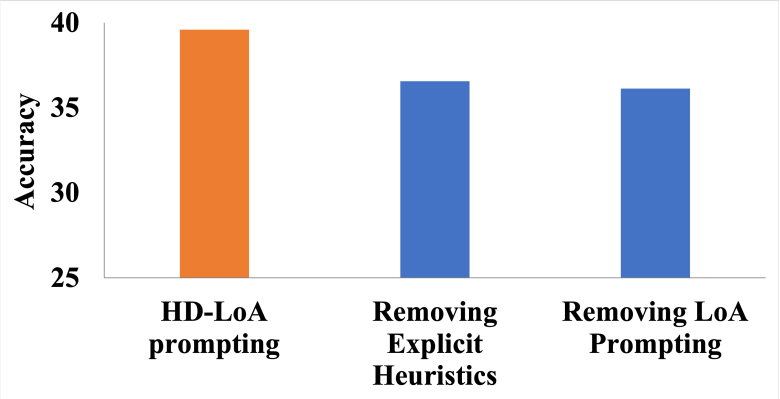


FIGURE 4.6: Experimental results of ablations.

4.5.4 Comparison with Fully Supervised Methods

We compare our HD-LoA method with supervised learning method that trained on the entire dataset for document-level EAE task. As illustrated in Table 4.6, it is anticipated that these models trained on thousands of samples would exhibit higher accuracy compared to our HD-LoA method, which employs only a single labeled sample. Nevertheless, HD-LoA prompting demonstrates competitive performance against supervised methods and even outperform these extensively trained models on the DocEE dataset in the cross-domain setting. This finding also illustrates the effectiveness of our HD-LoA prompting strategy, particularly in scenarios where it is impractical and costly to build large annotated datasets.

4.5.5 Ablations

To address **RQ4**, we conduct further experiments as follows:

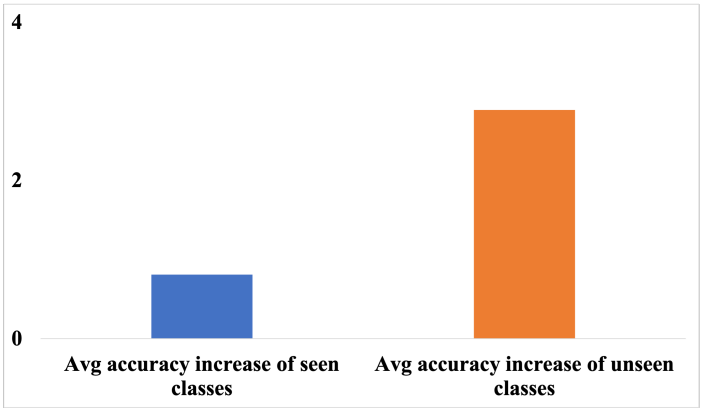


FIGURE 4.7: Seen classes and unseen classes accuracy increase comparison with LoA prompting.

- **Ablation Experiments**: We conduct ablation studies on removing the explicit heuristics and removing the link-of-analogy prompting strategy from our prompt. As presented in Figure 4.6, experimental results on the RAMS dataset demonstrate that removing either the task-specific heuristics or link-of-analogy prompting will significantly degrade the ICL performance of the HD-LoA prompting, suggesting the effectiveness of each component of our prompting strategy.

- **Seen Classes and Unseen Classes Accuracy Increase Comparison for LoA**: To further validate the objective of the LoA prompting strategy, we conduct experiment to validate the effectiveness of the LoA prompting strategy in enhancing ICL performance for unseen classes. Given that in-context examples can only capture a narrow subset of classes (seen classes), leaving the majority of argument roles unseen, we assess and compare the accuracy increase of adopting LoA prompting for seen classes against unseen classes. Experimental results in Figure 4.7 show that LoA prompting results in a more significant accuracy increase on unseen classes compared to seen classes. It indicates that the LoA prompting is indeed effective in enhancing ICL performance on classes unseen in the prompt.

4.6 Understanding Why HD-LoA Prompting Works

Following the empirical validation of the effectiveness of our HD-LoA prompting, this section delves into an analysis to elucidate why our method works.

Analysis of heuristic-driven demonstration construction method: Firstly, our method naturally incorporates diverse distinct heuristics in prompt. As shown in Section 4.2.2,

inclusion of diverse heuristics can significantly boost the ICL performance. In addition, cognitive research finds that humans use heuristics as efficient cognitive pathways to achieve more accurate inferences compared to complex methods [192, 193]. Paralleling this human cognitive strategy, we enable LLMs to learn from explicit heuristics to enhance inference. Specifically, for LLMs demonstrating suboptimal performance with Standard Prompting and in non-reasoning tasks where definitive rationales are elusive, the provision of explicit heuristics offers LLMs helpful strategies to use and enhance inference. Moreover, as discussed in Section 4.2, LLMs use implicit heuristics embedded conventional prompts to facilitate inference. By converting these implicit heuristics to explicit heuristics offers a more straightforward way to utilize heuristics and may potentially simplify the utilization of heuristics by LLMs.

Analysis of the link-of-analogy prompting: The LoA prompting, which is inspired by the analogical reasoning of human cognition, enables LLMs to process new situations by drawing analogies to known situations. This ability is particularly useful in ICL, where LLMs are always facing unseen samples and unseen classes. As evidenced by experiments in our ablation studies, the LoA prompting is indeed effective in enhancing ICL performance for classes unseen in the prompt.

4.7 Related Work

Document-level EAE Existing document-level EAE studies are mostly based on supervised learning methods, which relies on the extensive collection of labeled data [27, 165, 188, 189, 205, 214]. Only Agrawal et al. [208] exploits adopting LLMs on clinical EAE though standard prompts that not involve any reasoning strategies. Considering the potential of ICL to reduce the dependency on large-scale labeled datasets and the revolutionize impact of LLMs, it is lack of study on prompting strategy tailored for the EAE task.

In-context learning ICL enables LLMs to perform a target task by feeding a few prompted examples as part of the input [49]. As the mechanism of ICL is fundamentally different from supervised ML, the working mechanism of ICL remains an open question [4]. Few studies have conducted preliminary explorations: Min et al. [125] showed that the label space, input text distribution and overall format contribute to the ICL performance. Liu et al. [51] concluded that examples that are semantically similar to the test sample are more effective. Akyürek et al. [131] found that transformer-based ICL

can implement standard finetuning implicitly. In this work, we further hypothesize and validate that LLMs learn task-task specific heuristics from examples via ICL.
Moreover, the performance of ICL is very sensitive to example selection [124] and the optimal selection criteria remains unclear. Various studies proposed different ways: selecting examples based on complexity [126], mutual information [127], diversity [128], labeled dataset [8], etc. In this work, we convert the indiscriminate example selection process into a methodical approach that emphasizes task heuristics, making the example selection process more transparent.

4.8 Conclusion

In this work, we hypothesize and validate that LLMs learn task-specific heuristics from demonstrations during ICL, which can provide a guidance and simplify the example selection process. Building upon this hypothesis, we introduce an explicit heuristic-driven demonstration construction strategy, and propose a link-of-analogy prompting method. These methods shed light on the heuristic learning of LLMs and the challenge of handling unseen classes in ICL. Extensive experimentation demonstrates the effectiveness and adaptability of our HD-LoA prompting.

Limitations

Dependency on advanced reasoning abilities of LLMs. In this work, we aim to explore the upper bounds of in-context learning performance on EAE task in the few-shot setting. Our method's reliance on using the sophisticated reasoning capabilities in LLMs makes it unsuitable for models with limited reasoning capabilities. For example, the limited reasoning ability of the `gpt-3.5-turbo-instruct` model could hinder the performance of our method. However, our findings that LLMs can learn heuristics from in-context examples is applicable to diverse LLMs.
Heuristic Quality. The heuristic quality is important for our method. We address this issue by enhancing the probability of generating high-quality heuristics and filtering out low-quality heuristics. We generates an excessive number of heuristic candidates to increase the chances of including high-quality heuristics. Subsequently, we filter out

low-quality heuristics by assessing the accuracy of each heuristic candidate on a small set of samples. Future work could explore more sophisticated heuristic generation strategies, such as generating heuristics with diverse granularity or refining heuristics based on feedback from misclassified examples.

Part II

Important Issues Prevalent Across a Broader NLP Context

Chapter 5

Closed Boundary Learning for Classification Tasks with the Universum Class

5.1 Introduction

During our investigation into EE, we identified a distinct class named the *other* class, which exhibits significantly different properties compared to the other classes. We presented a preliminary exploration of this class in EE scenarios in Chapter 3. However, this class is not unique to EE; it exists widely across various NLP tasks and poses significant challenges for accurate classification without specialized treatment. Therefore, in this chapter, we systematically illustrate the widespread presence of this class and its unique properties. We also introduce a closed boundary learning framework specifically designed to address the challenges posed by this class.

In classification-based tasks, quite often we encounter a class named as *other* class, *miscellaneous* class, *neutral* class or *outside (O)* class. Such a class is a collection of samples that do not belong to any class of interest, such as samples of *no relation* class in relation extraction task. We adopt the terminology in [53] to designate all such classes as the *Universum class* (U). Universum class exits in various classification-based problems in NLP, such as relation extraction (RE) [54], named entity recognition (NER) [55], sentiment analysis (SA) [55], and natural language inference (NLI) [56]. To distinguish

the Universum class and the rest of the classes, we call the classes of interest as *target classes* (T). The set of all classes (A) in the data can be expressed as $A = U \cup T$

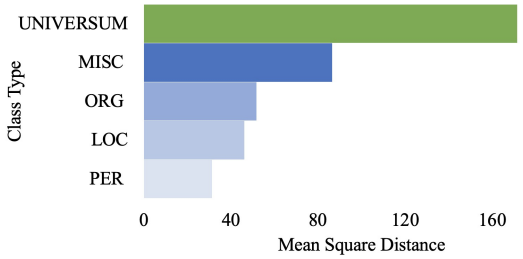
- *Universum class*: A collection of samples that do not belong to any class of interest.
- *Target class*: A class of interest in the task, i.e., one of the classes other than the Universum class.

The sample compositions of the Universum class and target classes are usually very different. Figure 5.1a provides some samples of a target class (*entity-destination*) and the Universum class (*other*) in relation extraction. Intuitively, we can observe that the *entity-destination* samples adhere to an **intra-class pattern**: an entity goes somewhere. However, the three examples of the *other* relation type are vastly dissimilar and do not exhibit any intra-class pattern. In fact, the Universum samples are labeled according to an **inter-class pattern**: they do not belong to any of the predefined target classes.

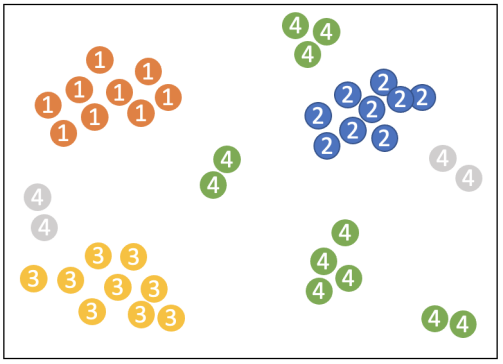
We further highlight the differences between the Universum class and target classes in two properties.
(1) **Heterogeneity**: The Universum class is composed of heterogeneous samples, which may form multiple clusters in the feature space of the test set, as illustrated by the green samples in Figure 5.1c. This is because the Universum class, as the class name "other" implies, contains all potential implicit classes that are not explicitly defined in the task. For example, in samples of the *other* class in Figure 5.1a, implicit classes may include the entity-parallel relationship, the entity-fill relationship, and the entity-narrative relationship.
Although such heterogeneous samples are easily mapped into a compact cluster for the training set, it is problematic for the test set. This is because the inherent predictive rule of the Universum class follows an unique *inter-class* pattern: a sample is labeled as Universum if it does not belong to any target classes. This sharply contrasts with the conventional *intra-class* patterns seen in target classes. Considering human annotation practices, an entity is labeled as *Location* when it aligns with established patterns of *Location* entities. In contrast, a sample is labeled as *Others* not due to intra-class patterns specific to the *Others* class, but because it fails to conform to the patterns of *Location*, *Person*, or *Organization*. Consequently, when current classification models treat the Universum class and target classes in the same manner, they tend to overfit the noise in the Universum class by memorizing various peculiarities of the heterogeneous

Entity-Destination Relation	*Other* Relation
The famous actress arrived at the airport.	The captain and crews are grateful for the support.
Quake survivors moved into makeshift houses.	The room was filled with huge canvases.
The research team is going into the deep jungle.	The stories are narrated through dance.
...	...

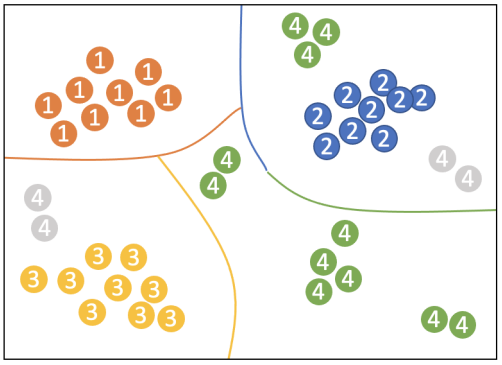
(A) Samples selected from the SemEval 2010 Task 8 dataset on relation extraction.



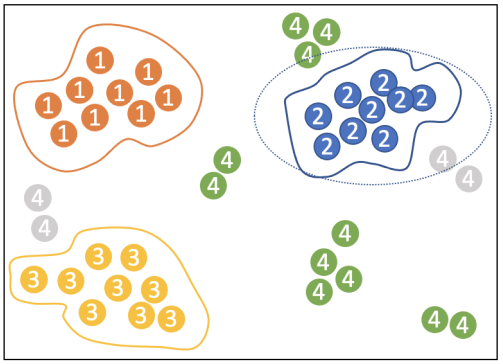
(B) Compactness comparison between the test data of the Universum class (green) and target classes (blue).



(C) The distribution of target classes (class 1, 2, 3) and the Universum class (class 4).



(D) The open decision boundaries obtained by traditional classifiers.



(E) The arbitrary closed boundaries obtained by our proposed method.

FIGURE 5.1: Illustration of distinction between the Universum class.

samples rather than recognizing the general predictive rule. Given the variations in data distributions between the test and training sets, only memorizing various peculiarities can easily lead to *overfitting*, causing a decline in accuracy. Furthermore, this inability to discern the genuine predictive rule for the Universum class can also compromise the model's robustness.

(2) **Lack of Representativeness in Training Data**: The Universum class is the complementary set of predefined target classes in the task. Therefore, it contains all possible implicit classes, i.e., classes not explicitly defined in the task but may appear in the real world. In this case, Universum samples in the training data are unable to sufficiently

represent all possible patterns of the genuine distribution of the Universum class. As depicted in Figure 5.1c, gray samples represent Universum samples in the test set that are not represented by the training data. Classifiers with open boundaries are prone to misclassifying unseen samples in the test set that is not represented by the training data.

Additionally, we provide a quantitative comparison of the average compactness between the Universum class and the target classes within the test data for the NER task [12], as depicted in Figure 5.1b. Notably, even though the Universum class is the class with the most samples, it exhibits significantly poorer compactness in its learned representations. This empirical observation aligns with our earlier theoretical analysis. Both the inability to discern the genuine *inter-class* predictive rule of the Universum class and the lack of representativeness in the training data contribute to this compromised compactness for the Universum class. Experiment details are in Section 5.4.8.

Despite the substantial difference between the target classes and the Universum class, this issue has long been neglected by the research community. The majority of works [12, 13, 57, 59, 215] treat the Universum class and target classes equally. Typically, a linear layer and a softmax function are applied at the end of the model to generate open decision boundaries, which we believe are inappropriate for tasks containing the Universum class.

How can we account for the distinct properties of the Universum class and target classes to derive better representations and classifiers? We think the key lies in **conforming to the inherent properties of the Universum class**. In this work, we propose a closed boundary learning method for classification-based tasks with the Universum class. Traditional methods often employ open boundary classifiers and constrain the representations of Universum samples to be distributed into a compact cluster. However, the open decision boundaries can easily misclassify Universum samples, as illustrated in Figure 5.1d. In addition, the restriction on compact space violates the inherent inter-class pattern of the Universum class. Therefore, we propose to use closed boundary classifiers as shown in Figure 5.1e. We constrain the space of target classes to be closed spaces and designate the area outside all closed boundaries in the feature space as the space of the Universum class. The treatment perfectly fits the nature of the Universum class: a sample is marked as the Universum if it does not belong to any target class during labeling.

The main contributions of this work are summarized as follows:

- We address an understudied problem in this paper. The Universum class widely exists in many NLP tasks and general machine learning tasks, but hasn't received significant attention in these contexts.
- Methodologically, we generate closed boundaries with arbitrary shape, propose the inter-class rule-based probability estimation for the Universum class to cater to the inherent properties of the Universum class, and propose a boundary learning loss to learn the decision boundary based on the balance of misclassified samples inside and outside the boundary.
- In adherence to the natural properties of the Universum class, our method improves both accuracy and robustness of classification models, which is validated on six state-of-the-art (SOTA) works across three different tasks.

5.2 Related Works

5.2.1 Classification Tasks with the Universum Class

The Universum class widely exists in classification based tasks in NLP, such as relation extraction (RE) [54], named entity recognition (NER) [55], and aspect category sentiment analysis (ACSA) [66], as summarized in Table 5.1. It should be noted that the span-based methods [57, 58] enumerate all possible spans for classification, which introduces an extra *other* class. Despite the heterogeneity and lack of representativeness of the Universum class, current works [12, 13, 57, 59–63] solve the classification problems containing the Universum class as normal multi-class classification problems and treat the Universum class and target classes equally.

5.2.2 Closed Boundary Learning Methods

Closed boundaries are often adopted in research fields of out-of-distribution (OOD) detection [72–74], open set recognition [75, 76], anomaly detection [77], and outlier detection [78, 79]. We borrow the term "generalized OOD detection" from [80] to encapsulate these problems and discern their differences from our proposed classification with the Universum class problem.

Task	Dataset	Label Name	Proportion
RE	SemEval 2010 Task 8 [65]	Other	17.4%
RE	TARCED [54]	No relation	79.5%
NER	CoNLL-2003 [55]	Miscellaneous	14.6%
NER (span based method)	CoNLL-2003 [55]	Other	>90%
ACSA	MAMS [66]	Neutral	43.4%

TABLE 5.1: The tasks and datasets that the Universum class exists.

5.2.2.1 Difference in Problem Setting

Classification tasks can be categorized into problems based on closed-world assumption and open-world assumption [80]. The generalized OOD detection is treated under the open-world assumption, while the classification problem with the Universum class is treated under the closed-world assumption. In addition, the OOD samples are not available in the training data in generalized OOD detection, whereas a considerable number of Universum samples are included in the training data in our problem setting. The information of existing Universum samples is important to generate accurate decision boundaries in our problem.

