Title	堅牢で説明可能かつ真実な法的AIシステムに向けて
Author(s)	Nguyen Minh Chau
Citation	
Issue Date	2024-09
Type	Thesis or Dissertation
Text version	ETD
URL	http://hdl.handle.net/10119/19391
Rights	
Description	Supervisor: NGUYEN, Minh Le, 先端科学技術研究科, 博士

Doctoral Dissertation

Towards Robust, Explainable, and Truthful Legal AI Systems

Nguyen Minh Chau

Supervisor : Nguyen Le Minh

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
Information Science

September, 2024

Abstract

The increasing complexity of legal problems and the vast amount of legal information necessitate the development of advanced Artificial Intelligence (AI) systems that can assist in the legal domain with robustness, explainability, and truthfulness. This dissertation, titled *Towards Robust, Explainable, and Truthful Legal Systems*, addresses two pivotal tasks in the legal field using novel AI approaches: Legal Information Retrieval and Legal Textual Entailment. Addressing these tasks is instrumental for enabling advanced applications such as automated legal question answering, legal decision support systems, and predictive analytics in the legal sector.

Legal Information Retrieval (LIR) forms a fundamental component of legal AI systems by ensuring the retrieval of comprehensive and relevant legal articles, which serve as critical inputs for subsequent tasks like Legal Textual Entailment. Our approach to LIR focuses on developing a robust retrieval model that can achieve high coverage of pertinent articles while maintaining a high level of precision. Especially, our Retrieve-Revise-Refine framework achieved the macro F2 scores of 0.8517 and 0.8069 on the COLIEE 2022 and 2023 datasets, respectively, representing improvements of 3.17% and 4.24% over previous state-of-the-art methods. The experimental results affirm that our LIR system significantly outperforms existing benchmarks by retrieving broader yet more precise sets of legal documents. This high coverage is crucial as it ensures that downstream applications, such as legal textual entailment models, are grounded on a reliable and extensive corpus of legal information, ultimately enhancing the overall performance and reliability of the AI system.

In the context of Legal Textual Entailment (LTE), the challenge extends beyond robustness to incorporate the dimensions of explainability and truthfulness. Our LTE models are designed to provide robust predictions regarding the entailment relationships between legal texts. Additionally, we place a strong emphasis on building systems that can offer natural language explanations for their decisions, thereby enhancing transparency and user trust. This ability to explain decisions is vital in the legal domain, where transparency and the rationale behind decisions are of paramount importance. Our proposed method achieved an accuracy of 76.15%, representing

a significant improvement of 8.26% over the previously established state-of-the-art benchmark. Furthermore, addressing the issue of truthfulness, we propose innovative methods aimed at reducing hallucinations and ensuring that the system's outputs remain true to the input data. The truthfulness of an AI system is particularly critical in legal applications, where inaccurate or misleading information can have severe consequences. Our proposed methods demonstrate a substantial improvement in reducing untruthful outputs, thereby enhancing the reliability of the system. Our Self-itemize method exhibits a significant enhancement in accuracy, as evidenced by a 5.50% increase. Furthermore, the truthfulness of logical reasoning has substantially improved, as indicated by an 8.30% rise in the accuracy of reasoning steps.

The rigorous experimental evaluations conducted as part of this research underscore the efficacy of our proposed approaches. The findings reveal that our LIR system sets new standards in terms of coverage and precision, positioning it as a highly effective tool for legal information retrieval. Similarly, our LTE models exhibit strong performance metrics, coupled with the ability to provide clear and accurate explanations for their predictions. The improvements in the truthfulness of the system are evident, further validating the effectiveness of our methods.

In conclusion, this dissertation makes significant contributions towards the development of robust, explainable, and truthful legal AI systems. By addressing the critical tasks of legal information retrieval and legal textual entailment with novel AI approaches, we pave the way for more advanced and reliable applications in the legal field. These advancements have the potential to transform legal processes, making them more efficient, transparent, and trustworthy, thereby contributing to the broader goal of harnessing AI for societal benefit.

Keywords: Legal Information Retrieval, Legal Textual Entailment, Language Models, Explainable Legal Systems, Truthful Legal Systems

Acknowledgements

First and foremost, I extend my most sincere gratitude to my principal advisor, Professor Nguyen Le Minh, for his unwavering encouragement, support, and kind guidance throughout my doctoral studies. His invaluable advice, insightful comments, and engaging discussions have been instrumental in advancing my research. This thesis would not have been possible without his steadfast support.

I am profoundly grateful to the members of my committee—Professor Satoshi Tojo, Professor Ken Satoh, Professor Kiyoaki Shirai, and Professor Naoya Inoue—for their constructive comments and discussions regarding my thesis.

I would like to acknowledge Professor Naoya Inoue for his pertinent advice and suggestions concerning my minor research project, as well as for the enriching discussions we have had.

I am also deeply indebted to Professor Ha Quang Thuy from the University of Engineering and Technology, Vietnam National University (Hanoi), for recommending and encouraging me to pursue my studies at JAIST.

My heartfelt gratitude goes to the staff at JAIST for creating an environment conducive to both research and personal well-being. I would like to extend special thanks to Inada-san, our lab secretary, for her invaluable assistance and support. I am immensely appreciative of all members of Nguyen's laboratory, whose camaraderie and collaboration have made my tenure at JAIST a remarkable period in my research career.

I extend my appreciation to the Vietnamese community at JAIST for their support on numerous aspects of life. I would also like to thank the members of the JAIST Badminton Club and the JAIST International Football Club for providing meaningful and engaging recreational activities post-research hours.

Lastly, I wish to express my deepest gratitude to my family, particularly my wife Thu Ha and my daughter Su Su, for their great patience and love. Their support has been the cornerstone that enabled me to complete this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The legal domain is characterized by its complexity and the voluminous amount of information involved. Legal tasks often require the integration and interpretation of large bodies of legal texts, from statutes and regulations to case law and scholarly articles. The advent of Artificial Intelligence (AI) offers unprecedented opportunities to assist legal professionals in navigating this complexity. This dissertation, titled *Towards Robust, Explainable, and Truthful Legal Systems*, focuses on leveraging AI to address two critical tasks in the legal field: Legal Information Retrieval (LIR) and Legal Textual Entailment (LTE). The goal is to develop AI systems that are not only robust but also capable of providing explanations for their decisions and operating in a truthful manner without generating hallucinations or inaccuracies.

Artificial Intelligence has made significant strides in various domains, including healthcare, finance, and natural language processing, offering tools and solutions that simplify complex tasks and enhance decision-making processes. The legal field, however, presents a unique set of challenges due to its stringent requirements for accuracy, transparency, and reliability. Legal documents are often dense, jargon-filled, and intricately related to numerous other texts, making the extraction and interpretation of relevant information a daunting endeavor. This complexity underscores the need for specialized AI systems that can handle the intricacies of legal texts with a high degree of reliability.

Legal Information Retrieval (LIR) is the first task that this dissertation addresses. LIR involves the process of identifying and retrieving relevant legal documents from vast repositories. The efficacy of a legal AI system heavily depends on its ability to access comprehensive and pertinent information. Traditional information retrieval methods often fall short in the legal domain due to the nuances of legal language and the interconnected nature of legal texts. Our approach aims to overcome these limitations by developing robust retrieval models that ensure high coverage and precision, furnishing legal professionals with accurate and extensive information necessary for informed decision-making.

Once relevant legal documents are retrieved, the next crucial step is **Legal Textual Entailment (LTE)**. This task involves determining whether a given legal hypothesis logically follows from a set of legal premises. LTE is pivotal for tasks such as automated legal reasoning, legal question answering, and predictive analytics. The challenge is not only to achieve robust predictions but also to ensure that the AI system can elucidate the reasoning behind its decisions. Given the high-stakes nature of legal decisions, it is imperative that AI systems in this domain are transparent, providing natural language explanations that can be understood by legal practitioners.

Another critical aspect that this dissertation addresses is **truthfulness**. The truthfulness of an AI system refers to the extent to which its outputs are accurate and reliably reflect the input data. In the context of legal applications, untruthful outputs—such as hallucinations where the system generates information not grounded in the input texts—can have serious repercussions, potentially leading to unjust or incorrect legal outcomes. This dissertation proposes innovative methods to enhance the truthfulness of AI systems, thereby reducing the occurrence of hallucinations and ensuring that the system's outputs are trustworthy and grounded in legitimate legal texts.

The research presented in this dissertation is driven by the overarching goal of advancing the development of AI systems that can significantly transform legal processes. By focusing on robustness, explainability, and truthfulness, this work aims to address the primary concerns of legal practitioners and stakeholders, ultimately facilitating the adoption of AI technologies in the legal field. The impact of such advancements extends beyond individual tasks, as robust and explainable AI systems can support a wide array of applications, from legal research to automated decision-making, potentially reshaping the landscape of legal practice.

In conclusion, this dissertation explores the intersection of artificial intelligence and the legal domain, addressing critical tasks that form the backbone of legal AI applications. By developing advanced methods for legal information retrieval and legal textual entailment, and by emphasizing the importance of explainability and truthfulness, this research contributes to the broader effort of making legal processes more efficient, transparent, and reliable. The subsequent sections provide a detailed account of the motivations behind this work, the contributions made, and the outcomes of our experimental evaluations, setting the stage for future innovations in the field of legal AI.

1.2 Motivations

The motivations behind this research stem from the significant challenges and opportunities present in the legal domain.

- **Volume and Complexity of Legal Information:** Legal professionals often face the challenging task of locating and interpreting relevant legal documents from massive, complex datasets. Efficient legal information retrieval systems are crucial to reduce the time and effort required for legal research and decision-making.

- **Need for Explainability:** Given the high-stakes nature of legal proceedings, AI systems must be able to provide clear, understandable rationales for their decisions. Explainability is essential for building trust and ensuring that the decisions made by AI systems are transparent and justifiable.

- **Ensuring Truthfulness:** Truthfulness refers to the accuracy and reliability of the AI system's outputs. In the legal domain, untruthful or erroneous outputs can have severe consequences, potentially leading to unjust decisions or misinformed actions. Therefore, it is imperative to develop methods that reduce hallucinations and enhance the truthfulness of AI systems.

1.3 Contributions

Overall, this dissertation makes several key contributions towards the development of robust, explainable, and truthful AI systems in the legal domain:

- **Development of a Robust Legal Information Retrieval Systems (Chapter 3 and 4):** We propose a novel approaches to legal information retrieval that achieve high coverage and precision. Our systems effectively retrieve comprehensive sets of relevant legal documents, forming a reliable foundation for subsequent legal tasks.

- **Legal Textual Entailment Models with Explanations (Chapter 5 and 6):** We develop LTE models that not only provide robust predictions but also generate natural language explanations for their decisions. These models enhance the transparency and interpretability of AI systems in the legal domain.

- **Methods to Improve Truthfulness (Chapter 7):** We introduce innovative techniques to increase the truthfulness of AI systems, reducing the incidence of hallucinations and ensuring that outputs remain accurate and reliable.

- **Comprehensive Experimental Evaluation:** The effectiveness of our approaches is demonstrated through rigorous experimental evaluations. The results show significant improvements in terms of coverage, precision, explainability, and truthfulness, setting new benchmarks in the legal AI field.

The overarching objective of this dissertation has been to advance the development of robust, explainable, and truthful AI systems within the legal domain. The work presented herein has meticulously addressed two critical tasks: Legal Information Retrieval (LIR) and Legal Textual Entailment (LTE). Through novel methodologies and comprehensive experimental evaluations, this research has demonstrated significant improvements in performance, explainability, and truthfulness.

In Chapter 3, we focused on Legal Information Retrieval tailored to statute law, particularly for legal bar exam queries. Traditional methods fell short in achieving the high precision and coverage required for these tasks. By addressing two distinct challenges—the abstract language of legal articles versus the scenario-specific language of queries and the handling of long documents—we developed a system that achieved an F2 score of 76.87%, representing a 3.85% improvement over previous state-of-the-art methods. This chapter underscored the importance of specialized models in overcoming domain-specific challenges, setting a new standard for future research in legal information retrieval.

Chapter 4 introduced a specialized three-stage framework for retrieving entailing legal article sets, focusing on yielding concise sets of relevant articles. The Retrieve-Revise-Refine framework showcased the integration of small and large language models in a complementary manner. This chapter highlighted the shift from traditional information retrieval approaches towards more tailored solutions, resulting in improvements in macro F2 scores by 3.17% and 4.24% on the COLIEE 2022 and 2023 datasets, respectively. The detailed ablation studies provided insights into the indispensable roles of each stage, reinforcing the framework's efficacy and contribution to the retrieval task.

Chapter 5 examined the performance of ChatGPT in legal text entailment tasks and proposed leveraging label models to integrate its provisional answers into consolidated labels. By treating ChatGPT's deterministic and non-deterministic answers as noisy predictions, we managed to achieve a noteworthy accuracy of 76.15%, an improvement of 8.26% over previous benchmarks. Our analysis of instances where ChatGPT produced incorrect answers provided valuable insights for future enhancements and demonstrated the viability of employing label models for weak supervision in legal AI tasks.

In Chapter 6, we presented our approach for the VLSP 2023 Vietnamese Legal Textual Entailment Recognition (LTER) challenge, which entailed verifying if a statement could be deduced from related legal articles. Our framework employed label models to ensemble predictions from a fine-tuned Vietnamese Llama-2 model and an off-the-shelf mT0 model. This approach balanced accuracy and explanation, securing the first-place position by achieving an accuracy score of 76.98%. The success of this framework highlighted the potential of combining multiple models to enhance both predictive performance and explainability in legal AI systems.

In Chapter 7, we addressed the crucial issue of truthfulness in LLM outputs. Our Self-itemize approach involved itemizing complex legal texts prior to response generation, significantly outperforming existing methodologies by 5.5% in accuracy on the COLIEE 2022 dataset. Furthermore, this method demonstrated enhanced truthfulness by achieving higher average accuracy in reasoning steps, underscoring the potential of input itemization in mitigating comprehension challenges and improving the reliability of AI systems in the legal domain.

1.4 Structure of the Dissertation

This dissertation is organized into eight chapters, each focusing on different aspects of developing robust, explainable, and truthful legal AI systems.

- **Chapter 1: Introduction:** This chapter sets the stage by providing an overview of the research problem, motivations, and contributions of the dissertation. It outlines the importance of developing AI systems with robustness, explainability, and truthfulness in the legal domain.
- **Chapter 2: Preliminary:** This chapter reviews the background and foundational concepts essential for understanding the subsequent chapters. It also introduces the fundamental AI techniques and methodologies employed in this research.
- **Chapters 3 and 4: Development of a Robust Legal Information Retrieval Systems:** These chapters propose novel approaches to legal information retrieval that achieve high coverage and precision. Our systems effectively retrieve comprehensive sets of relevant legal documents, forming a reliable foundation for subsequent legal tasks.
- **Chapters 5 and 6: Legal Textual Entailment Models with Explanations:** These chapters develop LTE models that not only provide robust predictions but also generate natural language explanations for their decisions. These models enhance the transparency and interpretability of AI systems in the legal domain.
- **Chapter 7: Methods to Improve Truthfulness:** This chapter introduces innovative techniques to increase the truthfulness of AI systems, reducing the incidence of hallucinations and ensuring that outputs remain accurate and reliable. It provides a comprehensive

evaluation of these methods, demonstrating significant improvements in the truthfulness of the AI systems.

- **Chapter 8: Conclusion:** The final chapter summarizes the key findings and contributions of the dissertation. It discusses the implications of the research, potential applications, and avenues for future work in the development of robust, explainable, and truthful legal AI systems.

Chapter 2

Preliminary

2.1 Transformer Model

The Transformer model, introduced by Vaswani et al. in 2017, represents a monumental advance in deep learning for natural language processing tasks. Unlike recurrent neural networks (RNNs) and long short-term memory networks (LSTMs), the Transformer architecture relies entirely on a mechanism called self-attention to draw global dependencies between input and output.

Architecture Overview

The Transformer model consists of an encoder and a decoder, each composed of a stack of N identical layers. The encoder processes the input sequence and generates a set of continuous representations. The decoder takes these representations and produces the output sequence.

Encoder

Each encoder layer consists of two primary sub-layers:

1. **Multi-Head Self-Attention Mechanism**

2. **Position-wise Fully Connected Feed-Forward Network**

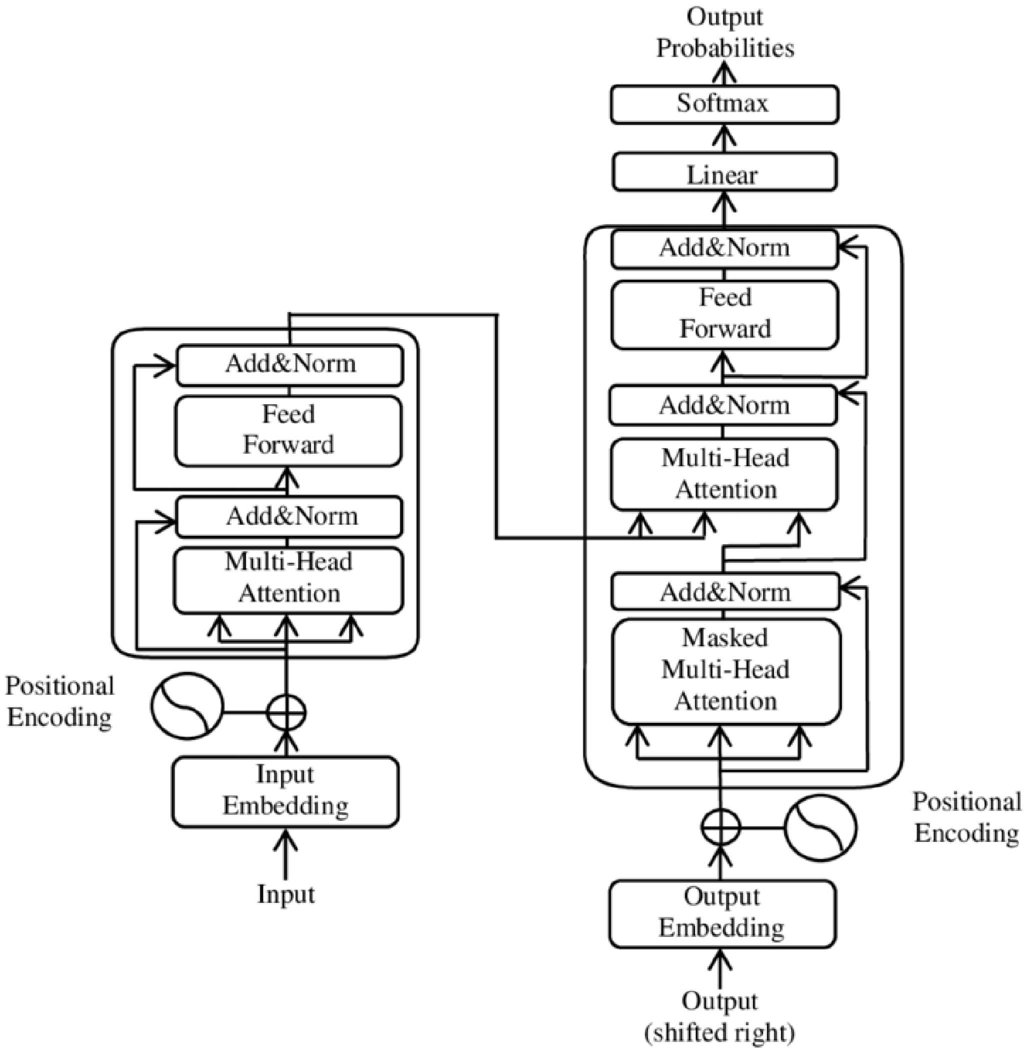

Figure 2.1: Architecture of the Transformer model.

These sub-layers are wrapped in residual connections followed by layer normalization. Formally, given an input sequence $\mathbf{X} \in \mathbb{R}^{n \times d}$, where n is the sequence length and d is the dimensionality of the input, the output of the i-th encoder layer can be represented as:

$$\mathbf{Z}^{'} = \text{LayerNorm}(\mathbf{X}^{(i)} + \text{MultiHead}(\mathbf{X}^{(i)}, \mathbf{X}^{(i)}, \mathbf{X}^{(i)})) \tag{2.1}$$

$$\mathbf{X}^{(i+1)} = \text{LayerNorm}(\mathbf{Z}^{'} + \text{FFN}(\mathbf{Z}^{'})) \tag{2.2}$$

where MultiHead denotes the multi-head self-attention mechanism, and FFN is the position-wise feed-forward network.

Decoder

The decoder also consists of N identical layers, each comprising:

1. **Masked Multi-Head Self-Attention Mechanism**
2. **Multi-Head Attention Mechanism with Encoder Outputs**
3. **Position-wise Fully Connected Feed-Forward Network**

Similar to the encoder, residual connections and layer normalization are applied. The mask in the first attention mechanism ensures that the prediction for a particular position depends only on the known outputs at that position and prior positions.

Given the target sequence embeddings $\mathbf{Y} \in \mathbb{R}^{m \times d}$ (where m is the target sequence length), and encoder output $\mathbf{H}$, a decoder layer's output can be expressed as:

$$\mathbf{Y}^{'} = \text{LayerNorm}(\mathbf{Y}^{(i)} + \text{MaskedMultiHead}(\mathbf{Y}^{(i)}, \mathbf{Y}^{(i)}, \mathbf{Y}^{(i)})) \tag{2.3}$$

$$\mathbf{Z} = \text{LayerNorm}(\mathbf{Y}^{'} + \text{MultiHead}(\mathbf{Y}^{'}, \mathbf{H}, \mathbf{H})) \tag{2.4}$$

$$\mathbf{Y}^{(i+1)} = \text{LayerNorm}(\mathbf{Z} + \text{FFN}(\mathbf{Z})) \tag{2.5}$$

Attention Mechanism

The self-attention mechanism is the core of the Transformer model. It maps a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The attention function can be described as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{2.6}$$

where $\mathbf{Q}$ (queries), $\mathbf{K}$ (keys), and $\mathbf{V}$ (values) are projections of the input representation, and d_k is the dimension of the keys.

Multi-Head Attention

Multi-head attention allows the model to focus on information from different representation subspaces at different positions. The multi-head attention computes the attention function multiple times in parallel with different parameters and concatenates the results:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}^O \tag{2.7}$$

where each $\text{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V)$, and $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, $\mathbf{W}_i^V$, and $\mathbf{W}^O$ are learned projection matrices.

Position-wise Feed-Forward Networks

Each position-wise feed-forward network is applied independently to each position and consists of two linear transformations with a ReLU activation in between:

$$\text{FFN}(x) = \max(0, x\mathbf{W}_1 + b_1)\mathbf{W}_2 + b_2 \tag{2.8}$$

where x is the input, $\mathbf{W}_1$, $\mathbf{W}_2$, b_1, and b_2 are parameters of the network.

Positional Encoding

Since the Transformer does not contain any recurrence or convolution, it needs a way to incorporate the order of the sequence. This is achieved through positional encoding, which adds a sine and cosine function of different frequencies to the input embeddings:

$$\text{PE}_{(i,2j)} = \sin\left(\frac{i}{10000^{2j/d}}\right) \tag{2.9}$$

$$\text{PE}_{(i,2j+1)} = \cos\left(\frac{i}{10000^{2j/d}}\right) \tag{2.10}$$

where i is the position and j is the dimension.

In summary, the Transformer model leverages self-attention and feed-forward layers to process sequences in a parallelized manner, effectively capturing long-range dependencies and complex patterns without the limitations of traditional sequence models such as RNNs.

2.2 Large Language Models

Large Language Models (LLMs) have become a cornerstone in the field of Natural Language Processing (NLP), exhibiting impressive capabilities in understanding and generating human language. These models are typically built on the foundation of the Transformer architecture but are scaled up in terms of parameters and training data, enabling them to perform a wide range of language tasks with high proficiency.

Definition and Characteristics

Large Language Models, such as GPT-3 (Generative Pre-trained Transformer 3) developed by OpenAI, are designed to process and generate text by learning intricate patterns in large corpora of text data. These models are characterized by:

1. **Scale:** LLMs tend to have billions or even trillions of parameters, allowing them to learn and store a vast amount of linguistic knowledge.

2. **Pre-training and Fine-tuning:** They are usually pre-trained on a large and diverse dataset in an unsupervised manner and then fine-tuned on specific tasks using supervised learning.

3. **Few-Shot and Zero-Shot Learning:** LLMs are adept at performing tasks with minimal task-specific data, leveraging their extensive pre-training to generalize well.

Training Large Language Models

The training of LLMs involves two primary phases: pre-training and fine-tuning.

Pre-training During the pre-training phase, the model learns from a large, unlabelled corpus of text. The goal is to predict the next word in a sequence, which requires understanding the context and semantics of the text. This is typically done using a language modeling objective such as the autoregressive or autoregressive masked language model objectives. Given a sequence of words $w_1, w_2, \ldots, w_n$, the model aims to maximize the likelihood:

$$\mathcal{L}_{\text{LM}} = \sum_{i=1}^{n} \log P(w_i \mid w_1, \ldots, w_{i-1}) \tag{2.11}$$

where $P(w_i \mid w_1, \ldots, w_{i-1})$ is the probability of the i-th word given the preceding words.

Fine-tuning Fine-tuning involves adapting the pre-trained model to a specific downstream task, such as sentiment analysis, question answering, or machine translation. This is done by training the model further on a smaller, task-specific dataset using supervised learning objectives. The loss function during fine-tuning is typically task-dependent.

Energy Efficiency and Computational Considerations

Training and deploying LLMs require enormous computational resources. The training process often involves thousands of GPUs or TPUs running in parallel over several weeks or months. Some considerations include:

1. **Model Parallelism:** Distributing a single model across multiple devices to manage the memory requirements.
2. **Data Parallelism:** Distributing the training data across multiple devices to accelerate the training process.
3. **Mixed Precision Training:** Using lower precision arithmetic to speed up computation and reduce memory usage without compromising model accuracy significantly.

Use Cases and Applications

LLMs have demonstrated state-of-the-art performance across a variety of NLP tasks:

1. **Text Generation:** Generating coherent and contextually relevant text, such as story writing, dialogue generation, and code generation.

2. **Machine Translation:** Translating text from one language to another with high accuracy.
3. **Question Answering:** Understanding and answering questions posed in natural language.
4. **Summarization:** Creating concise and meaningful summaries of longer texts.
5. **Sentiment Analysis:** Determining the sentiment expressed in a piece of text.

Ethical Considerations

The deployment of LLMs raises important ethical concerns:

1. **Bias and Fairness:** LLMs can inherit biases present in the training data, which can lead to unfair or discriminatory outcomes.
2. **Misinformation:** LLMs can generate plausible text that is factually incorrect, leading to the spread of misinformation.
3. **Privacy:** Training data may contain sensitive information, raising concerns about data privacy and security.
4. **Environmental Impact:** The computational resources required for training LLMs contribute to significant energy consumption and carbon footprint.

Addressing these issues requires ongoing research into fair and transparent model training methods, as well as guidelines for responsible usage and deployment.

In conclusion, Large Language Models represent a significant leap in NLP capabilities, enabling advanced language understanding and generation tasks. However, they also bring new challenges that need to be carefully managed to harness their potential responsibly.

Chapter 3

Legal Information Retrieval: A Divide-and-Conquer Approach

3.1 Overview

The information retrieval (IR) task for statute law requires a system to identify and extract relevant legal articles given a query of a legal bar examination. Approaches utilizing Transformer architecture have demonstrated significant advancements over traditional machine learning and IR methods when applied to legal documents. However, existing methods predominantly focus on domain adaptation rather than addressing intrinsic challenges particular to legal query and document characteristics. This paper identifies two such challenges and offers methods to address them effectively. The first challenge involves the variance in language (with articles written in an abstract form and queries often describing a specific scenario), which is tackled using a specialized model. The second challenge concerns to the length of documents and queries, addressed using another specialized model. Our experimental results indicate that our proposed system outperforms the previous best system, achieving a state-of-the-art F2 score of 76.87%, an enhancement of 3.85%.

3.2 Introduction

The Competition on Legal Information Extraction/Entailment (COLIEE) is a pioneering international competition focused on legal text processing, held annually since 2014. In 2021, COLIEE featured five tasks encompassing both case law and statute law. This study is focused on Task 3: information retrieval concerning statute law (defined as the written law enacted by a legislative body). The objective is to retrieve all relevant articles within a corpus for a given legal query as illustrated in Table 3.1. This chapter illustrates our methodology for addressing the information retrieval task concerning the English version of the Japanese Civil Code corpus.

During the initial stages of the competition, traditional machine learning and natural language processing techniques were prevalently used for Task 3 of COLIEE: Support Vector Machines (SVMs) (Hearst et al., 1998), BM25, TF-IDF, n-gram, Hidden Markov Model (HMM), and Random Forests. Subsequent years saw the introduction of Deep Learning approaches such as the employment of Long Short-Term Memory (LSTM) by (Kim et al., 2017) and the use of Recurrent Neural Networks (RNN) for text representation by SPABS (Yoshioka et al., 2018). Following the debut of BERT (Devlin et al., 2019), its robustness led to widespread adoption among participants who combined its retrieval results with those from well-known IR methods (TF-IDF, BM25, Indri (Strohman et al., 2005), or Word Movers' Distance (Kusner et al., 2015)). Despite significant improvements with Transformer models in Task 3, their application primarily focuses on domain adaptation without adequately addressing the peculiarities of legal documents and queries or providing appropriate rationale for the various pre-trained Transformer-based model combinations.

Through our analysis, we identify two key characteristics pertinent to legal texts in the statute law IR task that should be considered. Legal articles are inherently abstract, and legal bar queries can be either abstract or narrative of specific scenarios (illustrated in Table 3.1). Consequently, a single model may struggle to generalize the abstract relationship between legal articles and queries. To address this, we propose a method to identify specific-scenario queries and employ a specialized Transformer-based model for them, alongside another model for all queries. During inference, we ensemble the results from both models. Our experimental results show an increase of 3.78% in the F2 score with this approach for a Transformer-based model.

Table 3.1: Examples of training data for Task 3. Example 1 is about an abstract query and example 2 is about a specific-scenario query.

#	Query	Relevant Law Articles
1	Juristic act subject to a condition subsequent which is impossible shall be void.	**Article 133** (1) A juridical act subject to an impossible condition precedent is void. (2) A juridical act subject to an impossible condition subsequent is an unconditional juridical act.
2	The family court appointed B as the administrator of the property of absentee A, as A went missing without appointing the one. In cases where A owns land X, B needs to obtain the permission of a family court in order to sell Land X as an agent of A.	**Article 28** If an administrator needs to perform an act exceeding the authority provided for in Article 103, the administrator may perform that act after obtaining the permission of the family court. [...] **Article 103** An agent who has no specifically defined authority has the authority to perform the following acts only: (i) acts of preservation; [...]

Additionally, we note the presence of relatively long legal articles and queries (See Figures 3.1 and 3.2). Thus, we utilize Longformer (Beltagy et al., 2020), which can handle sequences exceeding the 512-token limit that constrains most off-the-shelf pre-trained models. This application adds a further 1.28% improvement in the F2 score. Combining our two proposals, we achieve a leading performance for COLIEE 2021 Task 3 with a 76.87% F2 score, marking a 3.85% improvement over the previous best system (Wehnert et al., 2021a).

Our contributions in this paper are:

- A method to identify and handle specific-scenario queries using Transformer-based models.
- Employment of Longformer to manage lengthy articles and queries.
- An extensive analysis of the impact of our proposed methods on system performance.

The structure of the chapter is as follows. Section 3 reviews related work. Section 4 details the system design. Section 5 discusses the experimental results. Section 6 concludes our work.