5.2.2.2 Difference in Methodology

By definition, the OOD detection problem assumes that the training data do not contain any OOD samples. However, a branch of the OOD studies, known as outlier exposure [96–101], introduces auxiliary outlier data during training. The introduced auxiliary data makes it close to the format of our raised classification problems with the Universum class. However, outlier exposure methods are not suitable for our problem. The outlier exposure method mostly adopts a two-step approach that consists of multi-class classification and OOD identification. Such two-step approach will suffer from error propagation problem. In addition,the OOD identification step distinguishes OOD and ID samples based on a score obtained by cross entropy or energy function. However, both cross entropy and energy function are monotonically varying. As a result, the decision boundary derived from a threshold score of the monotonically varying function is an open boundary, which leaves the heterogeneity and representativeness issues we pointed out in this paper still unresolved.

From a methodological point of view, our work is also different from the works in generalized OOD using closed boundaries. In generalized OOD studies, the closed boundaries are formulated by the classic density-based method [102, 103], one-class classification method [104, 105], or distance-based method [72, 75, 106–108]. The distance-based methods are limited to spherical boundary shapes but our method can generate arbitrary shape boundaries. The one-class classification method formulates only one closed boundary between positive and negative samples while our work generates closed boundaries for all target classes. Finally, only positive samples are used to learn decision boundaries in density-based method, while both target class samples and Universum samples are used in our work.

5.2.3 Universum Learning Methods

Although we adopt the terminology of Universum [53], the problem setting of our work is entirely different from that of previous studies on Universum learning [53, 67–69]. The idea of Universum learning studies is to exploit external, unlabeled Universum data to improve the accuracy of supervised tasks. However, in our problem, the Universum class is one of the internal, labeled classes of multi-class classification problems and we propose closed boundary learning to conform to its unique properties during learning. Furthermore, methodologically, the Universum learning method either employs open-boundary classifiers [69] or is incapable of distinguishing the Universum samples from labeled samples [53, 67, 68], neither of which are appropriate to the problem we presented.

5.3 Method

Problem Definition: The goal of our proposed method is to learn closed decision boundaries for target classes and meanwhile, jointly classify the Universum samples and target samples. We give a detailed description of how to recognize the Universum class in Section 5.3.1. In order to make our proposed method compatible with most existing classification methods, the starting point of our method is the representations of the final layer of classification models, which is a linear layer that maps data from high-dimensional feature space to a lower-dimension space. We denote the sample

Method	Pretrained Model	Reported F1/accuracy
SpanNER [12]	BERT-base	92.28
BS [57]	RoBERTa	93.65
A-GCN [60]	BERT-base	89.16
TaMM [61]	BERT-base	89.18
AC-MIMLLN [62]	Glove	76.42
SCAPT [13]	BERT-base	85.24

TABLE 5.2: The pretrained models chosen for each baseline model and the corresponding F1 score/accuracy reported in the original paper.

representations of the final linear layer as $\mathrm{H} = \{\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{N-1}\} \in \mathbb{R}^{N \times l}$, where N is the number of samples, and l is the output dimension of the linear layer.

5.3.1 Defining the Universum Class

Universum class exists in many tasks and datasets as we summarized in Table 5.1. Notably, the Universum class has various names such as *other* and *miscellaneous*, etc. In sentiment analysis, the *neutral* class can be considered as the Universum class because the word *neutral* is defined as "having no strongly marked or positive characteristics or features", which means the *neutral* class is a collection of all samples without strong emotions. Similarly, the *no relation* class in the relation extraction task can be considered as the Universum class.

5.3.2 Pretraining

Our method estimates the probability distribution of target classes based on their sample distributions. In order to avoid estimation based on randomly initialized weight and speed up the learning process, we employ N-pair loss [216] for pretraining, making sample representations be of small intra-class distance and large inter-class distance. Notably, in alignment with the inherent nature of the Universum class—not belonging to any specific classes of interest—and to help avoid the issue of overfitting, we make a change that does not require the model to reduce the intra-class distance of Universum samples during the pretraining.

5.3.3 Generating Closed Boundary of Arbitrary Shape for Target Classes

Existing closed boundary classification methods mainly use spherical shape boundaries [75, 76]; however, we argue that the spherical shape may not be the optimal solution because data samples are unlikely to perfectly fit into a sphere, and a spherical shape boundary may produce misclassifications. We adopt the Gaussian mixture model (GMM) and the threshold value to generate boundaries with arbitrary shapes.

5.3.3.1 Gaussian Mixture Model

We apply GMM with m components to estimate the class conditional probability distribution for each target class C_i, and further derive the joint probability estimation for each class.

$$p(\mathbf{h}_k \mid C_i) = \sum_{i=1}^{m} \pi_i \mathcal{N}\left(\mathbf{h}_k; \mu_i, \mathbf{\Sigma}_i\right) \tag{5.1}$$

$$p(\mathbf{h}_k, C_i) = p(\mathbf{h}_k \mid C_i) p(C_i) \tag{5.2}$$

where $\mathbf{h}_k$ denotes the input feature vector of the kth sample, μ_i and $\mathbf{\Sigma}_i$ are the estimated mean vector and covariance matrix of the ith Gaussian components, respectively. π_{ij} is the non-negative mixture weight under the constraint that $\sum_{j=1}^{m} \pi_{ij} = 1$. μ_i, $\mathbf{\Sigma}_i$, and π_{ij} are all learnable parameters in the model.
According to Bayes Theorem, the posterior probability $p(C_i \mid \mathbf{h}_k) = \frac{p(\mathbf{h}_k|C_i)p(C_i)}{p(\mathbf{h}_k)}$. Since we are interested in $\operatorname{argmax}_{C_i} \frac{p(\mathbf{h}_k|C_i)p(C_i)}{p(\mathbf{h}_k)}$, the decision can be made based on joint probability $p(\mathbf{h}_k, C_i)$.

5.3.3.2 Arbitrary Shape Boundary

Geometrical View: Inspired by the DENCLUE algorithm in generating arbitrary shape clusters [217], we introduce a threshold value ξ_i for each target class. A closed boundary of arbitrary shape is formulated by points satisfying $p(\mathbf{h}, C_i) = \xi_i$. Figure 5.2 is an illustration of formulating an arbitrary shape boundary in a two-dimensional space. A sample is assigned to class C_i if it is located inside the closed boundary. If the number of components of the GMM is set to one, then the shape of the boundary becomes spherical,

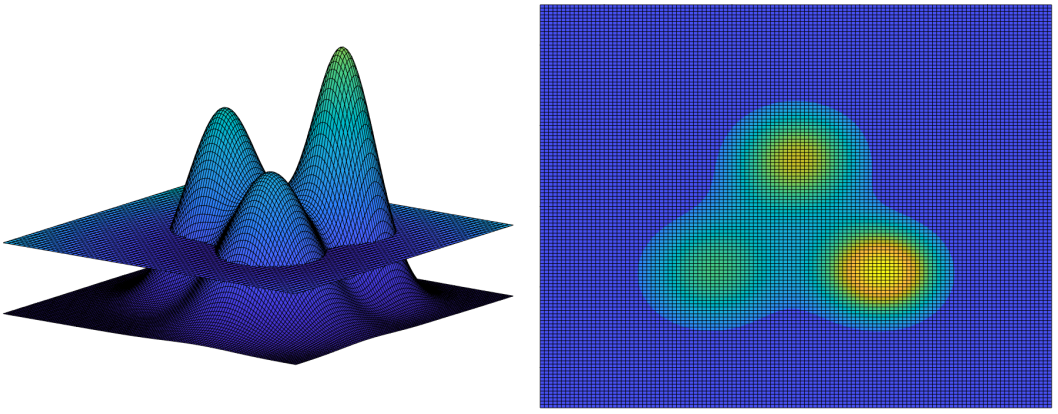


FIGURE 5.2: Illustration of generating arbitrary shape boundaries.

with its center and covariance matrix specified by μ_0 and $\mathbf{\Sigma}_0$, respectively. In this sense, the commonly used spherical shape boundary [75, 76] is a special case of our method. Notably, the threshold values $\Xi = \xi_1, \xi_2, \cdots, \xi_{n-1}$ are a learnable parameters, which eliminates the laborious process of hyperparameter tunning. Specifically, they are learned based on the balance of misclassified samples inside and outside the boundary through our proposed boundary learning loss, which is introduced later.

Probabilistic View: The above geometrical process can be described as:

$$\begin{cases} \mathbf{h}_k \in C_i \text{ if } p(\mathbf{h}_k, C_i) > \xi_i \\ \mathbf{h}_k \notin C_i \text{ if } p(\mathbf{h}_k, C_i) \leq \xi_i \end{cases} \tag{5.3}$$

5.3.4 Inter-Class Rule-Based Probability Estimation for the Universum Class

The main obstacle to properly addressing the issue of the Universum class is to estimate the probability of the Universum class based on its inherent *inter-class* property rather than *intra-class* sample distributions. We propose an inter-class rule-based probability estimation method to address this issue.

5.3.4.1 Motivation and the Estimation

We classify samples of Universum class and target classes based on the following rules we defined:

- *Rule 1*: A sample is assigned to the Universum class if it is not located inside any of the closed boundaries of target classes.
- *Rule 2*: A sample is assigned to the target class with the highest $p(\mathbf{h_k}, C_i)$ if it is located inside at least one closed boundary.

An intuitive way to incorporate the above rules is a two-step method that consists of Universum class detection and target class classification. However, such a pipeline method has the issue of error propagation. In addition, general probability estimation methods exploit intra-class sample distributions, which fail to overcome the natural inter-class property of the Universum class and do not conform to Rule 1. Therefore, a strategy need to be devised to convert Rule 1 and 2 into a probability expression, while simultaneously facilitates the learning of the neural network.
For compliance with Rule 1, the estimated probability of the Universum class must satisfy the following two conditions: for Universum class samples: $\forall i : p(\mathbf{h}_k, U) > p(\mathbf{h}_k, C_i)$ and for target class samples: $\exists i : p(\mathbf{h}_k, U) < p(\mathbf{h}_k, C_i)$. We can leverage the relationship between ξ_i and $p(\mathbf{h}_k, C_i)$ defined in Equation 5.3 to construct the estimation of $p(\mathbf{h}_k, U)$ that satisfies the above two conditions. In addition, to enhance the learning of neural networks, the gradient obtained from an Universum sample should move this sample away from its closest target class boundary. Therefore, we also involve $\max(p(\mathbf{h}_k, C_i))$, the probability of the closest target class of a Universum sample, to guide the Universum sample move away from target class boundaries. We propose to estimate the probability distribution of the Universum class as follows:

$$p(\mathbf{h}_k, U) = \lambda \frac{\xi_u^2}{p(\mathbf{h}_k, C_u)} + (1 - \lambda) \frac{\xi_v^2}{p(\mathbf{h}_k, C_v)} \tag{5.4}$$

$$\text{where} \quad \lambda = \begin{cases} 1, \; p(\mathbf{h}_k, C_u) > \xi_u \\ 0, \; p(\mathbf{h}_k, C_u) \leq \xi_u \end{cases} \tag{5.5}$$

$$\begin{cases} u = \text{argmax}_i \, \frac{p(\mathbf{h}_k, C_i)}{\xi_i}, \\ v = \text{argmax}_i \, p(\mathbf{h}_k, C_i) \end{cases} \tag{5.6}$$

ξ_i is the threshold value of target class i, and $u, v \in \{1, 2, \ldots, n-1\}$.

5.3.4.2 Analysis of the Proposed Estimation

For the estimated probability of the Universum class in Equation 5.4, two cases are possible.

Case 1: $p(\mathbf{h}_k, C_u) > \xi_u$, i.e., sample $\mathbf{h}_k$ is located inside at least one closed boundary. In this case, we have

$$p(\mathbf{h}_k, U) = \xi_u \frac{\xi_u}{p(\mathbf{h}_k, C_u)} < \xi_u < p(\mathbf{h}_k, C_u)$$

Since $p(\mathbf{h}_k, U) < p(\mathbf{h}_k, C_u)$, the model will select the target class i with the highest $p(\mathbf{h}_k, C_i)$, which fits perfectly with Rule 2.

Case 2: $p(\mathbf{h}_k, C_u) \leq \xi_u$, i.e., the sample $\mathbf{h}_k$ distribute outside every closed boundary. Combining the condition of case 2 and Equation 5.6, we have

$$\forall i \in \{1, 2, \ldots, n-1\} : \frac{p(\mathbf{h}_k, C_i)}{\xi_i} \leq \frac{p(\mathbf{h}_k, C_u)}{\xi_u} \leq 1$$

$$\text{i.e.}, \forall i \in \{1, 2, \ldots, n-1\} : p(\mathbf{h}_k, C_i) \leq \xi_i \tag{5.7}$$

Combining Equation 5.6 and Equation 5.7, we can derive that $\forall i \in \{1, 2, \ldots, n-1\}$:

$$p(\mathbf{h}_k, U) = \xi_v \frac{\xi_v}{p(\mathbf{h}_k, C_v)} \geq \xi_v \geq p(\mathbf{h}_k, C_v) \geq p(\mathbf{h}_k, C_i)$$

In case 2, from Equation 5.7 and Equation 5.3, we can learn that sample $\mathbf{h}_k$ is located outside all closed boundaries of target classes. In this case, the probability of Universum class $p(\mathbf{h}_k, U)$ obtains the largest value. Therefore, Rule 1 is perfectly expressed by the proposed probability estimation of the Universum class.

5.3.5 Boundary Learning Loss

To facilitate the learning of the closed decision boundaries, we propose a boundary learning loss below. Our intuition is that the decision boundary should be adjusted to the balance of misclassified samples inside and outside the boundary. For example, if samples of class j distribute inside the boundary of class i, then the boundary should

contract to exclude such samples and vice versa.

$$\mathrm{L_{bl}} = \frac{1}{M}\sum_{i=1}^{n-1}(\sum_{k\in\mathbb{O}} w_k \log\frac{\xi_i}{p(\mathbf{h}_k, C_i)} + \sum_{l\in\mathbb{I}} w_l \log\frac{p(\mathbf{h}_l, C_i)}{\xi_i})$$

M is the total number of misclassified samples for all boundaries, n is the number of classes, $\mathbb{O}$ and $\mathbb{I}$ denote the set of training samples misclassified outside and inside the decision boundary i, respectively. The weights in the loss function are $w_k = \frac{p(\mathbf{h}_k, C_i)}{p(\mathbf{h}_k, C_i)+\xi_i}$, $w_l = \frac{\xi_i}{p(\mathbf{h}_l, C_i)+\xi_i}$, and they are detached and cut off the gradient. Weights w_k and w_l have smaller values for samples located far from the boundary, enabling the boundary to be adjusted primarily on the basis of easily and semi-hard negatives instead of hard negatives.

During training, we sum the cross-entropy loss and boundary learning loss for optimization. In addition to balancing inside and outside misclassified samples, the boundary learning loss forces misclassified samples to be distributed in the proper region, which works well with cross-entropy loss.

5.3.6 Framework Overview

To provide better clarity of our method, we provide a succinct breakdown of the closed boundary learning framework's workings:

- **Initialization Post-Pretraining**: After pre-training, we employ the GMM for each target class. The Expectation-Maximization (EM) algorithm is employed to set the initial values for the GMM parameters μ_i, $\mathbf{\Sigma}_i$, and π_{ij}. As we transition to the training phase, the parameters of GMM are treated as learnable parameters. Contrary to traditional methods using the EM algorithm for continuous updates, these parameters are dynamically updated by the neural network throughout the training process.

- **Probability Estimation**: Probability distributions of target classes are estimated using GMM, as articulated in Equation 2. The probability distribution of the Universum class is computed through our Inter-Class Rule-Based Probability Estimation method, which is represented in Equation 4.

- **Training Optimization**: During the training process, we use a combined loss function, summing the cross-entropy loss with the boundary learning loss for optimization.

5.4 Experiments

5.4.1 Experimental Methodology

We demonstrate the efficacy of our method on six different SOTA models on three datasets of different NLP tasks, including SemEval 2010 Task 8 [65], MAMS [66], and CoNLL-2003 [55]. The proportion of Universum samples in the SemEval 2010 Task 8, MAMS, and CoNLL-2003 datasets are 17.4% (highest in 19 classes), 90%, and 43.4%. respectively. It is noteworthy that the ratio of Universum class in the NER task is not calculated from the *miscellaneous* samples in the dataset but from the *other* samples which are introduced by the span-based method [12, 57].
We evaluate the effectiveness of our proposed **C**l**O**sed b**O**undary **L**earning for classiciation with the **U**niversum class (**COOLU**) on 6 SOTA works, including SpanNER [12], BS [57], A-GCN [60], TaMM [61], AC-MIMLLN [62], and SCAPT [13].

5.4.2 Implementation Details

5.4.2.1 Baseline Models

We reproduce the baseline models based on the officially released source code, and apply closed boundary learning on the source code. All reported results are the average of three runs. It should be noted that some results of baseline models are slightly different from those given in the original papers due to the variations in random seeds and package versions when reproducing baseline models from their officially released codes. Nevertheless, baseline models and models with closed boundary learning are fairly compared in our work under the same random seed and deep learning environment.
In the six baseline models we selected, different results based on multiple language models are often reported in one work. We choose one of the pretrained models used in each work and reproduce the baseline models. The pretrained language model we used

Method	Pretrained Model	Reported F1/accuracy
SpanNER [12]	BERT-base	92.28
BS [57]	RoBERTa	93.65
A-GCN [60]	BERT-base	89.16
TaMM [61]	BERT-base	89.18
AC-MIMLLN [62]	Glove	76.42
SCAPT [13]	BERT-base	85.24

TABLE 5.3: The pretrained models chosen for each baseline model and the corresponding F1 score/accuracy reported in the original paper.

in each baseline and their reported results are summarized in Table 5.3.

5.4.2.2 Training Process

During pretraining process, all parameters of the original model θ are learned. We employ GMM estimation on training data after pretraining and obtain the initial value of μ_i, $\mathbf{\Sigma}_i$, and π_{ij}, where $i \in \{1, 2, \cdots, n-1\}$, $j \in \{1, 2, \cdots, m\}$. n is the number of classes, and m is the number of GMM components. Through preliminary experiments, we observed that the number of GMM components has a minimal impact on our model's performance, as a component count of four is sufficient for accurate approximation of arbitrary decision boundaries. Thus, we typically select $m = 4$ in our experiments. The threshold values ξ_i is initialized around the a quantile of $p(\mathbf{h}_k, C_i)$ values ($k \in \{0, 1, \cdots, N_i - 1\}$), where a is the accuracy or F1 score of the original model. With our inter-class rule-based probability estimation for the Universum class, we obtain $[p(\mathbf{h}_k, C_1), \cdots, p(\mathbf{h}_k, C_{n-1}), p(\mathbf{h}_k, U)]$. Then, the original model parameters θ, GMM parameters μ_i, $\mathbf{\Sigma}_i$, π_{ij} and threshold values ξ_i are learned by cross-entropy loss and our proposed boundary learning loss.

We use NVIDIA RTX A5000 GPUs to run the experiments and the model parameters are mostly follow the original baseline models.