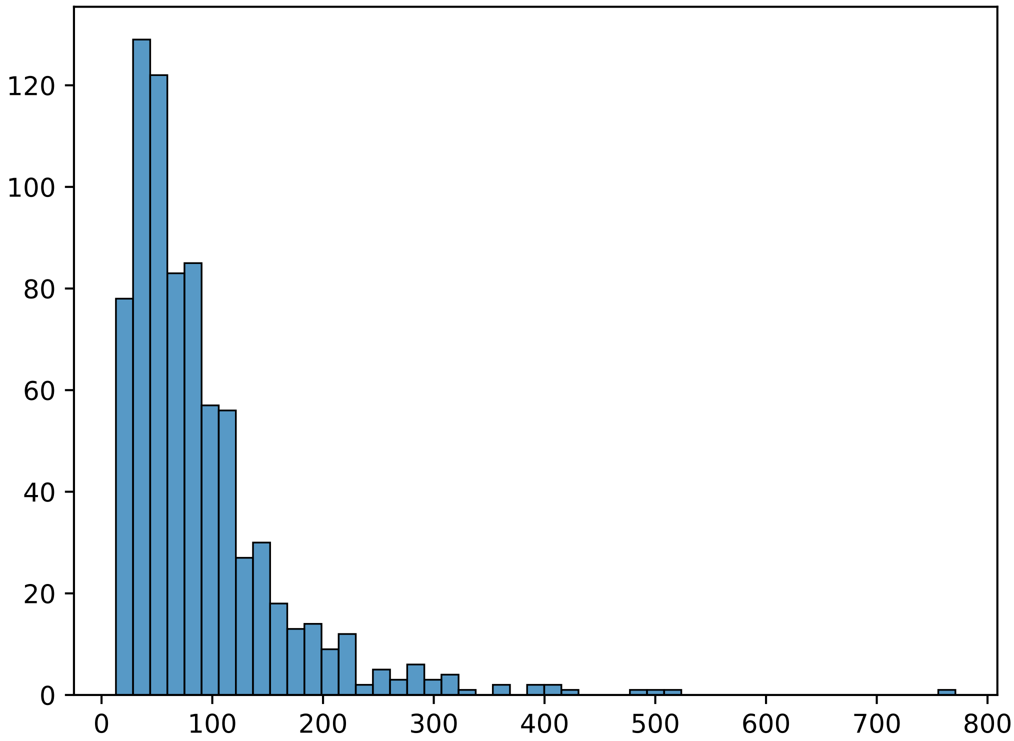

Figure 3.1: Distribution of number of tokens of Civil Code articles

3.3 Related Work

3.3.1 The COLIEE Competition

Tasks in COLIEE

In 2021, the COLIEE competition featured five tasks on case law and statute law. The case law segment included Task 1 (case law information retrieval) and Task 2 (entailment judgment between provided cases and an unseen case). The statute law segment encapsulated Task 3 (information retrieval for statute law), Task 4 (yes/no decision making concerning the entailment relationship between the retrieved articles and the query), and Task 5 (yes/no QA without retrieved articles from Task 3).

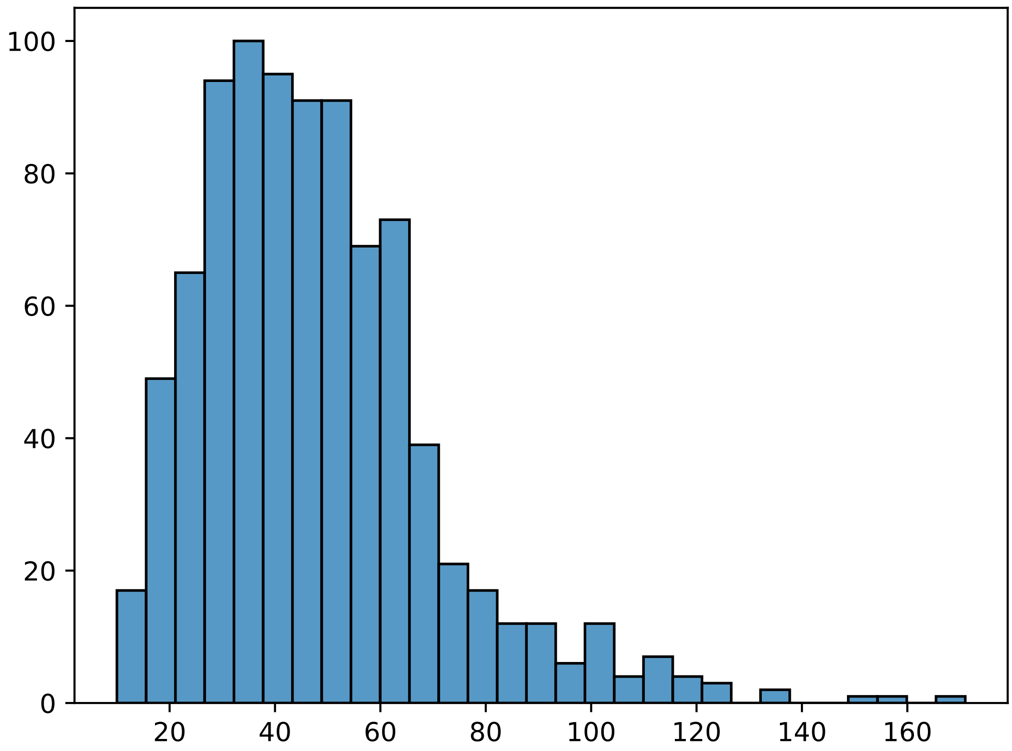

Figure 3.2: Distribution of number of tokens of queries

COLIEE Task 3

The goal of the statute law retrieval task (Task 3 in COLIEE) is to find relevant articles in a corpus that support a given legal bar exam query. The dataset consists of the Japanese Civil Code (also available in English) comprising 768 articles. The task input is a legal bar exam query Q with the system needing to identify a subset of Civil Code articles $A_{i|1\leq i\leq n}$ that can be used to determine a yes/no decision for the query, formulated as Entails($A_{i|1\leq i\leq n}$, Q) or Entails($A_{i|1\leq i\leq n}$, *not* Q). Given the retrieval task's emphasis on recall, the macro F2 score is used as the evaluation metric.

3.3.2 Transformer-based pre-trained models

The Transformer is a deep learning model that employs the self-attention mechanism to assess the significance of each component of the input data. BERT, a language model, utilizes the encoder architecture of the Transformer along with pre-training tasks to comprehend language patterns across extensive corpora. Although the pre-training phase demands substantial time and computational resources, a pre-trained BERT model facilitates the fine-tuning process on considerably smaller datasets, utilizing the acquired language representation for domain adaptation. Numerous BERT variants have been developed to address various challenges, including enhancements to the BERT training phase such as RoBERTa (Liu et al., 2019a) and AlBERT (Lan et al., 2020); efforts to reduce BERT's computational demands, such as DistilBERT (Sanh et al., 2019); models trained for specific domains, for instance, LegalBERT (Chalkidis et al., 2020) and BERTLAW[1] for the legal domain, and PubMedBERT (Gu et al., 2021) and BioBERT (Lee et al., 2019) for the biological domain; and solutions for processing extended documents, such as Longformer (Beltagy et al., 2020).

LegalBERT model

LegalBERT is a variation of the BERT model, pre-trained to support natural language processing within the legal domain, including computational law and legal technology applications. Instead of generic texts, it is pre-trained on various legal documents, including contracts, court cases, and legislation. This specialization allows it to outperform the original BERT in legal domain tasks.

Longformer model

Traditional BERT models are limited to processing sequences of up to 512 tokens. Longformer (Beltagy et al., 2020) addresses this limitation by offering an attention mechanism that scales linearly with sequence length, accommodating inputs exceeding 4,096 tokens. This capability is advantageous for statute law retrieval where concatenated query-article pairs often exceed 512 tokens.

[1] https://huggingface.co/nguyenthanhasia/BERTLaw

3.4 Methodology

This section outlines our system's methods. We begin by reformulating the retrieval task as a binary classification task. We then describe methods for handling specific-scenario queries and addressing long-text challenges, followed by our ensembling strategy for retrieval results (Figure 3.3).

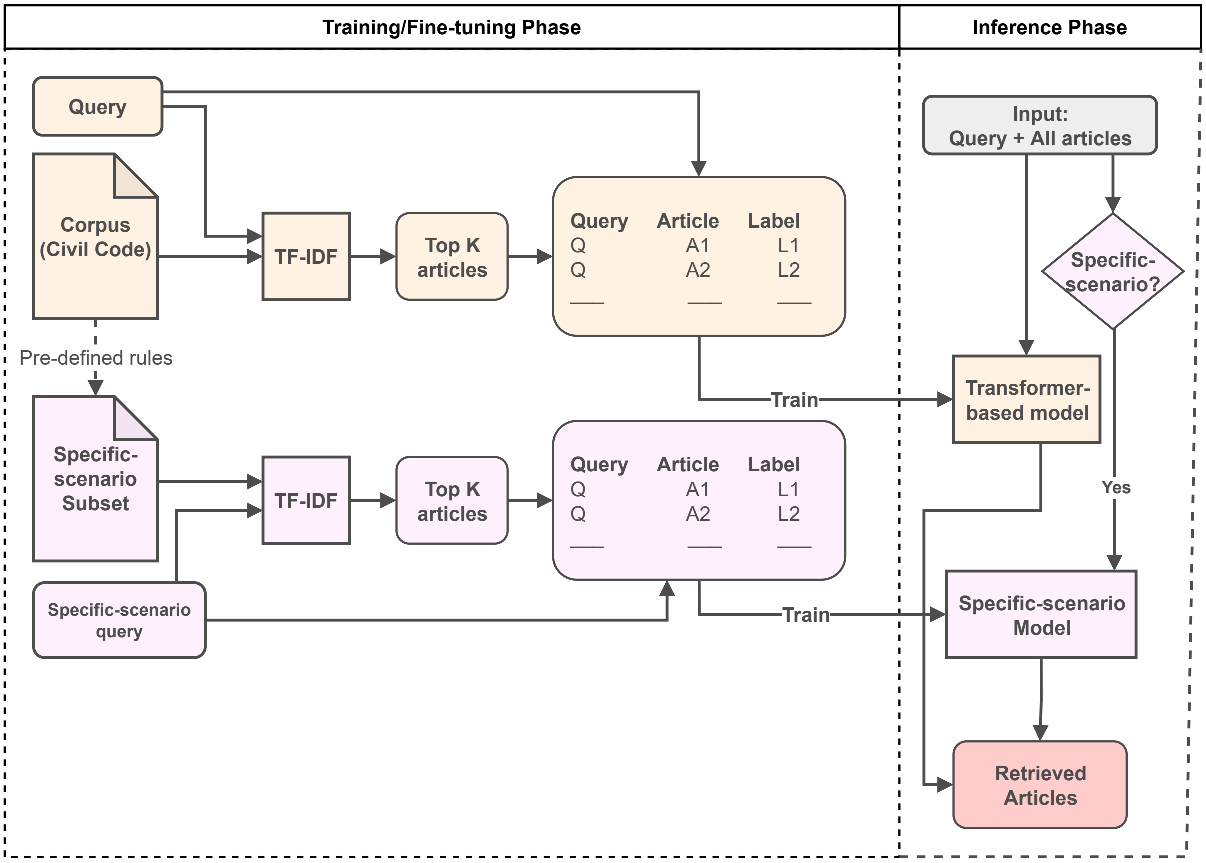

Figure 3.3: The proposed system. We use 2 Transformer-based models which are LegalBERT and Longformer in the experiments.

3.4.1 Fine-tuning a Transformer-based Model

This subsection elaborates on the procedures involved in data preparation and fine-tuning for a Transformer-based model aimed at statute law retrieval tasks. This methodology is applicable to any Transformer-based model, but given the legal context of our documents, we start with the pre-trained LegalBERT as our base model. LegalBERT, tuned for legal datasets, has demonstrated effectiveness across various legal processing tasks (Silveira et al., 2021; Chalkidis et al., 2021a).

The objective here is to identify relevant articles corresponding to a specific query. Essentially, for each article in the dataset, we need to ensure whether there is an entailment relationship between the query and the article. This transforms the retrieval task into a sentence pair classification problem, where the input consists of a query-article pair and the output label is binary (1/0), indicating the presence or absence of an entailment relationship. Specifically, the input is structured as *"[CLS]<tokens of query>[SEP]<tokens of article>"* with *[CLS]* and *[SEP]* serving as special tokens as described by (Devlin et al., 2019). Following the methodology in (Wehnert et al., 2021a), articles are represented not only by their textual content but also by their structural metadata (examples provided in Table 3.2).

For fine-tuning, both positive and negative samples are requisite. Positive samples are derived from the dataset, where each query typically corresponds to 1 or 2 relevant articles, though a maximum of six can exist rarely. For negative samples, we first convert the query and all articles into TF-IDF vectors and compute similarity scores. Subsequently, we select the top K articles with the highest TF-IDF similarity (excluding those labeled as relevant) as negative samples. This strategy aims to train the Transformer-based model to discern patterns that differentiate between highly similar but non-relevant articles. In our experiments, K is set to 100.

During the inference phase, we combine each query with each article as input. The model then outputs a yes/no prediction to determine the relevance of the article to the query. If the model predicts 0 relevant articles for a query, the article with the highest confidence score for being "relevant" is selected as the retrieved relevant article.

3.4.2 Identifying Specific-Scenario Queries and Training Specific-Scenario Models

Legal language generally possesses an abstract nature, particularly evident in the articles of the Civil Code. However, legal bar queries can vary; some are abstract, while others describe concrete scenarios (see Table 3.1). Specific-scenario queries require the capability to map real-life scenarios to relevant abstract legal articles. Lexical-based approaches often falter with these

Table 3.2: Examples on representation of articles. The *italic texts* indicates article's metadata.

#	Article representation
1	*Part I. General Provisions* *Chapter II. Persons* *Section 3. Capacity to Act* *Summary: Persons under Assistance; Assistants* *Content:* Article 16. A person subject to a decision for commencement of assistance becomes a person under assistance, and an assistant is appointed for that person.
2	*Part III. Claims* *Chapter V. Torts* *Summary: Capacity for Liability* *Content:* Article 713. A person who has inflicted damage on another person while in a condition wherein the person lacked the capacity to appreciate their own liability for their acts due to a mental disability is not liable to compensate for this; provided, however, that this does not apply if the person has temporarily caused that condition, intentionally or negligently.

queries due to variations in descriptive terms. Deep Learning models, given sufficient data, can generalize to handle specific-scenario queries, in addition to abstract ones. However, given the limited data in Task 3, the generalization ability of a Deep Learning model can be constrained if it must address all query types simultaneously. Consequently, we propose filtering and specially handling specific-scenario queries.

To identify specific-scenario queries, we observe that these queries often utilize a template style, capitalizing legal persons and objects. For instance, in example 2 of Table 3.1, the query "The family court appointed B as the administrator of the property of absentee A, as A went missing without appointing one. In cases where A owns land X, B needs to obtain the permission of a family court in order to sell Land X as an agent of A." uses uppercase letters to denote *the absentee* as A, *the appointed administrator* as B, and *A's land* as X. We implement rules to determine whether uppercase letters in queries represent legal persons or objects, excluding A and I to prevent errors.

Subsequently, specific-scenario queries and their relevant articles are separately processed using a Transformer-based model. We prepare fine-tuning data following the procedure described earlier and apply it to a LegalBERT model already fine-tuned on all Task 3 data, rather than just the pre-trained LegalBERT model. The rationale is that the fine-tuned LegalBERT model not only adapts to the legal domain but also to the specific statute law retrieval task across various queries, thus serving as an ideal starting point for fine-tuning on specific-scenario data.

3.4.3 Handling Long Articles and Queries

A notable limitation of many pre-trained Transformer-based models (including LegalBERT) is the maximum token limit, generally set at 512 tokens. Given that a large number of legal articles in the Civil Code are lengthy, concatenating a query with an article as input for LegalBERT often results in truncation of the latter portions of the article (refer to Figures 3.1 and 3.2 for token statistics of queries and articles). Consequently, the LegalBERT model struggles to learn the relevance patterns between the query and the article if the relevant segment is truncated. Furthermore, such truncation converts positive samples into noisy positive samples, potentially impairing the model's ability to generalize effectively.

To address this, we propose the adoption of the Longformer model for processing inputs with extended lengths. Unlike LegalBERT, Longformer can manage inputs up to 4,096 tokens. We fine-tune the pre-trained Longformer model similarly to LegalBERT, including the addition of metadata to articles and the preparation of positive and negative samples.

3.4.4 Ensemble the Retrieval Results

A past system Nguyen et al. (2021a) demonstrated that ensembling retrieval results from multiple models can enhance recall, subsequently improving the F2 score. We apply this approach to our final retrieval results, ensembling outputs from four models: two LegalBERT models (fine-tuned on distinct datasets: (1) the complete dataset and (2) scenario-specific data) and two Longformer models (fine-tuned similarly). The final retrieval results are thus an ensemble from these four models, accommodating different data characteristics.

Table 3.3: Number of queries and specific-scenario queries

Data	# queries	# specific-scenario queries
Training	748	103
Validation	58	16
Test	87	31

Table 3.4: Performance of the systems. The first row indicates the proposed system. The following lines indicate the participants' systems in Task 3 (COLIEE 2021). **Bold texts** indicate highest results. Underlined texts indicate second-highest results.

System	F2 score (%)	Precision (%)	Recall (%)
Ours	**76.87**	65.80	**85.19**
OvGU_run1 (Wehnert et al., 2021a)	<u>73.02</u>	67.49	77.78
JNLP.CrossLMultiLThreshold (Nguyen et al., 2021a)	72.27	60.00	<u>80.25</u>
BM25.UA (Kim et al., 2021b)	70.92	**75.31**	70.37
JNLP.CrossLBertJP (Nguyen et al., 2021a)	70.90	62.41	77.16
R3.LLNTU	70.47	66.56	74.38
R2.LLNTU	70.39	67.70	73.15
R1.LLNTU	68.75	63.68	73.15
JNLP.CrossLBertJPC15030C15050 (Nguyen et al., 2021a)	68.38	55.35	77.78
OvGU_run2 (Wehnert et al., 2021a)	67.17	48.57	<u>80.25</u>
TFIDF.UA (Kim et al., 2021b)	65.71	<u>67.90</u>	65.43
LM.UA (Kim et al., 2021b)	54.60	56.79	54.32
TR_HB (Schilder et al., 2021a)	52.26	33.33	61.73
HUKB-3 (Yoshioka et al., 2021b)	52.24	29.01	69.75
HUKB-1 (Yoshioka et al., 2021b)	47.32	23.97	65.43
TR_AV1 (Schilder et al., 2021a)	35.99	26.22	51.23
TR_AV2 (Schilder et al., 2021a)	33.69	14.90	55.56
HUKB-2 (Yoshioka et al., 2021b)	32.58	32.72	32.72
OvGU_run3 (Wehnert et al., 2021a)	30.16	15.70	70.06

3.5 Experiments

3.5.1 Dataset

The dataset for Task 3 contains 768 articles from the Civil Code. Table 3.3 provides information on the number of queries and corresponding scenario-specific queries in the training, validation, and test sets.

3.5.2 Experimental results

Table 3.4 shows the performance of our models on the Task 3 dataset. Our model surpasses the best system in the competition (Wehnert et al., 2021a) by 3.85%, demonstrating superior recall (important for this retrieval task) while maintaining fair precision.

We performed ablation studies to evaluate the impact of our approaches on retrieval per-

Table 3.5: Ablation studies. *SS model* indicates whether or not to ensemble with corresponding specific-scenario models. *Return* indicates the number of relevant articles returned. *Retrieve* indicates the number of correct returned relevant articles. **Bold texts** indicate highest results.

Model	SS model?	F2 score	Precision	Recall	Return	Retrieve
LegalBERT		71.81	69.96	74.69	102	67
LegalBERT	✓	75.59	69.96	80.25	117	72
Longformer		70.00	**70.37**	71.60	95	64
Longformer	✓	71.49	69.49	74.69	110	68
LegalBERT + Longformer		74.10	66.15	79.63	128	73
LegalBERT + Longformer	✓	**76.87**	65.80	**85.19**	149	**78**

formance. As shown in Table 3.5, ensembling predictions with the corresponding scenario-specific model consistently enhances the F2 score. Specifically, for LegalBERT, the F2 score increased by 3.78%, from 71.81% to 75.59%. For Longformer, this increase was 1.49%, from 70.00% to 71.49%. While the ensemble of LegalBERT and Longformer achieved a 74.10% F2 score, including their scenario-specific models further raised the F2 score to 76.87%. This improvement is primarily due to increased recall (5.56%, 3.09%, and 5.56% for LegalBERT, Longformer, and the combined LegalBERT + Longformer models, respectively), with minimal decrease in precision (0.88% at worst), thus significantly boosting the F2 score. Additionally, scenario-specific models increased the number of retrieved relevant articles by 4 to 5, greatly improving the macro-average F2 score, especially in scenarios where the base models failed to retrieve any relevant articles.

Table 3.6 illustrates examples of the contribution of scenario-specific models to the base models. It highlights cases where the base models (LegalBERT, Longformer, or their combination) failed to retrieve relevant articles, but the scenario-specific models succeeded.

3.6 Conclusion

This paper addresses two challenges related to the characteristics of legal bar exam queries and articles, previously overlooked by other methods. The first challenge is the linguistic disparity between legal articles, which are abstract, and legal bar exam queries, which may vary between abstract and scenario-specific. We propose identifying and addressing scenario-specific queries separately. The second challenge involves managing long documents, which restrict the efficacy of many pre-trained models. We leverage Longformer's capability to handle long documents to overcome this issue. By combining the retrieval outcomes of these models, our system achieved the best F2 score of 76.87% on the Task 3 (COLIEE 2021) dataset, highlighting the need for further research to address unique characteristics in legal domain documents.

Table 3.6: Examples on the contribution of specific-scenario (SS) models

LegalBERT and its SS model can retrieve but LegalBERT cannot
Query R02-8-E: A took the jewelry that B had forgotten, believing without negligence that it belonged to A. In this case, A may not obtain the ownership of the jewelry by good faith acquisition.
Retrieved article 192: Part II Real Rights - Chapter II Possessory Rights - Section 2 Effect of Possessory Rights (Good Faith Acquisition) Article 192 A person that commences the possession of movables peacefully and openly by a transactional act acquires the rights that are exercised with respect to the movables immediately if the person possesses it in good faith and without negligence.
Longformer and its SS model can retrieve but Longformer cannot
Query R02-23-O: If A delays in delivering X after the contract of sale stipulates that a third party (G) is to acquire the ownership of X and G manifests intention of availing of the benefit, B may not cancel the contract of sale without G's consent.
Retrieved article 538: Part III Claims - Chapter II Contracts - Section 1 General Provisions - Subsection 2 Effect of Contracts (Determination of Rights of the Third Party) Article 538 (1) After rights of the third party have accrued pursuant to the provisions of the preceding Article, the parties may not modify or extinguish those rights. (2) If, after rights of the third party accrue pursuant to the provisions of the preceding Article, the obligor does not perform the obligation to the third party, the other party to the contract referred to in paragraph (1) of that Article may not cancel the contract without the consent of the third party.
LegalBERT + Longformer and their SS models can retrieve but LegalBERT + Longformer cannot
Query R02-9-O: A had owned land in an area of holiday homes. B, who owned neighboring land, began to construct a fence on A's land crossing the boundary without A's consent, and two years later the fence was completed. As of this time, A may not demand the removal of the fence against B by filing an action for maintenance of possession.
Retrieved article 201: Part II Real Rights - Chapter II Possessory Rights - Section 2 Effect of Possessory Rights (Periods of Time for Filing Possessory Actions) Article 201 (1) An action for maintenance of possession must be filed during the obstruction or within one year after the obstruction stops; provided, however, that if the possessed thing has been damaged due to construction work and either one year has passed from the time when the construction was started or the construction has been completed, the action may not be filed. (2) An action for preservation of possession may be filed so long as the danger of obstruction exists. In this case, the provisions of the proviso to the preceding paragraph apply mutatis mutandis if the possessed thing is likely to be damaged by the construction work. (3) An action for recovery of possession must be filed within one year from the time when a possessor was forcibly dispossessed.

Chapter 4

Retrieve-Revise-Refine Framework for Retrieval of Concise Entailing Legal Article Set

4.1 Overview

Retrieval of entailing legal article set is a task characterized by the objective of identifying a concise (i.e., precise and compact) set of legal articles that holds an entailment relationship with a legal query or its negation. This task differs substantially from traditional information retrieval, which mainly focuses on ranking candidates by their relevance. However, prior research has predominantly approached the task of entailing legal article set retrieval with methodologies akin to those used in traditional information retrieval, thereby failing to adequately fulfill the essential demand for conciseness in the retrieved sets.

To address this gap, we propose a specialized three-stage framework which focuses on the key challenge of legal article set retrieval: yielding a concise set of entailing articles. The proposed Retrieve-Revise-Refine framework integrates a series of small and large language models (LMs) in a complementary manner. Initially, the retrieval stage employs various tailored fine-tuning strategies to small LMs, aiming to retrieve a comprehensive coverage of entailing articles. Subsequently, the revision stage utilizes large LMs to revise the initial retrieval, producing a narrower selection of legal articles. Finally, the refinement stage leverages the specialized insights offered by the small LMs to refine the output from the large LMs. This stage acknowledges a remarkable enhancement in precision over the second stage, at the expense of a marginal reduction in coverage.

Empirical results from evaluations conducted on the COLIEE 2022 and COLIEE 2023 datasets demonstrate the efficacy of the proposed framework, with notable improvements in the macro F2 score, achieving increases of 3.17% and 4.24% over the previous state-of-the-art methods, respectively. More importantly, our specialized framework has demonstrated superior effectiveness, as evidenced by its capability to identify more concise sets of legal articles in comparison to other methods, thereby aligning with the requirements of the task. These improvements mark a departure from previous methodologies, reinforcing the importance of a tailored approach in the task of retrieving entailing legal article set. Additionally, our ablation studies and subsequent analysis yield significant insights into the indispensable roles of each stage within the framework. The findings clearly demonstrate that every stage systematically contributes to the overall success of the retrieval process. The proposed framework represents a constructive step toward a promising direction for advancing the retrieval task.

4.2 Introduction

Artificial Intelligence (AI) continues to redefine the boundaries of legal technology, offering promise in automating advanced tasks such as legal question answering and consultation. In the domain of statute law, a particularly principal challenge is the task of retrieving the concise set of entailing legal articles to a query, a task essential to enhancing these advanced applications. In this context, we refer to this task as *entailing legal article set retrieval* or, more briefly, *legal article set retrieval*.

In our study, we undertake the task of legal article set retrieval, which involves reading a legal statement (denoted as query Q) and retrieving a set of legal articles from a corpus. The objective is to identify a concise (i.e., precise and compact) set of legal articles (A_1, A_2, $\ldots$, A_n) that holds an entailment relationship with the query Q or its negation *not* Q. We refer to those articles in the set as *entailing legal articles*, irrespective of whether they entail Q or *not* Q. Besides, since the query in this scenario is a legal statement, we use "*query*" and "*legal statement*" as synonymous terms.

Figure 4.1 presents an example of a query along with its corresponding set of entailing legal articles, which includes *Article 724* and *Article 724-2*. In this example, the set of *Article 724* and *Article 724-2* entails the negation of the query. To provide a comprehensive explanation, *Article 724* establishes the general rules of extinctive prescription for tort claims, which means that if a claim for compensation for loss or damage from a tortious act is not exercised within certain time limits, it is extinguished. Specifically, the article states two time limits: (i) three years from the time when the damage and perpetrator become known to the victim or their legal representative, or (ii) twenty years from the time of the tortious act itself. However, *Article 724-2* provides a special provision that modifies the three-year prescription period laid out in *Article 724* when it comes to cases involving death or injury to a person caused by tort. Specifically, it extends that prescription period from three years to five years. Therefore, if we interpret the query to suggest that claims for compensation for death or personal injury caused by tort are never extinguished by prescription, the combination of *Article 724* and *Article 724-2* shows that this is not accurate. In fact, the articles together establish that such claims do have prescription periods: five years from the time the damage and the perpetrator are known, or twenty years from the tortious act (as per the general rule in *Article 724*). Hence, if a claim is not exercised within these time periods, it is extinguished. Thus, *Article 724-2*, when read in conjunction with *Article 724*, negates the claim that compensation for loss or damage for death or injury to a person caused by tort is not extinguished by prescription. In other words, the provisions outlined in *Article 724* and *Article 724-2* collectively imply the negation of the query. The set of these two articles is concise (i.e., precise and compact) because neither article can be substituted with another without potentially compromising the accuracy of the conclusion regarding the validity of the query, and the exclusion of any one article may lead to an incorrect conclusion.

The task of retrieving entailing legal article set differs markedly from traditional information retrieval (IR) in two main points. Firstly, unlike the traditional IR which returns a ranked list of

QUERY

The claim for compensation for loss or damage for death or injury to person caused by tort is not extinguished by prescription.

SET OF ENTAILING LEGAL ARTICLES

Article 724 (Extinctive Prescription of Claim for Compensation for Loss or Damage Caused by Tort):
In the following cases, the claim for compensation for loss or damage caused by tort is extinguished by prescription:
(i) the right is not exercised within three years from the time when the victim or legal representative thereof comes to know the damage and the identity of the perpetrator; or
(ii) the right is not exercised within 20 years from the time of the tortious act.

Article 724-2 (Extinctive Prescription of Claim for Compensation for Loss or Damage Arising from Death to Person or Injury to Person Caused by Tort):
For the purpose of the application of the provisions of item (i) of the preceding Article with regard to the extinctive prescription of the claim for compensation for loss or damage for death or injury to person caused by tort, the term "three years" in the same item is deemed to be replaced with "five years".

Figure 4.1: An example of a query and the corresponding set of entailing legal articles.

articles, the legal article set retrieval task seeks a concise set of articles. This level of specificity extends to the nature of the legal queries and legal articles themselves: they are inherently complex and steeped in specialized legal language, demanding a retrieval system with deeper legal reasoning and linking capacity. Secondly, while traditional IR efforts primarily involve ranking candidates by relevance, our task requires that the retrieved articles not just relate to but jointly entail the contents of a query or its negation. These characteristics set this task apart from the broader goals and methods of traditional IR tasks.

Previous research in legal article set retrieval has predominantly employed two approaches. The first approach combines classical IR models with fine-tuned language models (LMs), and then ensembles the retrieval results to consolidate the final retrieved sets (Yoshioka and Aoki, 2021; Bui et al., 2024; Yoshioka and Aoki, 2023). Meanwhile, the second approach uses classical IR models exclusively for preliminary candidate filtering, which prepares inputs for further LM fine-tuning; the final results are often ensembled from various fine-tuned LMs (Bui et al., 2022b; Vuong et al., 2024; Nguyen et al., 2023b). Researchers have proposed numerous strategies to enhance model performance within these approaches. For example, manual feature selection, informed by domain expertise, was used to optimize retrieval systems (Yoshioka et al., 2022; Yoshioka and Aoki, 2023). The recognition of domain-specific entities, as explored by LLNTU (Goebel et al., 2023), has been evidenced as a crucial factor in contextualizing legal documents. Additionally, the augmentation of training data through external sources has been investigated as a means to improve the generalizability of language models in IR tasks (Wehnert et al., 2021b; Yoshioka and Aoki, 2021; Wehnert et al., 2022). Innovative approaches have also included sophisticated sentence embedding techniques to assess query and text semantic similarity (Schilder et al., 2021b; Wehnert et al., 2021b). Notably, a divide-and-conquer strategy, outlined by Nguyen et al. (2022), suggests categorizing queries and subsequently applying tailored deep learning models for each. Despite these advancements, the primary focus of earlier studies remains on traditional IR tasks, with less attention given to identifying concise set of legal articles that entails the query or its negation—a matter critical to the task at hand.

In this paper, we propose a novel three-stage framework, called Retrieve-Revise-Refine, to address the task of legal article set retrieval. In contrast to previous research, our framework is designed with the specific goal of pinpointing the concise (i.e., precise and compact) set of legal articles that either entail a query or its negation, thereby advancing the current understanding of this task. Furthermore, our approach leverages the unique advantages of combining both small language models and large language models (LMs) to improve the accuracy of the articles retrieved (i.e., precision), while endeavoring to limit the loss in coverage (i.e., recall).

At the initial Retrieval stage, our primary objective is to maximize the comprehensive retrieval of entailing articles, thereby minimizing the risk of omitting any entailing articles. To achieve this, our retrieval mechanism employs an ensemble of multiple small LMs, fine-tuned with various tailored strategies. These fine-tuning strategies are specifically designed to tackle different issues raised by the conventional fine-tuning strategy. By aggregating the predictions from these diverse models, we achieve over 90% coverage within the top-5 retrieval results. In the subsequent Revise stage, the objective is to revise the set of top retrieval results obtained from the initial stage into a more concise subset of legal articles. To this end, large language models (LMs) are utilized to meticulously assess the validity of the query with respect to each combination of articles from the top retrieval results. Following this assessment, a process of unification of large LMs' responses is conducted to derive the revised set of entailing legal articles, expecting a more compact subset. The final Refine stage aims to further distill the outputs from the second stage, yielding an even more precise set of legal articles. This stage involves an evaluation of the concordance between the predictions generated by the large LMs and the articles retrieved by the small LMs. In this stage, insights derived from the small LMs' predictions function as *refiners* for the predictions of the large LMs, with the goal of ensuring that the resultant set of entailing articles is highly concise while endeavoring to minimize the trade-off in retrieval quality. As shown in the empirical results, this stage exhibits a remarkable enhancement in precision compared to the second stage, though at the cost of a marginal reduction in coverage.

This paper highlights three main contributions, as detailed below:

- Firstly, we propose a novel three-stage framework, Retrieve-Revise-Refine, specifically designed to address the intricate challenge of legal article set retrieval. In particular, this structured approach uniquely focuses on retrieving a concise (i.e., precise and compact) set of entailing legal articles, setting it apart from existing methodologies, which follow the approaches for traditional information retrieval tasks.

- Secondly, we rigorously evaluate our framework using two datasets, where we observe notable improvements in the macro F2 score, achieving increases of 3.17% and 4.24% over the previous state-of-the-art methods, respectively. Moreover, the effectiveness of our approach is highlighted by its superior performance in identifying more concise sets of entailing articles, as shown in our experiments. These findings, along with further experimental results, demonstrate that our framework not only sets new benchmarks for the task but also underscores its robustness and adaptability across various contexts.

- Lastly, our comprehensive ablation studies and subsequent analysis provide valuable insights into the critical functions of each stage within the framework. In particular, the Retrieval stage ensures extensive coverage of the entailing articles, capturing over 90% within the top-5 retrieval results. The Revise stage enhances precision by revising the results, and the Refine stage further narrows down the set to the most concise set of entailing legal articles. As a whole, these stages work synergistically to enhance overall retrieval performance. This demonstrates that our Retrieval-Revise-Refine framework represents an advancement in the task of legal article set retrieval, and it offers a promising direction for future research on this task.

4.3 Methodology

Our framework demonstrates a three-stage approach, Retrieval-Revise-Refine, for retrieving the concise legal article set that entails a given query or its negation. In the Retrieval stage, our aim is to retrieve the broadest possible range of the entailing legal articles. This is accomplished by employing various tailored fine-tuning strategies to small LMs and ensembling their ranked retrieval results. In the Revise stage, large LMs are introduced to revise the broad set of articles obtained in the first stage, attempted to extract a more concise set. These large LMs leverage their extensive knowledge and capacity for complex understanding to revise the results, working on the top-ranking articles from the initial stage. Finally, the Refine stage involves cross-validating the large LMs' predictions with the articles found by the small LMs to further refine the selection. This step helps concentrate the results down to a even more concise set of entailing legal articles, without significantly compromising the retrieval quality. Figure 4.2 illustrates the overall of stages within our framework. The following subsections provide an in-depth description of each individual stage.

4.3.1 Retrieval Stage: Small LMs as Retrievers

In this stage, we treated our task as a traditional retrieval task. In other words, we simplified the legal article set retrieval task to be the task of ranking the articles by their relevance to a given query. Figure 4.3 illustrates the details of the Retrieval stage's workflow, highlighting various fine-tuning strategies aimed at maximizing the coverage of retrieved entailing articles. The retrieval results of this stage were an ensemble of results obtained from multiple model checkpoints. For a comprehensive understanding of the output flow for this stage (shown in Figure 4.3) and the other two stages for the example in Figure 4.1, please refer to Section Case Study.

Initial Fine-tuning

The initial fine-tuning process is highlighted in green in Figure 4.3. To construct the data for this fine-tuning process, firstly, we paired a query with each of its entailing article to be a

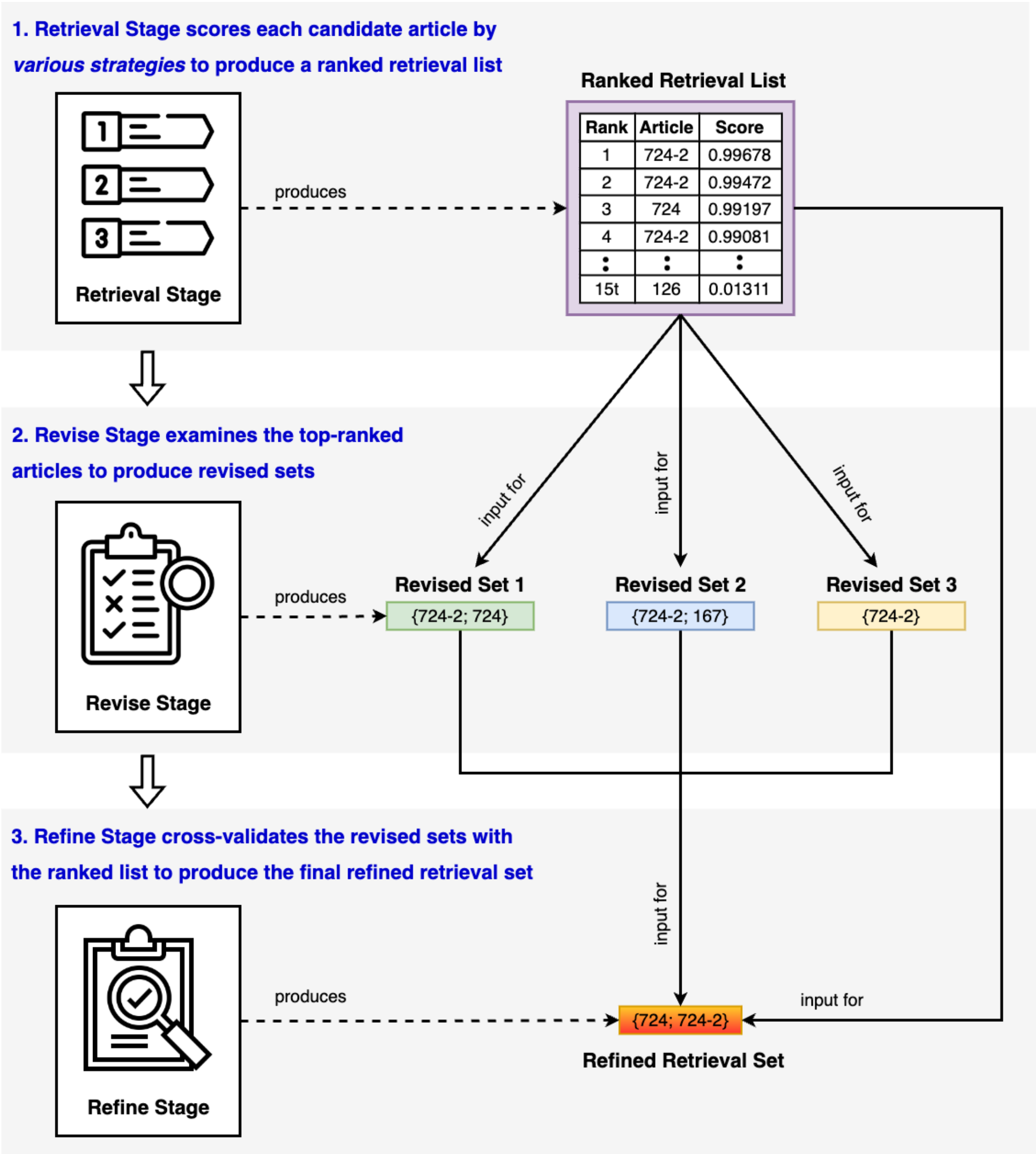

Figure 4.2: Overall of stages within our framework

sample of positive pair. As there was no explicit negative pair samples in the training data, we obtained negative pair samples through a TFIDF-based negative sampling strategy. For each query, we retrieved the top-k_1 (in experiments, we chose $k_1 = 150$, following Bui et al. (2022b)) TFIDF-ranked articles that did not include entailing articles. Each of them was paired with the query to be a negative pair sample. We performed fine-tuning with a BERT-based (Devlin et al., 2019) model on the constructed dataset (referred to as the *TFIDF dataset*) and picked the best five checkpoints (referred to as the best five *TFIDF checkpoints*) based on their performance on a validation set. Algorithm 1 details the procedure for obtaining best *TFIDF checkpoints*.

In accordance with the input format for BERT-based models, the article and the query were

Algorithm 1 Initial Fine-tuning

Require: $\mathcal{Q}$: Set of queries
Require: $\mathcal{A}$: Set of articles in corpus
Require: $\mathcal{E}$: Mapping of queries to entailing articles
Require: k_1: Number of top TFIDF-ranked negative samples (e.g., $k_1 = 150$)

Initialize sets to store positive and negative pairs
$\mathcal{D}_{\text{pos}} \leftarrow \{\}$ ▷ Initialize the set of positive pairs
$\mathcal{D}_{\text{neg}} \leftarrow \{\}$ ▷ Initialize the set of negative pairs
for each $q \in \mathcal{Q}$ **do** ▷ Iterate over each query
    $E_q \leftarrow \mathcal{E}(q)$ ▷ Get entailing articles for the query

    # Create positive pairs
    for each $a \in E_q$ **do** ▷ Iterate over each entailing article
        $\mathcal{D}_{\text{pos}} \leftarrow \mathcal{D}_{\text{pos}} \cup \{(q, a, 1)\}$ ▷ Add positive pair to set
    end for

    # Retrieve top TFIDF-ranked negative samples
    $A_{neg} \leftarrow$ Top k_1 TFIDF-ranked articles not in E_q ▷ Get top k_1 TFIDF results

    # Create negative pairs
    for each $a \in A_{neg}$ **do** ▷ Iterate over each TFIDF-ranked negative article
        $\mathcal{D}_{\text{neg}} \leftarrow \mathcal{D}_{\text{neg}} \cup \{(q, a, 0)\}$ ▷ Add negative pair to set
    end for
end for

Combine positive and negative pairs into final training dataset
$\mathcal{D}_{TFIDF} \leftarrow \mathcal{D}_{\text{pos}} \cup \mathcal{D}_{\text{neg}}$

Fine-tune BERT-based model on the created dataset
Fine-tune a BERT-based model on $\mathcal{D}_{TFIDF}$ ▷ Fine-tuning with the combined dataset
Pick the best five checkpoints based on validation set performance and assign to $TFIDF_{chkpts}$ ▷ Select best five checkpoints
return $TFIDF_{chkpts}$ ▷ Return the best five checkpoints

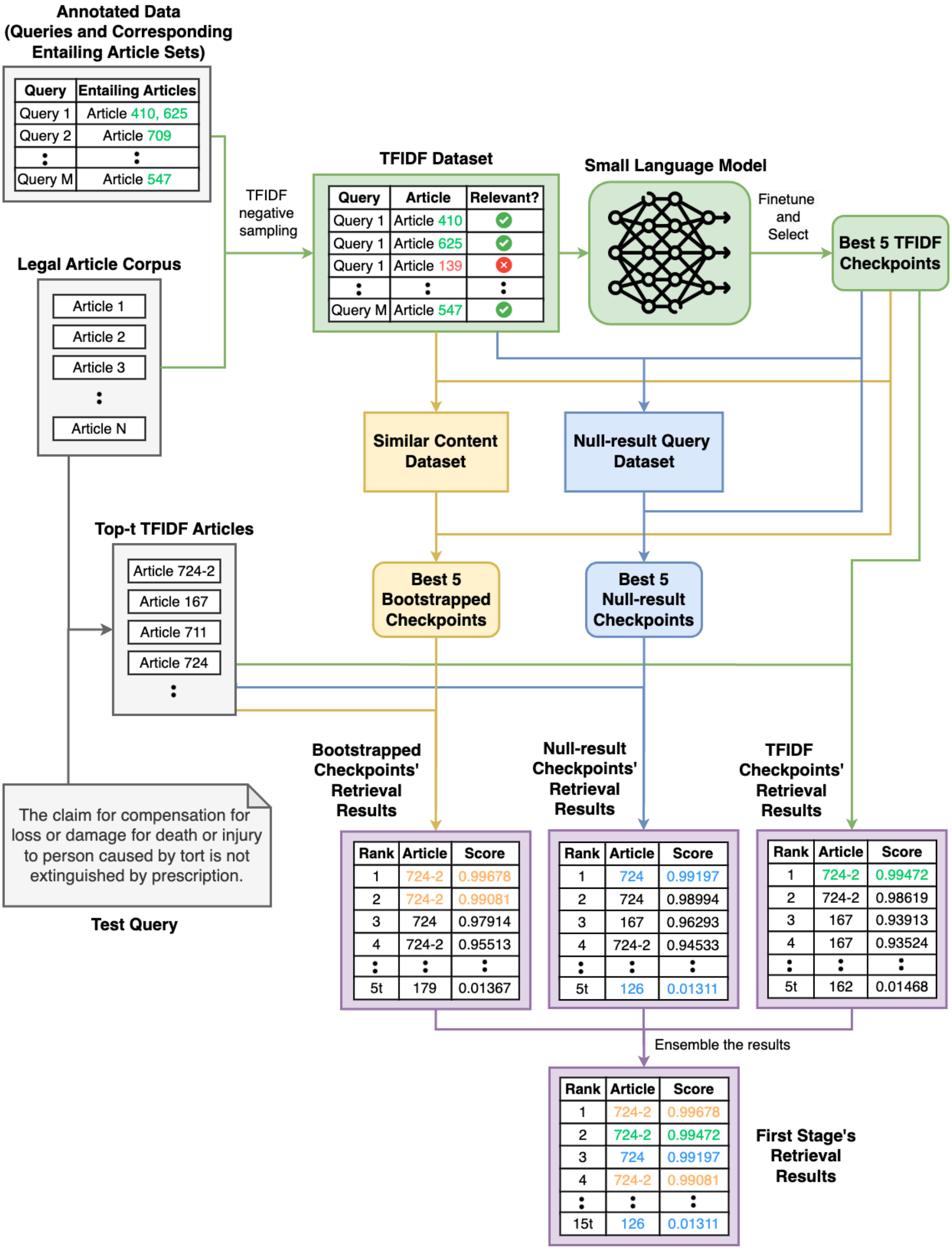

Figure 4.3: Details of the Retrieval stage's workflow

concatenated as demonstrated in Table 4.1.

To train the model, we utilized the cross-entropy loss function, which is appropriate for binary classification tasks. The formula for the binary cross-entropy loss function $\mathcal{L}$ is given below:

Table 4.1: Example pairs and corresponding class labels

Example of a pair	Class label
`[CLS] <an entailing article> [SEP] <the query>`	1 (Positive)
`[CLS] <an non-entailing article> [SEP] <the query>`	0 (Negative)

$$\mathcal{L} = -\left[y \cdot \log(p_1) + (1 - y) \cdot \log(p_0)\right] \tag{4.1}$$

where y is the ground truth label for the pair (with $y = 1$ for a positive pair and $y = 0$ for a negative pair), and p_1 and p_0 are the predicted probabilities of the pair being in the positive and negative classes, respectively. The probabilities p_1 and p_0 are derived from the model's logit outputs l_1 and l_0 using the softmax function:

$$p_0 = \frac{e^{l_0}}{e^{l_0} + e^{l_1}}, \quad p_1 = \frac{e^{l_1}}{e^{l_0} + e^{l_1}} \tag{4.2}$$

During the inference phase, given a new pair of article and query, the model outputted two logit values l_0 and l_1. These logit values were converted to probabilities p_0 and p_1 using the softmax function, as shown in Equation (2). The predicted class for the pair was determined by the higher probability:

$$\text{Predicted class} = \begin{cases} \text{Positive} & \text{if } p_1 > p_0 \\ \text{Negative} & \text{otherwise} \end{cases} \tag{4.3}$$

To obtain a ranking of articles, we sorted them based on the l_1 values (logits corresponding to positive class predictions) in descending order.

Addressing Null Results with Continual Fine-tuning

After analyzing the picked *TFIDF checkpoints*, we found that they encountered an issue where these checkpoints returned *null results* (i.e., all checkpoints predicted all article candidates as non-entailing) for multiple queries. This issue is critical for the task of legal article set retrieval because it can result in the complete failure of the system to retrieve any entailing legal articles for certain queries, thereby compromising the effectiveness and reliability of the retrieval pro-

Algorithm 2 Addressing Null Results with Continual Fine-tuning

Require: $\mathcal{Q}$: Set of queries
Require: $TFIDF_{chkpts}$: Set of picked TFIDF checkpoints
Require: $\mathcal{D}_{TFIDF}$: TFIDF dataset
Require: k_2 : Number of most similar queries to retrieve (e.g., $k_2 = 10$)

Identify queries that yield null results:
$\mathcal{Q}_{null} \leftarrow \{\}$ ▷ Initialize the set of null-result queries
for each $chkpt \in TFIDF_{chkpts}$ **do**
    $\mathcal{R}(chkpt) \leftarrow$ Predict relevance of all queries in $\mathcal{Q}$ using $chkpt$
    for each $q \in \mathcal{Q}$ **do**
        if $\mathcal{R}(chkpt, q) \equiv \emptyset$ **then** ▷ Check if there are null results for the query using this checkpoint
            $\mathcal{Q}_{null} \leftarrow \mathcal{Q}_{null} \cup \{q\}$ ▷ Add null-result query to the set
        end if
    end for
end for
# Find similar queries to null-result queries:
$\mathcal{Q}_{similar} \leftarrow \{\}$ ▷ Initialize set of similar queries for null-result queries
for each $q_{null} \in \mathcal{Q}_{null}$ **do** ▷ Iterate over each null-result query
    $sim_{queries} \leftarrow$ Retrieve the k_2 most similar queries to q_{null} using cosine similarity of embeddings
    $\mathcal{Q}_{similar} \leftarrow \mathcal{Q}_{similar} \cup sim_{queries}$ ▷ Add similar queries to the set
end for
# Combine null-result queries and their similar queries:
$\mathcal{Q}_{\text{new}} \leftarrow \mathcal{Q}_{null} \cup \mathcal{Q}_{similar}$
Extract a new dataset for continual fine-tuning:
$\mathcal{D}_{\text{null-result}} \leftarrow$ Extract dataset from $\mathcal{D}_{TFIDF}$ corresponding to $\mathcal{Q}_{\text{new}}$
Select the best TFIDF checkpoint and prepare for continual fine-tuning:
$best_{TFIDFchkpt} \leftarrow$ Best TFIDF checkpoint from the initial fine-tuning process
$nullresult_{chkpts} \leftarrow \{\}$ ▷ Initialize set for null-result checkpoints
Perform continual fine-tuning using the extracted dataset:
Continually fine-tune $best_{TFIDFchkpt}$ on $\mathcal{D}_{\text{null-result}}$
Pick the best five checkpoints from this process and assign to $nullresult_{chkpts}$
return $nullresult_{chkpts}$ ▷ Return the best five checkpoints

cess. To address this issue, we detected all null-result queries (i.e., queries where the picked *TFIDF checkpoints* returned *null results*) and retrieved their top-k_2 (in experiments, we chose $k_2 = 10$) most similar queries via the cosine similarity of their embeddings. Specifically, from the *TFIDF dataset*, we selected the samples corresponding to the null-result queries and their similar queries to form a new dataset (referred to as *null-result query dataset*). Then we performed continual fine-tuning on this dataset with the best *TFIDF checkpoint*, and picked the best five checkpoints (referred to as the *null-result checkpoints*). The refinement efforts concentrated on examining data from queries that yielded *null results*, along with closely related queries. The aim was to enhance the performance of the specially calibrated system checkpoints in handling instances of such queries. This process is highlighted in blue in Figure 4.3 and is detailed in Algorithm 2.

Tackling Similar Content Confusion with Bootstrap Fine-tuning

We additionally observed that the best five *TFIDF checkpoints* frequently exhibited confusion among multiple articles with similar content, often incorrectly identifying numerous articles as entailing when, in fact, only one was appropriate. To mitigate this phenomenon, we employed bootstrap fine-tuning. Queries were paired with their top-k_3 (in experiments, we chose $k_3 = 10$) closest relevant but non-entailing articles, identified by the best *TFIDF checkpoint*, to be the negative pair samples. The positive pair samples were retained from the original *TFIDF dataset*. We used this dataset (referred to as *similar content dataset*) to further fine-tune the best *TFIDF checkpoint*, then we also picked the best five checkpoints (referred to as the best five *bootstrapped checkpoints*). As a result of the bootstrap fine-tuning process, we expected those checkpoints with enhanced discriminative abilities. This process is highlighted in yellow in Figure 4.3 and is detailed in Algorithm 3.

Improving Coverage with Ensembling of Retrieval Lists

At this point, for a given query, we obtained a retrieval list for each of the fifteen best checkpoints (best five *TFIDF checkpoints*, best five *null-result checkpoints*, best five *bootstrapped checkpoints*). It should be noted that as the corpus could contain a significant number of ar-

Algorithm 3 Tackling Similar Content Confusion with Bootstrap Fine-tuning

Require: $\mathcal{Q}$: Set of queries
Require: $\mathcal{A}$: Set of articles in corpus
Require: $\mathcal{E}$: Mapping of queries to entailing articles
Require: $best_{TFIDFchkpt}$: Best TFIDF checkpoint from the initial fine-tuning process
Require: k_3: Number of closest non-entailing articles to select (e.g., $k_3 = 10$)

\# Initialize sets to store bootstrapped pairs (both positive and negative):
$\mathcal{D}_{\text{pos-bootstrap}} \leftarrow \{\}$ ▷ Initialize the set of bootstrapped positive pairs
$\mathcal{D}_{\text{neg-bootstrap}} \leftarrow \{\}$ ▷ Initialize the set of bootstrapped negative pairs
\# Iterate over each query to create bootstrapped pairs:
for each $q \in \mathcal{Q}$ **do** ▷ Iterate over each query
    $E_q \leftarrow \mathcal{E}(q)$ ▷ Get entailing articles for the query
    for each $a \in E_q$ **do** ▷ Iterate over each entailing article
        $\mathcal{D}_{\text{pos-bootstrap}} \leftarrow \mathcal{D}_{\text{pos-bootstrap}} \cup \{(q, a, 1)\}$ ▷ Add positive pair to set
    end for
    $A_{neg} \leftarrow$ Top k_3 closest relevant but non-entailing articles identified by $best_{TFIDFchkpt}$
    for each $a \in A_{neg}$ **do** ▷ Iterate over each non-entailing article
        $\mathcal{D}_{\text{neg-bootstrap}} \leftarrow \mathcal{D}_{\text{neg-bootstrap}} \cup \{(q, a, 0)\}$ ▷ Add negative pair to set
    end for
end for
\# Combine positive and negative bootstrapped pairs into the final dataset:
$\mathcal{D}_{\text{similar-content}} \leftarrow \mathcal{D}_{\text{pos-bootstrap}} \cup \mathcal{D}_{\text{neg-bootstrap}}$
\# Fine-tune the best TFIDF checkpoint on the created similar content dataset:
Fine-tune $best_{TFIDFchkpt}$ on $\mathcal{D}_{\text{similar-content}}$
Pick the best five checkpoints based on validation set performance and assign to $bootstrap_{chkpts}$
return $bootstrap_{chkpts}$ ▷ Return the best five checkpoints

ticles, we applied a filter to select the top-t articles with the highest TFIDF scores and then passed only these articles through to the checkpoints for ranking. The fifteen checkpoints each generated t predictions, resulting in a combined output of $15t$ predictions. As checkpoints from each type would be robust for a certain type of queries, emsembling of their retrieval lists would benefit the coverage. In other words, it would increase (or at least keep) the coverage of the true entailing articles. This process provided a solid foundation for the next stage of our methodology. The process, depicted in purple in Figure 4.3, is detailed in Algorithm 4.

Algorithm 4 Improving Coverage with Ensembling of Retrieval Lists

Require: $\mathcal{Q}$: Set of queries
Require: $\mathcal{A}$: Set of articles in corpus
Require: $TFIDF_{chkpts}$: Set of best five TFIDF checkpoints
Require: $nullresult_{chkpts}$: Set of best five null-result checkpoints
Require: $bootstrap_{chkpts}$: Set of best five bootstrapped checkpoints
Require: t: Number of top TFIDF articles to filter (e.g., $t = 150$)

Combine all checkpoints into a single set:
$\mathcal{CHKPTS} \leftarrow TFIDF_{chkpts} \cup nullresult_{chkpts} \cup bootstrap_{chkpts}$ ▷ Combine all checkpoints

Initialize the ensemble retrieval list:
$\mathcal{R}_{ensemble} \leftarrow \{\}$ ▷ Initialize the ensemble retrieval list

Iterate over each query to generate ensemble retrieval lists:
for each $q \in \mathcal{Q}$ **do** ▷ Iterate over each query
  $A_{top} \leftarrow$ Top t articles from $\mathcal{A}$ based on TFIDF scores for query q ▷ Filter top t articles
  $\mathcal{R}_q \leftarrow \{\}$ ▷ Initialize retrieval list for query q
  for each $chkpt \in \mathcal{CHKPTS}$ **do** ▷ Iterate over each checkpoint
    $R_{chkpt} \leftarrow$ Predict relevance of articles in A_{top} using $chkpt$ for query q ▷ Get predictions for top articles
    $\mathcal{R}_q \leftarrow \mathcal{R}_q \cup R_{chkpt}$ ▷ Add checkpoint predictions to query retrieval list
  end for
  $\mathcal{R}_{ensemble} \leftarrow \mathcal{R}_{ensemble} \cup \mathcal{R}_q$ ▷ Add query retrieval list to ensemble list
end for
return $\mathcal{R}_{ensemble}$ ▷ Return the ensemble retrieval list

In summary, the retrieval stage of our methodology was developed using a multi-tiered fine-tuning approach. We started by fine-tuning the initial BERT model, then moved on to targeted fine-tuning processes, and completed the stage with an ensemble of fine-tuned checkpoints. This structured approach allows for extensive coverage, which is crucial for the performance of the finer-grained stage that follow.

4.3.2 Revise Stage: Large LMs as Revisers

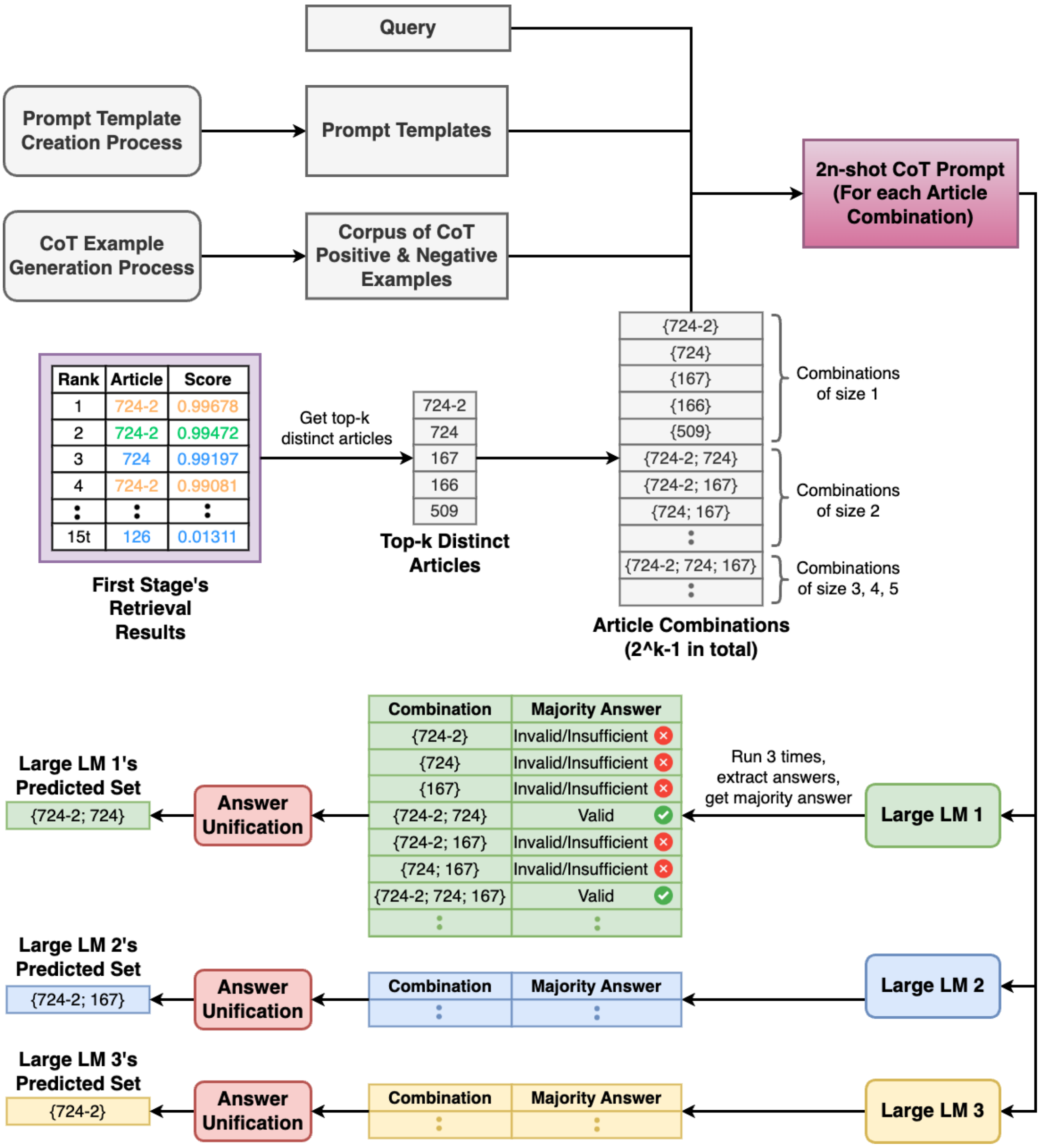

Figure 4.4: Details of the Revise stage's workflow

Large language models (large LMs) possess sophisticated language understanding capabilities, enabling them to comprehend and process complex legal texts. These models have been trained on extensive and diverse corpora, which encompass various domains, including legal language. Consequently, large LMs can interpret the nuanced semantics and intricate syntactic structures inherent in legal statements and articles. Their proficiency in natural language processing allows them to effectively match the entailment relationships between a query and a set

Algorithm 5 Step-by-step Instructions for a Single Large LM in the Revise Stage

Require: $\mathcal{Q}$: Set of queries
Require: k: Number of top distinct articles per query
Require: $\mathcal{T}$: Set of top-k distinct articles per query
Require: n: Parameter to determine number of few-shot examples ($2n$ examples in total) (e.g., $n = 2$)
Require: LLM: A large language model

Create multiple prompt templates:
$\mathcal{P}$: Create multiple prompt templates using the base prompt and GPT-4

Create base examples from training data and initialize storages of prompt-specific examples:
$\mathcal{X}_{\text{pos}} \leftarrow$ Positive examples (i.e., examples with *validate* answers)
$\mathcal{X}_{\text{neg}} \leftarrow$ Negative examples (i.e., examples with *refute* answers)
$\mathcal{X}_p \leftarrow \{\mathcal{X}_{\text{pos}}, \mathcal{X}_{\text{neg}}\}$ for each $p \in \mathcal{P}$

# Iterate over each query to generate revised sets:
for each $q \in \mathcal{Q}$ **do**
    # Generate all possible combinations of the top-k articles:
    $\mathcal{C}_q \leftarrow$ All possible article combinations of $\mathcal{T}_q$ $\triangleright$ $2^k - 1$ combinations in total
    Initialize empty list to track valid combinations: $\mathcal{V}_q \leftarrow \{\}$
    # Iterate over each combination and prompt template to get revised concise sets:
    for each combination $C \in \mathcal{C}_q$ **do**
        for each $p \in \mathcal{P}$ **do**
            Select n most similar positive and n most similar negative examples to q from $\mathcal{X}_p$
            Formulate final prompt with $2n$-shot examples and article combination C
            Run LLM on the generated prompt, 3 times and collect responses
            Extract high-frequency response across 3 runs: R_C
            if R_C indicates the combination C contains true entailing articles **then**
                $\mathcal{V}_q \leftarrow \mathcal{V}_q \cup \{C\}$
            end if
        end for
    end for
    Identify the most concise group in $\mathcal{V}_q$
    if multiple groups are equally concise **then**
        Combine all unique articles in those groups
    end if
end for
return Revised article sets for all queries for the setting of the LLM on the top-k distinct articles

of legal articles. By leveraging their extensive knowledge base and ability to perform contextual reasoning, large LMs can discern which articles in a corpus are entailing, either supporting or contradicting a given legal statement. Thus, large LMs are well-equipped to undertake the task of legal article set retrieval, identifying a concise set of entailing articles with a high degree of accuracy and efficiency.