5.4.2.3 Robustness Evaluation

We evaluate the robustness of the model based on TextFlint [218], a robustness evaluation toolkit for NLP tasks. There are two kinds of transformations provided by TextFlint

to generate the robust evaluation dataset, namely universal transformation and task-specific transformation. We adopt two universal transformations and two task-specific transformations to the test set of NER and RE task and generate four robustness evaluation datasets for each task. The terms of different transformations are explained below.

- "SpellingError": Universal transformation. Brings slight errors to words in the test samples.
- "AppendIrr": Universal transformation. Add irrelevant information to test samples.
- "CrossCategory": Task-specific transformation for NER. Replace the entity spans with substitutions from a different category.
- "OOV": Task-specific transformation for NER. Replace the entity spans with substitutions out of vocabulary.
- "InsertClause": Task-specific transformation for RE. Change sample sentences by appending adjuncts from the aspect of dependency parsing.
- "SwapEnt": Task-specific transformation for RE. Swap the named entities in a sentence into entities of the same type.

Specifically, the models are trained and validated on the original training set and validation set, but the test set is transormed into the robustness evaluation dataset by the transformations proposed by TextFlint. Then, modelw are tested on the transformed test set.

5.4.3 Overall Experimental Results

Our first research question (RQ) is *can COOLU achieve a "free" accuracy gain on tasks with the Universum class?* **(RQ1)** Table 5.4 shows the overall results for all 6 models. The reported results are the average of three runs. Models with our proposed closed boundary learning outperform the original models with open classifiers on NLP tasks containing the Universum class. The overall accuracy or F1 score is improved on all six models we evaluated, with the largest improvement from 92.09 to 93.50 in F1 score. We also notice that the improvement on the RE task is not as significant as on the NER and the ACSA tasks (0.66 against 1.41 and 1.22). This may be due to the fact that Universum

Task	Method	F1/accuracy	p-value
NER	SpanNER [12]	92.09±0.16	
	SpanNER [12] + ADB	77.22 ± 0.49	< 0.001
	SpanNER [12] + OECC	91.22 ± 0.12	
	SpanNER [12] + COOLU	**93.50**±0.13	
	BS [57]	92.53±0.02	
	BS [57] + ADB	75.52 ± 0.55	< 0.01
	BS [57] + OECC	91.88 ± 0.15	
	BS [57] + COOLU	**93.17**±0.13	
RE	A-GCN [60]	88.67±0.18	
	A-GCN [60] + ADB	85.99 ± 0.23	< 0.01
	A-GCN [60] + OECC	88.00 ± 0.09	
	A-GCN [60] + COOLU	**89.33**±0.20	
	TaMM [61]	88.76±0.23	
	TaMM [61] + ADB	85.08 ± 0.26	< 0.01
	TaMM [61] + OECC	88.17 ± 0.18	
	TaMM [61] + COOLU	**89.47**±0.21	
ACSA	AC-MIMLLN [62]	76.13±0.29%	
	AC-MIMLLN [62] + ADB	71.78 ± 0.85%	< 0.01
	AC-MIMLLN [62] + OECC	74.02 ± 0.68%	
	AC-MIMLLNN [62] + COOLU	**77.35**±0.42%	
	SCAPT [13]	84.13±0.19%	
	SCAPT [13] + ADB	79.67 ± 0.44%	< 0.01
	SCAPT [13] + OECC	83.36 ± 0.27%	
	SCAPT [13] + COOLU	**85.06**±0.23%	

TABLE 5.4: The overall performance of applying closed boundary learning on baseline models.

Method	ORG			PER			LOC			MISC			Other		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
SpanNER	89.92	89.81	89.87	97.74	96.47	97.11	93.05	93.88	93.46	78.99	82.83	80.87	99.87	99.83	99.85
SpanNER + COOLU	**94.29**	90.30	**92.30**	**98.54**	96.29	**97.40**	**96.17**	93.41	**94.77**	**88.02**	81.97	**84.89**	99.84	**99.89**	**99.86**

TABLE 5.5: The micro F1 score of SpanNER [12] with and without closed boundary learning.

samples only account for 17.4% of the SemEval 2010 Task 8, which is considerably less than the other datasets. In addition, statistical tests between the accuracy/F1 score of the baseline models and our method indicate that the improvement brought about by our COOLU method is statistically significant. The above experimental results answer **RQ1** in positive.

5.4.4 A Closer Look at the MicroF1, Precision and Recall

Another question is *does COOLU enhance classification accuracy for all classes or just the Universum class?* **(RQ2)** We show the micro F1 score of each class in applying closed boundary learning on SpanNER [12] in Table 5.5. The micro F1 score for the Universum (*other*) class, introduced by the span-based method, is excluded from the overall F1 score calculation as per the task requirements. The micro F1 score is improved in all classes, with the absolute improvement of 0.01, 2.43, 0.29, 1.31, and 4.02, respectively. The F1 improvement of the Universum class is very small compared to target classes because its sample size is more than 100 times larger than other classes, making the denominator very large when calculating. Consequently, an improvement of 0.01% of Universum samples being correctly classified leads to a significant enhancement in the precision score for the entity classes. The results answer **RQ2** positively: the improvement of overall performance is not only attributed to the improvement of the Universum class, but also to the improvement of all classes as a whole.

The third research question is *how does COOLU improve classification model's performance?* **(RQ3)** We find that our proposed closed boundary learning significantly improves the precision score of all target classes, with the largest absolute gain being 9.03 in Table 5.5. By analyzing the change in precision and recall, we can derive the following findings: Firstly, the misclassification of Universum samples as target samples results in low precision scores for target classes and low recall score for the Universum class in the baseline method, which proves our claim that the Universum class is easily misclassified if its unique properties are neglected. In addition, our proposed closed boundary learning method can effectively prevent the misclassification of the Universum class into target classes, which significantly improve the precision score of target classes at the expense of a very slight reduction in recall and similarly improve the recall score of the Universum class at the expense of a slight reduction in precision. The second finding answers **RQ3**.

5.4.5 Model Robustness Evaluation

We are more interest in the research question that *In adherence to the natural properties of the Universum class, does COOLU provide a more reasonable way of learning by improving both model's accuracy and robustness?* **(RQ4)** We attempt to demonstrate

	CrossCategory	OOV	SpellingError	AppendIrr
SpanNER	77.06	75.14	76.09	87.34
SpanNER + COOLU	**81.39**	**83.15**	**81.12**	**89.89**
	InsertClause	SwapEnt	SpellingError	AppendIrr
A-GCN	77.84	86.8	77.36	86.59
A-GCN + COOLU	**80.17**	**88.69**	**78.71**	**88.22**

TABLE 5.6: Comparison of model's robustness with and without closed boundary learning.

RQ4 by theoretical analysis and experimental evaluation of the robustness of the model. Theoretically, since the natural predictive rule of the Universum class is an *inter-class* pattern that does not belong to any target classes, traditional models are more likely to fit the noise in the Universum class by memorizing various peculiarities of *intra-class* heterogeneous samples rather than finding the general predictive rule. However, our method can identify the inter-class predictive rule of the Universum class and hence classify out-of-distribution Universum samples more accurately. In addition, the closed decision boundaries we learned are analogous to model's knowledge boundary of each target class: the space inside the boundary represents what the model knows about a certain class, i.e., the recognized patterns, whereas the space outside the boundary represents what the model doesn't know about this class from training data. Such knowledge boundary can avoid the misclassification of unseen non-target class samples as target class. The above two mechanism would contribute to better robustness of the model.

We evaluate the robustness of the model based on TextFlint [218], a robustness evaluation toolkit for NLP tasks. Specifically, TextFlint generate perturbations of the test data, and the robustness of the model is evaluated using the transformed test dataset. The terms such as "CrossCategory", "OOV", and "SpellingError" in Table 5.6 are different ways of transforming the test data. Detail information of these transformation methods are illustrated in Section 5.4.2.3.

It can be learned from the Table 5.6 that the robustness of SpanNER improved significantly by applying our proposed COOLU method, with the improvement of absolute F1 score of 4.44, 8.01, 5.03, and 2.55, respectively. In addition, although the improvement of the F1 score in A-GCN is less than 1, the improvement in robustness of the model is considerably large. The absolute F1 score on robustness evaluation datasets are improved at 2.33, 1.89, 1.35, and 1.63, respectively. Considering lots of studies theoretically identify a trade-off between robustness and accuracy [219–221], the improvement of both model's accuracy and robustness provides positive evidence for **RQ4**: COOLU can

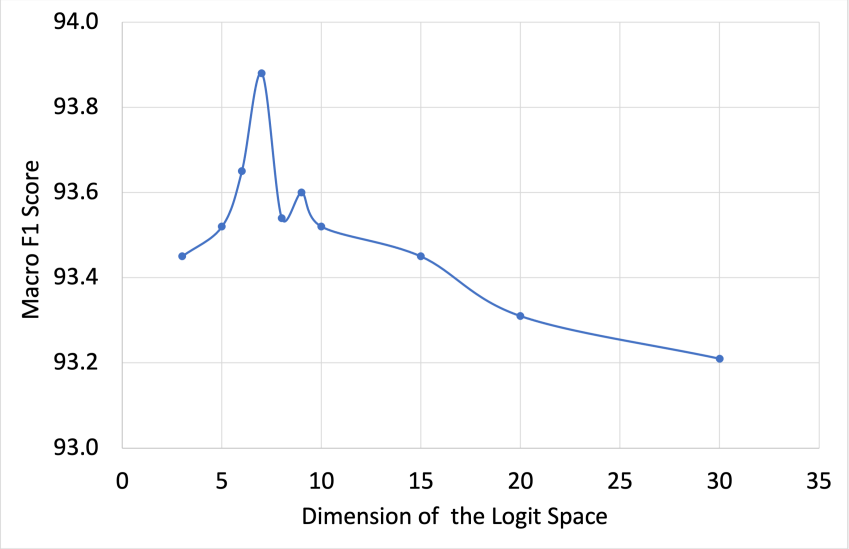


FIGURE 5.3: Impact of the last layer dimension on the accuracy of the model.

provide a more reasonable way of learning representations and classifiers.

5.4.6 The Impact of the Final Layer Dimension

The last layer's dimensionality can affect the performance of the model. Recalling the classic Hughes phenomenon [222] that the model accuracy is monotonically increasing first and then monotonically decreasing with the dimension of data, the dimension of the final layer may be chosen to boost model performance.
We investigate the effect of last layer dimension on the accuracy of the model on the SpanNER [12] with closed boundary learning and present the result in Figure 5.3. The F1 score of the test set grows with increasing of dimensions and reaches a maximum value of 93.88 when the dimension is seven, and then decreases with the dimension. The trend fits well with the Hughes phenomenon [222]. Our method is quite robust with the dimension and the overall result of SpanNER + COOLU reported in Table 5.4 is set as ten rather than the optimal value.

5.4.7 Ablations

We evaluated two recent OOD detection methods, ADB [75] and OECC [223], to demonstrate the inappropriateness of OOD detection approaches for our proposed problem. The experimental results reveal that integrating ADB and OECC with classification

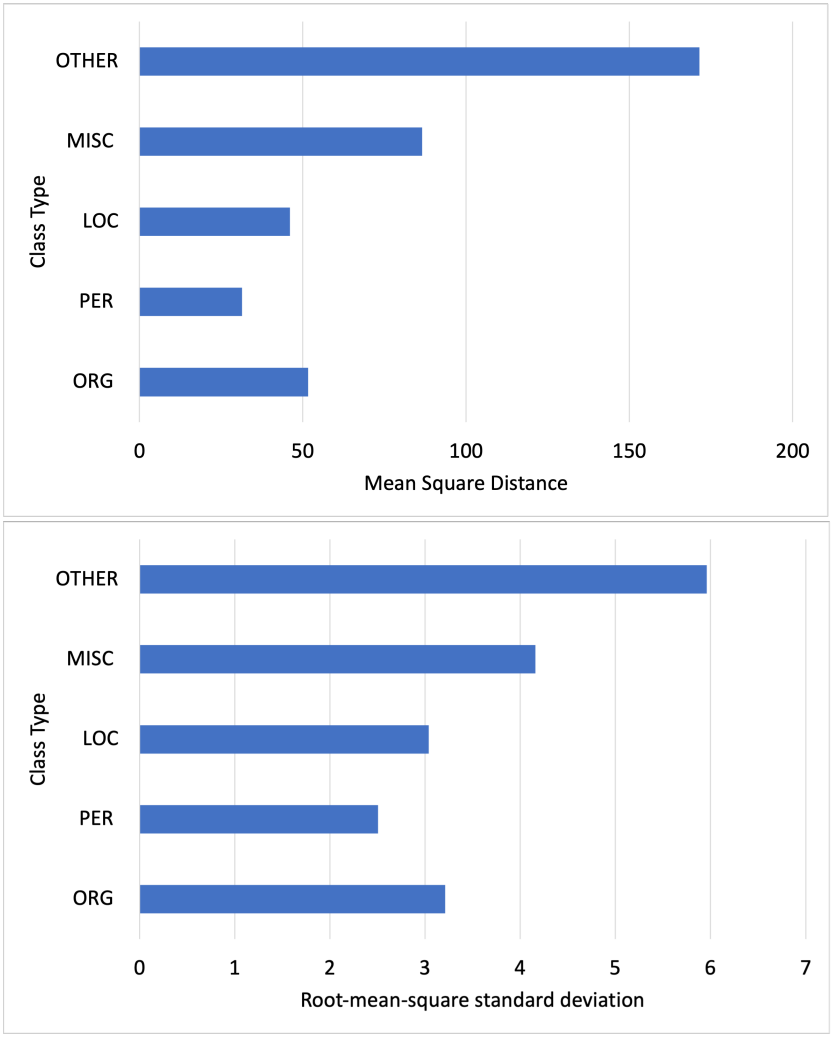


FIGURE 5.4: The compactness evaluation of the Universum class and target classes of the test data of NER task.

models considerably weakens the performance of the original models, as illustrated in Table 5.4. Given that both ADB and OECC were primarily designed for OOD detection and not the unique problem presented in this work, their unsuitability is anticipated. Specifically, due to the inherent difference in the problem settings, the ADB method cannot utilize the label information of the Universum class, leading to a significant decline in accuracy, especially when the Universum samples constitute a large portion in the span-based method. In addition, while OECC's problem setting, outlier exposure, aligns more closely with ours compared to other general OOD detection methods, it still shows inappropriateness, diminishing the accuracy of original classification models. This diminished accuracy can be attributed to: (1) error propagation in its two-step approach, and (2) the instability that arises from manually set thresholds.

5.4.8 Compactness of the Universum Class of the Test Set

We evaluate the compactness of the Universum class and target classes on the test data and depict the result in Figure 5.4. The representation of test samples after learning with open boundary classifiers by SpanNER [12] is used for evaluation. We evaluate the compactness based on the root-mean-square standard deviation (RMSSD) [224], and the mean square distance (MSD) [225], which are commonly used in clustering studies to evaluate compactness of a cluster. The smaller the RMSSD or MSD, the better the compactness. It is illustrated in Figure 5.4 that the compactness of the "OTHER" class is significantly worse than target classes. Notably, the class with the second-worse compactness is the "MISC" class, i.e., the "miscellaneous" class, which is also a type of the Universum class.

Given that the Universum class has the highest number of samples in the datasets we examined, it should be represented best through training and hence formulate the most compact cluster in the representation space for the test set. Yet, our empirical findings paint a different picture. Then, an interesting question was raised: **Why does the Universum class, despite being the largest, exhibit the worst compactness in its learned representations?**

Our research indicates that the answer is rooted in its **inherent inter-class pattern**. The Universum class is defined as a cluster of samples that do not belong to any of the predefined target classes. Considering human annotation practices, an entity is labeled as *Location* when it aligns with established patterns of *Location* entities. In contrast, a sample is labeled as *Others* not due to intra-class patterns specific to the *Others* class, but because it fails to conform to the patterns of *Location*, *Person*, or *Organization*. Consequently, current classification models are designed to recognize intra-class patterns and unable to discern the inherent inter-class predictive rule of the Universum class, which will result in the poorer compactness of the Universum class.

Additionally, from the lens of representation learning, the Universum class essentially encapsulates "everything else" in a given task. Thus, no matter the volume of this class, it's implausible to capture every nuance and pattern inherent to such a broadly defined class. This insight also sheds light on why, despite its significant size, the Universum class demonstrates the least compactness in its representations.

We also conduct an ablation study on N-pair loss pretraining. In our method, N-pair loss is adopted for pretraining to learn initial representations and to speed up the training

Method	F1/accuracy
SpanNER	92.09
SpanNER + N-pair pretraining	92.05
SpanNER + COOLU (N-pair pretraining)	**93.50**
SCAPT	84.13%
SCAPT + N-pair pretraining	84.16%
SCAPT + COOLU (N-pair pretraining)	**85.06%**

TABLE 5.7: The effect of pretraining on SpanNER [12] and SCAPT [13].

process. To demonstrate the effectiveness of our closed boundary learning method and to rule out the possibility that the improvement of our model is due to the N-pair loss pretraining process, we add an additional pretraining step to baseline models of SpanNER [12] and SCAPT [13]. Table 5.7 indicates that the pretraining process alone cannot improve the accuracy of the original baseline models. The improvement brought about by our proposed closed boundary learning is not a result of the pretraining step but the result of the entire system.

5.5 Conclusion

In this work, we highlight an understudied problem in classification-based tasks that the Universum class is treated equally with target classes despite their significant differences. As a solution, we propose a closed boundary learning method COOLU, which conforms the natural properties of the Universum samples. Specifically, we generate closed boundaries with arbitrary shapes, develop an inter-class rule-based strategy to estimate the probability of the Universum class, and propose a boundary learning loss to adjust decision boundaries. COOLU offers easy integration with most classification model, given that it operates on representations of the final layer of classification models. Our method not only boosts the accuracy of SOTA models but also significantly enhances their robustness.

5.6 Limitations

As a limitation, our method is not suitable for zero-shot or few-shot settings because the accuracy of GMM estimation is positively related to the number of samples used

[226]. Similarly, due to the inherent low-dimension constraints of the GMM, an extensive increase in the number of classes could pose challenges to the efficacy of our framework. Nevertheless, given that the last layer dimension, i.e. class number, in our method is usually small and the initialized GMM parameters will be fine-tuned by the neural network, most classification tasks are not limited by these constraints. Since the Universum class widely exists in NLP tasks and many general ML tasks, our method is applicable to most of these tasks.

Chapter 6

UniBias: Unveiling and Mitigating LLM Bias through Internal Attention and FFN Manipulation

6.1 Introduction

When applying LLMs to EE tasks in Chapter 4, we observe that LLMs are biased towards predicting certain answers. Consequently, we investigate this biased behavior and find that this inherent bias of LLMs undermines the effectiveness of them across diverse applications, leading to the issue of prompt brittleness—sensitivity to design settings such as example selection, order, and prompt formatting. Therefore, in this chapter, we explore the internal mechanisms of LLM bias and introduce a novel approach to mitigate LLM bias through manipulation of LLM inner components.

LLMs have shown exceptional capabilities in various natural language processing (NLP) tasks, employing the in-context learning (ICL) paradigm. This paradigm conditions LLMs on a context prompt comprising of a few example-label pairs [49, 227].

Despite their impressive performance, LLMs are prone to prompt brittleness, characterized by high sensitivity to the choice [6] and order [135] of examples, and prompt formatting [125], as demonstrated in Figure 6.1. Such prompt brittleness is found to arise from the bias in LLMs towards predicting certain answers [6]. The presence of the bias undermines the robustness and adaptability of LLMs in diverse applications.

Extensive research has focused on identifying factors that lead to LLM bias and strategies for mitigation. For instance, vanilla label bias [136] and recency bias [6] demonstrate the LLM's inherent non-contextual preference for certain labels and contextual preference for specific positions, respectively. Additionally, several calibration methods [6, 136, 139] are proposed to counteract the bias by adjusting decision boundaries of model output probabilities. However, these approaches are derived from *external* observations or adjustments of LLM outputs, leaving **the *internal* mechanisms within LLMs that cause such bias poorly understood**.

In this work, we investigate the internal mechanism of LLM bias, specifically how feedforward neural networks (FFNs) and attention heads contribute to such bias. Building on findings in mechanistic interpretability [141, 144], we assess the contribution of individual attention heads and FFN vectors[1] to label predictions in LLMs. By identifying FFN vectors and attention heads that convey biased influences towards label prediction, we reveal the internal mechanisms behind several key bias factors, including vanilla label bias [136], recency bias [6], and selection bias [137]. For instance, our analysis of FFN vectors without input context demonstrates that their cumulative impact biases the LLM towards specific labels, indicating a non-contextual preference for certain labels, i.e., vanilla label bias. We elaborate on the background of mechanistic interpretability in Section 6.2.1 and present our findings on the internal mechanisms of LLM biases in next section.

We pose the question: Can we identify the biased components of LLMs and mitigate their detrimental impact on label prediction? Motivated by this intuition, we propose **UniBias**, an inference-only method designed to identify and eliminate biased FFN vectors and attention heads in LLMs. Specifically, we begin by projecting each FFN vector and attention head into the vocabulary space to interpret the information conveyed by their outputs. We then detect biased components based on three criteria we defined: the relatedness criterion, the bias criterion, and the low variance criterion. After identification, we mitigate their impact by masking these biased components. Extensive experimental results demonstrate that LLMs, from which biased components have been removed, consistently outperform their original counterparts by a significant margin. Further, as illustrated in Figure 6.1, our method significantly improves both the performance and robustness of ICL with perturbations of various design settings.

[1]FFN vector refers to the value vector in the second weight matrix of the FFN layer. We elaborate on this in Section 6.2.1

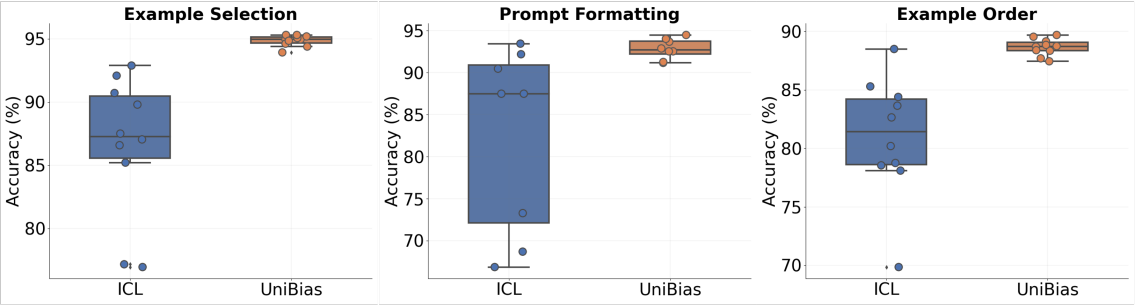


FIGURE 6.1: illustrates the prompt brittleness of ICL and the effectiveness of our method in mitigating this issue. Experiments are conducted in one-shot setting, using SST2 [9] dataset for experiments on example selection and prompt formatting and AGnews [10] dataset for example order experiment due to more diverse combination of orders.

The contributions of our work are summarized as follows:

- In contrast to existing works based on external observation of LLM outputs, we unveil the internal mechanisms within LLMs that lead to their bias towards predicting certain answers.
- We propose the UniBias method to mitigate LLM bias by identifying and eliminating biased FFN vectors and attention heads within LLMs. Moreover, our method demonstrate an effective way to manipulate internal structures of LLMs.
- Extensive experiments across 12 NLP datasets demonstrate that, by removing the biased components, our UniBias method significantly enhances ICL performance compared to the original LLM, achieving state-of-the-art results.

6.2 Internal Mechanisms Causing the Bias of LLMs

This section reveals the internal mechanisms within LLMs that lead to various bias factors.

6.2.1 Background

The theoretical background of this work is based on research on mechanistic interpretability [141–143], which aims to explain the internal processes in language models (LMs),

facilitating the interpretation of the contributions of individual model components to the final prediction.

We are focusing on decoder-only LMs in this paper. They are composed by a sequence of transformer layers, each composed of a multi-head self-attention layer and an feedforward neural network layer. The background knowledge for interpreting the contribution of each FFN vector and attention head to the models' prediction are demonstrated as follows.

The Residual Stream We interpret Transformers following the view of residual stream [141, 144]. Due to the residdual connection of Transformers, each layer takes a hidden state as input, and adds information obtained by its attention layer and FFN layer to the hidden state through residual connection. In this sense, the hidden state is a residual stream passed along layers, and each attention layer and FFN layer contribute to the final prediction by adding information to the residual stream.

Attention Heads Following Elhage et al. [141], Dar et al. [144], the output of each attention layer of LM can be computed as the sum of all its attention heads. Specifically, for l-th layer, the input is $X^l \in \mathbb{R}^{N\times d}$, and the attention layer is parameterized by four matrices $W_Q^l, W_K^l, W_V^l, W_O^l \in \mathbb{R}^{d\times d}$. The columns of each projection matrix and the rows of the output matrix can be split into H parts: $W_Q^{\ell,j}, W_K^{\ell,j}, W_V^{\ell,j} \in \mathbb{R}^{d\times \frac{d}{H}}$ and $W_O^{\ell,j} \in \mathbb{R}^{\frac{d}{H}\times d}$, where H is the number of attention heads. We then find that:

$$\begin{aligned}\text{Att}^\ell(X^\ell) &= \textbf{Concat}\left[A^{\ell,1}X^\ell W_V^{\ell,1}, A^{\ell,2}X^\ell W_V^{\ell,2}, \ldots, A^{\ell,H}X^\ell W_V^{\ell,H}\right] W_O^\ell \\ &= \sum_{j=1}^{H} A^{\ell,j}(X^\ell W_V^{\ell,j})W_O^{\ell,j}\end{aligned}$$

where $A^{\ell,j} = \text{softmax}\left(\frac{(X^\ell W_Q^{\ell,j})(X^\ell W_K^{\ell,j})^T}{\sqrt{d/H}} + M^{\ell,j}\right)$, $M^{\ell,j}$ is the attention mask. Therefore, the output of an attention layer is equivalent to computing attention heads independently, multiplying each by its own output matrix, and adding them into the residual stream of the LM.

FFN In line with Geva et al. [143, 145], transformer FFN layers can be cast as linear combination of vectors. Specifically, for an input vector $\mathbf{x}^\ell \in \mathbb{R}^d$, FFN parameter matrices $\mathbf{K}^\ell, \mathbf{V}^\ell \in \mathbb{R}^{d_m\times d}$, the FFN output can be derived as:

$$\text{FFN}^\ell(\mathbf{x}^\ell) = f(\mathbf{x}^\ell \mathbf{K}^{\ell^T})\mathbf{V}^\ell = \sum_{i=1}^{d_m} f(\mathbf{x}^\ell \cdot \mathbf{k}_i^\ell)\mathbf{v}_i^\ell = \sum_{i=1}^{d_m} m_i^\ell \mathbf{v}_i^\ell$$

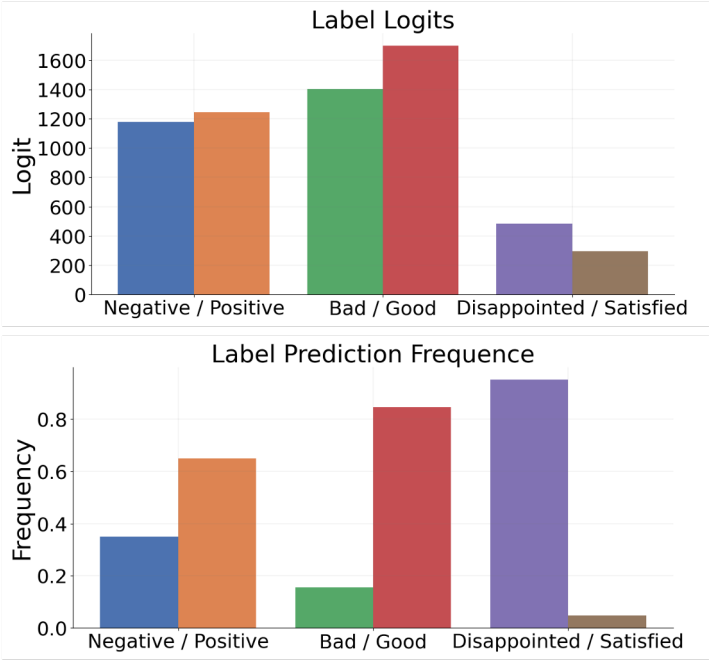


FIGURE 6.2: Unveiling vanilla label bias by uncontextual accumulated FFN logits.

where f is the activation function, i is the index of the vector. Then, the FFN layer can be viewed as a linear combination of vectors: the multiplication of $\mathbf{x}^\ell$ and the key vector $\mathbf{k}_i$ produces the coefficient m_i^ℓ that weights the corresponding value vector $\mathbf{v}_i$.

Logit Lens The logit lens [146] is a technique that directly decode hidden states into the vocabulary space using the unembedding matrix of the LLM for interpretation. This approach has been validated in various studies as an efficient method for interpreting the weight matrix or hidden states of LLMs [143, 144, 147, 148].

In summary, each attention layer and FFN layer contributes to the final prediction by adding their output hidden states to the residual stream. These outputs can be viewed as the sum of their respective attention heads and FFN vectors. Each attention head or FFN vector's output can be interpreted through the logit lens.

6.2.2 Internal Mechanisms of Bias Factors

We delve into the mechanisms behind several bias factors, analyzing the contributions of attention heads and FFN vectors to the biased predictions in LLMs. We explore vanilla label bias, position bias, and selection bias using the Llama-2 7B model [121].

Vanilla Label Bias The vanilla label bias [136], also known as common token bias [6], is the inherent, uncontextual preference of the model towards predicting certain label names. Given the contextual nature of attention layers, our investigation focuses on the FFN layers, where we identified a corresponding uncontextual preference. Specifically, by projecting the FFN value vectors into the vocabulary space, we compute the logits for various label names for each FFN vector. Utilizing the residual stream insight, we then aggregate these logits for all FFN vectors whose label logits rank within the top 10 over the vocabulary, reflecting uncontextual influences of FFN vectors that are effective in label prediction. This process yields what we term *uncontextual accumulated FFN logits*, revealing the intrinsic bias of the LLM towards predicting label names without the influence of input.

Figure 6.2 illustrates the accumulated uncontextual FFN logits across different label names in the sentiment analysis task, alongside their corresponding zero-shot prediction frequencies on the SST-2 dataset. For example, the label name 'positive' exhibits higher uncontextual accumulated FFN logits compared to 'negative,' leading to a higher frequency of 'positive' predictions. Additionally, when comparing the labels 'good' and 'bad', the difference in their uncontextual accumulated FFN logits is more pronounced than that between 'positive' and 'negative,' resulting in a larger discrepancy in prediction frequency. Conversely, the accumulated logits for the labels 'satisfied' and 'disappointed' show a reverse trend relative to 'positive' and 'negative', which results in a corresponding reverse trend in their prediction frequency ratios.

Recency Bias Recency bias refers to the tendency of LLMs to favor the label of the example at the end of the prompt [6]. By examining the behavior of attention heads within LLMs, we observe that specific heads consistently prioritize the example at the end of the prompt, providing an internal perspective on the origin of recency bias.
We identify the biased attention head using the method introduced in Section 6.3. We compare the behaviors of a biased attention head (layer 16, head 29) and an unbiased attention head (layer 16, head 19) in terms of the attention weight assigned to examples at different positions and the label logits of the corresponding attention head's output. Specifically, we use the SST-2 dataset, including one positive and one negative example in the prompt, and test with 40 samples, evenly split between positive and negative examples. More experimental details are provided in Section 6.4.1.3.

Experimental results in Figure 6.3 reveal that the biased attention head (layer 16, head

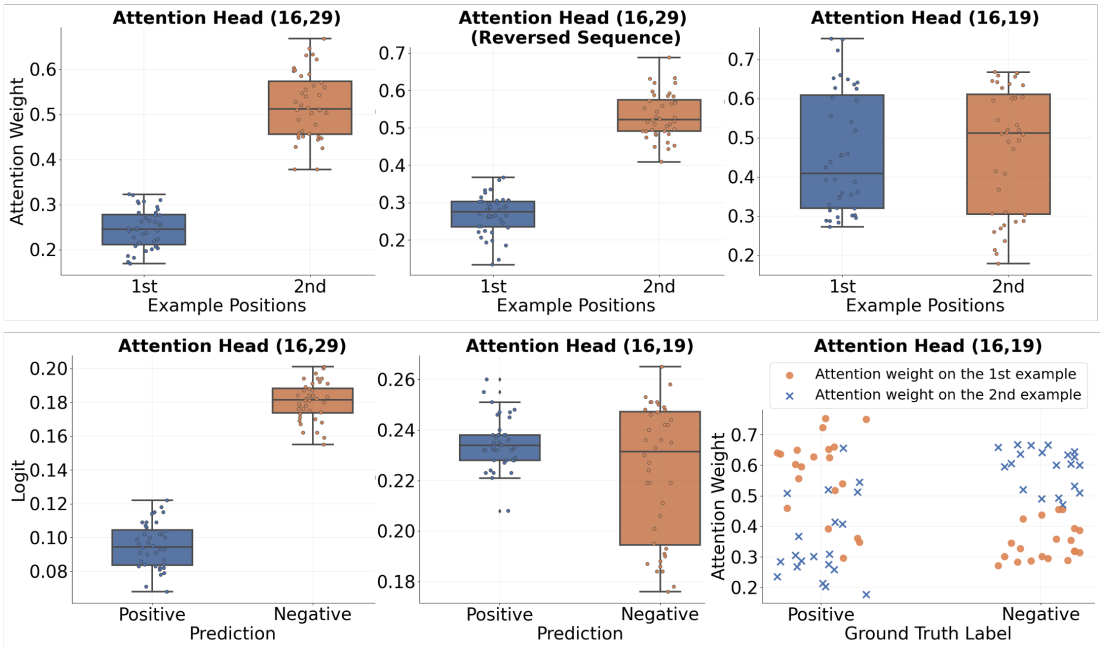


FIGURE 6.3: The internal mechanism of the recency bias.

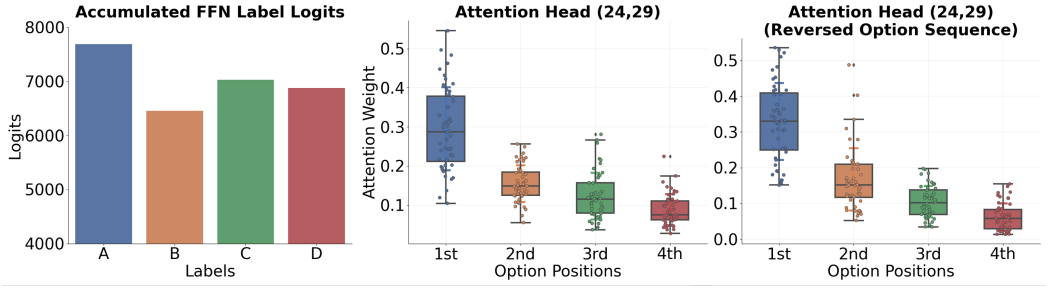


FIGURE 6.4: The internal mechanism of the selection bias.

29) consistently assigns significantly larger attention weights to the final example, irrespective of the ground truth labels of the test samples. This bias persists even when the sequence of examples is reversed, as shown in the second subfigure on the first row, indicating a biased preference of this attention head for the last example in the prompt. Furthermore, the biased attention weight assignment leads to biased logits, as shown in the third subfigure on the first row. In contrast, the unbiased attention head (layer 16, head 19) assigns very close averaged attention weights to both examples in the prompt. Interestingly, we observe that this unbiased head generally assigns larger weights to the example whose label matches the ground truth label of the test sample, resulting in 35 out of 40 samples being correctly classified based on this pattern by this single attention head. The preference shown by specific attention heads for the example at the end of the prompt reveals the internal mechanism of recency bias.

Selection Bias The selection bias refers that LLMs prefer to select specific option ID (like "Option A") as answers for multiple choice questions [137]. We have identified both FFN vectors and attention heads that consistently favor a specific option regardless of the ground truth label of the test sample, revealing the internal mechanism of selection bias.
We evaluate the Llama-2 7B model on the ARC dataset, which contains four options (A, B, C, D). We use a zero-shot setting to avoid the influence of position bias from multiple examples. More details are provided in Section 6.4.1.3.

Experimental results are illustrated in Figure 6.4. Firstly, we observe that the LLM exhibits a vanilla label bias favoring option "A", as shown in the first subfigure. Additionally, we identify a biased attention head that demonstrates a position bias consistently favoring the first option regardless of the ground truth labels of the test samples (second subfigure) or changes in the sequence of options (third subfigure). Since option A is usually the first option, these two biases both lead to the LLM's preference for option A.

6.3 Methodology

In the previous section, we unveil that various bias factors are stem from the biased behaviors of attention heads and FFN vectors. Naturally, we pose the question: Can we identify the biased components of LLMs and mitigate their impact on label prediction? Therefore, we propose our **UniBias** method to **Un**veil and m**i**tigate LLMs' label **Bias** through internal attention and FFN manipulation. Notably, our method is proposed for decoder-only LLMs.

6.3.1 Biased FFN Vectors Identification

Identifying biased FFN vectors in LLMs hinges on whether the contribution of each FFN vector is independent and interpretable. As discussed in Section 6.2.1, the output of an FFN layer can be cast as a linear combination of FFN vectors. Each FFN vector contributes to the final prediction by adding information encoded in its value vector, $\mathbf{v}_i^\ell$, weighted by its corresponding coefficient, m_i^ℓ. This information within $\mathbf{v}_i^\ell$ can be interpreted through the logit lens, enabling us to interpret it as a distribution of logits across the vocabulary space.