Figure 4.4 depicts the workflow involving multiple large LMs used in this stage, while Algorithm 5 provides the step-by-step instructions for a single large LM.

Based on the confidence scores in the Retrieval stage's retrieval results, we selected collection of top-k distinct articles per query. The top-k parameter is a hyperparameter that will be optimized individually for each large LM. It should be noted that repetitions among the articles with the highest confidence ratings may occur, as the ranked list of articles was generated based on outputs from multiple checkpoints. For example, it is possible that only three distinct articles may be represented within the top ten ranked articles.

These top-k articles underwent a revision process with the aim of leveraging the sophisticated language understanding capabilities of large LMs to identify a more concise set of entailing articles. In a preliminary analysis, we explored three approaches: Reranking, Selection, and Validation. For the Reranking approach, the complete set of retrieved legal articles (five in our case) was provided to a large LM, which was then tasked with reranking them. A notable drawback of this approach is the requirement to heuristically determine the top-n articles post-reranking. Additionally, the reranking process necessitates that the large LM compare all articles with each other, thus introducing complexity. In the Selection approach, a large LM was given the entire set of legal articles and instructed to select a concise subset. This approach demands the model to simultaneously undertake multiple tasks: (i) compare each article with the query and (ii) identify the relevant articles. Our observations indicated that this approach tends to: (i) generate redundancy, often producing a larger set than necessary, and (ii) occasionally omit relevant articles, yet reference them in the reasoning chain. The Validation approach involved considering subsets of the article list and querying the model to validate if these subsets entail the query. This method simplifies the task for the model by limiting the comparison to a subset of articles relative to the query. Furthermore, we refrained from asking which specific articles were entailing; instead, the model was only asked whether the entire subset was entailing, thus simplifying the task. Based on our observation, the Validation approach demonstrated better performance in identifying concise sets of entailing articles.

Therefore, we simplified the challenge by shifting our focus to a less complex task: validation rather than selection. Specifically, we provided the large LMs with various combinations of the top-k articles and instructed the models to assess whether the information contained in each particular combination was adequate to either validate or refute the query. The aforementioned procedure returned numerous binary (*validate*/*refute*) responses for the possible combinations, necessitating a step of unifying these responses.

In the content that follows, we elaborate on every aspect of this stage, including the creation of prompts, the technique of prompting large LMs and extracting their responses, and the strategy for unifying the diverse responses.

Prompt Templates Creation

We employed the Chain-of-Thought (CoT) prompting method, known for its effectiveness in enhancing large LM performance. A typical CoT prompt encourages the model to provide reasoning prior to concluding, a method known as "*reason-then-answer*". It has shown superiority over the "*answer-then-explain*" and "*answer-only*" techniques, especially in complex domains such as law where methodical reasoning is essential (Nguyen and Nguyen, 2024).

We created our base prompt template which instructs the large LMs to elaborate its reasoning process before delivering an answer, following to the structure below:

> Given the following legal article(s) and legal statement:
>
> Legal article(s): `{content_of_articles}`
>
> Legal statement: `{query}`
>
> Is it possible to verify the correctness of the legal statement using the provided legal article(s), or is the content of the legal article(s) insufficient?
>
> **Please respond with either "The statement is true" or "The statement is false" or**

"Not enough information". Explain first then answer later.

This process yielded a collection of five diverse prompts designed to address issues of prompt sensitivity, a necessary step given the susceptibility of large LMs' performance to the specific prompts employed. The four prompts produced by GPT-4 are:

Assessment of Legal Claim:

Relevant Law(s): `{content_of_articles}` Claim to Verify: `{query}`

Is the claim substantiated by the law(s) cited?

Choose an appropriate response from: "Claim substantiated", "Claim unsubstantiated", or "More information required". Provide the reasoning first, followed by the answer.

Law Conformity Assessment:

Statute(s) Provided: `{content_of_articles}` Assertion: `{query}`

Determine if the assertion is supported by the given statute(s).

Respond with "Assertion valid", "Assertion invalid", or "Insufficient legal context". Start with the reasoning, followed by presenting the conclusion at the end.

Statute Compliance Test:

Legal Text(s): `{content_of_articles}` Hypothesis: `{query}`

Please assess if the hypothesis aligns with the legal text(s).

Your options are "Hypothesis compliant", "Hypothesis non-compliant", or "Can-

not determine compliance". Begin by explaining your reasoning, followed by your conclusion.

Legal Verification:

Legal Provision(s): `{content_of_articles}` Conjecture: `{query}`

Is the conjecture in accordance with the supplied legal provision(s)?

Record your answer as "Conjecture verified", "Conjecture unverified", or "Unable to verify". Provide an explanation first and then state your conclusion.

It could be seen that the prompts adhere to a pattern that would be advantageous for the subsequent extraction of the responses below.

Prompting Large LMs and Extracting Answers from Responses

It was observed that using a few-shot prompting approach is effective when interacting with large LMs. Consequently, we employed this method on large LMs, supplying a set of examples before presenting the main task for the model to resolve. Specifically, we used a $2n$-shot Chain-of-Thought (CoT) prompting, which includes providing n positive examples (i.e., examples with *validate* answers) and n negative examples (i.e., examples with *refute* answers).

Firstly, we created a collection of examples for five different prompts using our training data. In particular, for each prompt, we randomly selected queries from the training data and applied zero-shot CoT prompting to them and their corresponding entailing legal articles until we obtained n examples with *validate* answers, referred to as n base positive examples. To obtain n base negative examples, we paired the queries with the highest-ranked TFIDF articles that are not in the correct set of entailing legal articles. As a result, for every prompt, we had a total of $2n$ base examples in the storage. For the remaining training data, we executed $2n$-shot CoT prompting. For each instance, we chose $2n$ examples from the pre-constructed prompt-specific storages — updating the positive or negative example storages as needed, depending on whether the responses were as expected. These examples were selected based on their relevance to the query under consideration, with the similarity assessed by the LEGAL-BERT model (Chalkidis et al., 2020).

After that, when handling a new query, we provided the large LMs with n positive and n negative examples prior to presenting the main prompt. This way, the models could generate their responses after being exposed to a balanced set of precedents.

The prompts were structured in such a way that responses are composed of a rationale part followed by a conclusion (i.e., an answer for the question whether the provided articles *validates* or *refutes* the given query). Our focus was to interpret the large LMs' binary decision from this conclusion, specifically assessing if the large LMs perceived the provided articles as *validating* or *refuting* the validity of the query. The prompts offered the large LMs with three response options: (i) to affirm that the articles *validates* the query (e.g., "*Assertion valid*", "*Hypothesis compliant*"), (ii) to assert the contrary, that the articles *refutes* the query (e.g., "*Assertion invalid*", "*Hypothesis non-compliant*"), or (iii) to declare that there's insufficient information to judge whether the articles *validate* or *refute* the query (e..g, "*Insufficient legal context*", "*Cannot determine compliance*"). By adhering to this structured approach and reviewing the large LMs' responses, we could extract its choice (i.e., answer) through pattern matching. A choice of either the first or second option indicated the large LM's prediction that the articles are entailing; however, if the third option was chosen, the large LM was indicating an inconclusive or incomplete correlation between the retrieved article set and the true article set, indicating that while some or none of the articles in the provided set might be correct, the set as a whole was not entirely accurate.

Since the large LMs could yield varying generated text upon each execution, we performed each prompt three times and considered the most frequent extracted answer as the model's prediction for the query.

Answer Unification

Up until this point, we had chosen the top-k most relevant articles for each test case using small LMs; after that, for each potential group of articles, we had obtained predictions from large

LMs on whether the group contains the true entailing articles.

For each large LM, we considered only the groups that the large LM recognized as containing the true entailing articles. The problem was to figure out which group of articles was the most concise (i.e., smallest but still complete). Our method involved quantifying how often a particular group was found within the selected groups (including itself). In this step, a group containing a larger number of articles (e.g., group of articles 1, 2, and 4) was assigned a lower score compared to a group with fewer articles from the same set (e.g., group of only articles 2 and 4). To elucidate further, if the model indicated that a group consisting of articles 1, 2, and 4 and a group consisting solely of articles 2 and 4 are both deemed as *validating/refuting* the query, preference is given to the latter group because of its greater conciseness. In the infrequent occurrence where multiple groups achieve the highest score, all the articles from these top-scoring groups are aggregated to form the predicted set.

4.3.3 Refine Stage: Small LMs as Refiners

Algorithm 6 details the process of identifying the optimal prompt template and top-k selection for each large LM, as well as the agreement examination process in the Refine stage. Figure 4.5 provides a visual representation of the agreement examination process.

At this point, for every prompt and top-k value, our framework had generated prediction sets from multiple large LMs. We determined the optimal prompt and top-k value for each of these large LMs by assessing their performance on the validation set. Subsequently, our task was to compile the final set of entailing legal articles, derived from all our predictions.

While all large LMs demonstrated satisfactory overall performance, they exhibited distinct strengths in response to varying queries. Merging their predicted sets can potentially enhance recall; however, this approach results in a significant decline in precision due to a substantial rise in errors (see Table 4.11 in Section 4.4.6). To address this issue, we employed the predictions from small LMs as *refiners* for the predictions extracted from large LMs. This Refine stage notably improved precision relative to the second stage, though with a slight decrease in coverage.

Initially, we merged the prediction sets of large LMs. To refine these predictions, we pro-

Algorithm 6 Refine Stage: Small LMs as Refiners

Require: $\mathcal{Q}$: Set of queries
Require: $\mathcal{P}$: Set of prompt templates
Require: $\mathcal{K}$: Set of different top-k values
Require: $LLMs$: Set of large language models
Require: S_{chkpts}: Set of fifteen small LM checkpoints
Require: r: Top-r value for small LM retrieval (e.g., $r = 6$)

Initialize storage for predictions from all combinations of prompts and top-k values for each query
Initialize storage for large LM predictions across all prompts and top-k values
for each $p \in \mathcal{P}$ **do** ▷ Iterate over each prompt template
  for each $k \in \mathcal{K}$ **do** ▷ Iterate over each top-k value
    for each $q \in \mathcal{Q}$ **do** ▷ Iterate over each query
      Run $LLMs$ on query q with prompt p and top-k articles to get predictions
      Store predictions $P_{q,p,k}$ for query q ▷ Store the predictions for query, prompt, and top-k combination
    end for
  end for
end for

Determine the optimal prompt template and top-k value for each large LM based on validation set performance
Determine the optimal prompt p^* and top-k value k^* for each large LM based on validation set performance
$\mathcal{P}^* \leftarrow \{p^*_{LLM}\}$ for each $LLM \in LLMs$ ▷ Store the optimal prompt for each large LM
$\mathcal{K}^* \leftarrow \{k^*_{LLM}\}$ for each $LLM \in LLMs$ ▷ Store the optimal top-k value for each large LM

Initialize a merged set of predictions from all large LMs for each query
Initialize a merged set of large LM predictions $\mathcal{M}_q$ for each query q: $\mathcal{M}_q \leftarrow \{\}$
for each $LLM \in LLMs$ **do** ▷ Iterate over each large LM
  for each $q \in \mathcal{Q}$ **do** ▷ Iterate over each query
    $\mathcal{M}_q \leftarrow \mathcal{M}_q \cup P_{q,p^*_{LLM},k^*_{LLM}}$ ▷ Merge predictions using optimal prompt and top-k value
  end for
end for

Initialize the final refined set of entailing articles for each query
Initialize final refined set $\mathcal{F}_q$ for each query q: $\mathcal{F}_q \leftarrow \{\}$
for each $q \in \mathcal{Q}$ **do** ▷ Iterate over each query
  $A_{top-r} \leftarrow$ Top r predictions from small LMs for query q (considering duplicate selections) ▷ Get top-r results
  for each article $a \in \mathcal{M}_q$ **do** ▷ Iterate over each article in the merged set
    if $a \in A_{top-r}$ **then** ▷ Check if article is in the top-r results from small LMs
      $\mathcal{F}_q \leftarrow \mathcal{F}_q \cup \{a\}$ ▷ Add article to the final refined set
    end if
  end for
end for
return Final refined sets of entailing articles $\mathcal{F}$ for all queries ▷ Return the final refined sets

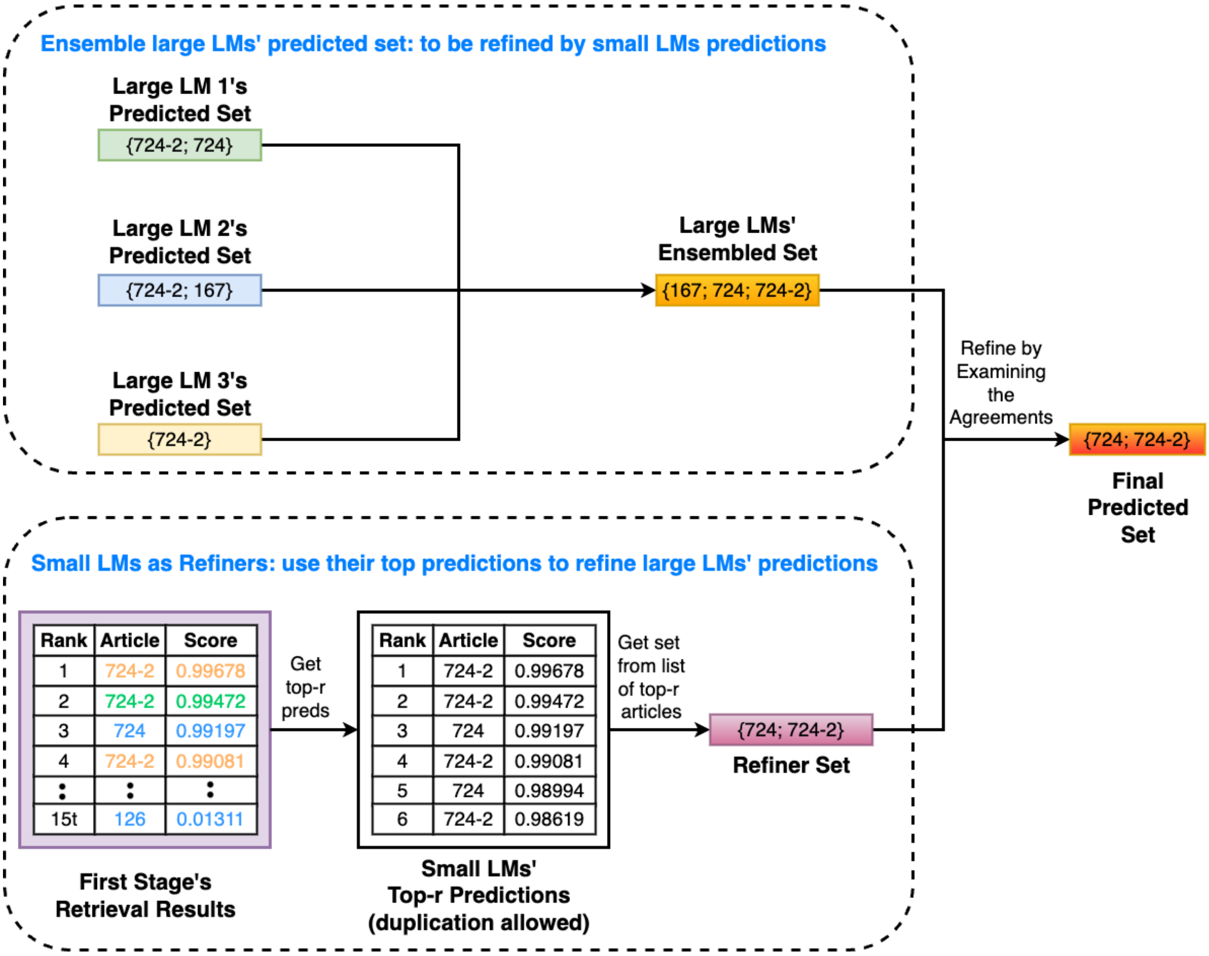

Figure 4.5: The agreement examination process in the Refine stage

posed to utilize the top-r retrieval (duplicate selections allowed) from fifteen small LM checkpoints as *refiners* for the large LM predictions. These smaller checkpoints served to refine the predictions from the larger LMs by filtering out any predictions that did not align with the top-r retrieved articles from the small LMs, resulting in a curated set of legal articles. Essentially, the top-r predictions from the small LMs functioned as a validation step for the predictions made by the large LMs, thereby enhancing the quality by excluding potential errors. Using this approach, we emphasized the agreement between the large group of small LMs and the small group of large LMs. For instance, if the merged predictions from the larger LMs include the articles 1, 2, 4, 7, and 9, but the top-6 (r = 6) results from the small LMs only include the articles 2 and 4 (e.g., four predictions of article 2 and two predictions of article 4), then our final selection would comprise only those two articles. As indicated in Section 4.4.6, it was shown that by selecting the optimal value for r in the top-r approach, the process could improve precision with the trade-off of minor recall score, leading to better retrieved article sets, hence improving the overall performance.

4.4 Experiments

4.4.1 Datasets

For our research, we leveraged the dataset provided by the COLIEE competition (Competition on Legal Information Extraction/Entailment), encompassing the years from 2014 to 2023 (for more details, refer to Section 4.7.2). The dataset was divided into three sections for the purpose of training, validating, and testing our framework. The training set encompassed the years 2014 to 2020 from the COLIEE dataset and was used for model training. The validation of the model and adjustment of its hyperparameters were conducted using the data from COLIEE 2021. To evaluate the efficacy of our framework and test its robustness and generalizability, we utilized two distinct test sets corresponding to the datasets of COLIEE 2022 and COLIEE 2023. Through this temporal splitting approach to our dataset, we guaranteed that our model was subject to legal questions that were not previously introduced in the training phase.

The queries contained within the COLIEE dataset originate from the annual Japanese Bar Examination. The corpus of legal articles comprises the articles from the Japanese Civil Code, amounting to a total of 768 distinct articles. The dataset was initially presented in Japanese; however, the competition organizers supplied participants with high-quality English translations for both the queries and the legal article corpus. Consequently, participants were afforded the opportunity to utilize either the original Japanese versions, the English translations, or a combination of both in the development of their systems.

The statistics provided in Tables 4.2 and Table 4.3 offer insights into the COLIEE dataset's queries for both the Japanese and English versions. The Japanese dataset character counts and the English dataset word counts both exhibit significant variability in query length. However, the nearness of the median to the average values across the datasets suggests that there was no significant skewness in the query length distribution. When considering the average query length within each data split—training, validation, and test—the Japanese queries tend to consist of approximately 75 to 88 characters, reflecting a moderate consistency in length across different years. Similarly, the English queries comprise about 39 to 42 words on average.

It is important to distinguish between the range and the average query lengths. The range, indicated by the minimum and maximum values, shows the broad variation in query lengths. In contrast, the average query length points to the typical size of a query within each subset. Our analysis reveals that the average query length are relatively consistent between different subsets—training, validation, and test—for both the Japanese and English versions of the dataset.

Table 4.2: Sample sizes and query lengths for Japanese version of COLIEE dataset

Data	# samples	Query length (**# characters**)			
		min	max	median	average
Training (COLIEE 2014 - 2020)	806	14	294	69.0	75.5
Validation (COLIEE 2021)	81	21	219	83.0	88.6
Test (COLIEE 2022)	109	29	173	68.0	77.9
Test (COLIEE 2023)	100	25	162	77.0	80.2

Table 4.4 presents statistics illustrating the variations in lengths of articles in the corpus for both Japanese and English datasets. Despite a notable range in article lengths – stretching from as few as 16 to as many as 1120 characters for Japanese data, and from 8 to 663 words for English data – there is a discernible discrepancy between the median and average values in both languages. This disparity suggests that the distribution of article lengths is skewed. Specifically, the data points to a skewed distribution where most articles are shorter, yet a few articles are substantially longer, thereby pushing the average higher than the median.

Histograms depicted in Figures 4.6 and Figure 4.7 support this claim, depicting a concentration of articles on the lower end of the length spectrum, with sparser occurrences of lengthier articles tailing off toward the right. This is reinforced by the statistic that approximately 90% of articles contain 253 or fewer characters in the Japanese dataset and 145 or fewer words in the

Table 4.3: Sample sizes and query lengths for English version of COLIEE dataset

Data	# samples	Query length (**# words**)			
		min	max	median	average
Training (COLIEE 2014 - 2020)	806	6	149	37.0	39.4
Validation (COLIEE 2021)	81	10	79	38.0	39.1
Test (COLIEE 2022)	109	13	92	39.0	42.3
Test (COLIEE 2023)	100	11	87	42.0	42.2

Table 4.4: Lengths of articles in corpus for Japanese and English dataset

Article length	# samples	min	max	median	average
Japanese data (# characters)	768	16	1120	99.5	128.9
English data (# words)	768	8	663	55.0	72.9

English dataset. These figures indicate that while short articles dominate the corpus, outliers of much longer articles contribute to the skewness observed in the distribution of article lengths.

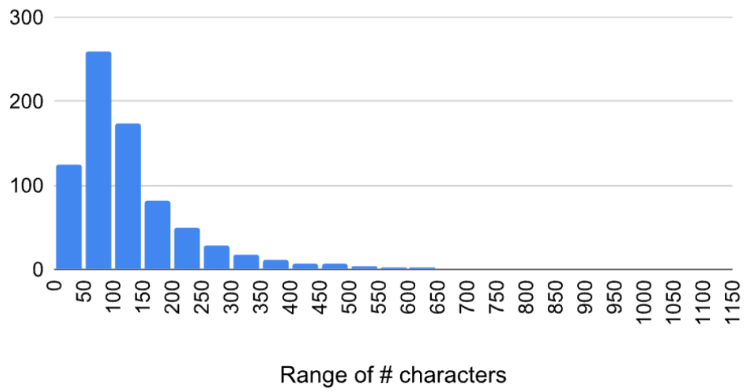

Figure 4.6: Frequency of article lengths (# characters) in Japanese corpus

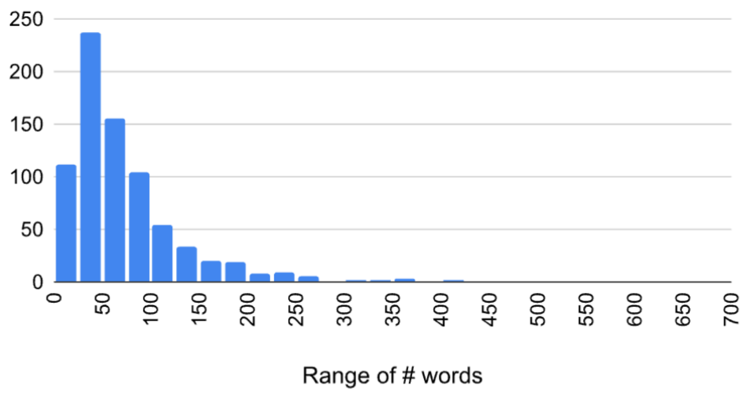

Figure 4.7: Frequency of article lengths (# words) in English corpus

Table 4.5 presents the distribution of the number of entailing articles within the true sets, ranging from one to six, across all dataset splits. This table indicates a consistent trend across all dataset splits of the COLIEE competition, where the overwhelming majority of legal cases are associated with either one or two entailing articles. Notably, in every dataset split, combinations of cases requiring one or two articles account for at least 95% of the cases (see the last column of Table 4.5). It underscores the importance of efficient and accurate retrieval of one or two key articles from a potentially large corpus.

Table 4.5: Proportions of number of entailing articles in the true sets. The mark "-" indicates 0.00%.

Data	Number of entailing articles in the true sets						
	1	2	3	4	5	6	1 and 2
Training (COLIEE 14 - 20)	77.42%	17.74%	3.72%	0.74%	0.25%	0.12%	95.16%
Validation (COLIEE 2021)	80.25%	17.28%	-	2.47%	-	-	97.53%
Test (COLIEE 2022)	86.24%	10.09%	1.83%	0.92%	0.92%	-	96.33%
Test (COLIEE 2023)	72.00%	28.00%	-	-	-	-	100.00%

4.4.2 Evaluation Metrics

In the context of legal article set retrieval, the macro F2 score was specially chosen by the COLIEE organizers as the primary evaluation metric. This choice stems from its alignment with the fundamental objectives of the retrieval process, which functions as a preliminary step in identifying candidate articles for subsequent tasks, such as entailment evaluation. Due to the critical nature of this preliminary stage, it is imperative to prioritize recall to maximize the chance that all true entailing articles are included in the retrieved set. In other words, high recall is essential in minimizing the risk of omitting pertinent information required for subsequent evaluations. Moreover, the adoption of the macro F2 score, which incorporates the macro aspect, ensures a balanced and fair assessment by treating the performance on each query's retrieved set with equal significance, regardless of the variability in the number of relevant articles across different queries. This metric thus facilitates a comprehensive evaluation of the system's efficacy in recalling relevant legal articles, providing a robust foundation for a reliable and thorough entailment analysis in the subsequent stages of the legal document processing workflow.

Given a set of queries $\{Q_1, Q_2, \ldots, Q_M\}$ and their respective sets of entailing legal articles, the macro F2 score is calculated by first computing the F2 score for each individual query and then averaging these scores across all queries. The F2 score for each query Q_i is given by:

$$F2_{Q_i} = \frac{5 \cdot P_{Q_i} \cdot R_{Q_i}}{4 \cdot P_{Q_i} + R_{Q_i}}, \tag{4.4}$$

where P_{Q_i} and R_{Q_i} are the precision and recall for the query Q_i, respectively, calculated as follows:

$$P_{Q_i} = \frac{TP_{Q_i}}{TP_{Q_i} + FP_{Q_i}}, \tag{4.5}$$

$$R_{Q_i} = \frac{TP_{Q_i}}{TP_{Q_i} + FN_{Q_i}}. \tag{4.6}$$

In these equations, TP_{Q_i} represents the true positives (the number of relevant articles correctly retrieved), FP_{Q_i} represents the false positives (the number of irrelevant articles incorrectly retrieved), and FN_{Q_i} represents the false negatives (the number of relevant articles not retrieved) for query Q_i.

The macro F2 score is then calculated as the arithmetic mean of the F2 scores for all queries:

$$\text{Macro F2} = \frac{1}{M}\sum_{i=1}^{M} F2_{Q_i}. \tag{4.7}$$

4.4.3 Competitive Baselines

This section examines the competitive baselines proposed in the COLIEE competitions of 2022 and 2023, with a particular focus on the top-performing systems that demonstrated robust performance. Notably, COLIEE participants were permitted to submit up to three prediction files for evaluation, with each submission potentially employing different methodologies or identical methodologies with varied parameter settings. In our experiments, we compare our framework against the highest-scoring submissions from the baselines to evaluate the efficacy of our proposed framework. The methodologies employed by these baseline models are described below.

- For COLIEE 2022, various methodologies were adopted:
  - **HUKB** (Yoshioka et al., 2022) employed the BM25 information retrieval (IR) model utilizing diverse article databases, including original articles, articles rewritten with references, and the judicial decision parts of articles. The final results were synthesized by combining these outcomes.
  - **JNLP** (Bui et al., 2022b) implemented a deep learning (DL) based approach and differentiated between use-case questions and other question types. They developed

two distinct models—one tailored for ordinal questions and another for use-case questions.

- **LLNTU** employed BERT-based models with feature selections for their submissions.
- **OvGU** (Wehnert et al., 2022) combined scores from a TF-IDF model with sentence-embedding based similarity scores to generate answer rankings. They also incorporated external knowledge to enhance sentence-embedding calculations.
- **UA** utilized both the TF-IDF model and the BM25 model as parts of their IR module.

• Moving on to COLIEE 2023, participants again developed innovative approaches:

- **CAPTAIN** (Nguyen et al., 2023b) employed large LM-based ranking models, using Tohoku BERT for the Japanese language and monoT5 (Nogueira et al., 2020) for English. The most effective system was an ensemble of these two models.
- **HUKB** (Yoshioka and Aoki, 2023) used ensembles of keyword-based IR with varied configurations alongside large LM-based ranking models employing Tohoku BERT.
- **JNLP** (Bui et al., 2024) utilized ensembles that combined BM25 for Japanese and large LM-based ranking model for English, specifically monoT5.
- **LLNTU** employed an ordinal keyword-based system (TFIDF) and introduced keyword emphasis through a named entity recognition system.
- **NOWJ**'s (Vuong et al., 2024) strategy involved a two-stage retrieval approach starting with BM25 followed by re-ranking using an large LM-based model; they utilized the BERT base multilingual model for both English and Japanese languages. Both English and Japanese texts contributed to their final scoring calculation.
- **UA** (Rabelo et al., 2023) used BM25 and TF-IDF, for their IR modules.

4.4.4 Implementation Details

For the initial fine-tuning process in the Retrieval stage, we selected the BERT base Japanese model[1] to fine-tune on the Japanese version of the COLIEE dataset. The justification for choosing a Japanese model, rather than its English counterpart, is substantiated by numerous prior studies (Rabelo et al., 2022a; Kim et al., 2022; Goebel et al., 2023). These studies have consistently indicated marginally superior outcomes when fine-tuning LMs on the Japanese dataset as opposed to the English version. This observation likely arises from the possibility that, despite the high quality of the English translation, it may not achieve absolute perfection. During the same stage, for the purpose of ensembling retrieval lists, we we configured each checkpoint to generate a predetermined number of predictions, denoted as t. Consistent with the value chosen for top-k_1, we set this parameter to $t = 150$.