How to identify an FFN vector as biased? We assess whether it consistently introduces a biased preference towards specific labels into the residual stream, regardless of variations in the test samples. Such consistent biases can skew the LLM's predictions. We introduce the following criteria to detect biased components in LLMs, which are also applicable for identifying biased attention heads:

- **Relatedness Criterion**: The information introduced by the FFN vector (or attention head) should closely relate to label prediction.
- **Biased Criterion**: The information contributed to the residual stream by the FFN vector (or attention head) exhibits a biased distribution, favoring certain labels over others.
- **Low Variance Criterion**: The label prediction information added by the FFN vector (or attention head) to the residual stream is almost identical across a set of test samples with different labels, i.e., exhibits very small variance.

The third criterion is key to identifying biased FFN vectors (or attention heads), as consistently low variance indicates that the FFN vector is not adequately responsive to varying inputs. Combined with the second criterion, this suggests a bias towards certain predictions regardless of the input's contextual differences.

To examine these criteria, we interpret the information contributed by each FFN vector, i.e., $m\mathbf{v}$. For simplicity, we omit the layer number ℓ and FFN index i. Since the FFN value vector $\mathbf{v}$ is fixed, changes in the FFN coefficient m across different samples reflect the change in information brought by the FFN vector. We interpret this information by projecting each FFN value vector into the vocabulary space and analyzing the logit distribution over label tokens, termed *label logits*.

Specifically, given an FFN value vector $\mathbf{v} \in \mathbb{R}^d$, the unembedding matrix $E \in \mathbb{R}^{d \times d_e}$, a label token mapping matrix $L \in \mathbb{R}^{N \times d_e}$, where each row is a one-hot vector indicating the token id of the first token of each label name, the label logits $\mathbf{g^{(k)}} = [g_0^{(k)}, g_1^{(k)}, \ldots, g_{c-1}^{(k)}]^\top$ (where c is the class number) corresponding to the FFN value vector $\mathbf{v}$ of k-th sample can be obtained by:

$$\mathbf{g} = \mathbf{v} \cdot E \cdot L^\top$$

We use p unlabeled samples from the task to assess the three criteria we defined. The coefficients and label logits of an FFN vector for these samples are denoted as $\mathbf{m} =$

$[m_0, m_1, \ldots, m_{p-1}]$ and $\mathbf{G} = [\mathbf{g}^{(0)}, \mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(p-1)}]^\top \in \mathbb{R}^{p \times c}$, respectively. An FFN vector is considered biased if it meets the following conditions:

$$\begin{cases} \frac{1}{p}\sum_{k=0}^{p-1} \text{Sum}\left(\mathbf{G}_{k,:}\right) = \frac{1}{p}\sum_{k=0}^{p-1} \text{Sum}\left(\mathbf{g^{(k)}}\right) = \frac{1}{p}\sum_{k=0}^{p-1}\sum_{j=0}^{c-1} g_j^{(k)} > th_{FFN}^1 \\ \frac{1}{p}\sum_{k=0}^{p-1} \text{Bias}\left(\mathbf{G}_{k,:}\right) = \frac{1}{p}\sum_{k=0}^{p-1} \text{Bias}\left(\mathbf{g^{(k)}}\right) = \frac{1}{p}\frac{1}{c}\sum_{k=0}^{p-1}\sum_{j=0}^{c-1}\left(g_j^{(k)} - \mu(\mathbf{g^{(k)}})\right) > th_{FFN}^2 \\ CV\left(\mathbf{m}\right) = \frac{\sigma(\mathbf{m})}{\mu(\mathbf{m})} = \frac{\sqrt{\frac{1}{p}\sum_{j=0}^{p-1}\left(m_k - \mu(\mathbf{m})\right)^2}}{\frac{1}{p}\sum_{k=0}^{p-1} m_k} < th_{FFN}^3 \end{cases}$$

where $\mu(\mathbf{g}^{(k)}) = \frac{1}{c}\sum_{j=0}^{c-1} g_j^{(k)}$, $\mu(\mathbf{m}) = \frac{1}{p}\sum_{k=0}^{p-1} m_k$. The thresholds $th_{FFN}^1, th_{FFN}^2, th_{FFN}^3$ are set by grid search, which is elaborated in Section 6.3.4

The relatedness criterion is measured by the sum of label logits. The biased criterion is measured by the logit difference between each label logit and the average label logit. The low variance criterion is measured by the coefficient variance (CV) of the FFN vector coefficient across different samples, which is the standard deviation normalized by the mean of the data.

6.3.2 Biased Attention Heads Identification

The identification of biased attention heads closely resembles the process of identifying biased FFN vectors. As discussed in Section 6.2.1, each attention head's contribution to the final prediction is independent and interpretable. Therefore, we project the output hidden states of each attention head into the vocabulary space to interpret the information they contribute.

To identify biased attention heads, we use the same three criteria introduced for identifying biased FFN vectors. To apply these criteria, we project the output hidden states from each attention head into the vocabulary space and analyze their label logits as the information contributes to label prediction. The output from each attention head consists of hidden states generated for every token in the sequence. For our analysis, we specifically use the hidden state of the last token preceding the prediction of label names, interpreting it as the most direct contribution of the attention head to the prediction, given the autoregressive nature of LLMs.

Specifically, to obtain the label logits for an attention head, consider the output hidden states $H \in \mathbb{R}^{N\times d}$ of this head, the unembedding matrix $E \in \mathbb{R}^{d\times d_e}$, and the label token mapping matrix $L \in \mathbb{R}^{N\times d_e}$. Given the token position $p_{\text{label}} \in \{0, 1, \ldots, N-1\}$, which indicates the index of the first token of the predicted label names, the label logits $\mathbf{a}^{(k)} = [a_1^{(k)}, a_2^{(k)}, \ldots, a_c^{(k)}]^\top$ of the attention head for the k-th sample are derived by:

$$\mathbf{a}^{(k)} = H_{(p_{\text{label}}-1),:} \cdot E \cdot L^\top.$$

we employ the same p unlabeled samples from the task to assess the criteria for identifying baised attention head. The label logits for these samples are formed as $A = [\mathbf{a}^{(0)}, \mathbf{a}^{(2)}, \ldots, \mathbf{a}^{(m-1)}]^\top \in \mathbb{R}^{m\times c}$. An attention head is considered biased if it meets the following conditions:

$$\begin{cases} \dfrac{1}{p}\displaystyle\sum_{k=0}^{p-1} \text{Sum}\left(A_{k,:}\right) = \dfrac{1}{p}\sum_{k=0}^{p-1} \text{Sum}\left(\mathbf{a}^{(k)}\right) = \dfrac{1}{p}\sum_{k=0}^{p-1}\sum_{j=1}^{c} \mathbf{a}_j^{(k)} > th_{Att}^1 \\ \dfrac{1}{p}\displaystyle\sum_{k=0}^{p-1} \text{Bias}\left(A_{k,:}\right) = \dfrac{1}{p}\sum_{k=0}^{p-1} \text{Bias}\left(\mathbf{a}^{(k)}\right) = \dfrac{1}{p}\dfrac{1}{c}\sum_{k=0}^{p-1}\sum_{j=0}^{c-1} \left(a_j^{(k)} - \mu(\mathbf{a}^{(k)})\right) > th_{Att}^2 \\ \displaystyle\sum_{j=0}^{c-1} w_j \cdot CV\left(A_{:,j}\right) = w_j \cdot \dfrac{\sigma(A_{:,j})}{\mu(A_{:,j})} < th_{Att}^3 \end{cases}$$

where $w_j = \sum_{j=0}^{c-1} \frac{\mu(A_{:,j})}{\sum \mu(A_{:,j})}$, $\mu(A_{:,j}) = \frac{1}{p}\sum_{k=0}^{p-1} A_{i,j}$, $\sigma(A_{:,j}) = \sqrt{\frac{1}{p}\sum_{k=0}^{p-1}(A_{i,j} - \mu(A_{:,j}))^2}$. The functions of the first two criteria are identical to those for biased FFN vector identification. The third function is the weighted sum of the coefficient variance of each label across test samples. The thresholds for biased attention head identification are also derived by grid search.

6.3.3 Biased FFN Vectors and Attention Heads Manipulation

After identifying the biased components of the LLM, we eliminate their influence by masking these biased FFN vectors and attention heads. Specifically, we create masks for the attention heads in each attention layer and reset the coefficient of the biased FFN vector and biased attention head mask.

6.3.4 Grid Searching

Specifically, we utilize a small subset of training data as a support set, with 20 samples for each class. We then perform a grid search over all combinations of threshold values and select the combination that results in the most balanced distribution of average label logits. Specifically, let $\mathbf{T}$ represents the set of threshold combinations, and $P(t)$ denote the average label logits for a threshold combination $t \in \mathbf{T}$, we aim to find the combination t^* that minimizes the bias of label logits: $t^* = \arg\min_{t \in \mathbf{T}} \text{Bias}(\mathbf{P}(t))$.

It is noteworthy that although there are multiple combinations of thresholds, they usually result in a few set of different biased components. Many different combinations of grid search thresholds yield the same set of biased LLM components. For example, for a grid search of thresholds of FFN vectors with 80 combinations, it only result in 4 different sets of biased FFN vectors that need to be examined with the support set on the SST-2 dataset. Additionally, during the inference stage of evaluating test samples, the computation time of the UniBias method is completely identical to that of the original LLMs.

6.4 Experiments

In this section, we aims to investigate a few research questions (RQ). **RQ 1**: After eliminating biased components from LLMs, does the ICL performance improve compared to the original LLM? Additionally, how does our UniBias method compare to existing calibration methods? **RQ 2**: Given that ICL suffers from prompt brittleness, can our UniBias method contribute to more robust ICL performance? **RQ 3**: Are there any observable patterns of biased FFN vectors and attention heads within and across tasks? **RQ 4**: What is the performance of LLMs after eliminating only the biased FFN vectors and only the biased attention heads, respectively? **RQ 5**: What is the impact of support set size on the performance of the UniBias method?

6.4.1 Experimental Setup

6.4.1.1 Datasets

We evaluate our UniBias method on 12 diverse natural language processing datasets across various tasks, including sentiment analysis, topic classification, natural language inference, reasoning, and word disambiguation, as presented in Table 6.1. In our experiments, we utilize k (where $k = 0, 1, 2, 4$) training samples per class as prompt examples for k-shot ICL. For testing, we randomly select 2000 samples for MMLU and 3000 samples for MNLI and MR, while employing the original testing sets for other datasets. Detailed dataset statistics are available in Table 6.1.

Dataset	**# Classes**	**# Testing Size**
Sentiment classification		
SST2 [9]	2	872
SST-5 [9]	5	2210
MR [228]	2	3000
CR [229]	2	376
Topic classification		
AGNews [10]	4	7600
TREC [230]	6	500
Natural language inference		
MNLI [231]	3	3000
RTE [232]	2	277
Reasoning		
ARC-Challenge [233]	4	1170
MMLU [234]	4	2000
COPA [235]	2	100
Word disambiguation		
WiC [236]	2	638

TABLE 6.1: Detailed Dataset information

6.4.1.2 Baselines

In addition to the standard ICL, we compare our proposed UniBias with state-of-the-art LLM debiasing and calibration baselines, including **Contextual Calibration (CC)** [6], **Domain-Context Calibration (DC)** [136], and **Prototypical Calibration (PC)** [139]. We reproduce all baselines strictly follows the authors' instructions and recommendations to ensure a fair comparison.

6.4.1.3 Models and implementation details

Models: We evaluate our method on Llama-2 7b and Llama-2 13b models [121]. For all experiments, unless stated otherwise, we use 1-shot ICL setting, i.e. one example per class, and repeat five times under different random seeds. We use $k = 20$ sampes per class as the support set to obtain all threshold values by grid searching, as mentioned in the method section.

Experiments on internal mechanisms of biased factors: All experiments are conducted on Llama-2 7b model. For the vanilla label bias experiment, we projected all FFN value vectors into the vocabulary space and sum the label logits for all FFN vectors whose label logits rank within the top 10 over the vocabulary to calculate uncontextual accumulated FFN logits. We change different set of label words in prompt to derive the label prediction frequency of different label pairs. For the recency bias experiment, based on findings in [191], instead of the summed attention weights over the whole example, we adopt the sum of attention weights on label words of the example, e.g. "Answer: positive" as the effective attention weight on each example. For the selection bias experiment, we use zeroshot ARC dataset prompts in Table A.3, and we use 12 samples for each class. The attention weight is also summed on label words instead of the whole option.

Baselines: We reproduce all baselines using the publicly available code released by the authors to ensure a fair comparison. For the PC method, instead of using test samples as in the original work, we employ 200 training samples per class as the estimate set for parameter estimation using the EM algorithm. This adjustment is made to reflect real-world scenarios where test samples are not readily available. Additionally, the number of samples used by the PC method is significantly larger than that used by our UniBias method.

Unibias: In our method, all threshold values are determined through grid searching as described in the methodology section. Specifically, we use 20 samples per class as the support set for grid searching in all experiments. For each repetition of the experiment, the support set is randomly selected based on different random seeds. Additionally, to manipulate biased FFN vectors and attention heads, we create masks for the attention heads of all attention layers and adjust the FFN coefficient values and attention head masks using the hook operation. Additionally, we conduct the experiment on four A5000 GPUs. The prompts used in this paper are provided in Appendix A.3.

Dataset	Llama-2 7b					Llama-2 13b				
Method	ICL	CC	DC	PC	UniBias	ICL	CC	DC	PC	UniBias
SST-2	$87.22_{6.03}$	$92.24_{3.39}$	$94.15_{1.22}$	$93.90_{1.54}$	$\mathbf{94.54}_{0.62}$	$93.90_{1.79}$	$95.25_{0.93}$	$95.37_{0.70}$	$94.56_{1.71}$	$\mathbf{95.46}_{0.52}$
MNLI	$53.83_{2.22}$	$53.36_{3.16}$	$52.19_{2.55}$	$45.38_{5.01}$	$\mathbf{54.97}_{0.88}$	$62.43_{1.49}$	$63.89_{0.81}$	$61.86_{1.23}$	$57.47_{3.53}$	$\mathbf{64.65}_{2.73}$
WiC	$50.00_{0.16}$	$52.19_{2.00}$	$52.40_{1.69}$	$\mathbf{57.11}_{2.49}$	$53.71_{1.16}$	$54.48_{3.19}$	$50.63_{1.73}$	$49.72_{0.30}$	$55.67_{1.67}$	$\mathbf{57.93}_{1.70}$
COPA	$67.60_{2.30}$	$67.80_{2.17}$	$60.40_{2.79}$	$67.80_{3.70}$	$\mathbf{69.00}_{2.74}$	$67.50_{10.40}$	$75.20_{7.80}$	$71.00_{8.80}$	$76.80_{6.30}$	$\mathbf{83.20}_{2.70}$
CR	$91.54_{0.39}$	$92.13_{0.40}$	$92.61_{0.44}$	$91.97_{0.35}$	$\mathbf{92.61}_{0.11}$	$91.01_{1.30}$	$92.13_{0.88}$	$92.23_{0.76}$	$91.65_{0.64}$	$\mathbf{92.34}_{0.74}$
AGNews	$85.59_{1.87}$	$83.54_{1.96}$	$\mathbf{89.08}_{0.86}$	$86.81_{2.92}$	$88.29_{1.24}$	$89.14_{0.44}$	$88.23_{1.14}$	$\mathbf{89.34}_{0.61}$	$86.03_{0.65}$	$88.68_{0.43}$
MR	$89.37_{1.83}$	$91.77_{1.42}$	$\mathbf{92.35}_{0.23}$	$91.39_{1.65}$	$92.19_{0.37}$	$90.10_{2.10}$	$\mathbf{93.20}_{0.57}$	$93.00_{0.52}$	$92.80_{0.86}$	$92.23_{1.12}$
RTE	$66.21_{7.30}$	$64.33_{3.68}$	$65.49_{2.09}$	$62.59_{4.71}$	$\mathbf{67.65}_{6.44}$	$76.10_{4.73}$	$71.99_{5.02}$	$66.21_{1.09}$	$75.31_{2.90}$	$\mathbf{78.23}_{2.13}$
SST-5	$46.97_{0.87}$	$51.36_{1.69}$	$51.92_{1.77}$	$\mathbf{55.41}_{1.51}$	$53.79_{1.46}$	$51.03_{1.25}$	$47.20_{1.69}$	$48.98_{2.11}$	$\mathbf{53.63}_{0.95}$	$51.80_{1.00}$
TREC	$72.92_{12.42}$	$76.44_{3.21}$	$77.16_{3.94}$	$74.92_{5.78}$	$\mathbf{80.80}_{3.17}$	$74.70_{12.10}$	$\mathbf{83.80}_{3.86}$	$80.50_{9.07}$	$81.85_{9.53}$	$81.25_{6.86}$
ARC	$51.90_{0.60}$	$\mathbf{53.10}_{0.40}$	$53.00_{0.60}$	$40.40_{0.50}$	$\mathbf{53.10}_{0.60}$	$66.54_{0.33}$	$64.33_{0.99}$	$64.88_{0.59}$	$59.47_{1.07}$	$\mathbf{66.81}_{0.37}$
MMLU	$41.73_{2.25}$	$43.72_{0.97}$	$43.57_{1.38}$	$34.12_{3.41}$	$\mathbf{44.83}_{0.24}$	$53.53_{1.55}$	$50.84_{1.57}$	$51.81_{1.24}$	$45.50_{1.65}$	$\mathbf{53.55}_{1.05}$
Avg.	67.07	68.49	68.70	66.81	**70.46**	72.54	73.06	72.08	72.56	**75.51**

TABLE 6.2: Comparison of one-shot ICL performance for different methods across datasets using Llama-2 7b and Llama-2 13b models. The mean and standard deviation are reported for five repetitions with different ICL examples.

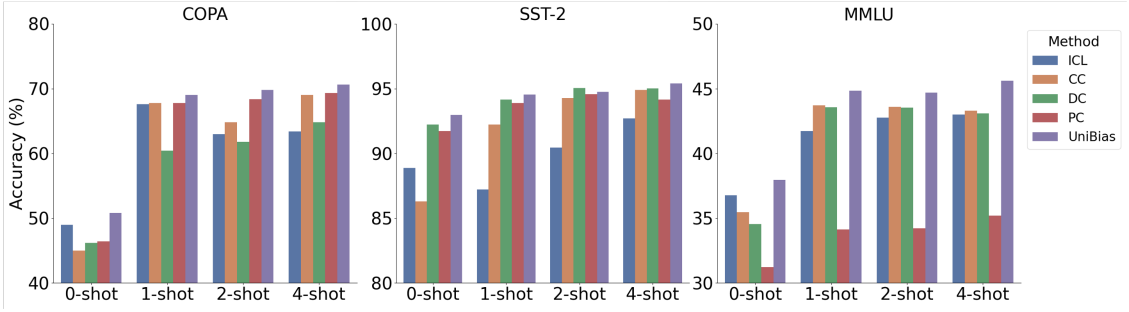


FIGURE 6.5: The performance comparison under different numbers of ICL shots.