In the Revise stage, due to the limited availability of robust LLMs specifically tailored for the Japanese language, we incorporated three prominent large LMs: Orca-2 13B (Mitra et al., 2023), Qwen 14B (chat version) (Bai et al., 2023), and Mistral 7B v0.2 (Jiang et al., 2023a). As indicated by their names, the Orca model has 13 billions parameters, the Qwen model has 14 billions parameters, and the Mistral model has 7 billions parameters. These models exhibit exceptional proficiency in processing prompts across multiple languages, although their training has predominantly centered around English datasets. Consequently, we employed these three models for the analysis of the English version of the COLIEE dataset. Within this same stage, we employed a set of five distinct prompt templates. For generating prompts to the large LMs, we applied $2n$-shot chain-of-thought (CoT) prompting. This includes n positive examples and n negative examples, with n set at 2, leading to a 4-shot prompting procedure. If the combined length of these texts exceeded the large LMs' maximum context length, the earlier parts of the text were truncated to fit. Furthermore, the Revise stage involved the selection of a top-k subset of distinct articles for each query, prioritizing those with the highest confidence scores from the Retrieval stage. Through hyperparameter optimization, we trialed values of k from 1 to 5, adopting the value for each large LM that yielded the highest macro F2 score on the validation split.

[1] https://huggingface.co/tohoku-nlp/bert-base-japanese-whole-word-masking

In the process of prompting the large language models (LLMs), the *temperature* parameter was configured to 1, and the *top P* parameter set to 1, in order to promote a diverse range of responses. Each prompt was executed three times, and the majority answer was taken as the prediction for that prompt.

In the Refine stage, we selected the top-r predictions from small LMs to further refine the predictions of the large LMs. The value of r was empirically determined by evaluating a range from 1 to 15. The optimal value, which yielded the best performance on the validation dataset, was found to be 6.

4.4.5 Retrieval-Revise-Refine Framework's Performance

The performance of our framework is compared with the previously mentioned benchmarks for the task of retrieving sets of legal articles, as shown in Table 4.6 and Table 4.7. Within these tables, "Returned" signifies the total count of articles yielded by a given system, while "Retrieved" denotes the subset of those articles that are correctly identified.

Table 4.6: Performance on the COLIEE 2022 data.
We highlight the **highest value** and <u>second-highest value</u> of each column.

Method	Returned	Retrieved	Precision	Recall	Macro F2
LLNTU	114	74	0.6743	0.6391	0.6416
UA	115	90	0.8073	0.7641	0.7638
JNLP (Bui et al., 2022b)	**178**	<u>101</u>	0.6865	0.8378	0.7699
OVGU (Wehnert et al., 2022)	<u>161</u>	96	0.7781	0.8054	0.7790
HUKB (Yoshioka et al., 2022)	136	<u>101</u>	<u>0.8180</u>	<u>0.8405</u>	<u>0.8204</u>
Our framework	131	**106**	**0.8654**	**0.8671**	**0.8517**

In analyzing the presented experimental results, it's clear that the performance of the proposed framework is superior to the competitive benchmarks across both the COLIEE 2022 and COLIEE 2023 datasets for the task of legal article set retrieval.

For the COLIEE 2022 data (Table 4.6), the proposed Retrieval-Revise-Refine framework not only retrieved the highest number of correct articles (106) but also achieved the highest precision (0.8654) and recall (0.8671). *Precision* measures the number of correct returned articles out of all returned articles, which indicates the framework's ability to exclude non-

Table 4.7: Performance on the COLIEE 2023 data.
We highlight the **highest value** and second-highest value of each column.

Method	Returned	Retrieved	Precision	Recall	Macro F2
UA (Rabelo et al., 2023)	108	67	0.6267	0.5700	0.5698
LLNTU	99	74	0.7400	0.6500	0.6600
HUKB (Yoshioka and Aoki, 2023)	172	85	0.6342	0.7150	0.6793
NOWJ (Vuong et al., 2024)	154	90	0.6892	0.7750	0.7345
JNLP (Bui et al., 2024)	**194**	98	0.6517	0.8300	0.7526
CAPTAIN (Nguyen et al., 2023b)	141	92	0.7333	0.8000	0.7645
Our framework	141	**99**	**0.7925**	**0.8350**	**0.8069**

entailing articles effectively. *Recall* measures the number of correct returned articles out of all articles that should have been returned, which shows the framework's ability to capture entailing articles. Consequently, our framework attained a macro F2 score of 0.8517, representing a 3.17% enhancement over the preceding state-of-the-art performance by Yoshioka et al. (2022) (HUKB). Comparatively, HUKB's methodologies achieved the second-best macro F2 score of 0.8204, with a notable high precision (0.8180) and recall (0.8405). Although JNLP (Bui et al., 2022b) returned the highest number of articles (178), its precision was lower than UA, HUKB and our framework. The high number of articles returned by JNLP did, however, contribute to the third-highest recall in the benchmark group (0.8378), but this approach sacrificed precision, leading to a lower F2 score compared to other methods.

For the COLIEE 2023 data (Table 4.7), the proposed framework again demonstrates superior results, retrieving the highest number of correct articles (99) the highest precision (0.7925), and the highest recall (0.8350) among the compared methods. For the given dataset, our model achieved a macro F2 score of 0.8069, representing an improvement of 4.24% over the previous state-of-the-art performance attained by Nguyen et al. (2023b) (CAPTAIN). The CAPTAIN's methodologies showcased solid performance with the second-best macro F2 score (0.7645) and a good balance between precision and recall. JNLP (Bui et al., 2024) returned the highest number of articles (194) in this dataset as well; however, this did not translate into the highest number of correct articles, as they achieved lower precision when compared to our framework.

Overall, the proposed framework exhibits an enhancement over the previous leading systems by a margin of 3.17% for the COLIEE 2022 dataset and 4.24% for the COLIEE 2023 dataset.

The consistent outperformance of the proposed framework across both datasets suggests that the underlying methodology is effective to varying data conditions. The ability to maintain a high level of precision while scaling up recall indicates an advantageous capability to discern relevancy in legal documents, which can be particularly challenging given the complexity and specificity of legal language.

In summary, the experimental results indicate that the proposed Retrieval-Revise-Refine framework establishes a new state-of-the-art benchmark for the task of retrieving concise sets of entailing legal articles in COLIEE 2022 and COLIEE 2023. Additionally, the proposed framework demonstrates superior performance when compared to previous methodologies, both in terms of precision—in accurately identifying relevant articles—and recall—in compiling a comprehensive set of these entailing articles.

4.4.6 Ablation Studies

To clarify the contribution of each stage within our Retrieval-Revise-Refine framework, we conducted a systematic ablation study. We deconstruct our framework into its constitutive elements and evaluate the individual and cumulative effects on the overall retrieval performance. Initially, we assess the standalone capacity of the Retrieval stage. Subsequently, we examine the incremental addition of the Revise stage, namely the Retrieval-Revise combination. Finally, we present the performance metrics when all stages of the framework jointly operate, under various configurations. This step-by-step evaluation enables us to discern the importance of each individual stage and provides an understanding of how they interact and contribute to the overall performance in the context of legal article set retrieval task.

Standalone Retrieval Stage

Utilizing the confidence scores yielded from the initial Retrieval Stage, we obtained a set of top-k unique articles for each query. We experimented with various k values, ranging from 1 to 5. The empirical findings obtained from processing solely the retrieval stage to gather the top-k unique articles per query are demonstrated in Table 4.8 and Table 4.9. To enhance comprehension and visualization of these results, Figures 4.8 and Figure 4.9 graphically depict the changes in precision, recall, and the macro F2-score, using the data presented in the tables.

Table 4.8: Performance of Standalone Retrieval Stage on COLIEE 2022.
We highlight the **highest value** of each column.

# top articles (k)	Returned	Retrieved	Precision	Recall	Macro F2
1	109	92	**0.8440**	0.7671	**0.7744**
2	218	104	0.4771	0.8500	0.7230
3	327	112	0.3425	0.8939	0.6599
4	436	115	0.2638	0.9026	0.5898
5	**545**	**116**	0.2128	**0.9118**	0.5332

Table 4.9: Performance of Standalone Retrieval Stage on COLIEE 2023.
We highlight the **highest value** of each column.

# top articles (k)	Returned	Retrieved	Precision	Recall	Macro F2
1	100	83	**0.8300**	0.7250	**0.7367**
2	200	102	0.5100	0.8500	**0.7367**
3	300	110	0.3667	0.9000	0.6818
4	400	112	0.2800	0.9100	0.6125
5	**500**	**114**	0.2280	**0.9250**	0.5598

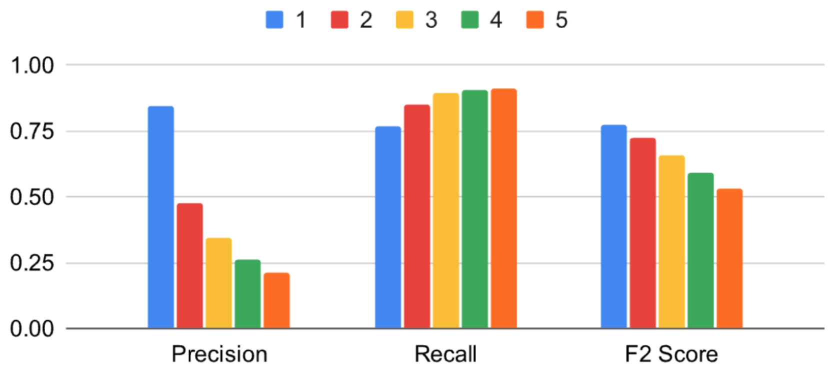

Figure 4.8: Changes in precision, recall, and macro F2-score with different # of top articles for COLIEE 2022

Across both data, an upward trend in recall is observed with the increase in the number of articles retrieved. For instance, the recall value has significantly risen from 0.7671 to 0.9118 for COLIEE 2022, and from 0.7250 to 0.9250 for COLIEE 2023, as we go from retrieving the top-1 to the top-5 articles. These increases illustrate that as we expand the scope of retrieved articles, the system is more likely to include relevant articles among the returned results. It is noteworthy that by retrieving only the top 5 articles, the system is able to achieve a recall score of over 0.9 in both data (0.9118 for COLIEE 2022 and 0.9250 for COLIEE 2023).

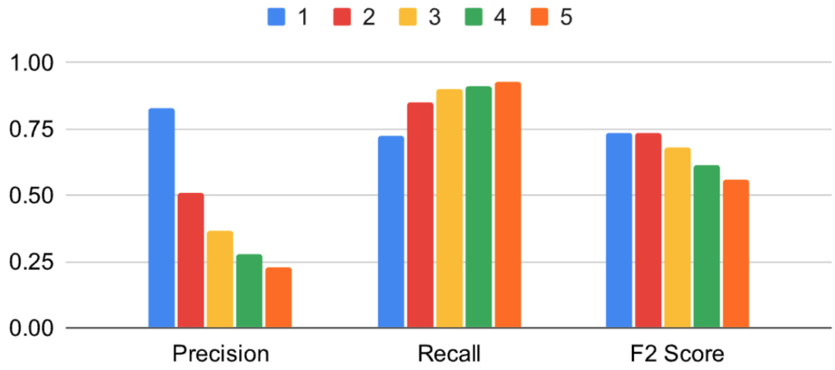

Figure 4.9: Changes in precision, recall, and macro F2-score with different # of top articles for COLIEE 2023

Conversely, precision displays a downward trend with the expansion of the article set size. In COLIEE 2022, precision declines steeply from 0.844 for the top-1 article to 0.2128 for the top-5, while a similar trend in COLIEE 2023 shows a decrease from 0.8300 to 0.2280. This trend underscores a trade-off between precision and recall in the retrieval process: increasing the number of articles retrieved improves the chances of including relevant ones but concurrently retrieves more irrelevant articles, hence diminishing the precision.

The macro F2 Score, which gives more weight to recall than precision, exhibits a peak when selecting one top article and then a decline as more articles are selected. In COLIEE 2022, the macro F2 score achieves 0.7744 when retrieving the top-1 article but then falls down with each subsequent increase in k, dropping to 0.5332 for the top-5. A similar trend is observed in 2023, with the macro F2 score peaks at 0.7367 for the top-1 (and top-2) before decreasing to 0.5598 for the top-5.

Additionally, we have contrasted the performance of our Retrieval stage in terms of coverage with that of BM25, as detailed in Table 4.10. The comparative results of the BM25 algorithm versus our Retrieval stage indicate a consistent trend of outperformance by our Retrieval stage in both COLIEE 2022 and COLIEE 2023 datasets. Across the range of top-1 to top-5 retrieved articles, our Retrieval stage demonstrates a superior recall score. Notably, the analysis of the recall metrics between BM25 and our Retrieval stage reveals a disparity in the performance improvement across the two datasets, COLIEE 2022 and COLIEE 2023. For COLIEE 2022, the coverage gap between BM25 and our retrieval stage is relatively narrow, ranging from 4.89% to 6.48%. In contrast, the gap for COLIEE 2023 is significantly wider, spanning from 10.50% to 22.00%. This pronounced difference suggests that the queries in the COLIEE 2022 dataset likely have a higher degree of keyword overlap with relevant articles, making them more amenable to retrieval by traditional term-matching algorithms like BM25. Conversely, the COLIEE 2023 dataset appears to contain queries with less keyword overlap, benefiting more from our enhanced retrieval stage, which potentially incorporates semantic analysis and other advanced techniques to bridge the gap between query terms and relevant documents. This observation indicates that our Retrieval stage is not only more excel at identifying relevant legal documents but also shows robustness in adapting to new datasets, making it a more reliable option for legal information retrieval tasks.

Table 4.10: Recall comparison between BM25 and our Retrieval stage

	COLIEE 2022		COLIEE 2023	
# top articles	BM25	Our Retrieval Stage	BM25	Our Retrieval Stage
1	0.7182	0.7671 (↑ 4.89%)	0.6200	0.7250 (↑ 10.50%)
2	0.8008	0.8500 (↑ 4.92%)	0.6400	0.8500 (↑ 21.00%)
3	0.8291	0.8939 (↑ 6.48%)	0.6800	0.9000 (↑ 22.00%)
4	0.8382	0.9026 (↑ 6.44%)	0.6900	0.9100 (↑ 22.00%)
5	0.8474	0.9118 (↑ 6.44%)	0.7200	0.9250 (↑ 20.50%)

Retrieval-Revise Combination

During the Revise stage of our experiments, we focused on selecting a top-k group of unique articles in response to each query. This selection was based on the highest confidence levels from the previous Retrieval stage. We experimented with different values of k, varying from 1 to 5, along with five diverse prompt templates, in a process of hyperparameter optimization. We were looking for the combination that would produce the highest macro F2 score on our validation dataset for each large LM.

The results were quite specific to each LM. The Orca model delivered its optimal performance when using the first prompt template and a k value of 5. The Qwen model, on the other hand, reached its peak with the fourth prompt template and a k of 1. Meanwhile, the Mistral model performed best when paired with the third prompt template and a k of 5. These outcomes demonstrate that the large LMs have varying responses to the prompts provided, which highlights the importance of a thorough search for the most effective prompt for each individual large LM.

Table 4.11: Performance of Retrieval-Revise combination on COLIEE datasets.
We highlight the **highest value** of each column.

Data	Large LM	Returned	Retrieved	Precision	Recall	F2 Score
COLIEE 2022	Orca	195	105	0.5765	0.8714	0.7673
	Qwen	109	95	**0.8716**	0.7946	0.8020
	Mistral	130	102	0.8471	0.8515	**0.8350**
	Ensemble of 3	**217**	**111**	0.5459	**0.8965**	0.7712
COLIEE 2023	Orca	171	97	0.6017	0.8250	0.7419
	Qwen	100	85	**0.8500**	0.7400	0.7522
	Mistral	136	96	0.7767	0.8050	**0.7787**
	Ensemble of 3	**207**	**108**	0.5733	**0.8950**	0.7772

Table 4.11 presents the performance of the Retrieval-Revise combination for the COLIEE 2022 and COLIEE 2023 across three different large LMs: Orca, Qwen, and Mistral.

For the COLIEE 2022 dataset, although Orca correctly retrieved the highest number of entailing articles (105) and its recall was the highest among the three models (0.8714), its precision was the lowest among the models (0.5765) and the macro F2 score was 0.7673, the lowest among the three models. Qwen returned fewest correctly retrieved entailing articles (109) (as it only considered top k=1 article), but with the highest precision (0.8716) compared to Orca. Mistral showcased an macro F2 score of 0.8350, the highest among the three models, resulting from a solid combination of precision (0.8471) and recall (0.8515). The results suggest Mistral's robustness for this task compared to the other two.

For the COLIEE 2023 dataset, a similar trend is observed with some shifts in the absolute numbers. The Orca model achieved a macro F2 score of 0.7419. Qwen again showcased a strong precision rate (0.8500). This indicates that precision metric remained as one of Qwen's strong points. Mistral model still outperformed the other two models with an F2 score of 0.7787.

The simple ensemble of predictions from the three large LMs — Orca, Qwen, and Mistral — yielded a decline in the macro F2 score for the COLIEE 2022 dataset, decreasing notably from the highest individual model score of 0.8350, as achieved by Mistral, to 0.7712 for the collective ensemble. For the COLIEE 2023 dataset, the F2 score underwent a minor reduction from Mistral's best single model performance of 0.7787 to 0.7772 when aggregating predictions across the three models. These observations suggest that while a single large LM can yield impressive results, the straightforward combination of multiple LMs does not necessarily enhance overall performance and may even have a negative effect. Therefore, the observed drop in F2 score upon ensembling underscores the necessity for a Refine stage — a methodical approach to aggregate outputs from multiple large LMs and effectively refine them by small LMs' perspectives.

Retrieval-Revise-Refine Framework

During the Refinement stage, we incorporated the top-r predictions from smaller LMs with the objective of refining the output from the large LMs. The optimal r was established through experimental tuning, where we tested a range from 1 to 15 to identify the value that most effectively improved performance on the validation set. The experimentation indicated that the value which yielded the most favorable results on our validation dataset was $r = 6$.

Table 4.12: Performances of best settings for different stage combinations. We highlight the **highest value** of each column. RTN means Returned, RTR means Retrieved.

Data	Stage	RTN	RTR	Precision	Recall	F2 Score
2022	Retrieval (best)	109	92	0.8440	0.7671	0.7744
	Retrieval-Revise (best)	130	102	0.8471	0.8515	0.8350 (+ 8.05%)
	Retrieval-Revise-Refine (best)	**133**	**106**	**0.8654**	**0.8671**	**0.8517** (+ 1.67%)
2023	Retrieval (best)	100	83	**0.8300**	0.7250	0.7367
	Retrieval-Revise (best)	136	96	0.7767	0.8050	0.7787 (+ 4.20%)
	Retrieval-Revise-Refine (best)	**141**	**99**	0.7925	**0.8350**	**0.8069** (+ 2.82%)

The data presented in Table 4.12 summarizes the performance metrics of various stage combinations under the best setting for each stage combination on the COLIEE 2022 and COLIEE 2023 datasets. In the 2022 dataset, the "Retrieval" stage alone returns 109 articles with a precision of 0.8440 and recall of 0.7671, leading to an macro F2 Score of 0.7744. Upon integrating the "Revise" stage, while there is a slight increase in the number of returned articles to 130, both precision and recall values improve to 0.8471 and 0.8515, respectively, resulting in a significant macro F2 Score enhancement to 0.8350, marking an 8.05% improvement. The inclusion of the "Refine" stage further augments performance with 133 returned articles, achieving the highest precision (0.8654), recall (0.8671), and F2 Score (0.8517), noting an additional 1.67% gain. Similarly, in the 2023 dataset, the foundational "Retrieval" stage results in returning 100 articles with a precision of 0.8300 and recall of 0.7250, leading to a F2 Score of 0.7367. Incorporating the "Revise" stage expands the number of returned articles to 136, while maintaining a high recall (0.8050) compared to the best "Retrieval" (0.7250), but with a slight decrease in precision to 0.7767. This stage combination improves the macro F2 Score to 0.7787, reflecting an increase of 4.20%. Progressing to the "Retrieval-Revise-Refine" combination, the framework outputs 141 articles, achieving precision of 0.7925 and notably the highest recall of 0.8350, thereby obtaining an optimal macro F2 Score of 0.8069, marking an additional 2.82% improvement. These consistent trends underscore the benefit of sequentially incorporating "Revise" and "Refine" stages to enhance retrieval accuracy and overall system efficacy across datasets.

Table 4.13: Performances of different stage combinations for best setting for Retrieval-Revise-Refine framework. We highlight the **highest value** of each column. RTN means Returned, RTR means Retrieved.

Data	Stage	RTN	RTR	Precision	Recall	F2 Score
2022	Retrieval	**545**	**116**	0.2128	**0.9118**	0.5332
	Retrieval-Revise	217	111	0.5459	0.8965	0.7712 (+23.80%)
	Retrieval-Revise-Refine	133	106	**0.8654**	0.8671	**0.8517** (+ 8.05%)
2023	Retrieval	**500**	**114**	0.2280	**0.9250**	0.5598
	Retrieval-Revise	207	108	0.5733	0.8950	0.7772 (+21.74%)
	Retrieval-Revise-Refine	141	99	**0.7925**	0.8350	**0.8069** (+ 2.97%)

Table 4.13 showcases performances of different combinations of stages under the best setting for Retrieval-Revise-Refine framework. Analyzing the performance across different stages for the COLIEE 2022 and COLIEE 2023 datasets, we observe a consistent trend within each year as the system progresses from the initial Retrieval stage through subsequent stages (Revise and Refine). For both years, starting with the Retrieval stage only, the system returns the largest number of articles but with the lowest precision. As we proceed to the "Retrieval-Revise" stage combination, there is a notable increase in precision with a slight decrease in recall, indicating that the system is becoming more selective and accurate in retrieving relevant articles, which consequently enhances the macro F2 Score significantly. This trend continues

as we advance to the "Retrieval-Revise-Refine" combination. In this case, while the number of returned articles decreases further, precision sees substantial improvement, indicating the framework's capability to effectively filter out irrelevant articles while maintaining its ability to retrieve a high proportion of relevant articles. The consistent F2 Score improvement at each stage for both years emphasizes that refining the retrieval process contributes to the overall efficacy.

4.4.7 Analysis on Conciseness of Retrieved Sets

Table 4.14: Number of concise retrieved entailing article sets of different set sizes. The mark "-" indicates that there were no queries with that set size. We highlight the **highest value** and second-highest value of each column.

Data	Method	Set size						Macro F2 score
		1	2	3	4	5	6	
COLIEE 2022	JNLP (Bui et al., 2022b)	47	**2**	0	0	0	-	0.7699
	OVGU (Wehnert et al., 2022)	71	1	0	0	0	-	0.7790
	HUKB (Yoshioka et al., 2022)	70	**2**	0	0	0	-	0.8204
	Retrieval-Revise-Refine	**74**	**2**	0	0	0	-	**0.8517**
COLIEE 2023	NOWJ (Vuong et al., 2024)	44	3	-	-	-	-	0.7345
	JNLP (Bui et al., 2024)	36	3	-	-	-	-	0.7526
	CAPTAIN (Nguyen et al., 2023b)	45	1	-	-	-	-	0.7645
	Retrieval-Revise-Refine	**50**	**6**	-	-	-	-	**0.8069**

The data presented in Table 4.14 provides a comparative analysis of the performance of our proposed Retrieve-Revise-Refine framework against the top three methods for each dataset, COLIEE 2022 and COLIEE 2023. It highlights our system's efficacy in retrieving concise sets of entailing legal articles. Specifically, our approach achieves the highest number of concise sets with a set size of 1 and set size of 2 across both datasets. For COLIEE 2022, the Retrieve-Revise-Refine framework retrieved 74 sets of size 1 and 2 sets of size 2, surpassing the HUKB method, which retrieved 70 sets of size 1 and 2 sets of size 2, and achieving a superior macro F2 score of 0.8517 compared to HUKB's 0.8204. For COLIEE 2023, our method retrieved 50 sets of size 1 and 6 sets of size 2, again outperforming the CAPTAIN method's 45 sets of size 1 and 1 set of size 2, leading to a higher macro F2 score of 0.8069 against CAPTAIN's 0.7645.

Besides, for set sizes of 3, 4, 5, and 6, no method was able to achieve a concise set of legal articles, indicating the inherent difficulty in retrieving larger concise sets for this task. While the table specifically focuses on the number of concise sets retrieved, the performance of non-concise larger sets with sizes of 3, 4, 5, and 6 does still impact the final macro F2 score, affecting the quality and comprehensiveness of the retrieved sets. These results collectively underscore the strength of our three-stage Retrieve-Revise-Refine framework in improving conciseness for the task of entailing legal article retrieval, which is crucial this task and for further practical legal applications.

4.5 Case Study

This section presents a detailed case study to illustrate the flow of outputs across the three stages of our proposed Retrieval-Revise-Refine framework, which ultimately leads to the identification of the concise entailing article set for a query under consideration. The specific example analyzed here, as depicted in Figure 4.1, was initially discussed in Section 4.2 where it was demonstrated how *Articles 724* and *Article 724-2* collectively entail the negation of the legal statement presented in the query. Visual representations in Figure 4.2, Figure 4.3, Figure 4.4, and Figure 4.5 significantly facilitate the understanding of this section.

During the Retrieval stage, we combined results from multiple small LM checkpoints (i.e., *TFIDF checkpoints*, *null-result checkpoints*, *bootstrapped checkpoints*). As illustrated in Figure 4.3, the retrieval results consistently ranked *Article 724-2* among the top results. This can be attributed to the fact that while the query comprises 22 words, the first 17 of which (i.e., *"The claim for compensation for loss or damage for death or injury to person caused by tort"*) precisely match the content in *Article 724-2* (Figure 4.10). The top results from these checkpoints also include *Article 724* and *Article 167*, where the latter is incorrectly identified as an entailing article.

A comparative analysis of *Articles 724-2* and *167* is presented in Figure 4.11, revealing significant content overlap. This overlap led to erroneous retrieval of *Article 167* by *TFIDF checkpoints* and *null-result checkpoints*. Notably, this issue was addressed by the bootstrapped

Article 724-2 (Extinctive Prescription of Claim for Compensation for Loss or Damage Arising from Death to Person or Injury to Person Caused by Tort): For the purpose of the application of the provisions of item (i) of the preceding Article with regard to the extinctive prescription of the claim for compensation for loss or damage for death or injury to person caused by tort, the term "three years" in the same item is deemed to be replaced with "five years".
QUERY The claim for compensation for loss or damage for death or injury to person caused by tort is not extinguished by prescription.

Figure 4.10: Comparative analysis of *Article 724-2* and the query, with exact matches highlighted in blue.

checkpoints, which successfully discerned the subtleties in highly similar content.

Article 724-2 (Extinctive Prescription of Claim for Compensation for Loss or Damage Arising from Death to Person or Injury to Person Caused by Tort): For the purpose of the application of the provisions of item (i) of the preceding Article with regard to the extinctive prescription of the claim for compensation for loss or damage for death or injury to person caused by tort, the term "three years" in the same item is deemed to be replaced with "five years".
Article 167 (Extinctive Prescription of Claim for Compensation for Loss or Damage Resulting from Death or Injury to Person): To apply the provisions of paragraph (1), item (ii) of the preceding Article to the extinctive prescription of a claim for damages arising from the death or injury to persons, the phrase "10 years" in that item is deemed to be replaced with "20 years".

Figure 4.11: Comparative analysis of *Article 724-2* and *Article 167*. Identical n-grams ($n \geq 4$) are highlighted in blue.

It is observed that within the first 31 ranked results (duplicate articles allowed), only *Article 724-2*, *Article 166*, and *Article 724* are present. The other two articles within the top-5 distinct retrieval results included *Article 166* and *Article 711*, with *Article 166*'s highest rank was 32 and this number of *Article 711* was 37. This indicates that the Retrieval stage performed commendably in distinguishing entailing from non-entailing articles (as menntioned, this stage achieved a recall score exceeding 0.9 with top-5 distinct retrieval results alone).

As demonstrated in Figure 4.4, the subsequent Revise stage was applied to the top-5 distinct retrieval results from the Retrieval stage using three large LMs. Each LM revised the article set differently: one produced the revised set of 724-2; 724, another produced 724-2; 167, and one produced 724-2. Notably, the large LMs successfully excluded non-entailing articles *Article 166* and *Article 711*. However, only one LM accurately identified the concise entailing articles *Article 724-2* and *724*, while the others still made errors in this stage.

In Figure 4.5, the ensemble of these revised sets produced the set 167; 724; 724-2, which, while comprehensive, was not entirely concise due to the inclusion of redundant *Article 167*.

Revisiting the Retrieval stage results, the top-6 retrieval results (the hyperparameter $r = 6$) included only *Articles 724-2* and *724*. By examining the agreements between the aforementioned sets, we refined the selection to the final set of entailing articles, consisting solely of *Articles 724-2* and *Articles 724*, thus achieving the desired concise set.

4.6 Discussion

4.6.1 Rationale Behind the Refine Stage

The effectiveness of the Refine stage in our Retrieve-Revise-Refine framework is attributed to its strategic utilization of small LMs to validate and consolidate the predictions made by large LMs. This stage capitalizes on the inherent strengths of tailored fine-tuned small LMs in discerning fine-grained legal nuances, thus refining the broader, more generalized predictions from large LMs. By examining the agreement between the retrieved articles from the small LMs and the revised outputs from the large LMs, our framework leverages synergies between diverse models, leading to a more precise and compact set of entailing articles. This approach not only enhances precision by eliminating potentially erroneous or non-relevant articles but also ensures the coverage of the true entailing articles is marginally trade-off, as evidenced by our empirical results. Consequently, this layered methodology helps achieve a notable improvement in the retrieval performance, demonstrating significant gains in the macro F2 score across multiple datasets.

4.6.2 Theoretical Implications

Our proposed three-stage framework, Retrieval-Revise-Refine, employs small LMs for the Retrieval stage, large LMs for revising the outputs of small LMs, and utilizes both small and large LMs for the refinement process. In the Retrieval stage, ensembling predictions from multiple small LMs achieves over 90% coverage with only the top-5 results, indicating that combining multiple small LMs offers broader coverage due to their individual strengths in specific query types. This observation aligns with previous studies (Yoshioka et al., 2022; Nguyen et al., 2023b).

In the Revise stage, large LMs are utilized to revise the predictions made by small LMs. It is observed that different large LMs can exhibit varying performance levels. Interestingly, we found that a simple ensemble of multiple large LMs' predictions may not outperform a single robust large LM. Even though LMs are generally effective, their performance can fluctuate considerably for a specialized task like legal article set retrieval. This variability underscores the importance of training quality and data rather than merely the size of the LMs, which aligns with the findings of Shu et al. (2023). Notably, we observed that smaller-sized LMs, such as Mistral 7B v0.2, can outperform larger ones like Orca 13B or Qwen 14B, again emphasizing that model quality is paramount.

The Refine stage addresses this variability by leveraging the agreement between a large number of small LMs and a smaller ensemble of large LMs. This approach mitigates errors made by larger LMs, reinforcing the importance of integrating decisions from multiple diverse-strength high-quality sources.

4.6.3 Practical Implications

The proposed framework constitutes a substantial advancement in the field of legal article set retrieval, significantly improving upon the current state-of-the-art methodologies. Specifically, our approach achieves an impressive macro F2 score of 0.8517 when evaluated on the COLIEE 2022 dataset and a commendable score of 0.8069 on the COLIEE 2023 dataset. These performance metrics underscore the robustness and efficacy of our framework in accurately retrieving entailing legal articles.

This marked improvement not only establishes a new benchmark for legal retrieval tasks but also provides a solid foundation for subsequent research endeavors aimed at further enhancing automated legal processing capabilities. The high performance across different evaluation datasets signifies the generalizability and adaptability of our model to diverse sets of legal texts.