6.4.2 Main Experiments

Table 6.2 presents the performance of various datasets and model sizes under the 1-shot setting. Our proposed UniBias method consistently achieves the highest accuracies in most cases. In terms of the overall average accuracy, UniBias improves upon the standard ICL by a substantial margin of 3.39% and exceeds the state-of-the-art (SOTA) DC by 1.76% using Llama-2 7b. With Llama-2 13b, UniBias surpasses the standard ICL and the SOTA CC by 2.97% and 2.45%, respectively. Figure 6.5 further illustrates the results under zero-shot and various few-shot settings for COPA, SST2, and MMLU. Our proposed UniBias consistently surpasses other baselines in all scenarios, underscoring its effectiveness.

In response to **RQ 1**, UniBias not only enhances the performance of original LLMs but also outperforms existing methods. We attribute this success to its internal analysis and bias mitigation techniques, which leverage FFNs and attentions, unlike other methods that rely solely on external observations.

6.4.3 Alleviating Prompt Brittleness

Existing studies have found that LLMs are prone to prompt brittleness, with various factors such as the selection and order of examples, as well as the prompt formatting. To address **RQ 2**, we simulate these brittle scenarios by choosing different demonstration samples, using different prompt formats, and changing the example order to observe variations in LLM performance.

Figure 6.1 presents Llama-2 7b's performance both with and without UniBias. Without UniBias, the standard ICL's performance varies significantly, ranging from 8% to 26%,

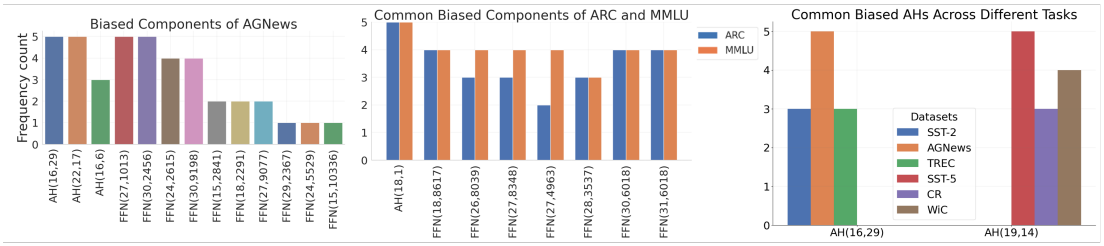


FIGURE 6.6: Analysis of biased attention heads (AHs) and FFN vectors (FFNs). The frequency count of biased LLM components across five repeat experiments with different example selections is reported.

demonstrating its instability. After applying UniBias, the accuracy stabilizes, with variations consistently less than 4% under perturbations of various design settings. This evidence verifies that UniBias effectively reduces prompt brittleness and enhances robustness.

Method	SST-2	MNLI	WiC	COPA	CR	AGNews	MR	RTE	SST-5	TREC	ARC	MMLU
ICL	87.22	53.83	50.00	67.60	91.54	85.59	89.37	66.21	46.97	72.92	51.90	41.73
FFN-only	94.17	54.59	50.88	69.20	92.57	85.52	91.78	67.33	47.09	73.04	51.92	42.62
Attention-only	94.22	52.83	52.76	68.50	91.49	86.25	**92.61**	66.55	52.68	80.68	53.00	44.67
UniBias	**94.54**	**54.97**	**53.71**	**69.00**	**92.61**	**88.29**	92.19	**67.65**	**53.79**	**80.80**	**53.10**	**44.83**

TABLE 6.3: Performance comparison of only removing biased FFN vectors (FFN-only), only removing biased attention heads (attention-only), our Unibias method, and the ICL of original LLM.

6.4.4 Biased LLM Components Analysis

In response to **RQ3**, we present the frequency counts of identified biased attention heads (AHs) and FFNs under repeated experiments in Figure 6.6. A large frequency count for an LLM component indicates a higher repeat of being identified as biased in the corresponding dataset. The first subfigure displays the biased components for various example selections, revealing several commonly biased LLM components across different prompts within a single dataset. The second subfigure highlights the common biased components across different datasets (ARC and MMLU) for the reasoning task, indicating that different datasets with similar tasks could share common biased LLM components. The third subfigure demonstrates the presence of common biased LLM components across different tasks. Experimental results suggest an interesting future direction: we may identify global biased components that would mitigate bias across multiple tasks and diverse prompt design settings.

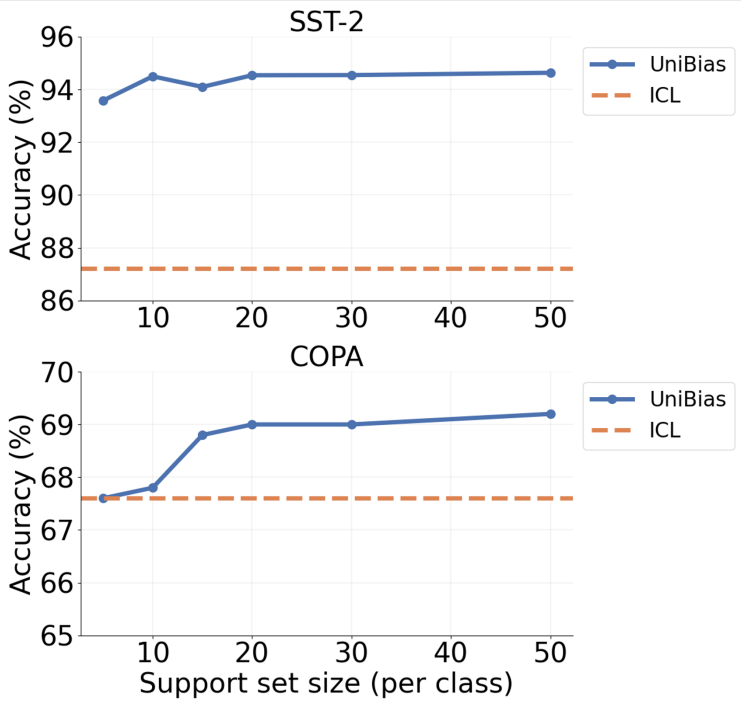


FIGURE 6.7: Performance of Unibias under different support set.

6.4.5 Ablations

We conduct ablation studies to analyze the impact of exclusively eliminating biased AHs or FFNs to address **RQ 4**. Table 6.3 presents the results of removing only biased FFN vectors (FFN-only) and only biased attention heads (attention-only). Both FFN-only and attention-only methods outperform the standard ICL, demonstrating their effectiveness. When combined as UniBias, the method achieves the best results across most datasets, indicating that the two approaches are complementary.

Additionally, we conduct experiments to investigate the impact of support set size (**RQ 5**) as our proposed UniBias method employs a small support set for grid searching. To analyze its effect, we vary the size of the support set. Figure 7 illustrates Unibias's performance with support set sizes ranging from 5 to 50 samples. The results indicate that the performance stabilizes when the support set contains 20 or more samples per class. Notably, for the SST2 dataset, even with much fewer support samples, Unibias significantly outperforms the standard ICL.

6.5 Related Work

Bias in LLMs: It is well recognized that LLMs are unstable under various ICL design settings, and this instability arises from biases in LLMs toward predicting certain answers [6, 135]. To understand these biases, existing studies have identified various bias factors, including recency bias, majority label bias, common token bias [6], and domain label bias [136] in classification tasks. More recently, selection bias, which consistently favors specific options in multiple-choice questions, has also been identified [137, 138]. To address these biases, several calibration methods have been proposed, including contextual calibration [6], domain-context calibration [136], and prototypical calibration [139]. However, these identified bias factors and calibration methods are derived from external observations or adjustments of LLM outputs, leaving the underlying mechanisms within LLMs that cause such biases poorly understood.

Mechanistic Interpretability: Mechanistic interpretability [141, 142] aims to explain the internal processes in language models, facilitating the interpretation of the contributions of individual model components to the final prediction. Our work builds on the understanding of the residual stream [141], the logit lens [146], and the interpretation of LLM components in the vocabulary space [143, 144].

6.6 Conclusion

In this work, we have deepened the understanding of biases in LLMs by unveiling the internal mechanisms that contribute to various bias factors. Building on this understanding, we proposed our UniBias method to mitigate these biases by identifying and eliminating biased FFN vectors and attention heads, demonstrating an effective way to manipulate the internal structures of LLMs. Extensive experiments show that our UniBias method achieves state-of-the-art performance across 12 NLP datasets and different ICL settings. Additionally, our method successfully alleviates prompt brittleness and enhances the robustness of ICL.

There are many interesting avenues for future research. For instance, instead of identifying biased components for each ICL prompt, future work could explore the identification of global biased components that mitigate bias across multiple tasks and diverse prompt design settings. Additionally, the biased FFN vectors and attention heads we identify could potentially serve as sensors for guiding effective prompt generation. We expect

that this internal perspective on LLM bias will inspire more innovative applications in both bias mitigation methods and prompt engineering.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis addresses the challenges of the EE task to achieve more accurate and practically applicable event extraction. Additionally, it tackles challenges that are prevalent not only in EE but extend across various NLP tasks, such as the classification problem involving the Universum class, thereby expanding the benefits of this work to a broader context. Moreover, following the revolutionary impact of LLMs since the release of ChatGPT in November 2022, our investigation has similarly undergone a paradigm shift towards exploring LLMs. Consequently, this thesis presents pioneering explorations into the application of LLMs in EE and efforts to mitigate inherent biases within LLMs.

For event extraction, our contributions are listed as follows:

- Chapter 3 introduces a novel method for document-level event argument extraction. This method capitalizes on redundant event information within documents to enhance accuracy in document-level EE. It also utilizes coreference information to enhance comprehension of the document. Additionally, a new loss function is proposed to appropriately handle the *Others* class in the EE task.
- Chapter 4 presents pioneering work in developing a prompting strategy tailored for EE. Traditional event extraction approaches based on supervised learning require substantial volumes of annotated training data, which pose challenges in the development and scalability of EE systems in real-world applications. By

leveraging the ICL capabilities of LLMs, we significantly reduce the reliance on large-scale annotated data and substantially enhance performance in a few-shot setting.

This thesis also investigates some fundamental issues that are prevalent not only in event extraction (EE) but also extend across various NLP tasks. The contributions are outlined as follows:

- In Chapter 5, we address an often overlooked issue concerning the classification problem involving the Universum class, which exists in many classification-based tasks in NLP. This class is characterized by its heterogeneity and lack of representativeness in training data. Traditional methods frequently treat the Universum class on par with the target classes, leading to issues such as overfitting, misclassification, and diminished model robustness. We propose a closed boundary learning framework that more appropriately handles this unique class.
- In Chapter 6, we explore the inherent biases of LLMs, which often compromise their effectiveness, leading to prompt brittleness—sensitivity to design settings such as example selection, order, and prompt formatting. Our research delves into how feedforward neural networks (FFNs) and attention heads contribute to the bias of LLMs. We introduce the UniBias method, an innovative approach that effectively identifies and eliminates biased FFN vectors and attention heads to mitigate LLM bias.

In terms of exploring LLMs, our contributions are as follows:

- To deepen our understanding of the In-Context Learning (ICL) paradigm, Chapter 4 investigates what LLMs learn from ICL demonstrations. Specifically, we hypothesize and validate that LLMs learn task-specific heuristics from demonstrations in ICL. Building upon this hypothesis, we introduce an explicit heuristic-driven demonstration construction approach, which transforms the haphazard example selection process into a systematic method that emphasizes task heuristics.
- In Chapter 6, we delve into the inherent biases of LLMs that frequently lead to prompt brittleness. We present the first work to explore the internal mechanisms contributing to LLM bias, focusing particularly on how feedforward neural network (FFN) vectors and attention heads contribute to the bias. By Interpreting

the contribution of individual FFN vectors and attention heads, we identify and eliminate biased components of LLMs, thereby mitigate the LLM bias.

7.2 Future Works

The potential for future research in event extraction, classification tasks involving the Universum class, and LLM bias mitigation is vast. The following areas have been identified as particularly promising:

- For EE task, future research could significantly benefit from more rigorously examined datasets. Existing document-level EE datasets often contain many missing arguments due to multiple occurrences of event arguments within a document. This can mislead both the model training process and the evaluation of model performance.
- The evaluation metric for EE could also be further refined. Current methods predominantly use exact match criteria, which do not account for semantically identical arguments that differ textually. This limitation can obscure the true effectiveness of EE methods.
- Given the comprehensive reasoning and understanding capabilities of LLMs, integrating these models into event extraction shows promise. Investigating more approaches based on ICL or fine-tuning LLMs for EE could yield fruitful results.
- An interesting aspect of the closed boundary framework we proposed is its indication of the models' knowledge boundaries. Within the closed boundaries are spaces that the models understand about the task, while spaces beyond represent areas of unknown. Future research could explore using closed boundaries to characterize these knowledge boundaries, potentially enhancing the reliability of model predictions.
- Our findings in Chapter 4 reveal that LLMs learn task heuristics from demonstrations. Future works could continue to explore prompting strategies from the novel perspective of task heuristics, simplifying the example selection process and potentially enhancing model performance and adaptability.

- Our pioneering work investigating the internal mechanisms of LLM bias and mitigating it by manipulating LLM components opens many avenues for future research. One area could involve identifying global biased components that mitigate bias across multiple tasks and diverse prompt settings. Additionally, the biased FFN vectors and attention heads we've identified could potentially serve as sensors to guide effective prompt generation. This internal perspective on LLM bias is expected to inspire innovative applications in both bias mitigation methods and prompt engineering.

Appendix A

Prompts Adopted in This Thesis

A.1 Recognize Implicit Heuristics of In-Context Examples by GPT-4

The prompt we use in Chapter 4 to recognize implicit heuristics inherent in examples and the output of GPT-4 is given below.

Prompt Input

```
What is the most critical and profound heuristic at play in each of the following
    examples? If any two examples share the same heuristic, please indicate this
    connection. Use no more than two sentences to illustrate the heuristics of
    each example.

example 1
Q: Did the death of Helen Palmer have a significant effect on Dr. Seuss?
A: Helen Palmer was Dr. Seuss's wife. The death of a spouse can have a significant
     effect on a person. Thus, the death of Helen Palmer could have a significant
    effect on Dr. Seuss. So the answer is yes.

example 2
Q: Would a moose hypothetically be too much for a minotaur to devour whole?
A: A moose is about 1,000 pounds. A minotaur is about 1,000 pounds. Thus, a moose
    would be too much for a minotaur to devour whole. So the answer is yes.

example 3
Q: Yes or no: Is H's most common two letter pair partner a freebie in Wheel of
    Fortune bonus round?
A: The most common two letter pair partner for H is E. E is a freebie in the Wheel
     of Fortune bonus round. Thus, H's most common two letter pair partner is a
    freebie in the Wheel of Fortune bonus round. So the answer is yes.

example 4
Q: Yes or no: Would bald eagle deliver an urgent message before B-52?
A: The top speed of a bald eagle is about 80 mph. The top speed of a B-52 is about
     650 mph. Thus, a B-52 would be faster than a bald eagle. So the answer is no.

example 5
Q: Yes or no: Does an individual oceanographer study many sciences?
A: Oceanography is the study of the ocean. The ocean is a complex system. Thus,
    oceanography is a multidisciplinary science. So the answer is yes.

example 6
Q: Yes or no: Could Steven Spielberg send emails as a child?
A: Steven Spielberg was born in 1946. Email was invented in 1971. Thus, Steven
    Spielberg could not send emails as a child. So the answer is no.
```

LLM Output

```
Example 1: This employs the "empathetic reasoning" heuristic, assuming that
    because most people would be significantly affected by the death of a spouse,
    so too would Dr. Seuss.

Example 2: The "comparison heuristic" is used here, comparing the size and weight
    of a moose to that of a minotaur to make a conclusion about the feasibility of
     consumption.

Example 3: This uses the "knowledge-based" heuristic, using known facts about a
    game show's rules to answer the question.

Example 4: This is another instance of the "comparison heuristic," this time
    comparing the top speeds of a bald eagle and a B-52 to determine which would
    deliver a message faster.

Example 5: The "definition-based" heuristic is applied here, relying on the
    understanding of what the field of oceanography entails to conclude that it
    involves the study of multiple sciences.

Example 6: The "chronological heuristic" is used here, comparing the timelines of
    Spielberg's childhood and the invention of email to answer the question.
```

A.2 Explicit Heuristic Generation by GPT-4

The prompt we utilize in Chapter 4 to generate explicit heuristics for document-level EAE and the output by GPT-4 is provided below.

Prompt Input

```
Generate 10 high-level and concise heuristics for identifying the event argument
    of  'giver' in a document, as part of the event argument extraction task. Each
     heuristic should offer a distinct perspective and adhere to the format
    demonstrated in the provided example.

Example heuristic:
Semantic heuristic: The 'giver' is identified as the individual, group, or
    organization mentioned in the document that is responsible for providing a
    gift or grant.
```

LLM Output

```
Grammatical heuristic: The 'giver' is typically the subject in a sentence
    structure where a verb of giving (e.g., 'donate', 'provide', 'offer') is used,
     indicating the action performer.
Lexical heuristic: The 'giver' can often be identified by keywords or phrases that
     signify giving, such as 'donor', 'benefactor', or 'sponsor', appearing in
    close proximity to the action of giving.
Syntactic heuristic: The 'giver' may be recognized by analyzing sentence structure
    , often appearing before prepositional phrases starting with 'to' that
    introduce the recipient (e.g., "X gives Y to Z", X is the 'giver').
Semantic role labeling heuristic: In sentences where an action of giving is
    present, the 'giver' is likely the entity filling the semantic role of 'Agent'
     or 'Source', initiating the action.
Coreference resolution heuristic: The 'giver' might be determined by identifying
    pronouns or noun phrases that refer back to an individual, group, or
    organization previously mentioned as performing the action of giving.
Dependency parsing heuristic: In parsing the sentence structure, the 'giver' is
    often connected through a dependency relation (e.g., 'nsubj' for nominal
    subject) to the main verb representing the giving action.
Discourse analysis heuristic: The 'giver' can be inferred from the broader context
     or narrative of the document, where a party is described or implied to be in
    a position to give, support, or provide resources.
Named Entity Recognition (NER) heuristic: The 'giver' is often a named entity
    categorized as a person, organization, or geopolitical entity, involved in a
    transactional or giving action.
Pragmatic heuristic: The 'giver' may be implied through indirect speech acts or
    politeness strategies, where the action of giving is not explicitly stated but
     understood from the context or social norms.
Co-occurrence heuristic: The 'giver' might be associated with frequent co-
    occurrence with verbs or nouns related to giving in a large text corpus,
    indicating a statistical likelihood of being the giving party.
```

A.3 Prompt Templates

The prompt templates used in Chapter 6 are provided below. We generate few-shot ICL templates follow the template styles in [136, 139], as illustrated in Table A.1.

Dataset	**Template**	**Label Space**
SST-2 CR MR	Review: {*sentence*} Sentiment: {*label*}	negative / positive
MNLI	Premise: {*premise*} Hypothesis: {*hypothesis*} Answer: {*label*}	yes / maybe / no
ARC MMLU	Question: {*question*} {*options*} Answer: {*label*}	A / B / C / D
SST-5	Review: {*sentence*} Sentiment: {*label*}	terrible / bad / okay / good / great
AGNews	Article: {*passage*} Answer: {*label*}	world / sports / business / technology & science
TREC	Question: {*sentence*} Answer Type: {*label*}	abbreviation / entity / description / person / location / number
COPA	Premise: {*premise*} Choice1: {*choice1*} Choice2: {*choice2*} Answer: {*label*}	1 / 2
RTE	Premise: {*sentence1*} Hypothesis: {*sentence2*} Answer: {*label*}	yes / no
WiC	Sentence1: {*sentence1*} Sentence2: {*sentence2*} Word: {*word*} Answer: {*label*}	false / true

TABLE A.1: Prompt templates for all k-shot ICL experiments.

ID	Template	Label Space
1	Review: {Sentence} Sentiment: {Label}	Positive / Negative
2	Input: {Sentence} Prediction: {Label}	Positive / Negative
3	Review: {Sentence} Sentiment: {Label}	good / bad
4	{Sentence} It was {Label}	good / bad
5	Review: {Sentence} Positive Review: {Label}	Yes / No
6	{Sentence} My overall feeling was that the movie was {Label}	good / bad
7	Review: {Sentence} Question: Is the sentiment of the above review Positive or Negative? Answer: {Label}	Positive / Negative
8	My review for last night's film: {Sentence}The critics agreed that this movie was {Label}	good / bad

TABLE A.2: Templates of different prompt formatting used in the prompt brittleness experiment for SST-2.

Dataset	Template	Label Set
SST-2	Review: {sentence} Sentiment: {label}	negative / positive
COPA	Premise: {premise} Choice1: {choice1} Choice2: {choice2} Answer: {label}	1 / 2
MMLU	Question: {*question*} {*options*} Answer: {*label*}	A / B / C / D

TABLE A.3: Prompt templates for the 0-shot experiments.

List of Author's Publications

Conference Proceedings

- **Hanzhang Zhou**, and Kezhi Mao. "Document-level event argument extraction by leveraging redundant information and closed boundary loss." *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2022)*, pp. 3041-3052. 2022.

- **Hanzhang Zhou**, Zijian Feng, and Kezhi Mao. "Closed Boundary Learning for Classification Tasks with the Universum Class." *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 15522–15536. 2023.

- **Hanzhang Zhou**, Junlang Qian, Zijian Feng, Hui Lu, Zixiao Zhu and Kezhi Mao. "LLMs Learn Task Heuristics from Demonstrations: A Heuristic-Driven Prompting Strategy for Document-Level Event Argument Extraction". *Accepted to The 62ND Annual Meeting Of The Association For Computational Linguistics (ACL 2024): Long Papers*. 2024.

- **Hanzhang Zhou**, Zijian Feng, Zixiao Zhu, Junlang Qian, and Kezhi Mao. "UniBias: Unveiling and Mitigating LLM Bias through Internal Attention and FFN Manipulation". *Submitted to Advances in Neural Information Processing Systems (NeuIPS 2024)*. 2024.

- Zijian Feng, **Hanzhang Zhou**, Kezhi Mao, Zixiao Zhu. "FreeCtrl: Constructing Control Centers with Feedforward Layers for Learning-Free Controllable Text Generation." *Accepted to the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024): Long Papers*. 2024.

- Zijian Feng, **Hanzhang Zhou**, Zixiao Zhu, Junlang Qian, Kezhi Mao. "Unveiling and Manipulating Prompt Influence in Large Language Models." *The Twelfth International Conference on Learning Representations (ICLR 2024)*. 2024

- Zixiao Zhu, Junlang Qian, Zijian Feng, **Hanzhang Zhou**, Kezhi Mao. "An Entailment-based Few-shot Text Classification with Extensional Definition". *Findings of the Association for Computational Linguistics: (NAACL 2024)*, pp. 1124-1137. 2024

- Zijian Feng, **Hanzhang Zhou**, Zixiao Zhu, Kezhi Mao. "PromptExplainer: Explaining Language Models through Prompt-based Learning". *Findings of the Association for Computational Linguistics: (EACL 2024)*, pp. 882-895. 2024.

- Zixiao Zhu, Zijian Feng, **Hanzhang Zhou**, Junlang Qian, Kezhi Mao, "MICL: Improving In-Context Learning through Multiple-Label Words in Demonstration". *Submitted to the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024)*. 2024.

- Junlang Qian, Zixiao Zhu, **Hanzhang Zhou**, Zijian Feng, Zepeng Zhai, Kezhi Mao, "Next-Token is Not All You Need: Unleashing Subsequent Tokens Parallelly for Zero-Shot Text Classification". *Submitted to the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024)*. 2024.

Journal Articles

- Zijian Feng, Kezhi Mao, and **Hanzhang Zhou**. "Adaptive micro-and macro-knowledge incorporation for hierarchical text classification." *Expert Systems with Applications*, 2024, 248: 123374.

- Zijian Feng, **Hanzhang Zhou**, Zixiao Zhu, Kezhi Mao. "Tailored text augmentation for sentiment analysis." *Expert Systems with Applications*, 2022, 205: 117605.

- Zeju Li, Jinjua Yu, Yuanyuan Wang, **Hanzhang Zhou**, Haowei Yang, and Zhongwei Qiao, "DeepVolume: Brain Structure and Spatial Connection-Aware Network for Thin-Section Brain MRI Reconstruction". *IEEE Transactions on Cybernetics*, 2019, 51(7): 3441-345

Bibliography

[1] Thien Huu Nguyen and Ralph Grishman. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 365–371, 2015. xix, 12, 13

[2] Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 300–309, 2016. xix, 10, 13, 14

[3] Xiao Liu, Zhunchen Luo, and He-Yan Huang. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, 2018. xix, 14, 15

[4] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022. xix, 24, 26, 28, 58, 74

[5] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022. xix, 16, 27, 63, 67, 69, 70, 71

[6] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/zhao21c.html`. xix, 28, 29, 103, 104, 108, 115, 121

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. xix, 15, 31

[8] Kashun Shum, Shizhe Diao, and Tong Zhang. Automatic prompt augmentation and selection with chain-of-thought from labeled data. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational*

Linguistics: EMNLP 2023, pages 12113–12139, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.811. URL https://aclanthology.org/2023.findings-emnlp.811. xx, 27, 59, 63, 64, 65, 67, 69, 71, 75

[9] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard, editors, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170. xx, 64, 69, 71, 105, 115

[10] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf. xx, 105, 115

[11] Jian Liu, Chen Liang, Jinan Xu, Haoyan Liu, and Zhe Zhao. Document-level event argument extraction with a chain reasoning paradigm. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9570–9583, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.532. URL https://aclanthology.org/2023.acl-long.532. xxi, 58, 69, 70, 71

[12] Jinlan Fu, Xuanjing Huang, and Pengfei Liu. SpanNER: Named entity re-/recognition as span prediction. 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 7183–7195, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.558. URL https://aclanthology.org/2021.acl-long.558. xxii, 19, 82, 83, 86, 92, 93, 95, 96, 98, 100, 101

[13] Zhengyan Li, Yicheng Zou, Chong Zhang, Qi Zhang, and Zhongyu Wei. Learning implicit sentiment in aspect-based sentiment analysis with supervised contrastive pre-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 246–256, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.22. URL https://aclanthology.org/2021.emnlp-main.22. xxii, 19, 82, 83, 86, 92, 93, 95, 101

[14] Ralph Grishman. Twenty-five years of information extraction. *Natural Language Engineering*, 25(6):677–692, 2019. 1, 57

[15] Sunita Sarawagi et al. Information extraction. *Foundations and Trends® in Databases*, 1(3):261–377, 2008. 1

[16] Wei Xiang and Bang Wang. A survey of event extraction from text. *IEEE Access*, 7:173111–173137, 2019. 1, 9

[17] Matthias Meyer, Timo Farei-Campagna, Akos Pasztor, Reto Da Forno, Tonio Gsell, Jérome Faillettaz, Andreas Vieli, Samuel Weber, Jan Beutel, and Lothar Thiele. Event-triggered natural hazard monitoring with convolutional neural networks on the edge. In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, pages 73–84, 2019. 1

[18] Dhekar Abhik and Durga Toshniwal. Sub-event detection during natural hazards using features of social media data. In *Proceedings of the 22nd international conference on world wide web*, pages 783–788, 2013. 1

[19] Wijnand Nuij, Viorel Milea, Frederik Hogenboom, Flavius Frasincar, and Uzay Kaymak. An automated framework for incorporating news into stock trading strategies. *IEEE transactions on knowledge and data engineering*, 26(4):823–835, 2013. 2

[20] Philippe Capet, Thomas Delavallade, Takuya Nakamura, Agnes Sandor, Cedric Tarsitano, and Stavroula Voyatzi. A risk assessment system with automatic extraction of event types. In *Intelligent Information Processing IV: 5 th IFIP International Conference on Intelligent Information Processing, October 19-22, 2008, Beijing, China 5*, pages 220–229. Springer, 2008. 2

[21] Barlogis Rodolphe, Ouedraogo Cheik, Aurelie Montarnal, and Didier Gourc. Employing bert model backed by expert knowledge to extract from textual media event of interest along container shipping supply chain. *IFAC-PapersOnLine*, 56 (2):11117–11122, 2023. 2

[22] George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon, 2004. 9

[23] Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 10, 14

[24] Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. 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 3533–3546, 2021. 10, 14

[25] Yuan Liang, Zhuoxuan Jiang, Di Yin, and Bo Ren. Raat: Relation-augmented attention transformer for relation modeling in document-level event extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4985–4997, 2022. 10, 15

[26] Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. Document-level event extraction via parallel prediction networks. 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 6298–6308, 2021.

[27] Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. 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 3533–3546, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.274. URL https://aclanthology.org/2021.acl-long.274. 37, 39, 40, 41, 45, 52, 53, 74

[28] Hai-Long Trieu, Thy Thy Tran, Khoa NA Duong, Anh Nguyen, Makoto Miwa, and Sophia Ananiadou. Deepeventmine: end-to-end neural nested event extraction from biomedical texts. *Bioinformatics*, 36(19):4910–4917, 2020. 10

[29] Ellen Riloff et al. Automatically constructing a dictionary for information extraction tasks. In *AAAI*, volume 1, pages 2–1, 1993. 11

[30] K Bretonnel Cohen, Karin Verspoor, Helen L Johnson, Christophe Roeder, Philip Ogren, William A Baumgartner Jr, Elizabeth White, and Lawrence Hunter. High-precision biological event extraction with a concept recognizer. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 50–58, 2009. 11

[31] Quoc-Chinh Bui and Peter MA Sloot. A robust approach to extract biomedical events from literature. *Bioinformatics*, 28(20):2654–2661, 2012. 11

[32] Jethro Borsje, Frederik Hogenboom, and Flavius Frasincar. Semi-automatic financial events discovery based on lexico-semantic patterns. *International Journal of Web Engineering and Technology*, 6(2):115–140, 2010. 11

[33] David Ahn. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, 2006. 11

[34] Hai Leong Chieu and Hwee Tou Ng. A maximum entropy approach to information extraction from semi-structured and free text. *Aaai/iaai*, 2002:786–791, 2002. 11

[35] Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. Using cross-entity inference to improve event extraction. In *Proceedings of*

the 49th annual meeting of the association for computational linguistics: human language technologies, pages 1127–1136, 2011. 11

[36] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10): 1995, 1995. 12

[37] Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, 2015. 13

[38] Grégoire Burel, Hassan Saif, Miriam Fernandez, and Harith Alani. On semantics and deep learning for event detection in crisis situations. 2017. 13

[39] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 13

[40] Diya Li, Lifu Huang, Heng Ji, and Jiawei Han. Biomedical event extraction based on knowledge-driven tree-lstm. 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 1421–1430, 2019. 14

[41] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, 2015. 14

[42] Kung-Hsiang Huang, Mu Yang, and Nanyun Peng. Biomedical event extraction with hierarchical knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1277–1285, 2020. 14

[43] Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. 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 4925–4936, 2019. 14

[44] Shuaicheng Zhang, Qiang Ning, and Lifu Huang. Extracting temporal event relation with syntax-guided graph transformer. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 379–390, 2022. 15

[45] I-Hung Hsu, Zhiyu Xie, Kuan-Hao Huang, Prem Natarajan, and Nanyun Peng. Ampere: Amr-aware prefix for generation-based event argument extraction model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10976–10993, 2023. 16

[46] Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. Prompt for extraction? paie: Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6759–6774, 2022. 16

[47] Zhiyang Xu, Jay Yoon Lee, and Lifu Huang. Learning from a friend: Improving event extraction via self-training with feedback from abstract meaning representation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10421–10437, 2023. 16

[48] Ge Shi, Yunyue Su, Yongliang Ma, and Ming Zhou. A hybrid detection and generation framework with separate encoders for event extraction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3163–3180, 2023. 16

[49] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 16, 26, 58, 70, 74, 103

[50] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023. 16

[51] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.deelio-1.10. URL https://aclanthology.org/2022.deelio-1.10. 16, 28, 58, 74

[52] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022. 16, 58, 67

[53] Jason Weston, Ronan Collobert, Fabian Sinz, Léon Bottou, and Vladimir Vapnik. Inference with the universum. In *Proceedings of the 23rd international conference on Machine learning*, pages 1009–1016, 2006. 17, 19, 20, 39, 41, 79, 85

[54] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45, 2017. URL https://nlp.stanford.edu/pubs/zhang2017tacred.pdf. 17, 19, 79, 83, 84

[55] Erik F. Tjong Kim Sang and Fien De Meulder. 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, 2003. URL https://aclanthology.org/W03-0419. 17, 19, 79, 83, 84, 92

[56] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015. 17, 79

[57] Enwei Zhu and Jinpeng Li. Boundary smoothing for named entity recognition. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7096–7108, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.490. URL https://aclanthology.org/2022.acl-long.490. 18, 19, 82, 83, 86, 92, 93, 95

[58] Fei Li, ZhiChao Lin, Meishan Zhang, and Donghong Ji. A span-based model for joint overlapped and discontinuous named entity recognition. 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 4814–4828, 2021. 18, 83

[59] Juncheng Wan, Dongyu Ru, Weinan Zhang, and Yong Yu. Nested named entity recognition with span-level graphs. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 892–903, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.63. URL https://aclanthology.org/2022.acl-long.63. 19, 82, 83

[60] Yuanhe Tian, Guimin Chen, Yan Song, and Xiang Wan. Dependency-driven relation extraction with attentive graph convolutional networks. 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 4458–4471, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.344. URL https://aclanthology.org/2021.acl-long.344. 86, 92, 93, 95

[61] Guimin Chen, Yuanhe Tian, Yan Song, and Xiang Wan. Relation extraction with type-aware map memories of word dependencies. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2501–2512, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.221. URL https://aclanthology.org/2021.findings-acl.221. 86, 92, 93, 95

[62] Yuncong Li, Cunxiang Yin, Sheng-hua Zhong, and Xu Pan. Multi-instance multi-label learning networks for aspect-category sentiment analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language*

Processing (EMNLP), pages 3550–3560, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.287. URL https://aclanthology.org/2020.emnlp-main.287. 86, 92, 93, 95

[63] Juntao Yu, Bernd Bohnet, and Massimo Poesio. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.577. URL https://aclanthology.org/2020.acl-main.577. 19, 83

[64] Virginia McLean. Fourth message understanding conference (muc-4). 1992. 19, 49

[65] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid O Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. *arXiv preprint arXiv:1911.10422*, 2019. 19, 84, 92

[66] Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. A challenge dataset and effective models for aspect-based sentiment analysis. 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 6280–6285, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1654. URL https://aclanthology.org/D19-1654. 19, 83, 84, 92

[67] Olivier Chapelle, Alekh Agarwal, Fabian Sinz, and Bernhard Schölkopf. An analysis of inference with the universum. *Advances in neural information processing systems*, 20, 2007. 19, 20, 85

[68] Zhiquan Qi, Yingjie Tian, and Yong Shi. Twin support vector machine with universum data. *Neural Networks*, 36:112–119, 2012. 19, 20, 41, 85

[69] Xiang Zhang and Yann LeCun. Universum prescription: Regularization using unlabeled data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017. 20, 85

[70] Chunhua Shen, Peng Wang, Fumin Shen, and Hanzi Wang. {\cal U} boost: Boosting with the universum. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):825–832, 2011. 19

[71] Bharat Richhariya and Muhammad Tanveer. Eeg signal classification using universum support vector machine. *Expert Systems with Applications*, 106:169–182, 2018. 19

[72] Eduardo Dadalto Camara Gomes, Florence Alberge, Pierre Duhamel, and Pablo Piantanida. Igeood: An information geometry approach to out-of-distribution detection. In *International Conference on Learning Representations*, 2022. 20, 21, 23, 83, 85

[73] Jie Ren, Stanislav Fort, Jeremiah Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. A simple fix to mahalanobis distance for improving near-ood detection. *arXiv preprint arXiv:2106.09022*, 2021. 21

[74] Xingyu Chen, Xuguang Lan, Fuchun Sun, and Nanning Zheng. A boundary based out-of-distribution classifier for generalized zero-shot learning. In *European Conference on Computer Vision*, pages 572–588. Springer, 2020. 20, 21, 83

[75] Hanlei Zhang, Hua Xu, and Ting-En Lin. Deep open intent classification with adaptive decision boundary. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14374–14382, 2021. 20, 23, 83, 85, 87, 88, 98

[76] Bo Liu, Hao Kang, Haoxiang Li, Gang Hua, and Nuno Vasconcelos. Few-shot open-set recognition using meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2020. 20, 83, 87, 88

[77] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018. 20, 83

[78] Vatsal Sharan, Parikshit Gopalan, and Udi Wieder. Efficient anomaly detection via matrix sketching. *Advances in neural information processing systems*, 31, 2018. 20, 83

[79] Mahito Sugiyama and Karsten Borgwardt. Rapid distance-based outlier detection via sampling. *Advances in neural information processing systems*, 26, 2013. 20, 83

[80] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021. 20, 22, 83, 84

[81] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009. 20

[82] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal processing*, 99:215–249, 2014. 20

[83] Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 20, 22

[84] Karanjit Singh and Shuchita Upadhyaya. Outlier detection: applications and techniques. *International Journal of Computer Science Issues (IJCSI)*, 9(1):307, 2012. 21

[85] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2022. 21

[86] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33: 21464–21475, 2020. 21

[87] Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. *Advances in Neural Information Processing Systems*, 34: 144–157, 2021. 21

[88] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018*, 2018. 21

[89] KIMIN LEE, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *32nd Conference on Neural Information Processing Systems (NIPS)*. Neural Information Processing Systems Foundation, 2018. 21

[90] Ev Zisselman and Aviv Tamar. Deep residual flow for out of distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13994–14003, 2020. 21

[91] Dihong Jiang, Sun Sun, and Yaoliang Yu. Revisiting flow generative models for out-of-distribution detection. In *International Conference on Learning Representations*, 2021. 21

[92] Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. *Advances in neural information processing systems*, 32, 2019. 21

[93] Zhisheng Xiao, Qing Yan, and Yali Amit. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. *Advances in neural information processing systems*, 33:20685–20696, 2020. 21

[94] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pages 9690–9700. PMLR, 2020. 21

[95] Haiwen Huang, Zhihan Li, Lulu Wang, Sishuo Chen, Bin Dong, and Xinyu Zhou. Feature space singularity for out-of-distribution detection. *arXiv preprint arXiv:2011.14654*, 2020. 21

[96] Julian Katz-Samuels, Julia B Nakhleh, Robert Nowak, and Yixuan Li. Training ood detectors in their natural habitats. In *International Conference on Machine Learning*, pages 10848–10865. PMLR, 2022. 23, 84

[97] Yifei Ming, Ying Fan, and Yixuan Li. Poem: Out-of-distribution detection with posterior sampling. In *International Conference on Machine Learning*, pages 15650–15665. PMLR, 2022.