The framework not only addresses the computational challenges inherent in legal information retrieval but also paves the way for tackling more complex and nuanced tasks within the legal domain. Consequently, this research represents a critical step forward, contributing significantly to the development of sophisticated legal informatics solutions that can more effectively support legal professionals and scholars in their work.

4.6.4 Strengths and Weaknesses of the Proposed Framework

Strengths

Firstly, the Retrieval-Revise-Refine framework offers considerable flexibility in model selection and the number of models used. It allows experimentation with various model types in the Retrieval stage to maximize coverage by ensembling models with diverse strengths across different query types. Additionally, large LMs in the Revise stage can be easily replaced or updated (with newer models) without substantial modifications. Furthermore, the framework exclusively utilizes data provided by the COLIEE competition, demonstrating its robustness without needing external datasets. It obviates the requirement for domain experts to manually select features or specific information such as Named Entity Recognition (NER).

Weaknesses

One of the primary weaknesses of the proposed framework is the necessity for inferences from multiple models, including both small and large LMs. This requirement may pose computational challenges. However, modern advancements in infrastructure and the availability of servers with advanced GPUs can effectively mitigate this issue, ensuring that the framework remains viable for practical applications.

4.7 Related Work

4.7.1 Approaches to Legal Information Retrieval Tasks

Since the concept of *relevance* in legal information retrieval (LIR) differs markedly from that in traditional information retrieval (Van Opijnen and Santos, 2017), the methods employed for LIR are consequently diverse. LIR has developed through various methods, and here we focus on methods based on natural language processing (NLP) and deep learning.

Natural Language Processing-based Methods

The analysis of legal texts using Natural Language Processing (NLP) techniques has obtained significant interest due to the complex nature of extracting meaningful information from these documents. Early approaches were heavily based on *deontic* logic, distinguishing permissions from obligations, and the Hohfeldian system (Hohfeld, 1913), which pairs terms with their opposites and correlatives to analyze legal semantics. For instance, Giorgini et al. (2005) formalized security requirements using Datalog based on ownership and delegation, while Breaux et al. (2006) used Semantic Parameterization (Breaux et al., 2008) to extract rights and obligations from regulations. These methods, however, involve extensive manual rule creation.

The integration of NLP with machine learning has been observed to improve the semantic understanding and retrieval of legal documents. A notable example is the LexNLP package (Bommarito II et al., 2021), which combines an NLP pipeline with machine learning to extract structured information and perform Named Entity Recognition (NER). Soria et al. (2007) created an automated process for legal text analysis, employing tokenization, normalization, and POS-tagging to classify legal paragraphs by their normative content.

Machine learning has also been employed in frameworks such as the Provision Automatic Classifier (Biagioli et al., 2005), which uses a Multiclass Support Vector Machine (Weston and Watkins, 1998) to classify legal provisions after feature selection and weighting operations. Another approach by Sleimi et al. (2021) involved a semantic framework based on an NLP pipeline and heuristic analysis to extract semantic metadata from legal provisions, supporting systematic legal requirement analysis.

Other advanced techniques include the SANAPHOR system by Prokofyev et al. (2015), which enhances reference resolution using a knowledge base to type noun phrases and manage co-references. Gifford (2017) developed *LexrideLaw*, a legal search engine utilizing query expansion and litigation issues ontology to retrieve appellate cases.

Deep Learning-based Methods

Recent advances in deep learning have significantly impacted the processing of legal documents, with applications spanning from classification and summarization to case forecasting and information retrieval. One of the foundational works by Mikolov et al. (2013) introduced the *Feedforward Neural Net Language Model* and *Recurrent Neural Net Language Model*, which efficiently compute word vector representations and maximize the probability of word co-occurrence within a given context (*word2vec*). This methodology laid the groundwork for further research in the legal domain.

Various approaches have leveraged word embeddings for different tasks. For instance, *word2vec* embedding was employed to enhance Named Entity Recognition (Sienčnik, 2015), while Grbovic et al. (2015) combined context and content information to improve query expansion and information retrieval tasks. Nalisnick et al. (2016) explored the use of a Dual Embedding Space Model for legal document ranking by mapping query words and documents into distinct input and output spaces. Bansal et al. (2019) conducted a comprehensive study of various deep learning models, including CNNs (Chen, 2015), RNNs (Liu et al., 2016), LSTMs (Hochreiter and Schmidhuber, 1997), and GRUs (Chung et al., 2014), applied to different tasks within the legal domain.

Domain-specific embedding models have also been developed to better capture the semantics of legal texts. Shao et al. (2020) created a Transformer-based (Vaswani et al., 2017) model named BERT-PLI, a pre-trained BERT model fine-tuned on a law entailment dataset to infer relevance between cases at the paragraph level. Chalkidis et al. (2020) presented LEGAL-BERT, a specialized version of BERT with 12 layers, 768 hidden units, and 12 attention heads, designed to support legal NLP research. Moreover, Tran et al. (2020) developed a methodology that integrates lexical and latent features for legal information retrieval. This approach encodes documents into a continuous vector space using deep neural networks.

Several studies have focused on deep learning models for legal text classification and retrieval. Hammami et al. (2019) used Convolutional Neural Networks (CNNs) with word2vec embeddings to classify French legal content. Da Silva et al. (2018) implemented a CNN-based classification system for court documents, utilizing text extracted via *Optical Character Recognition* (OCR) and NLP pipelines, though the performance was influenced by OCR accuracy.

These approaches highlight the evolving landscape of deep learning applications in the legal domain, emphasizing the need for domain-specific adaptations and hybrid models to improve the accuracy and relevance of legal information retrieval.

4.7.2 COLIEE Competition and Dataset

The field of legal artificial intelligence (AI) has acquired increasing attention from the research community in recent years, motivated by advancements in natural language processing and machine learning. One of the pivotal initiatives driving this development is the Competition on Legal Information Extraction/Entailment (COLIEE). Established in 2014, COLIEE is an annual competition dedicated to the processing and understanding of legal texts. Over the years, the competition has focused on two primary types of legal data: case law and statute law.

The case law data, which are sourced from the Federal Court of Canada, are provided by Compass Law. This dataset includes annotated legal documents, which are used to engage participants in various tasks that assess the systems' capabilities for information retrieval and entailment within the realm of case laws. On the other hand, the statute law data is derived and annotated from the annual Japanese bar exam, which presents a different set of challenges due to the structured nature of statutory texts.

For each type of legal data, the COLIEE competition offers two distinct tasks: retrieval and entailment. The retrieval task involves identifying *relevant* legal documents or parts of documents in response to a query, while the entailment task requires determining whether a particular statement logically follows from the provided legal texts. The subject of our study, legal article set retrieval, specifically corresponds to the retrieval task on statute law data.

Since its establishment in 2014, for the legal article set retrieval, COLIEE has utilized data from the Japanese bar exam, encompassing a broad range of examination years from 2007 to the present. The dataset for this task is thoroughly prepared to include both Japanese text and its corresponding high-quality English translation. This bilingual dataset facilitates international participation and ensures that language barriers do not impede the ability to engage with the task fully.

Moreover, the quality and comprehensiveness of the dataset provided by COLIEE have been fundamental in fostering rigorous research and development. Participants in the competition can leverage these standardized datasets to train their models, allowing for a fair comparison of different approaches.

The COLIEE competition serves as a crucial platform for advancing the field of legal AI by providing standardized datasets and well-defined tasks. It offers a unique opportunity for researchers to benchmark their methodologies against others and to contribute to the progress of legal information extraction and entailment technologies. Over the years, the competition has not only highlighted significant milestones and achievements in the field but has also identified key challenges that continue to drive research efforts. By bringing together a diverse group of participants, from academia to industry, COLIEE acts as a catalyst for innovation and collaboration in the realm of legal AI. It could be said that the COLIEE competition and its carefully curated datasets have played a fundamental role in the advancement of legal AI research. By continuously providing challenging tasks and high-quality data, COLIEE has established itself as a vital event that propels the state-of-the-art in legal text processing and understanding.

4.8 Conclusion

In this paper, we presented a novel three-stage Retrieve-Revise-Refine framework specifically designed to address the challenges in the task of legal article set retrieval, distinguishing it from traditional information retrieval tasks. Our approach effectively combines the strengths of both small and large LMs to achieve a precise and compact set of legal articles that either entail a query or its negation.

Initially, the Retrieval stage leverages multiple fine-tuning strategies on small LMs to ensure comprehensive coverage, thereby maximizing recall. Subsequently, the Revise stage employs large LMs to refine the results from the small LMs by carefully revising combination of the top-k articles. Finally, the Refine stage further improves the precision by cross-referencing the predictions from large LMs with the top-r sets retrieved by smaller LMs.

Empirical evaluations on the COLIEE 2022 and COLIEE 2023 datasets have demonstrated the efficacy of our framework, achieving remarkable improvements in macro F2 scores by 3.17% and 4.24%, respectively over existing methodologies. Our framework also exhibits exceptional performance in retrieving concise sets of entailing articles, as evidenced by the results of our experiments. Furthermore, our ablation studies underscored the systematic contributions of each stage, validating the effectiveness and flexibility of our framework.

The proposed framework represents a constructive step towards improving the automation of legal information processing, effectively addressing the need for a concise set of entailing articles in legal queries. Moving forward, this framework sets a strong foundation for further enhancements in legal technology applications, potentially extending to more complex tasks within the legal domain.

Chapter 5

Legal Text Entailment: A Study on LLM Explanations and Ensemble Predictions

5.1 Overview

The objective of legal text entailment is to ascertain whether the assertions in a legal query logically follow from the information provided in one or multiple legal articles. ChatGPT, a large language model, is robust in many natural language processing tasks, including legal text entailment: when we set the *temperature* = 0 (the ChatGPT answers are deterministic) and prompt the model, it achieves 70.64% accuracy on COLIEE 2022 dataset, which outperforms the previous SOTA of 67.89%. On the other hand, if the *temperature* is larger than zero, ChatGPT answers are not deterministic, leading to inconsistent answers and fluctuating results. We propose to leverage Generative model (Ratner et al., 2016) as an ensemble model to integrate the provisional answers by ChatGPT into consolidated labels. By that way, we treat ChatGPT provisional answers as noisy predictions which can be consolidated by Generative model. The experimental results demonstrate that this approach can attain an accuracy of 76.15%, marking a significant improvement of 8.26% over the prior state-of-the-art benchmark. Additionally, we perform an analysis of the instances where ChatGPT produces incorrect answers, then we classify the errors, offering insights that could guide potential enhancements for future research endeavors.

5.2 Introduction

Legal text entailment is a task in natural language processing (NLP) that involves determining whether a given statement logically follows from the facts stated in a legal text. The development of automated systems for addressing the legal text entailment task is of critical significance, as it has the potential to provide substantial benefits to individuals with varying legal needs. For example, it can help lawyers and legal professionals save time and effort analyzing large volumes of legal texts. Traditionally, lawyers have had to manually read and analyze legal documents to determine the relevant facts and legal arguments. With the help of automated legal text entailment systems, lawyers can quickly identify the most relevant information and arguments, which can help them make more informed decisions. Besides, it is crucial for the development of advanced legal applications such as legal chatbots or legal question-answering systems. These applications can help make legal services more accessible and affordable, particularly for people who cannot afford expensive legal advice.

ChatGPT[1] is a large language model developed by OpenAI[2] that is capable of understanding natural language text and generating human-like responses to prompts. Trained on a massive corpus of text data, ChatGPT has shown impressive performance across a wide range of natural language processing tasks, including language translation, summarization, and question-answering. We are interested in using ChatGPT to analyze legal texts, given its ability to understand the complex and nuanced language used in legal documents. Legal text entailment is one such task where ChatGPT's natural language processing capabilities can be particularly useful.

Generative model (Ratner et al., 2016) is a model that serves as a crucial bridge between data with noisy labels and accurate model predictions. In the case where no gold data is available, and we only have the noisy labels from a variety of information sources like heuristics, expert rules, or crowdsourced annotations, the role of Generative model is to generate probabilistic labels for the data by integrating these weak signals.

In the context of using ChatGPT or similar language models, *temperature* refers to a parame-

[1] https://chat.openai.com/

[2] https://openai.com/

ter that controls the randomness of the generated text. When generating text, higher *temperature* values (e.g., 0.8 or 1.0) make the output more creative and varied, as the model is more likely to select less probable words and phrases. Lower *temperature* values (e.g., 0.1 or 0.3) make the output more deterministic and focused, as the model tends to choose more probable words, resulting in more predictable responses. It is widely recognized that when the *temperature* variable is set to a value other than zero, ChatGPT may yield inconsistent responses for the same prompt. In simpler terms, despite ChatGPT's strong language comprehension capabilities, it remains somewhat unpredictable and prone to variability. Consequently, we can regard the responses from ChatGPT as uncertain provisional answers. Hence, we propose to employ Generative model to refine these provisional answers and generate the final consolidated answers. Our experimental results demonstrate a significant improvement of 5.51% compared to no Generative model employed and 8.26% compared to the previous state-of-the-art benchmark, implying that Generative model is suitable for integrating responses generated by large language models, such as ChatGPT.

In this paper, we preliminarily conduct experiments using the prompt-based configuration of ChatGPT to tackle the task of legal text entailment. The goal is to identify the most effective prompt type among three options: (i) *Answer-only*, (ii) *Answer-then-Explain*, and (iii) *Reason-then-Answer*. The findings reveal that the *Reason-then-Answer* prompt type outperforms the others. Specifically, ChatGPT with the *Reason-then-Answer* prompt achieves a performance boost of 2.75% on the COLIEE 2022 dataset (Kim et al., 2023), achieving an accuracy of 70.64% compared to the previous accuracy of 67.89%.

Subsequently, we employ ChatGPT (utilizing the *Reason-then-Answer* prompt type) to generate multiple answers for each query, resulting in uncertain provisional predictions. To enhance these less-certain provisional predictions, we propose to employ Generative model to refine the results. The proposed strategy leads to a refined prediction that elevates the accuracy to 76.15%, showcasing a substantial improvement of 8.26% over the previous state-of-the-art benchmark.

Furthermore, we conduct an analysis of cases in which ChatGPT generates inaccurate responses. Subsequently, we categorize these errors, providing valuable insights that could in-form potential improvements for future research endeavors.

5.3 Related work

5.3.1 The COLIEE competition

The development of automated systems for addressing legal text entailment is an emerging area of research that has the potential to revolutionize legal services. However, this field is still in its infancy, and much work remains to be done to develop accurate and efficient systems. To this end, the Conference on Legal Information Extraction and Entailment (COLIEE Kim et al. (2023)) has emerged as a prominent forum for advancing the development of automated legal text entailment systems. This annual international competition provides a platform for researchers and practitioners to showcase their latest advances in this field while promoting collaboration and knowledge sharing among participants.

5.3.2 Approaches to legal text entailment in COLIEE competition

The Conference on Legal Information Extraction and Entailment (COLIEE) has facilitated the development of a diverse range of approaches for the task of legal textual entailment. In COLIEE 2020 Rabelo et al. (2021), participants employed a range of NLP techniques and models such as BERT Devlin et al. (2018), RoBERTa Liu et al. (2019b), GloVe Pennington et al. (2014), and LSTM Hochreiter and Schmidhuber (1997). The winning team, JNLP Nguyen et al. (2020), fine-tuned BERT-based models with Japanese legal data and utilized TF-IDF to achieve superior performance. Rule-based ensembles, SVM Cortes and Vapnik (1995), and attention mechanisms with word embeddings were also used to tackle the legal text classification task. In COLIEE 2021 Rabelo et al. (2022a), the winning team HUKB Yoshioka and Aoki (2021) employed an ensemble of BERT models and utilized data augmentation, which outperformed the other approaches Nguyen et al. (2021c); Wehnert et al. (2021b); Kim et al. (2021a); Fujita et al. (2021). The 2022 competition saw further innovations, such as a method for selecting relevant parts from articles and employed an ensemble of BERT with data augmentation Yoshioka and Aoki (2023), an ensemble of rule-based and BERT-based methods with data augmentation and person name inference Fujita et al. (2023), used the longest uncommon subsequence similarity comparison model Lin et al. (2022b), or employed an ensemble of graph neural networks with textbook nodes and sentence embeddings Wehnert et al. (2023). These advances demonstrate the ongoing efforts to improve the performance of automated systems for legal text entailment, with significant implications for the future of legal services.

5.4 Methods

5.4.1 Preliminary experiment: Prompting ChatGPT for legal textual entaiment

The utilization of Chain-of-Thought prompting Wei et al. (2022) has the potential to encourage a more profound level of reasoning within a large language model, thereby leading to improved responses. However, the applicability of Chain-of-Thought prompting might not be suitable for all scenarios Chen et al. (2023a). In certain cases, asking ChatGPT to provide only the answer, without detailing each step of reasoning, could yield better outcomes. As a preliminary experiment, we would like to test ChatGPT with different types of prompts: (i) *Answer-only*, (ii) *Answer-then-Explain*, and (iii) *Reason-then-Answer* (similar to Chain-of-Thought prompting). Figure 5.1 provides an overview of the procedure for prompting ChatGPT to obtain an answer.

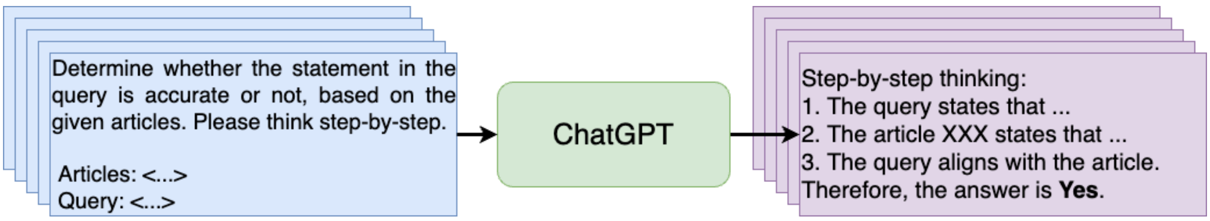

Figure 5.1: ChatGPT prompting procedure

The designed prompts are as follows:

1. *Answer-only*: ChatGPT only outputs the answer.

```
{
  "role": "user",
```

```
  "content": "Given a query (which is delimited with triple
  backticks) and the related articles (which is also delimited
  with triple backticks). Is the query entailed by the related
  articles? Please provide a simple answer of either "Yes" or
  "No", without any explanation.

  Query: ```{query}```

  Related articles: ```{related_articles}```"
}
```

2. *Answer-then-Explain*: ChatGPT outputs the answer and provides an explanation for its reasoning.

```
{
  "role": "user",
  "content": "Given a query (which is delimited with triple
  backticks) and the related articles (which is also delimited
  with triple backticks). Is the query entailed by the related
  articles? Please provide the answer of "Yes" or "No", then
  provide an explanation.

  Query: ```{query}```

  Related articles: ```{related_articles}```"
}
```

3. *Reason-then-Answer*: ChatGPT provides a step-by-step reasoning process and concludes with an answer.

```
{
```

```
    "role": "user",
    "content": "Given a query (which is delimited with triple
    backticks) and the related articles (which is also delimited
    with triple backticks). Is the query entailed by the related
    articles? To answer, please use the following format:
      Step-by-step reasoning: <your step-by-step reasoning>
      Answer: <a clear "Yes" or "No" response>

    Query: ```{query}```

    Related articles: ```{related_articles}```"
  }
```

When prompting ChatGPT, the *temperature* parameter determines the level of randomness in the generated output. In this experiment, in order to ensure deterministic responses, we set the *temperature* parameter to a value of 0.

We experiment with the test data of COLIEE 2022 Kim et al. (2023) and compare them with the previous systems' highest performances. For each test sample, the task involves assessing whether a given statement in a query can be inferred from the related legal articles provided in a list. The obtained experimental results are presented in Table 5.1.

Table 5.1: Results when prompting ChatGPT compared to previous methods

Method	Accuracy
Previous methods	
JNLP	53.21%
UA	54.13%
OvGU	57.80%
LLNTU	60.55%
HUKB	66.97%
KIS	67.89%
Prompt type	
Answer-only	66.97%
Answer-then-Explain	67.89%
Reason-then-Answer	**70.64%**

The results table demonstrates ChatGPT's robustness, with competitive accuracy compared to previous methods. In particular, ChatGPT using the *Reason-then-Answer* prompt demonstrates a performance improvement of 2.75%, reaching an accuracy of 70.64% on the COLIEE 2022 dataset Kim et al. (2023). This is in contrast to the previous accuracy of 67.89%. Notably, the *Reason-then-Answer* prompt yields the highest accuracy, indicating its appropriateness as a prompt type. One possible explanation for this outcome could be as follows: the *Answer-only* prompt relies solely on the model's likelihood to predict tokens such as "Yes" or "No." In contrast, the *Answer-then-Explain* approach, while furnishing explanations for the model's predictions, presents the answers upfront, with the subsequent explanation serving to support the anticipated response. Conversely, the *Reason-then-Answer* approach offers a systematic, step-by-step analysis before arriving at a conclusion, closely emulating human reasoning processes. This emulation appears to contribute to the model's favorable performance. Hence, we use the *Reason-then-Answer* prompt for further experiments.

Table 5.2: Comparison of end-to-end retrieval-entailment performance with entailment using gold standard retrieval

Method	Accuracy
Reason-then-Answer (end-to-end)	67.89%
Reason-then-Answer (gold retrieval)	**70.64%**

Table 5.2 provides a comparative analysis of the performance between end-to-end retrieval-entailment and entailment utilizing gold standard retrieval. The end-to-end system leverages the retrieval outputs generated by the Retrieve-Revise-Refine framework as detailed in Chapter 4. The Retrieve-Revise-Refine framework has demonstrated a commendable macro F2 score of 0.8517, indicating substantial robustness of the end-to-end system. This robustness is evidenced by a mere 2.75% reduction in accuracy when compared to the system utilizing gold standard retrieval. Notably, this performance level, with an accuracy of 67.89%, is on par with the benchmark set by the previous best system, KIS.

5.4.2 Generative model as ensemble model for integrating provisional ChatGPT answers into consolidated answers

In these experiments, we set the *temperature* variable to be non-zero to enable possibilities of ChatGPT producing different answers. As mentioned above, we use the *Reason-then-Answer* prompt for prompting ChatGPT. We experiment with many values of *temperature*: 0.1, 0.2, 0.3, ..., 0.9, 1.0. For each *temperature*, we prompt ChatGPT 10 times. In other words, for each value of *temperature*, there are 10 provisional answers for the queries in the dataset.

Table 5.3 shows some information on the results of prompting ChatGPT 10 times with different *temperature* values. It can be seen that within a single run, the best accuracy could be 76.15% when the *temperature* equals 0.4, 0.6, 0.8, 0.9 or 1.0. However, the corresponding *min* accuracy could be down to 65.14%, 66.06%, or 67.89%. It suggests that the best accuracy above may just because the model got lucky. We can look at the median values to demonstrate this point: the median values are 71.56%, 71.10%, 68.81%, 72.02%, 70.64% for the *temperature* equals 0.4, 0.6, 0.8, 0.9, 1.0, respectively. Another interesting observation is the differences of the performances of the runs (see the row **max-min**): while with *temperature* equals 0.1, the difference is only 4.59%, these values when the *temperature* equals 0.2 to 1.0 ranges from 8.26% to 12.84%.

Table 5.3: Accuracies when prompting ChatGPT 10 times with different *temperature* values. Values in **bold**/<u>underline</u> indicate the **highest**/<u>lowest</u> value in each row.

Temperature	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
max	73.39	75.23	75.23	**76.15**	<u>72.48</u>	**76.15**	75.23	**76.15**	**76.15**	**76.15**
min	**68.81**	66.97	<u>62.39</u>	65.14	64.22	65.14	65.14	66.06	67.89	66.06
max-min	<u>4.59</u>	8.26	**12.84**	11.01	8.26	11.01	10.09	10.09	8.26	10.09
avg	71.83	71.47	69.45	71.74	<u>69.72</u>	71.01	70.46	70.09	**71.93**	71.01
median	**72.48**	71.56	<u>68.81</u>	71.56	69.72	71.10	70.18	<u>68.81</u>	72.02	70.64

Based on those observations, it can be said that ChatGPT could achieve a high but fluctuating performance. Hence, we propose to treat ChatGPT answers as provisional answers, and to leverage the Generative model as the ensemble model to integrate the provisional answers to consolidated answers (Figure 5.2). We perform experiments for each of *temperature* values above with the Generative model.

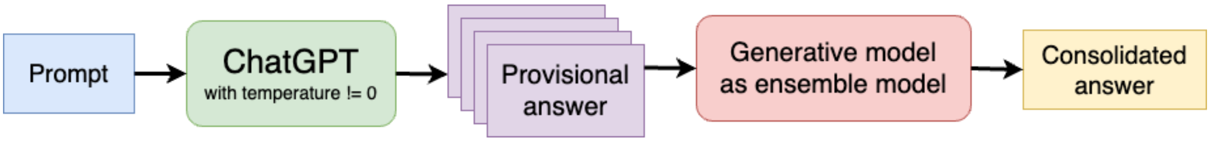

Figure 5.2: Employ Generative model as ensemble model on provisional answers produced by ChatGPT

Table 5.4: Results when employing Generative model on provisional answers. Value in **bold** indicates the highest average accuracy over all *temperature* values.

Ensemble by\Temp	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
Majority voting	72.48	73.39	70.64	73.39	74.31	73.39	74.31	73.39	73.39	71.56
Generative model	74.31	73.39	72.48	73.39	**76.15**	73.39	74.31	71.56	74.31	75.23

Table 5.4 shows the results when employing Generative model as ensemble model on provisional answers. It can be seen that with *temperature* = 0.5, the model could achieve an accuracy of 76.15%. It suggests that the *temperature* = 0.5 may be a good trade-off between the "creativity" value and the deterministic value for ChatGPT in our context.

In summary, we found that the Generative model employed on 10 provisional ChatGPT answers with the *temperature* = 0.5 achieves 76.15% accuracy on the COLIEE 2022 legal text entailment dataset Kim et al. (2023). This accuracy improves 5.51% compared to a single ChatGPT prompting of 70.64% and improves 8.26% over the prior state-of-the-art benchmark of 67.89%, suggesting the effectiveness of Generative model in integrating provisional answers.

We further investigate the performance of Generative model when integrating different numbers of provisional answers. In particular, instead of integrating 10 provisional answers (which achieves 76.15%), we set the number of provisional answers to be from 3 to 9, and employ the Generative model, then report the average accuracy. For example, in case of three provisional answers provided, we consider all combinations of 3 answers out of 10 answers, employ the Generative model for each combination, then calculate the average accuracy. The results are shown in Table 5.5. Interestingly, when the number of provisional answers equals 3, the performance is the lowest while the performance is highest when there are 8 provisional answers. We can also see a trend that, increasing the number of provisional answers, the performance tends to be higher and more stable.

Table 5.5: Performance of Generative model when integrating different numbers of provisional answers. Values in **bold**/<u>underline</u> indicate the **highest**/<u>lowest</u> value in each row.

# provisional answers	**3**	**4**	**5**	**6**	**7**	**8**	**9**
max	6.15	77.06	**77.98**	**77.98**	77.06	**77.98**	**77.98**
min	3.21	66.97	3.21	67.89	68.81	**70.64**	**70.64**
avg	1.51	72.64	72.53	73.59	73.49	**74.29**	73.39
median	1.56	72.48	72.48	73.39	73.39	**74.31**	72.94
max-min	22.94	10.09	**24.77**	10.09	8.25	.34	.34

5.5 Error analysis and classification

To gain a better understanding of the errors made by ChatGPT for the *Reason-then-Answer* prompt, we conducted an analysis over the mis-classified query by ChatGPT and classify the errors into three categories. These errors in ChatGPT's reasoning can be classified as follows:

- Category 1: Incorrect provision of facts by ChatGPT leading to incorrect conclusions. This category accounts for 17.24% of the errors, indicating that ChatGPT has a problem with hallucinations, and it may make up facts to support its hallucinations. For instance, in the scenario below, ChatGPT fabricates a fact during the reasoning process in step 6 (highlighted in yellow), consequently arriving at an incorrect conclusion:

> Related articles:
>
> Article 143:
>
> (1) If a period is provided for in weeks, months, or years, the period is calculated in accordance to the calendar.
>
> (2) If a period does not commence at the beginning of the week, month, or year, that period expires on the day immediately preceding the day which corresponds to the commencement day in the last week, month or year; provided, however, that if the period is provided for in months or years and the last month does not contain a corresponding day, the period expires on the last day of that month.
>
> Query:

If an agreement is made at 11:00 a.m. on July 12 of a given year that the obligation will be fulfilled within one month, the period expires at 12:00 p.m. on August 13 of the same year, both of which do not fall on Sundays, holidays as provided in the Act on National Holidays, or any other holidays.

ChatGPT step-by-step reasoning:

1. The query states that if an agreement is made on July 12 of a given year that the obligation will be fulfilled within one month, the period expires at 12:00 p.m. on August 13 of the same year.

2. The query specifies that the period does not fall on Sundays, holidays as provided in the Act on National Holidays, or any other holidays.

3. Article 143(1) states that if a period is provided for in months, the period is calculated in accordance with the calendar.

4. Article 143(2) states that if a period does not commence at the beginning of the month, that period expires on the day immediately preceding the day which corresponds to the commencement day in the last month.

5. Since the agreement is made on July 12 and the period is for one month, the period will end on August 12.

6. Since August 13 is not a Sunday, holiday as provided in the Act on National Holidays, or any other holiday, the period can be extended to August 13.

7. Therefore, the query is entailed by the related articles.

ChatGPT answer: Yes

Gold answer: No

- Category 2: The inability of ChatGPT to draw correct conclusions from correct premises. This category accounts for 72.42% of the errors, demonstrating that ChatGPT still has limitations in understanding legal texts and reasoning legal scenarios. For instance, in the following scenario, ChatGPT can accurately identify the premises presented in both the query and the related article; however, it fails to draw an appropriate conclusion:

Related articles:

Article 548:

The right to cancel is extinguished if the holder of the right to cancel, intentionally or negligently, causes significant damage to, or makes it impossible to return the object of the contract, or converts the object into another kind of thing by processing or alteration; provided, however, that this does not apply if the holder of the right to cancel does not know of the holder's right to cancel.

Query:

If the holder of the right to cancel negligently makes it impossible to return the object of the contract, the right to cancel is extinguished even if the holder of the right to cancel does not know of the holder's right to cancel.

ChatGPT step-by-step reasoning:

1. The related article states that the right to cancel is extinguished if the holder of the right to cancel intentionally or negligently makes it impossible to return the object of the contract.

2. The related article also states that this provision does not apply if the holder of the right to cancel does not know of their right to cancel.

3. The query states that if the holder of the right to cancel negligently makes it impossible to return the object of the contract, the right to cancel is extinguished even if the holder of the right to cancel does not know of their right to cancel.

4. Therefore, the query is entailed by the related article.

ChatGPT answer: Yes

Gold answer: No

- Category 3: Difficulties faced by ChatGPT while reasoning on "mutatis mutandis" articles. This category accounts for 10.34% of the errors, suggesting that ChatGPT has limitations in linking the non-obvious claims. In the following case, Article 558 pertains to contracts for sale, while Article 559 indicates that these provisions also apply to contracts for value other than contracts for sale with necessary modifications. However, ChatGPT fails to deduce the application of mutatis mutandis in this context:

Related articles:

Article 558:

The expenses associated with contracts for sale are borne equally by both parties.

Article 559:

The provisions of this Section apply mutatis mutandis to contracts for value other than contracts for sale; provided, however, that this does not apply when it is not permitted by the nature of the contract for value.

Query:

The expenses associated with conclusions of lease contracts are borne equally by both parties.