[98] Jingkang Yang, Haoqi Wang, Litong Feng, Xiaopeng Yan, Huabin Zheng, Wayne Zhang, and Ziwei Liu. Semantically coherent out-of-distribution detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8301–8309, 2021.

[99] Sunil Thulasidasan, Sushil Thapa, Sayera Dhaubhadel, Gopinath Chennupati, Tanmoy Bhattacharya, and Jeff Bilmes. An effective baseline for robustness to distributional shift. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 278–285. IEEE, 2021.

[100] Sina Mohseni, Mandar Pitale, JBS Yadawa, and Zhangyang Wang. Self-supervised learning for generalizable out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5216–5223, 2020.

[101] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2018. 23, 84

[102] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto. Generative probabilistic novelty detection with adversarial autoencoders. *Advances in neural information processing systems*, 31, 2018. 23, 85

[103] Weiming Hu, Jun Gao, Bing Li, Ou Wu, Junping Du, and Stephen Maybank. Anomaly detection using local kernel density estimation and context-based regression. *IEEE Transactions on Knowledge and Data Engineering*, 32(2):218–233, 2018. 23, 85

[104] Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. Panda: Adapting pretrained features for anomaly detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2806–2814, 2021. 23, 85

[105] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018. 23, 85

[106] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. *arXiv preprint arXiv:2204.06507*, 2022. 23, 85

[107] Alireza Zaeemzadeh, Niccolo Bisagno, Zeno Sambugaro, Nicola Conci, Nazanin Rahnavard, and Mubarak Shah. Out-of-distribution detection using union of 1-dimensional subspaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9452–9461, 2021.

[108] Yu Shu, Yemin Shi, Yaowei Wang, Tiejun Huang, and Yonghong Tian. P-odn: Prototype-based open deep network for open set recognition. *Scientific reports*, 10(1):1–13, 2020. 23, 85

[109] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024. 24, 26

[110] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022. 24

[111] Frederick Jelinek. *Statistical methods for speech recognition*. MIT press, 1998. 25

[112] Ronald Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278, 2000. 25

[113] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013. 25

[114] Stefan Kombrink, Tomas Mikolov, Martin Karafiát, and Lukás Burget. Recurrent neural network based language modeling in meeting recognition. In *Interspeech*, volume 11, pages 2877–2880, 2011. 25

[115] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000. 25

[116] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013. 25

[117] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. 26

[118] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. 26

[119] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020. 26

[120] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 26

[121] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 26, 107, 116

[122] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. 26

[123] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 26

[124] Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. Demystifying prompts in language models via perplexity estimation. *arXiv preprint arXiv:2212.04037*, 2022. 27, 75

[125] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.759. URL `https://aclanthology.org/2022.emnlp-main.759`. 27, 28, 60, 74, 103

[126] Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*, 2022. 27, 58, 75

[127] Taylor Sorensen, Joshua Robinson, Christopher Rytting, Alexander Shaw, Kyle Rogers, Alexia Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. An information-theoretic approach to prompt engineering without ground truth labels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 819–862, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.60. URL `https://aclanthology.org/2022.acl-long.60`. 27, 75

[128] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*, 2023. 27, 58, 63, 75

[129] Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, GUOTONG XIE, Xiaoling Wang, and Xipeng Qiu. Unified demonstration retriever for in-context learning. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023. 27

[130] Yinpeng Liu, Jiawei Liu, Xiang Shi, Qikai Cheng, and Wei Lu. Let's learn step by step: Enhancing in-context learning ability with curriculum learning. *arXiv preprint arXiv:2402.10738*, 2024. 27

[131] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022. 28, 60, 74

[132] Vipula Rawte, Amit Sheth, and Amitava Das. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*, 2023. 28

[133] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023. 28

[134] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021. 28

[135] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.556. URL https://aclanthology.org/2022.acl-long.556. 28, 103, 121

[136] Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. Mitigating label biases for in-context learning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14014–14031, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.783. URL https://aclanthology.org/2023.acl-long.783. 29, 30, 104, 108, 115, 121, 130

[137] Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*, 2023. 29, 104, 110, 121

[138] Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*, 2023. 29, 121

[139] Zhixiong Han, Yaru Hao, Li Dong, Yutao Sun, and Furu Wei. Prototypical calibration for few-shot learning of language models. In *The Eleventh International Conference on Learning Representations*, 2023. 30, 104, 115, 121, 130

[140] Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine A Heller, and Subhrajit Roy. Batch calibration: Rethinking calibration for in-context learning and prompt engineering. In *The Twelfth International Conference on Learning Representations*, 2024. 30

[141] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html. 32, 104, 105, 106, 121

[142] Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. In *The Eleventh International Conference on Learning Representations*, 2022. 121

[143] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446. URL https://aclanthology.org/2021.emnlp-main.446. 32, 33, 105, 106, 107, 121

[144] Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. Analyzing transformers in embedding space. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16124–16170, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.893. URL https://aclanthology.org/2023.acl-long.893. 32, 33, 104, 106, 107, 121

[145] Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.3. URL https://aclanthology.org/2022.emnlp-main.3. 33, 106

[146] Nostalgebraist. Interpreting gpt: the logit lens, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. 33, 107, 121

[147] Michael Hanna, Ollie Liu, and Alexandre Variengien. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 76033–76060. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/efbba7719cc5172d175240f24be11280-Paper-Conference.pdf. 33, 107

[148] Qinan Yu, Jack Merullo, and Ellie Pavlick. Characterizing mechanisms for factual recall in language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9924–9959, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.615. URL https://aclanthology.org/2023.emnlp-main.615. 33, 107

[149] Shasha Liao and Ralph Grishman. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://aclanthology.org/P10-1081. 37

[150] Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1034. URL https://aclanthology.org/N16-1034. 40

[151] Xiao Liu, Zhunchen Luo, and Heyan Huang. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1156. URL https://aclanthology.org/D18-1156.

[152] Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1522. URL https://aclanthology.org/P19-1522.

[153] Xinya Du and Claire Cardie. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in*

Natural Language Processing (EMNLP), pages 671–683, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.49. URL https://aclanthology.org/2020.emnlp-main.49. 69, 70, 72

[154] Kaiwen Wei, Xian Sun, Zequn Zhang, Jingyuan Zhang, Guo Zhi, and Li Jin. Trigger is not sufficient: Exploiting frame-aware knowledge for implicit event argument extraction. 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 4672–4682, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.360. URL https://aclanthology.org/2021.acl-long.360.

[155] Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. CLEVE: Contrastive Pre-training for Event Extraction. 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 6283–6297, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.491. URL https://aclanthology.org/2021.acl-long.491. 40

[156] Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.42. URL https://aclanthology.org/2021.acl-short.42. 37

[157] Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction. 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 337–346, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1032. URL https://aclanthology.org/D19-1032. 37, 39, 40, 53

[158] Xinya Du and Claire Cardie. Document-level event role filler extraction using multi-granularity contextualized encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8010–8020, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.714. URL https://aclanthology.org/2020.acl-main.714. 40, 49, 51, 58, 72

[159] X. Du, Alexander M. Rush, and Claire Cardie. Document-level event-based extraction using generative template-filling transformers. In *EACL*, 2021. 49, 50, 51, 53

[160] Dongfang Lou, Zhilin Liao, Shumin Deng, Ningyu Zhang, and Huajun Chen. MLBiNet: A cross-sentence collective event detection network. 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 4829–4839, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.373. URL https://aclanthology.org/2021.acl-long.373. 40

[161] Sha Li, Heng Ji, and Jiawei Han. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.69. URL https://aclanthology.org/2021.naacl-main.69. 40, 68, 72

[162] Kung-Hsiang Huang and Nanyun Peng. Document-level event extraction with efficient end-to-end learning of cross-event dependencies. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 36–47, Virtual, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.nuse-1.4. URL https://aclanthology.org/2021.nuse-1.4. 40, 41

[163] Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. Document-level event extraction via parallel prediction networks. 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 6298–6308, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.492. URL https://aclanthology.org/2021.acl-long.492. 39, 40, 41

[164] Wasi Uddin Ahmad, Nanyun Peng, and Kai-Wei Chang. Gate: Graph attention transformer encoder for cross-lingual relation and event extraction. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*, 2021.

[165] Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.718. URL https://aclanthology.org/2020.acl-main.718. 37, 68, 69, 74

[166] Sanghamitra Dutta, Liang Ma, Tanay Kumar Saha, Di Liu, Joel Tetreault, and Alejandro Jaimes. GTN-ED: Event detection using graph transformer networks. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, pages 132–137, Mexico City, Mexico, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.textgraphs-1.13. URL https://aclanthology.org/2021.textgraphs-1.13. 40

[167] Bharat Richhariya and Muhammad Tanveer. A reduced universum twin support vector machine for class imbalance learning. *Pattern Recognition*, 102:107150, 2020. 41

[168] Chien-Liang Liu, Wen-Hoar Hsaio, Chia-Hoang Lee, Tao-Hsing Chang, and Tsung-Hsun Kuo. Semi-supervised text classification with universum learning. *IEEE transactions on cybernetics*, 46(2):462–473, 2015. 41

[169] Yanshan Xiao, Junyao Feng, and Bo Liu. A new transductive learning method with universum data. *Applied Intelligence*, pages 1–13, 2021. 41

[170] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4393–4402. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/ruff18a.html`. 42

[171] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International Conference on Pattern Recognition*, pages 475–489. Springer, 2021. 42

[172] Radu Tudor Ionescu, Fahad Shahbaz Khan, Mariana-Iuliana Georgescu, and Ling Shao. Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7842–7851, 2019. 42

[173] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016. 42

[174] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 42

[175] Mehadi Hassen and Philip K Chan. Learning a neural-network-based representation for open set recognition. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pages 154–162. SIAM, 2020. 42

[176] Leo GM Noordman and Wietske Vonk. Memory-based processing in understanding causal information. *Discourse Processes*, 26(2-3):191–212, 1998. 44

[177] Bertram Opitz and Angela D Friederici. Brain correlates of language learning: the neuronal dissociation of rule-based versus similarity-based learning. *Journal of Neuroscience*, 24(39):8436–8440, 2004. 44

[178] Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. A general framework for information extraction using dynamic span graphs. 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 3036–3046, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1308. URL https://aclanthology.org/N19-1308. 45, 52

[179] Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay. GraphIE: A graph-based framework for information extraction. 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 751–761, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1082. URL https://aclanthology.org/N19-1082. 45, 52

[180] Joseph Fisher and Andreas Vlachos. Merge and label: A novel neural network architecture for nested NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5840–5850, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1585. URL https://aclanthology.org/P19-1585. 45

[181] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020. doi: 10.1162/tacl_a_00300. URL https://aclanthology.org/2020.tacl-1.5. 45

[182] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 45

[183] Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. Simple and effective text matching with richer alignment features. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4699–4709, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1465. URL https://aclanthology.org/P19-1465. 48

[184] David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. Entity, relation, and event extraction with contextualized span representations. 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 5784–5789, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1585. URL https://aclanthology.org/D19-1585. 49, 51

[185] Hui Li, Xin Zhao, Lin Yu, Yixin Zhao, and Jie Zhang. Deedp: Document-level event extraction model incorporating dependency paths. *Applied Sciences*, 13(5): 2846, 2023. 50

[186] Hao Wang, Miao Li, Jianyong Duan, Li He, and Qing Zhang. Document-level event role filler extraction using key-value memory network. *Applied Sciences*, 13(4):2724, 2023. 50, 51

[187] Yubing Ren, Yanan Cao, Ping Guo, Fang Fang, Wei Ma, and Zheng Lin. Retrieve-and-sample: Document-level event argument extraction via hybrid retrieval augmentation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 293–306, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.17. URL https://aclanthology.org/2023.acl-long.17. 58

[188] Amir Pouran Ben Veyseh, Minh Van Nguyen, Franck Dernoncourt, Bonan Min, and Thien Nguyen. Document-level event argument extraction via optimal transport. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1648–1658, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.130. URL https://aclanthology.org/2022.findings-acl.130. 74

[189] Hanzhang Zhou and Kezhi Mao. Document-level event argument extraction by leveraging redundant information and closed boundary loss. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3041–3052, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.222. URL https://aclanthology.org/2022.naacl-main.222. 58, 74

[190] Yiming Zhang, Shi Feng, and Chenhao Tan. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.622. 58

[191] Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. Label words are anchors: An information flow perspective for understanding in-context learning. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9840–9855, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.609. URL https://aclanthology.org/2023.emnlp-main.609. 58, 116

[192] Gerd Gigerenzer and Wolfgang Gaissmaier. Heuristic decision making. *Annual review of psychology*, 62:451–482, 2011. 59, 66, 67, 74

[193] Robin M Hogarth and Natalia Karelaia. Heuristic and linear models of judgment: Matching rules and environments. *Psychological review*, 114(3):733, 2007. 59, 74

[194] Gal Shachaf, Alon Brutzkus, and Amir Globerson. A theoretical analysis of fine-tuning with linear teachers. *Advances in Neural Information Processing Systems*, 34:15382–15394, 2021. 59

[195] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of big data*, 2(1):1–21, 2015. 59

[196] Jiashuo Sun, Yi Luo, Yeyun Gong, Chen Lin, Yelong Shen, Jian Guo, and Nan Duan. Enhancing chain-of-thoughts prompting with iterative bootstrapping in large language models. *arXiv preprint arXiv:2304.11657*, 2023. 63

[197] Silviu Pitis, Michael R Zhang, Andrew Wang, and Jimmy Ba. Boosted prompt ensembles for large language models. *arXiv preprint arXiv:2304.05970*, 2023. 63

[198] Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. Active prompting with chain-of-thought for large language models. *arXiv preprint arXiv:2302.12246*, 2023. 63, 64, 67

[199] Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021. 64, 65

[200] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023. 67

[201] Dedre Gentner and Linsey A Smith. Analogical learning and reasoning. In *The Oxford Handbook of Cognitive Psychology*. Oxford University Press, 2013. 67

[202] Brian H Ross. This is like that: The use of earlier problems and the separation of similarity effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 13(4):629, 1987. 67

[203] Dedre Gentner and Kenneth D Forbus. Computational models of analogy. *Wiley interdisciplinary reviews: cognitive science*, 2(3):266–276, 2011. 68

[204] Dedre Gentner and Arthur B Markman. Structure mapping in analogy and similarity. *American psychologist*, 52(1):45, 1997. 68

[205] Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers), pages 6759–6774, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.466. URL https://aclanthology.org/2022.acl-long.466. 68, 69, 70, 72, 74

[206] Runxin Xu, Peiyi Wang, Tianyu Liu, Shuang Zeng, Baobao Chang, and Zhifang Sui. A two-stream AMR-enhanced model for document-level event argument extraction. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5025–5036, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.370. URL https://aclanthology.org/2022.naacl-main.370. 69, 70

[207] Xianjun Yang, Yujie Lu, and Linda Petzold. Few-shot document-level event argument extraction. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8029–8046, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.446. URL https://aclanthology.org/2023.acl-long.446. 69, 70, 71

[208] Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David Sontag. Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1998–2022, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.130. 69, 70, 71, 74

[209] MeiHan Tong, Bin Xu, Shuai Wang, Meihuan Han, Yixin Cao, Jiangqi Zhu, Siyu Chen, Lei Hou, and Juanzi Li. DocEE: A large-scale and fine-grained benchmark for document-level event extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3970–3982, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.291. URL https://aclanthology.org/2022.naacl-main.291. 68, 69, 72

[210] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075. 69, 71

[211] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. Rationale-augmented ensembles in language models. *arXiv preprint arXiv:2207.00747*, 2022. 69, 70

[212] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022. 70

[213] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. 70

[214] I-Hung Hsu, Zhiyu Xie, Kuan-Hao Huang, Prem Natarajan, and Nanyun Peng. AMPERE: AMR-aware prefix for generation-based event argument extraction model. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10976–10993, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.615. URL https://aclanthology.org/2023.acl-long.615. 74

[215] Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. Packed levitated marker for entity and relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4904–4917, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.337. URL https://aclanthology.org/2022.acl-long.337. 82

[216] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016. 86

[217] Alexander Hinneburg and Daniel A Keim. An efficient approach to clustering in large multimedia databases with noise. In *Knowledge Discovery and Datamining (KDD'98)*, pages 58–65, 1998. 87

[218] Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, Qinzhuo Wu, Zhengyan Li, Chong Zhang, Ruotian Ma, Zichu Fei, Ruijian Cai, Jun Zhao, Xingwu Hu, Zhiheng Yan, Yiding Tan, Yuan Hu, Qiyuan Bian, Zhihua Liu, Shan Qin, Bolin Zhu, Xiaoyu Xing, Jinlan Fu, Yue Zhang, Minlong Peng, Xiaoqing Zheng, Yaqian Zhou, Zhongyu Wei, Xipeng Qiu, and Xuanjing Huang. TextFlint: Unified multilingual robustness evaluation toolkit for natural 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: System Demonstrations*, pages 347–355, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-demo.41. URL https://aclanthology.org/2021.acl-demo.41. 93, 97

[219] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, number 2019, 2019. 97

[220] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and

accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019.

[221] Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. In *Proceedings of the 37th International Conference on Machine Learning*, pages 7909–7919, 2020. 97

[222] Gordon Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE transactions on information theory*, 14(1):55–63, 1968. 98

[223] Aristotelis-Angelos Papadopoulos, Mohammad Reza Rajati, Nazim Shaikh, and Jiamian Wang. Outlier exposure with confidence control for out-of-distribution detection. *Neurocomputing*, 441:138–150, 2021. 98

[224] Sughash Sharma. Applied multivariate techniques. 1996. 100

[225] Xuanli Lisa Xie and Gerardo Beni. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(08):841–847, 1991. 100

[226] Josef V Psutka and Josef Psutka. Sample size for maximum-likelihood estimates of gaussian model depending on dimensionality of pattern space. *Pattern Recognition*, 91:25–33, 2019. 102

[227] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. A survey on in-context learning, 2023. 103

[228] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Kevin Knight, Hwee Tou Ng, and Kemal Oflazer, editors, *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219855. URL `https://aclanthology.org/P05-1015`. 115

[229] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, 2004. 115

[230] Ellen M Voorhees and Dawn M Tice. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207, 2000. 115

[231] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi:

10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101. 115

[232] Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer, 2005. 115

[233] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. 115

[234] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2020. 115

[235] Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*, 2011. 115

[236] Mohammad Taher Pilehvar and Jose Camacho-Collados. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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 1267–1273, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1128. URL https://aclanthology.org/N19-1128. 115