ChatGPT step-by-step reasoning:

1. The query is about the expenses associated with lease contracts.

2. Article 559 states that the provisions of the section apply to contracts for value other than contracts for sale.

3. However, it also states that this does not apply when it is not permitted by the nature of the contract for value.

4. Article 558 states that the expenses associated with contracts for sale are borne equally by both parties.

5. Since the query is specifically about lease contracts and there is no mention of lease contracts in the related articles, we cannot infer that the expenses associated with conclusions of lease contracts are borne equally by both parties. Therefore, the answer is No.

ChatGPT answer: No

Gold answer: Yes

Besides, the lack of relevant articles in the dataset prevents ChatGPT from reaching conclusive answers or may lead to incorrect responses. It suggests that the data could be improved in the future. In the following case, the lack of sufficient related articles led to confusion in ChatGPT, resulting in the model's inability to reach a conclusion. In this case, the articles 140, 141, 142 should be added to the list of related articles.

Related articles:

Article 143:

(1) If a period is provided for in weeks, months, or years, the period is calculated in accordance to the calendar.

(2) If a period does not commence at the beginning of the week, month, or year, that period expires on the day immediately preceding the day which corresponds to the commencement day in the last week, month or year; provided, however, that if the period is provided for in months or years and the last month does not contain a corresponding day, the period expires on the last day of that month.

Query:

If an agreement is made at 5:00 p.m. on January 30 of a non-leap year that the obligation would be fulfilled within one month, the period expires at 12:00 p.m. on February 28 of the same year, both of which do not fall on Sundays, holidays as provided in the Act on National Holidays, or any other holidays.

ChatGPT step-by-step reasoning:

1. The query states that if an agreement is made on January 30 of a non-leap year that the obligation would be fulfilled within one month, the period expires at 12:00 p.m. on February 28 of the same year.

2. Article 143(1) states that if a period is provided for in months, the period is calculated in accordance with the calendar.

3. Article 143(2) states that if a period does not commence at the beginning of the month, the period expires on the last day of that month.

4. Since the agreement in the query is made on January 30 and the period is for one month, the period would expire on February 28.

5. The query specifies that the period would expire at 12:00 p.m. on February 28, but neither Article 143(1) nor Article 143(2) mention anything about a specific time of day.

6. However, the query also specifies that the expiration date must not fall on Sundays, holidays as provided in the Act on National Holidays, or any other holidays, which is not addressed in the related articles.

7. Therefore, it cannot be definitively concluded whether the query is entailed by the related articles.

ChatGPT answer: No

Gold answer: Yes

5.6 Conclusion

In summary, our proposed method involves employing Generative model to integrate the provisional answers produced by ChatGPT into consolidated answers. The results in our experiments show that employing the Generative model to 10 provisional ChatGPT answers, with a *temperature* value of 0.5, yields an accuracy of 76.15% in our task. This showcases a notable enhancement of 8.26% compared to the previously established state-of-the-art benchmark. Furthermore, we conduct an analysis of situations in which ChatGPT provides inaccurate responses. Subsequently, we categorize these errors, providing valuable insights that could direct potential improvements for future research endeavors.

Chapter 6

Robust Predictions with Enhanced Explainability for Legal Textual Entailment

6.1 Overview

We present our winning approach to the VLSP 2023 Vietnamese Legal Textual Entailment Recognition (LTER) challenge. The LTER task examines the ability of models to verify if a given statement can be deduced from related legal articles. Our research outlines a unique framework that employs label models to ensemble predictions from large language models (LLMs), including a fine-tuned Vietnamese Llama-2 model and an off-the-shelf mT0 model (a multi-lingual LLM). Specifically, firstly, we enhance a Vietnamese Llama-2 model for the task of reason-then-answer for LTER, based on the example responses from a more robust LLM. On the other hand, we employ the off-the-shelf mT0 model to provide binary predictions ("entailment" or "not entailment" exclusively) for each sample. The combination of these models seeks to prioritize both accuracy and explanation in the predictions and their corresponding reasonings. To this end, we ensembled these often-noisy predictions using label models, which help adjudicate predictions based on agreements and disagreements among them. In the testing phase, while mT0 demonstrated higher accuracy, it lacked in providing explanations, which our fine-tuned Vietnamese Llama-2 successfully delivered. We employed label models to obtain a balance between model performance and explanatory capability, resulting in an accuracy score of 76.98% in the VLSP 2023 LTER challenge, and securing the first rank in the competition.

6.2 Introduction

In the era of rapid advancements in Artificial Intelligence (AI), particularly in the domain of Natural Language Processing (NLP), the application of AI in legal text analysis and processing has emerged as a critical and indispensable endeavor. Notably, research in NLP for widely spoken languages such as English, Chinese has seen substantial progress (Chalkidis et al., 2021b; Tuggener et al., 2020; Xiao et al., 2018; Duan et al., 2019). However, there exists a conspicuous gap when it comes to the Vietnamese language, especially within the complex and nuanced realm of legal discourse.

In response to this gap, Association for Vietnamese Language and Speech Processing (VLSP), along with ALQAC organizers (Automated Legal Question Answering Competition) (Nguyen et al., 2023a), embarks on pioneering efforts to facilitate the establishing of the foundational groundwork for the application of NLP techniques in the Vietnamese legal domain. Specifically, the VLSP 2023 challenge on Legal Textual Entailment Recognition (LTER)[1] is designed for the intricate task of determining the legal entailment relationship between a given legal statement and the corresponding legal passages — an indispensable foundational element for a spectrum of Legal AI applications, such as legal question answering.

Vietnamese Legal Textual Entailment Recognition (LTER) confronts many challenges. The intricate nature of legal language, which is written with specialized terms and complex structures, poses a substantial challenge for accurate entailment recognition. Next, the scarcity of annotated data for training models in the legal domain, particularly in Vietnamese, hinders the development of robust systems. The shortage of labeled examples limits the model's ability to navigate the nuanced intricacies of Vietnamese legal texts. Additionally, the long and complex sentences in legal documents further amplifies the difficulty, demanding advanced natural language processing capabilities to interpret the intricate relationships within the text.

[1] https://vlsp.org.vn/vlsp2023/eval/lter

Previous work related to this task has explored various approaches, each with its strengths and limitations. One category of models relies on logic programming (Satoh et al., 2010; Nguyen et al., 2023c), incorporating legal knowledge and often requiring human intervention. While these models may demonstrate competence in certain scenarios, their limitations become apparent in terms of flexibility, as they heavily depend on predefined rules and may struggle to adapt to evolving legal contexts. Moreover, the need for human expertise and intervention can be time-consuming and impractical for large-scale applications, hindering their scalability. Over the past few years, the deep learning approach has been widely applied to LTER, starting with small- to medium-sized deep learning models like BERT-based (Devlin et al., 2018), or RoBERTa-based (Liu et al., 2019b). However, these small- and medium-sized deep learning models often exhibit limitations in terms of robustness, leading to suboptimal performance (Yoshioka et al., 2021a; Nguyen et al., 2021b; Fujita et al., 2022; Bui et al., 2022b). Their reliance on a limited capacity for feature extraction and representation learning may make them less capable at capturing the intricacies of legal texts. The scarcity of annotated data, particularly in the context of Vietnamese legal language, further compounds the challenges faced by these smaller models, limiting their ability to generalize effectively. On the other end of the spectrum, large language models (LLMs) (Muennighoff et al., 2022; Touvron et al., 2023a,b), fueled by advancements in deep learning, have showcased impressive performance in various natural language processing tasks, including LTER. However, their success is not without drawbacks. For most of them, one significant limitation lies in their lack of explainability, a critical factor in the field of legal domain applications. For some LLMs with instruction fine-tuning or feedback from human, the explanation may be included, but the ability to provide explanation for legal questions can still be improved (more advanced LLMs like GPT-4 (OpenAI, 2023) or ChatGPT may provide explanations for its answer to a legal question, but unfortunately, they are not open-source). Understanding the reasoning behind a model's decision is very important in legal contexts where transparency and explainability are essential. Hence, there is a need for a framework that is not only robust, but also provide explanations to its predictions.

In addressing the aforementioned challenges posed by existing models in the legal text entailment recognition (LTER) task, our research proposes a novel framework that attempts to balance the strengths of large language models (LLMs) (robustness and explainability) with the employment of label models to ensemble the predictions from LLMs. The ensemble approach in our framework combines predictions from two distinct LLMs: the off-the-shelf mT0 model (Muennighoff et al., 2022) (a multi-lingual LLM) and a fine-tuned Vietnamese Llama-2 model. At the first step, we perform knowledge distillation so that our Vietnamese Llama-2 model[2] is specifically optimized for the reason-then-answer paradigm in LTER, leveraging example responses from a more robust LLM. This fine-tuning process aims to enhance the model's understanding of the intricate nuances present in legal texts, contributing to its effectiveness in providing accurate and contextually relevant predictions. In contrast, the mT0 model, while demonstrating higher accuracy in the testing phase, falls short in terms of providing detailed explanations for its predictions. It exclusively yields binary outcomes, either "entailment" or "not entailment", limiting its explainability. Recognizing the importance of explanation in legal contexts, our framework strategically incorporates the predictions of the two models to deliver a more comprehensive justifications for the final predictions. The decision to employ label models in our ensemble strategy stems from their capacity to address prediction noise and enhance overall model performance. Label models act as adjudicators, leveraging agreements and disagreements among predictions to refine the final output. This approach is particularly valuable in scenarios where individual models may exhibit limitations, such as the trade-off between accuracy and explanatory capability observed in our chosen LLMs.

As demonstrated in the VLSP 2023 LTER challenge, our framework achieved an accuracy score of 76.98%, securing the first rank in the competition. The integration of label models not only contributes to the robustness of our framework but also aligns with the imperative for transparency in legal domain applications. Looking ahead, we expect the advancement of a comprehensible system for the LTER task specifically, and also for other tasks in the legal domain.

[2]https://huggingface.co/chaumng/legal-rte-vn-llama2-7b-instruction

6.3 Related work

6.3.1 The mT0-XXL model

Developed by Google, mT5 (Xue et al., 2020) is a family of pre-trained language models available in various sizes. Notably, mT5 is a multilingual adaptation of T5 (Raffel et al., 2020), undergoing pre-training on a Common Crawl-based dataset that encompasses 101 languages (including Vietnamese), without any supervised training. Consequently, fine-tuning is necessary to make the model suitable for downstream tasks. Based on mT5, mT0 emerges as a Multitask Prompted Finetuning (MTF) variant, innovatively developed by BigScience. mT0 constitutes a diverse set of models proficient in comprehending human instructions across numerous languages in a zero-shot manner. The acquisition of mT0 involves finetuning the mT5 pre-trained multilingual language models on a crosslingual task mixture (namely xP3[3]). Remarkably, the resultant mT0 models showcase crosslingual generalization capabilities for tasks and languages not encountered during finetuning.

The mT0 model family comes with various sizes; we utilize the mT0-XXL[4] model (with 13 billion parameters, representing the largest configuration available) in our framework.

6.3.2 Label models

The label model, a pivotal element within the framework of weak supervision Zhang et al. (2022), plays a crucial role in tackling the challenges posed by limited or noisy labeled datasets. In situations where obtaining precise annotations proves to be costly or impractical, the label model provides an effective solution for generating pseudo-labels from diverse sources of noisy supervision, such as heuristics, crowd-sourcing, or distant supervision. Its primary objective is to deduce the true underlying labels of data points by capitalizing on the consensus or patterns inherent in the noisy annotations.

The necessity for label models emerges from the increasing demand for resilient and scalable machine learning methods, particularly when traditional manual labeling becomes excessively

[3] https://huggingface.co/datasets/bigscience/xP3
[4] https://huggingface.co/bigscience/mt0-xxl

expensive or time-intensive. Label models bridge this gap by furnishing a systematic approach to harnessing the collective insights from multiple noisy sources, resulting in more dependable labeled data for training models.

Functionally, label models operate by ensembling input from various sources and applying statistical techniques to probabilistically estimate the true labels. Notable examples of label models encompass the FlyingSquid model Fu et al. (2020), Dawid-Skene model Dawid and Skene (1979), Hyper label model Wu et al. (2022), FABLE model Zhang et al. (2023a), and Generative model Ratner et al. (2016).

6.4 Methodology

6.4.1 Our legal-rte-vn-llama2-7b-instruction model

Llama-2 (Touvron et al., 2023b) represents an auto-regressive language model family employing an optimized transformer architecture. Through a combination of supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF), the tuned versions of Llama 2 are designed to align with human preferences for both helpfulness and safety. This model family encompasses pretrained and fine-tuned generative text models, ranging from 7 billion to 70 billion parameters. Specifically, the Llama2-chat 7B model[5], a 7-billion-parameter large language model, is optimized for dialogue applications.

The BKAI-HUST Foundation Models Lab has developed the vietnamese-llama2-7b-40GB model[6] based on the aforementioned architecture. This model undergoes a single-epoch continual pre-training, also known as incremental pre-training, utilizing a mixed dataset totaling 40.5 GB. However, practical usage of this model necessitates further supervised fine-tuning (SFT).

In the context of enhancing explanations for answers in a legal textual entailment recognition task, we introduce the legal-rte-vn-llama2-7b-instruction model[7]. This model is developed by conducting further fine-tuning on the vietnamese-llama2-7b-40GB model, specifically on an instruction dataset focused on explainable legal textual entailment. The instruction dataset

[5]https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
[6]https://huggingface.co/bkai-foundation-models/vietnamese-llama2-7b-40GB
[7]https://huggingface.co/chaumng/legal-rte-vn-llama2-7b-instruction

is generated by prompting a more robust large language model. The legal-rte-vn-llama2-7b-instruction model is leveraged to provide explanations for predictions, thereby augmenting the model's explainability.

6.4.2 Our framework

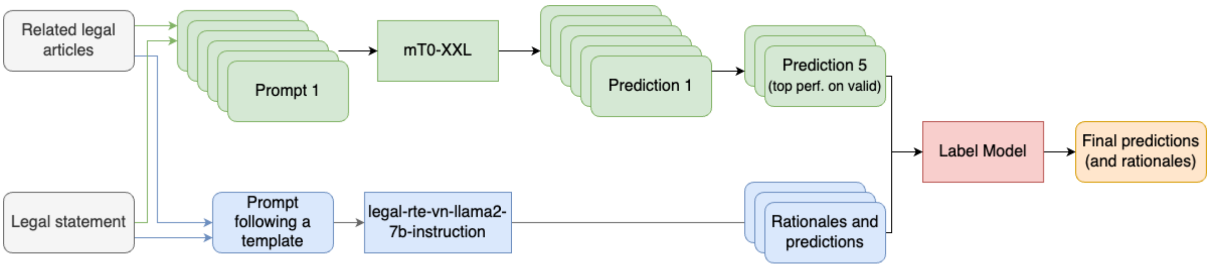

Figure 6.1: Overview of our framework

The effectiveness of large language models has been demonstrated prominently, including in tasks like Legal Textual Entailment Recognition (LTER). Notably, the mT0-XXL model, utilized by the AIEPU team (Hoang et al., 2023), has secured the top rank in the ALQAC 2023 legal textual entailment task, showcasing its capability. Our attempt involved applying this model to the training dataset provided by VLSP. Following the methodology of AIEPU, we employed 50 distinct prompts[8] in Vietnamese, prompting the model to determine if related Vietnamese legal articles entail a given legal statement (without any explanations).

The mT0-XXL model has advantages such as not requiring further fine-tuning while still achieving good performance on the LTER task. This is attributed to its fine-tuning on a crosslingual task mixture, conveying crosslingual generalization capabilities for tasks not encountered during finetuning. However, there are two drawbacks when using this model. The first one is the model's inability to provide explanations for its predictions. In our preliminary experiments, although we attempted to extract explanations from mT0-XXL model, the quality observed was sub-optimal. This limitation hinders the ability to "debug" prompts, as the model outputs are binary (only "entail" or "not entail") without insightful reasoning. Another challenge is the model's sensitivity to the chosen prompt, leading to unstable and fluctuating performances among the 50 prompts.

[8]https://github.com/DucLong06/Legal-Prompts/blob/main/prompts_vn.json

To compensate the lack of explanatory power of mT0-XXL, we incorporated the legal-rte-vn-llama2-7b-instruction model, specifically designed for the Vietnamese legal textual entailment task. This model stands out by providing reasoning before presenting its conclusion on entailment or not. Unlike mT0-XXL, this model's results exhibit less fluctuation across different runs, as it relies on a single prompt template. This model used the prompt template described in figure 6.2 (the English-translation follows the Vietnamese prompt).

Điều luật: "[Một hoặc nhiều điều luật liên quan]"
Legal articles: "[One or many relevant legal articles]"

Dựa vào điều luật trên, đưa ra suy luận của bạn, sau đó đưa ra kết luận về nhận định dưới đây là đúng hay sai.
Based on the above legal articles, provide your rationale, then draw a conclusion about whether the statement below is correct or incorrect.

Nhận định: "[Một nhận định pháp lý]"
Legal statement: "[A legal statement]"

Đưa ra câu trả lời của bạn dưới định dạng sau:
Suy luận: <suy luận của bạn>
Kết luận: <chỉ ghi "Hoàn toàn đúng", "Hoàn toàn sai", hoặc "Có ý sai">
Give your answer in the following format:
Rationale: <your rationale>
Conclusion: <write only "Absolutely correct", "Absolutely incorrect", or "Partly incorrect">

Figure 6.2: Prompt template for the legal-rte-vn-llama2-7b-instruction model

This prompt template above is of the type "reason-then-answer" (give rationale before coming into the conclusion), which is pointed out to be better than the answer-only type (only give the conclusion) and the answer-then-explain (give answer first, then explain later) (Nguyen and Nguyen, 2023). One interesting point of the above prompt is that the legal-rte-vn-llama2-7b-instruction model introduces a nuanced approach in the "Conclusion" section of the prompt. It includes a third option, "Có ý sai" ("Partly incorrect"), recognizing the possibility that legal statements may contain small inaccuracies. This addresses the challenge of large language models considering an entire statement correct, even if a small part is incorrect.

To tackle the issue of unstable results with mT0-XXL, arising from varying prompts, checkpoints, or multiple runs, we employed label models to ensemble predictions from different runs. Several label models were explored, each with unique characteristics:

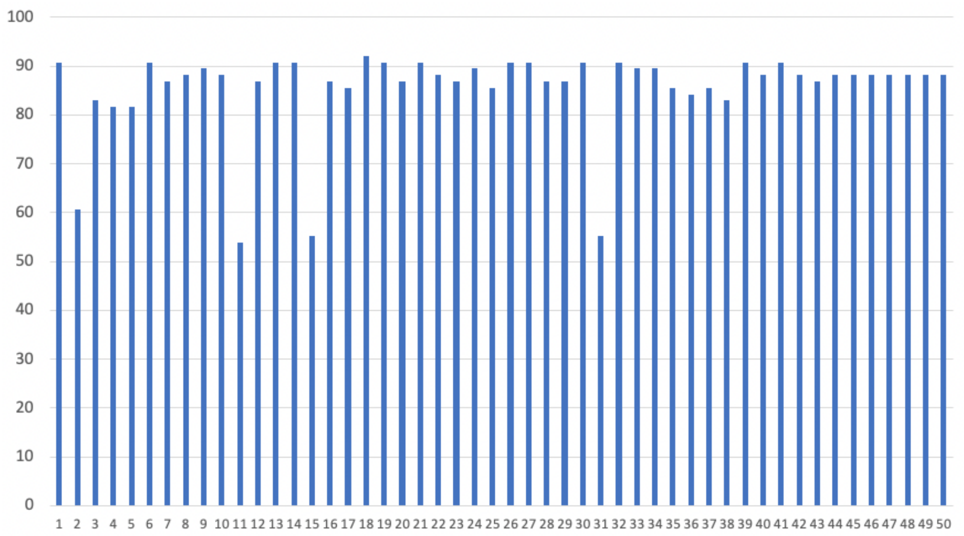

Figure 6.3: Accuracy of mT0-XXL with 50 prompts on the validation set

- Majority voting: Simple label model relying on majority votes from provisional answers.

- FlyingSquid (Fu et al., 2020): It utilizes both agreements and disagreements within provisional answers to construct a labeling model. This model evaluates the accuracy of labeling functions, ultimately designating the answer with the highest probability as the final consolidated answer.

- Dawid-Skene (Dawid and Skene, 1979): It is a probabilistic label model tailored for crowdsourcing scenarios. It estimates true labels and worker reliabilities in situations involving multiple noisy annotators.

- Hyper label model (Wu et al., 2022): It acts as an analytical method for label integration. Constructed using a Graph Neural Network, this model ensures that its predictions remain unchanged or are appropriately adjusted, even when the order of provisional answers is altered.

- FABLE (Zhang et al., 2023a): It stands as a statistical label model built on a mixture of Bayesian label models. Each Bayesian label model aligns with a global pattern of correlation, and the coefficients of the mixture components are predicted by a Gaussian Process classifier based on instance features.

- Generative model (Ratner et al., 2016): It is applied in weak supervision. This model estimates true labels through a "denoising" process applied to the provided provisional answers.

For the legal-rte-vn-llama2-7b-instruction model, we selected the top 3 checkpoints and ran each checkpoint 3 times, resulting in 9 sets of predictions. This approach aimed to enhance prediction coverage. In contrast, for mT0-XXL, we ran 50 prompts and selected the 8 best-performing prompts based on validation dataset accuracy. Given the perceived reliability of the legal-rte-vn-llama2-7b-instruction model in reasoning before answering, more predictions were generated from this model. The final step involved using label models to ensemble the results of all predictions. Figure 1 summarizes our framework.

6.5 Experiments

6.5.1 Dataset

The VLSP LTER training dataset comprises only 76 samples, which is relatively limited. To address this, participants are encouraged to crawl or augment the data if necessary. As our approach does not hinge on a training set, we utilize these 76 samples as the validation dataset to select the optimal label model. The private test dataset includes 140 samples, with one example (id: Hf9ySHQdjI) excluded by the organizers due to inadequate information, leaving 139 samples for testing.

An example from the validation set when applied to the prompt template is shown in figure 6.4.

6.5.2 Experiments

Initially, we employ mT0-XXL with 50 prompts on the validation set (76 samples). Figure 6.3 illustrates that model performance can vary significantly, ranging from the top performance of 92.11% to the lowest achieving 53.95%. The performances across the 50 prompts exhibit notable fluctuations, yet many prompts achieve high accuracy, exceeding 90.00%.

For the legal-rte-vn-llama2-7b-instruction model, we select the top 3 checkpoints and run

Điều luật: "Các hành vi bị nghiêm cấm trong cơ sở giáo dục
1. Xúc phạm nhân phẩm, danh dự, xâm phạm thân thể nhà giáo, cán bộ, người lao động của cơ sở giáo dục và người học.
2. Xuyên tạc nội dung giáo dục.
3. Gian lận trong học tập, kiểm tra, thi, tuyển sinh.
4. Hút thuốc; uống rượu, bia; gây rối an ninh, trật tự. [...]

Dựa vào điều luật trên, đưa ra suy luận của bạn, sau đó đưa ra kết luận về nhận định dưới đây là đúng hay sai.

Nhận định: "Được phép gian lận trong học tập và thi cử. Đồng thời cho phép học sinh được hút thuốc, uống rượu bia và cờ bạc."
Legal statement: "Cheating in studies and exams is allowed. Students are also allowed to smoke, drink alcohol and gamble."

Đưa ra câu trả lời của bạn dưới định dạng sau:
Suy luận: <suy luận của bạn>
Kết luận: <chỉ ghi "Hoàn toàn đúng", "Hoàn toàn sai", hoặc "Có ý sai">

Figure 6.4: Example of prompt for the legal-rte-vn-llama2-7b-instruction model

each checkpoint 3 times (results differ due to randomness in model inference). Table 6.1 displays the performances of these 9 runs. While the top performance does not match that of the mT0-XXL model, the difference is not substantial. Additionally, the legal-rte-vn-llama2-7b-instruction model provides explanations for its predictions, a crucial aspect for the legal field.

To create a stable, high-performance, explanation-provided framework, we experiment with various label models: MajorityVoting, FlyingSquid, DawidSkene, HyperLabelModel, Fable, and Generative model. Table 6.2 showcases the accuracy (on the validation set) of ensem-

Checkpoint	Run	Accuracy (%)
Top checkpoint 1	Run 1	84.21
	Run 2	**88.16**
	Run 3	82.89
Top checkpoint 2	Run 1	85.53
	Run 2	**88.16**
	Run 3	86.84
Top checkpoint 3	Run 1	80.26
	Run 2	80.26
	Run 3	82.89

Table 6.1: Performances of legal-rte-vn-llama2-7b-instruction with 3 checkpoints. **Bold text** indicates highest accuracy.

bling 17 predictions. Generative model stands out with 92.11% accuracy after ensembling the predictions, marking the highest among the label models.

Method	Accuracy (%)
MajorityVoting	90.79
FlyingSquid	90.79
DawidSkene	90.79
HyperLabelModel	89.47
Fable	90.79
Generative model	**92.11**

Table 6.2: Accuracy of different label models

We also experiment with Generative model to ensemble the predictions of mT0-XXL model only (8 predictions) and the predictions of the legal-rte-vn-llama2-7b-instruction model only (9 predictions). Results in table 6.3 suggest that applying the label model (Generative model in this case) to the combined predictions from the two models is favorable. Thus, in the submission, we use Generative model as the label model to ensemble the predictions from the two models combined.

Predictions from	Accuracy (%)
`mT0-XXL`	90.79
`legal-rte-vn-llama2`	86.84
Combined	**92.11**

Table 6.3: Accuracy when employing Generative model to different set of predictions

After using the label model to derive final predictions, we not only maintain high accuracy but also provide explanations for the predictions. For instance, the generated rationale for the example in figure 6.4 is demonstrated in figure 6.5 (generated by the legal-rte-vn-llama2-7b-instruction model).

For the private test submission, we employ Generative model to ensemble 17 predictions, achieving an accuracy of 76.98% (table 6.4). Our framework attains the highest performance on the private test set among the participants. Except for the runner-up (NOWJ team), who achieved a very close performance, our method outperforms other teams by a significant margin. This suggests that the ensemble method used by our team may be a promising approach for this task.

Suy luận: Dựa vào điều luật trên, không có thông tin nào cho phép học sinh được hút thuốc, uống rượu bia và cờ bạc trong cơ sở giáo dục. Điều luật chỉ đề cập đến các hành vi bị nghiêm cấm trong cơ sở giáo dục, bao gồm xúc phạm nhân phẩm, danh dự, xâm phạm thân thể, Gian lận trong học tập, kiểm tra, thi, tuyển sinh, hút thuốc, uống rượu bia và gây rối an ninh, trật tự. Không có thông tin nào cho phép học sinh được vi phạm các quy định này.
Ratioinale: Based on the above legal article, there is no information allowing students to smoke, drink alcohol and gamble in educational institutions. The legal article only mentions prohibited acts in educational establishments, including insulting dignity and honor, physical abuse, cheating in studies, tests, exams, admission, smoking, etc. drinking alcohol and disrupting security and order. There is no information that gives students permission to violate these regulations.

Kết luận: Hoàn toàn sai.
Conclusion: Absolutely incorrect.

Figure 6.5: Example of rationale generated by the legal-rte-vn-llama2-7b-instruction model

Team	Accuracy
HNO3	53.24%
Angels	54.68%
A3N1	71.94%
NOWJ	76.26%
CAN NOT STOP (Ours)	**76.98%**

Table 6.4: Accuracy of participated teams

6.6 Conclusion

In conclusion, our winning approach in the VLSP 2023 LTER challenge, combining the strengths of the fine-tuned Vietnamese Llama-2 and mT0 models through label models, showcases a robust balance between accuracy and explanatory capability. This strategic ensemble resulted in an impressive 76.98% accuracy, securing the first rank in the competition.

Looking forward, potential future work could explore enhancements to the ensemble methodology, perhaps incorporating additional models or refining label model strategies to further improve the system's performance. Additionally, deeper investigations into the interpretability of legal textual entailment models could contribute to the development of more transparent and accountable AI systems in the legal domain. As the field evolves, continued efforts in fine-tuning language models for specific legal tasks and expanding datasets can further refine the system's capabilities and generalizability. Overall, our success in this challenge lays a foundation for future endeavors, urging researchers to delve deeper into the nuanced intersection of language processing and legal text understanding.

Chapter 7

Enhancing Truthfulness in Legal Text Entailment with the Self-itemize Approach for LLMs

7.1 Overview

Maintaining truthfulness in Large Language Models (LLMs) is crucial, particularly in the legal domain where precision and reliability of information are paramount. This research introduces a novel Self-itemize approach, aimed at enhancing the accuracy and truthfulness of LLM outputs. Unlike traditional methods that correct responses post-generation, our approach simplifies complex legal texts prior to response generation via itemizing the content of the input. Experimental results utilizing the COLIEE 2022 dataset reveal that our method achieves a significant improvement in accuracy, outperforming the current state-of-the-art by 5.5%. Additionally, our method demonstrates enhanced truthfulness, achieving a higher average accuracy in reasoning steps compared to existing methodologies. The findings suggest that input itemization can substantially mitigate comprehension challenges in LLMs, particularly in legal entailment tasks.

7.2 Introduction

A response from a large language model (LLM) is truthful when it accurately captures the factual details provided in the input and logically builds upon both the input and established facts without introducing errors. This includes ensuring that the response remains consistent with verified information and that the content generated is logically coherent and contextually aligned with the input. By adhering to these principles, the model ensures the reliability and clarity of the information, avoiding the creation of false or misleading content.

Truthfulness in a large language model is crucial for legal applications due to the high stakes involved in interpreting and applying legal texts. Accurate and reliable information is essential to ensure fair and just outcomes, as even minor errors can misinform decisions and lead to unjust results. Maintaining truthfulness upholds the integrity of legal processes, supports precise interpretations of statutes and case law, and reinforces trust in the legal system.

To enhance LLMs' truthfulness and reduce their hallucinations, many research has explored the integration of human feedback for the refinement of models Kreutzer et al. (2018); Glaese et al. (2022); Ouyang et al. (2022); Scheurer et al. (2023). However, these work involved manual labor that make them costly and lacked of capability to provide instantaneous feedback. To mitigate the dependence on human intervention, an alternative approach involves the deployment of self-correcting large language models (LLMs) that leverage automated feedback. In this iterative process, the model learns from automatically generated feedback signals to comprehend the consequences of its actions and subsequently adapt its behaviors. The sources of automated feedback are diverse, ranging from the LLM itself functioning as the feedback provider (Madaan et al., 2024; Schick et al., 2022), to the utilization of a separately trained feedback model (Yang et al., 2022b; Paul et al., 2023). Additional sources include readily accessible external tools (Gou et al., 2023; Chen et al., 2023b) and external knowledge bases such as Wikipedia or the internet (Yu et al., 2023; Li et al., 2023). Several strategies have been proposed to enhance LLM correction through automated feedback. These include methodologies such as self-training (Huang et al., 2022; Bai et al., 2022), generate-then-rank processes (He et al., 2022; Weng et al., 2022), feedback-guided decoding (Yang et al., 2022a; Xie et al., 2024), and iterative post-hoc revision (Zhang et al., 2023b; Jiang et al., 2023b).

The "self-correct" approaches, although effective when Language Learning Models (LLMs) comprehend the input query well, often encounter difficulties in logical reasoning. This method attempts to enhance the initial response through critiques and refinements. However, in the realm of legal discourse, the intricate structure and domain-specific language of legal texts frequently pose significant comprehension challenges for LLMs, including advanced model like GPT-3.5. As highlighted in Nguyen and Nguyen (2024), GPT-3.5 exhibited deficiencies in legal text entailment tasks by (i) generating erroneous information that led to incorrect conclusions, (ii) demonstrating an inability to logically deduce conclusions from given information, and (iii) struggling with the nuances of legal terminology.

To address these challenges, we introduce the *Self-itemize* approach for LLMs. In contrast to the *self-correct* approach, which amends the initial response post-generation, our method itemizes the input information prior to generating responses. This process involves (i) analyzing the complexities of the input query and identifying the areas where the LLM may falter, and (ii) prompting the LLM to itemize the input query itself before formulating a response based on the itemized query.

Experimental results indicate that our method achieves an absolute improvement of 5.5% in accuracy compared to the preceding state-of-the-art results. Furthermore, it demonstrates enhanced truthfulness in logical reasoning processes.

7.3 Methodology

7.3.1 Legal Text Entailment: Input Query Analysis

We hypothesize that itemizing complex legal articles into more straightforward legal statements will facilitate the task of legal text entailment, making it easier to match the content of such statements with the given query. To test this hypothesis, we analyze a legal corpus, identifying instances where the legal articles are excessively complex and could be broken down into simpler forms. Due to the structured nature and fixed terms of legal articles, we can summarize the phrases that typically contribute to the complexity of legal texts as follows:

Phrase	Description	Example
provided, however	This phrase introduces an exception or a specific condition to the preceding statement. It usually follows a main clause to highlight a significant exception or condition.	A minor must obtain the consent of the minor's legal representative to perform a juridical act; provided, however, that this does not apply to a juridical act for merely acquiring a right or being released from an obligation.
mutatis mutandis	This Latin phrase means "with the necessary changes having been made". It is used to indicate that a rule or statement applies in a comparable way to another situation while allowing for changes in details.	(1) A manifestation of intention that an agent makes indicating that they will be making a manifestation of intention on behalf of the principal within the scope of the agent's authority binds the principal directly. (2) The provisions of the preceding paragraph apply mutatis mutandis to a manifestation of intention that a third party makes to an agent.
The same applies	This phrase indicates that the rule or condition stated in the main clause also applies in a different but similar context.	No action for demand for rescission of fraudulent act may be filed if two years have passed from the time when the obligee came to know that the obligor committed the act knowing that it would be prejudicial to the obligee. The same applies if 10 years have passed from the time of the act.
In such a case	This phrase refers to a specific situation or condition just described and discusses the consequences or actions applicable in that particular scenario.	Even if the obligee exercises the subrogor's right, the obligor is not precluded from independently collecting or otherwise disposing of the subrogor's right. In such a case, the other party is not precluded from performing the obligation to the obligor with respect to the subrogor's right.
Notwithstanding the provisions	This phrase means "in spite of" or "despite." It introduces an exception to a preceding general rule or provision.	(1) A third party obligor of a claim which has been attached may not [...] (2) Notwithstanding the provisions of the preceding paragraph, if a claim acquired after the attachment has arisen from a cause that existed before the attachment, the third party obligor may [...]
In this case	This phrase refers to a particular situation just described and comments on what follows or what actions are required in that specific scenario.	The pledgee may subpledge the thing pledged within the duration of the pledgee's right, upon the pledgee's own responsibility. In this case, the pledgee is responsible for any loss arising from the subpledge even if the same is caused by force majeure.

Table 7.1: Legal phrases and their descriptions

Phrase	# times	in # articles
provided, however	185	170
mutatis mutandis	120	98
The same applies	48	43
In such a case	40	37
Notwithstanding the provisions	24	20
In this case	21	20

Table 7.2: Frequency of legal phrases and their appearance in articles

7.3.2 Self-itemize: Rewrite legal articles

Based on the statistics presented in Table 7.2, we observe that the phrases "provided, however" and "mutatis mutandis" constitute the majority of cases where itemization is feasible. Our focus in this study is on these two phrases. Our approach is straightforward: we instruct GPT-3.5 to first itemize the articles, and subsequently use this itemized version to address the legal text entailment task.

We employ different prompts for each phrase ("provided, however" and "mutatis mutandis"). The phrase "mutatis mutandis" typically references other paragraphs within the same article or different articles, while "provided, however" usually remains within the same legal statement. In scenarios where both phrases appear, we first apply the prompt for "mutatis mutandis," followed by the prompt for "provided, however" to further itemize the rewritten article.

The prompt for "provided, however" is relatively straightforward and is as follows:

Prompt for "provided however"

Legal article: """related_articles"""

Task: Rewrite any exceptions mentioned in this legal article into clear, affirmative statements.

Ensure that the statements are accurate and retain the original meaning.

For "mutatis mutandis," we provide the GPT-3.5 model with a more detailed guide and include an in-context example to enhance comprehension:

Prompt for "mutatis mutandis"

Task: Rewrite the entire article as follows:

- If a paragraph in the article does not contain the phrase "mutatis mutandis," keep it unchanged.

- If a paragraph in the article does contain the phrase "mutatis mutandis," rewrite that paragraph into standalone statements (i.e., statements that do not refer to any preceding paragraph). Ensure to retain the text "; provided, however," if it exists.

Ensure that the statements are accurate and retain their original meaning.

Give one in-context example

Perform the task on the following input:

"""related_articles"""

Following the rewriting of legal articles, we use the "Reason-then-Answer" prompt template as suggested in Nguyen and Nguyen (2024). We set the temperature parameter to 0 to eliminate randomness in the responses. Finally, we extract the answer from the model's response. The employed prompt is as follows:

Main prompt for GPT-3.5

Given a query (which is delimited with triple backticks) and the related articles (which is also delimited with triple backticks).

Is the query entailed by the related articles?

To answer, please use the following format:

Step-by-step reasoning: <your step-by-step reasoning>

Answer: <a clear "Yes" or "No" response>

Query: “‘query“‘

Related articles: “‘related_articles“‘

7.4 Experiments

7.4.1 Dataset

In line with the methodology outlined in Nguyen and Nguyen (2024), we conduct our experiments using the COLIEE 2022 dataset, specifically designed for the legal text entailment task.

7.4.2 Baselines

To benchmark the performance of our approach, we compare it against prior methodologies implemented in the COLIEE 2022 competition, as well as the GPT-3.5 approach detailed in Nguyen and Nguyen (2024). Below is a summary of the approaches utilized in the COLIEE 2022 competition (note that each participant could submit at most 3 submissions; in the result table, we compare our method with the highest performance of the submissions for each participant):

- HUKB (3 runs) (Yoshioka et al., 2022) proposed a method to select relevant part from the articles (HUKB-2) and a new data augmentation method (HUKB-1) in addition to their system in COLIEE 2021 (HUKB-3) which uses an ensemble of BERT with data augmentation, extracting judicial decision sentences, creating positive/negative data from articles.

- JNLP (3 runs) (Bui et al., 2022a) compared ELECTRA, RoBERTa, and LegalBERT, which is pretrained using large legal English texts. They also compared impacts of negation data augmentation, and paragraph-level entailments.

- KIS (3 runs) (Fujita et al., 2022) employed an ensemble of their rule-based method using predicate-argument structures which extends their previous work, and BERT-based methods. Their BERT-based methods commonly use data augmentation (KIS2), with data selection (KIS1), and with person name inference (KIS3). They also employed an ensemble of different trials of finetunings.

- LLNTU (2 runs) (Lin et al., 2022a) restructured given data to a dataset of the disjunctive union strings from training queries and articles, and established a longest uncommon subsequence similarity comparison model, without stopwords (LLNTUdiffSim), and with stopwords (LLNTUdeNgram). One of their runs was retracted because they used their dataset via web crawling that could potentially include the correct answers of the test dataset.

- OvGU (3 runs) (Wehnert et al., 2022) employed an ensemble of graph neural networks (GNNs) as their previous work (OvGU1), concatenated with referring textbook nodes (OvGU2 and averaging sentence embeddings (OvGU3). No submission for the past training datasets.

- UA (3 runs) (Rabelo et al., 2022b) provides no detail description.

7.4.3 Experimental Results

The presented results in Table 7.3 detail a comparative analysis of multiple methodologies employed for a particular task, within the legal domain given the context of COLIEE 2022. Firstly, we observe the accuracies for the approaches from COLIEE 2022, listing methods such as JNLP, UA, OVGU, LLNTU, HUKB, and KIS. Among these, the best-performing method by KIS, achieves an accuracy of 67.89%. This evidences some variability in the effectiveness of those methods, with accuracies ranging from 53.21% (JNLP) to the aforementioned 67.89%. In contrast, the GPT-3.5-based approaches demonstrate a marked improvement in performance. The Reason-then-Answer method (Nguyen and Nguyen, 2024) achieves an accuracy of 70.64%, already surpassing the highest accuracy of the method KIS. Furthermore, our reproduction of this method, yields an accuracy of 72.48%, due to the upgrade of GPT-3.5 along the time. Most notably, our Self-itemize approach attains a notable accuracy of 77.98%. This result distinctly outperforms all other methods listed, including both approaches in COLIEE 2022 and the GPT-3.5-based approach in Nguyen and Nguyen (2024), demonstrating the efficacy of the Self-itemize technique in this context.

Our proposed Self-itemize approach addresses a fundamental challenge faced by many large LMs: the comprehension of complex legal texts. By itemizing the input legal articles into discrete, standalone points, our method facilitates an improved understanding of the input content by the LLMs. This segmentation allows the LLMs to independently evaluate each point against the query, thereby enhancing input comprehension. This improved understanding leads to more accurate logical reasoning and consequently, superior outcomes. The efficacy of the Self-itemize approach is highlighted by a substantial 5.5% increase in accuracy, as compared to other methods, showcasing its potential in the realm of legal text analysis.

Method	**Acc (%)**
Previous approaches	
JNLP	53.21
UA	54.13
OVGU	57.80
LLNTU	60.55
HUKB	66.97
KIS	67.89
GPT-3.5-based approaches	
Reason-then-Answer (reported)	70.64
Reason-then-Answer (reproduced)	72.48
Self-itemize (Ours)	**77.98**

Table 7.3: Comparison of accuracy between methods.

Among all queries in the COLIEE 2022 data, 44.95% of them are with the rewrited legal articles. Because the other queries are not affected by our method compared to the Reason-then-Answer method, we will investigate the truthfulness of only the queries where the corresponding related articles are rewrited by our method. By this way, we serve a meaningful comparison between the reasoning produced by the Reason-then-Answer method versus our Self-itemize method.

For evaluation of truthfulness for GPT-3.5 reasoning, we use a state-of-the-art LLM GPT-4 to evaluate. Specially, we provide GPT-4 with the input query and the response of GPT-3.5 and ask GPT-4 to evaluate the correctness of each reasoning steps in there. We measure the average accuracy over all those queries.

Method	**Acc (%)**
Self-itemize (end-to-end)	73.39
Self-itemize (gold retrieval)	**77.98**

Table 7.4: Self-itemize: Comparison of end-to-end retrieval-entailment performance with entailment using gold standard retrieval

Table 7.4 presents a comparative analysis of the Self-itemize method's performance, contrasting end-to-end retrieval-entailment with entailment using gold standard retrieval. The Retrieve-Revise-Refine framework used in the end-to-end system has demonstrated a notable macro F2 score of 0.8517. The end-to-end Self-itemize approach achieves an accuracy of 73.39%, while the Self-itemize method using gold standard retrieval achieves a higher accuracy of 77.98%. Although there is a difference in performance, the end-to-end system still exhibits considerable robustness and efficacy, as evidenced by its strong accuracy figure that surpasses the performance of Reason-then-Answer approach.

Table 7.5 compares two methods, Reason-then-Answer (Nguyen and Nguyen, 2024) and Self-itemize, in terms of the average number of reasoning steps, the average number of correct reasoning steps, and the average accuracy of the reasoning steps. The Self-itemize method demonstrates a slight increase in the average number of reasoning steps (3.18) compared to the Reason-then-Answer (Nguyen and Nguyen, 2024) method (3.00). This marginal increment suggests that the Self-itemize method tends to utilize a more extended reasoning process. Despite this increase, the method's efficiency is evidenced by the higher average number of correct reasoning steps, which stands at 1.96 for the Self-itemize method, surpassing the 1.53 correctness count of the Reason-then-Answer (Nguyen and Nguyen, 2024) method. Most notably, the average accuracy of the reasoning steps is markedly superior in the Self-itemize method, reaching 59.93%, compared to 51.63% for the Reason-then-Answer (Nguyen and Nguyen, 2024) method. This 8.30% increase in accuracy underscores the enhanced truthfulness of the Self-itemize approach. In summary, while the Self-itemize method introduces a slight increase in the number of reasoning steps, it significantly enhances both the correctness and accuracy of the reasoning process. These findings illustrate the potential advantages of adopting the Self-itemize method for tasks requiring more truthful reasoning steps.

Method	**Avg. no. of RS**	**Avg. no. of correct RS**	**Avg. acc. of RS**
Reason-then-Answer	3.00	1.53	51.63%
Self-itemize (Ours)	3.18	1.96	**59.93%**

Table 7.5: Comparison of reasoning steps and accuracy between methods. RS denotes "reasoning steps"

7.5 Case Study

Table 7.6 presented illustrates the distinction between the "Reason-then-Answer" and "Self-itemize" methods in handling a query related to possession and liability of a damaged movable item. The query describes a scenario in which an individual, A, leases a movable object, X, owned by another individual, B, to a third party, C, without B's permission. The object X is subsequently damaged due to reasons attributable to C, and B requests the return of the item from C. The pivotal legal question is whether C must compensate for the entire loss or damage, even if C believed in good faith that A was the owner of X.

In examining the 'Reason-then-Answer' column, it becomes evident that the method incorrectly concludes with a response of "No." The reasoning begins accurately by acknowledging the query scenario and the belief held by C. However, the method misinterprets the scope of Article 191 by implying that the specific circumstances of leasing without permission and the related liability are not addressed within the article. In doing so, it fails to recognize the broader legal principle articulated in Article 191, which asserts that a possessor without the intention to own must compensate for the entire loss or damage, regardless of their good faith.

Conversely, the Self-itemize method provides a correct and thorough analysis, resulting in the accurate conclusion "Yes." This method meticulously deconstructs the query and aligns it with the nuanced stipulations of Article 191. It comprehensively outlines the principles of liability for possessors, distinguishing between good faith, bad faith, and lack of intention to own. Importantly, Self-itemize correctly identifies that the scenario described in the query falls under the provision that even a good faith possessor must compensate for the entire loss or damage if they did not intend to own the item. This precise alignment with Article 191's provisions ensures a truthful and accurate response to the legal query.

The disparity in reasoning outcomes underscores the importance of comprehensively interpreting legal texts and accurately mapping query scenarios to relevant statutory provisions. The Self-itemize method's success is attributed to its meticulous breakdown of legal principles and accurate contextualization of the query within the statutory framework, whereas the 'Reason-then-Answer' method's oversight leads to an incorrect conclusion due to a narrower interpretation of the legal article.

7.6 Limitations

While the *Self-itemize* approach has demonstrated significant improvements in the accuracy and truthfulness of LLM outputs, it is not without limitations. These limitations are multi-faceted and need to be thoroughly discussed to identify potential areas for future research and improvement.

7.6.1 Scope of Itemization

The primary limitation of our method lies in the scope of itemization. While our approach focuses on itemizing specific phrases such as "provided, however" and "mutatis mutandis," it does not account for the entire spectrum of complex legal terminology and phrasings that may appear in legal documents. Legal texts often contain a plethora of intricate expressions, cross-references, and conditional clauses that can complicate comprehension and entailment tasks. Future work should aim to broaden the range of target phrases and structures for itemization to enhance the overall efficacy of the method.

7.6.2 Generalizability to Other Datasets/Domains

The *Self-itemize* approach has been specifically tailored for legal text entailment dataset of COLIEE competition, which may limit its applicability to other datasets/domains with different textual characteristics and entailment requirements. Generalizing this method to other specialized fields could present unique challenges and would require domain-specific adaptations. Investigations into cross-domain applications and modifications will be essential to broaden the utility of this approach.

7.6.3 Dependence on Rewriting Accuracy

The accuracy of itemized legal articles generated by the LLM is crucial to the success of our *Self-itemize* approach. Misinterpretations or inaccuracies in the itemization stage can lead to erroneous outputs in subsequent entailment tasks. Therefore, while the current prompts have been designed to guide the itemization process effectively, there remains a risk of introducing errors during this stage. Enhancing the accuracy and reliability of the itemization process should be a focus of future research, possibly through improved prompt engineering or more advanced LLMs trained specifically on legal text itemization.

7.7 Conclusion

This paper introduces the Self-itemize approach to improve the truthfulness and accuracy of responses generated by Large Language Models, specifically within the legal domain. By analyzing and itemizing complex legal articles before response generation, this method markedly enhances the model's performance in legal text entailment tasks. Our experimentation using the COLIEE 2022 dataset demonstrates a significant increase of 5.5% in accuracy, positioning it above the current state-of-the-art approaches. Moreover, the truthfulness of logical reasoning is notably improved, evidenced by an 8.30% increase in the accuracy of reasoning steps. This approach not only reduces dependence on manual labor but also addresses the inherent complexity of legal texts, ensuring more reliable and cohesive responses.

Method	Reason-then-Answer	Self-itemize
Query	If A leases movable X owned by B to C without permission, and then X is damaged due to grounds attributable to C, C who is asked by B to return X, must compensate for the entire loss or damage, even if C believed without negligence that A was the owner of X.	
Related article	If a possessed thing has been lost or damaged due to grounds attributable to the possessor, a possessor in bad faith is liable to compensate the person recovering the thing for the entire loss or damage, and a possessor in good faith is liable to compensate that person for the loss or damage to the extent currently enriched as a result of that loss or damage; provided, however, that a possessor that is without the intention to own the thing must compensate the person for the entire loss or damage, even if the possessor has acted in good faith.	If a possessed thing has been lost or damaged due to grounds attributable to the possessor, a possessor in bad faith is liable to compensate the person recovering the thing for the entire loss or damage, and a possessor in good faith is liable to compensate that person for the loss or damage to the extent currently enriched as a result of that loss or damage. However: A possessor without the intention to own the thing must compensate the person for the entire loss or damage, regardless of whether they have acted in good faith.
LLM reasoning	1. The query discusses a scenario where A leases movable X owned by B to C without permission, and then X is damaged due to grounds attributable to C. 2. The query states that C, who is asked by B to return X, must compensate for the entire loss or damage, even if C believed without negligence that A was the owner of X. 3. Article 191 discusses the liability of possessors for lost or damaged things based on their possession status (bad faith, good faith, or without intention to own). 4. Article 191 does not specifically address the scenario of leasing a movable item without permission and the liability in such a situation. 5. The query and Article 191 address different scenarios and legal principles.	1. The query discusses a scenario where A leases movable X owned by B to C without permission, and then X is damaged due to grounds attributable to C. 2. The query states that C, who is asked by B to return X, must compensate for the entire loss or damage, even if C believed without negligence that A was the owner of X. 3. Article 191 of the related articles discusses the liability of possessors in cases where a possessed thing has been lost or damaged due to grounds attributable to the possessor. 4. Article 191 specifies that a possessor in bad faith is liable to compensate the person recovering the thing for the entire loss or damage caused by grounds attributable to the possessor. 5. Article 191 also states that a possessor without the intention to own the thing must compensate the person for the entire loss or damage, regardless of whether they have acted in good faith. 6. The scenario described in the query aligns with the principles outlined in Article 191, as it involves a possessor (C) causing damage to a possessed thing (X) and being required to compensate for the entire loss or damage, even if they believed they were the owner without negligence. 7. Therefore, based on the information provided in the related articles, the query is entailed by the principles outlined in Article 191.
LLM A	No	Yes
Gold A	Yes	

Table 7.6: Comparison of truthfulness in reasoning steps of two methods. LLM A denotes LLM Answer, Gold A denotes Gold Answer.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

In conclusion, this dissertation has laid a robust foundation for the further development and refinement of AI systems tailored to the complex requirements of the legal domain. By advancing methods that enhance performance, transparency, and trustworthiness, this work contributes significantly to the broader effort of harnessing AI for the benefit of the legal profession and society at large.

This dissertation has made several key contributions to the field of legal AI:

- Development of specialized models to address language and document length challenges in legal information retrieval, achieving state-of-the-art performance.
- Introduction of the Retrieve-Revise-Refine framework for retrieving concise legal article sets, significantly improving precision and aligning with task requirements.
- Presentation of a robust approach for legal textual entailment recognition involving the ensemble of multiple models, balancing accuracy and explanatory capability.
- Novel methodology for leveraging label models to enhance the accuracy and consistency of large language models in legal text entailment tasks.
- Innovative Self-itemize approach to enhance truthfulness and accuracy of LLM outputs in legal applications, demonstrating the efficacy of input itemization.

The findings and methodologies presented in this dissertation have several implications for the development of AI systems in the legal domain. The demonstrated improvements in retrieval and entailment tasks pave the way for more advanced applications such as automated legal question answering, legal decision support systems, and predictive analytics in legal contexts.

8.2 Future Work

Future work should continue to explore the integration of various AI models and techniques to further enhance robustness, explainability, and truthfulness. Here, we outline several directions for future research and development that can build on the foundations laid in this dissertation:

- **Advanced Retrieval Techniques**: Building on the Retrieve-Revise-Refine framework, future work could investigate more sophisticated retrieval techniques that incorporate real-time information, contextual relevance, and cross-referencing capabilities. This would enable more nuanced and precise retrieval of legal documents, further improving research efficiency and accuracy in legal work.

- **Explainability and Transparency**: The development of methods to enhance the explainability of legal AI systems remains a critical area of focus. Techniques such as model interpretability tools, coherent rationale extraction, and user-friendly explanations can help legal professionals better understand AI-driven recommendations and decisions. Balancing accuracy with transparency will be vital in gaining the trust of legal practitioners and stakeholders.

- **Ethical and Responsible AI**: Addressing the ethical implications of deploying AI in legal contexts is paramount. Future work should investigate fairness, accountability, and bias mitigation techniques to ensure that AI systems do not perpetuate or exacerbate existing biases within the legal system. Establishing guidelines and frameworks for the responsible use of AI in legal applications will be indispensable as these technologies become more integrated into legal processes.

- **Robustness and Generalization**: Further research should target improving the robustness of legal AI systems to function effectively across different sub-domains of law and diverse jurisdictions. This can be achieved by incorporating more varied and comprehensive legal datasets. Additionally, evaluating the generalization capabilities of the proposed methods on these datasets will help ensure that the systems perform well in real-world scenarios.

By pursuing these directions, future research can contribute to the continuous evolution of legal AI, addressing its present limitations while expanding its capabilities. This will facilitate the creation of more intelligent, reliable, and ethical AI systems that benefit the legal profession and society as a whole.

Chapter 9

Papers and Awards

Papers

1. **Chau Nguyen**, Nguyen-Khang Le, Dieu-Hien Nguyen, Phuong Nguyen, Le-Minh Nguyen, "A Legal Information Retrieval System for Statute Law," in *Asian Conference on Intelligent Information and Database Systems (ACIIDS)*, 2022.

2. **Chau Nguyen**, Phuong Nguyen, Thanh Tran, Dat Nguyen, An Trieu, Tin Pham, Anh Dang, Le-Minh Nguyen, "CAPTAIN at COLIEE 2023: Efficient Methods for Legal Information Retrieval and Entailment Tasks," in *Workshop of the Tenth Competition on Legal Information Extraction/Entailment (COLIEE'2023) in the 19th International Conference on Artificial Intelligence and Law (ICAIL)*, 2023.

3. **Chau Nguyen**, Le-Minh Nguyen, "Employing Label Models on ChatGPT Answers Improves Legal Text Entailment Performance," *New Frontiers in Artificial Intelligence*, 2023.

4. **Chau Nguyen**, Thanh Tran, Khang Le, Hien Nguyen, Truong Do, Trang Pham, Son T. Luu, Trung Vo, Le-Minh Nguyen, "Pushing the Boundaries of Legal Information Pro-

cessing with Integration of Large Language Models," *Lecture Notes in Artificial Intelligence*, 2024.

5. **Chau Nguyen**, Phuong Nguyen, Le-Minh Nguyen, "Retrieve-Revise-Refine: A Novel Framework for Retrieval of Concise Entailing Legal Article Set," *Information Processing and Management*, (revision submitted).

6. **Chau Nguyen**, Thanh Tran, Phuong Nguyen, Le-Minh Nguyen, "A Framework for Legal Textual Entailment Recognition: Robust Predictions With Enhanced Explainability," *Journal of Computer Science and Cybernetics*, (under review).

7. **Chau Nguyen**, Le-Minh Nguyen, "Explanation Generation in LLMs: Enhancing Faithfulness of Legal Text Entailment Explanations with the Self-simplify Method," in *The 16th IEEE International Conference on Knowledge and Systems Engineering (KSE 2024)*.

8. Binh Dang, **Chau Nguyen**, Le-Minh Nguyen, "An Approach to Personalized Legal Information Retrieval System," in *Fifteenth International Workshop on Juris-informatics (JURISIN 2021)*, 2021.

9. Truong Do, **Chau Nguyen**, Vu Tran, Ken Satoh, Yuji Matsumoto, Minh Nguyen, "CovRelex-SE: Adding Semantic Information for Relation Search via Sequence Embedding," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations (EACL)*, 2023.

Awards

- **First Prize - VLSP 2023 Challenge on Legal Textual Entailment Recognition**, VLSP, 2023.

- **First Prize - COLIEE 2023 Task 3: Statute Law Information Retrieval**, COLIEE, 2023.

- **First Prize - COLIEE 2023 Task 4: Legal Textual Entailment Recognition**, COLIEE, 2023.

- **Third Prize - Zalo AI Challenge 2023: Elementary Maths Solving**, Zalo Inc., 2023.

Fellowships

- **Doctoral Research Fellow (DRF)**, JAIST, 2021-2024.

Bibliography

Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., et al. (2023). Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., et al. (2022). Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Bansal, N., Sharma, A., and Singh, R. (2019). A review on the application of deep learning in legal domain. In *Artificial Intelligence Applications and Innovations: 15th IFIP WG 12.5 International Conference, AIAI 2019, Hersonissos, Crete, Greece, May 24–26, 2019, Proceedings 15*, pages 374–381. Springer.

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *arXiv:2004.05150*.

Biagioli, C., Francesconi, E., Passerini, A., Montemagni, S., and Soria, C. (2005). Automatic semantics extraction in law documents. In *Proceedings of the 10th international conference on Artificial intelligence and law*, pages 133–140.

Bommarito II, M. J., Katz, D. M., and Detterman, E. M. (2021). Lexnlp: Natural language processing and information extraction for legal and regulatory texts. In *Research handbook on big data law*, pages 216–227. Edward Elgar Publishing.

Breaux, T. D., Antón, A. I., and Doyle, J. (2008). Semantic parameterization: A process for modeling domain descriptions. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 18(2):1–27.

Breaux, T. D., Vail, M. W., and Anton, A. I. (2006). Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations. In *14th IEEE International Requirements Engineering Conference (RE'06)*, pages 49–58. IEEE.

Bui, M.-Q., Do, D.-T., Le, N.-K., Nguyen, D.-H., Nguyen, K.-V.-H., Anh, T. P. N., and Le Nguyen, M. (2024). Data augmentation and large language model for legal case retrieval and entailment. *The Review of Socionetwork Strategies*, pages 1–26.

Bui, M.-Q., Nguyen, C., Do, D.-T., Le, N.-K., Nguyen, D.-H., Nguyen, T.-T.-T., Nguyen, M.-P., and Nguyen, L.-M. (2022a). Using deep learning approaches for tackling legal's challenges (coliee 2022). In *Sixteenth International Workshop on Juris-informatics (JURISIN)*.

Bui, Q. M., Nguyen, C., Do, D.-T., Le, N.-K., Nguyen, D.-H., Nguyen, T.-T.-T., Nguyen, M.-P., and Nguyen, M. L. (2022b). Jnlp team: Deep learning approaches for tackling long and ambiguous legal documents in coliee 2022. In *JSAI International Symposium on Artificial Intelligence*, pages 68–83. Springer.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., and Androutsopoulos, I. (2020). LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Chalkidis, I., Fergadiotis, M., Manginas, N., Katakalou, E., and Malakasiotis, P. (2021a). Regulatory compliance through doc2doc information retrieval: A case study in eu/uk legislation where text similarity has limitations. *arXiv preprint arXiv:2101.10726*.

Chalkidis, I., Fergadiotis, M., Tsarapatsanis, D., Aletras, N., Androutsopoulos, I., and Malakasiotis, P. (2021b). Paragraph-level rationale extraction through regularization: A case study on european court of human rights cases. *arXiv preprint arXiv:2103.13084*.

Chen, J., Chen, L., Huang, H., and Zhou, T. (2023a). When do you need chain-of-thought prompting for chatgpt? *arXiv preprint arXiv:2304.03262*.

Chen, X., Lin, M., Schärli, N., and Zhou, D. (2023b). Teaching large language models to self-debug. *arXiv preprint arXiv:2304.05128*.

Chen, Y. (2015). Convolutional neural network for sentence classification. Master's thesis, University of Waterloo.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20:273–297.

Da Silva, N. C., Braz, F., De Campos, T., Gusmao, D., Chaves, F., Mendes, D., Bezerra, D., Ziegler, G., Horinouchi, L., Ferreira, M., et al. (2018). Document type classification for brazil's supreme court using a convolutional neural network. In *10th International conference on forensic computer science and cyber law (ICoFCS), Sao Paulo, Brazil*, pages 29–30.

Dawid, A. P. and Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Duan, X., Wang, B., Wang, Z., Ma, W., Cui, Y., Wu, D., Wang, S., Liu, T., Huo, T., Hu, Z., et al. (2019). Cjrc: A reliable human-annotated benchmark dataset for chinese judicial reading comprehension. In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 439–451. Springer.

Fu, D., Chen, M., Sala, F., Hooper, S., Fatahalian, K., and Ré, C. (2020). Fast and three-rious: Speeding up weak supervision with triplet methods. In *International Conference on Machine Learning*, pages 3280–3291. PMLR.

Fujita, M., Kiyota, N., and Kano, Y. (2021). Predicate's argument resolver and entity abstraction for legal question answering: Kis teams at coliee 2021 shared task. In *Proceedings of the COLIEE Workshop in ICAIL*.

Fujita, M., Onaga, T., Ueyama, A., and Kano, Y. (2022). Legal textual entailment using ensemble of rule-based and bert-based method with data augmentation by related article generation. In *JSAI International Symposium on Artificial Intelligence*, pages 138–153. Springer.

Fujita, M., Onaga, T., Ueyama, A., and Kano, Y. (2023). Legal textual entailment using ensemble of rule-based and bert-based method with data augmentation by related article generation. In *New Frontiers in Artificial Intelligence: JSAI-isAI 2022 Workshop, JURISIN 2022, and JSAI 2022 International Session, Kyoto, Japan, June 12–17, 2022, Revised Selected Papers*, pages 138–153. Springer.

Gifford, M. (2017). Lexridelaw: an argument based legal search engine. In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*, pages 271–272.

Giorgini, P., Massacci, F., Mylopoulos, J., and Zannone, N. (2005). Modeling security requirements through ownership, permission and delegation. In *13th IEEE International Conference on Requirements Engineering (RE'05)*, pages 167–176. IEEE.

Glaese, A., McAleese, N., Trebacz, M., Aslanides, J., Firoiu, V., Ewalds, T., Rauh, M., Weidinger, L., Chadwick, M., Thacker, P., et al. (2022). Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*.

Goebel, R., Kano, Y., Kim, M.-Y., Rabelo, J., Satoh, K., and Yoshioka, M. (2023). Summary of the competition on legal information, extraction/entailment (coliee) 2023. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 472–480.

Gou, Z., Shao, Z., Gong, Y., Shen, Y., Yang, Y., Duan, N., and Chen, W. (2023). Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.

Grbovic, M., Djuric, N., Radosavljevic, V., Silvestri, F., and Bhamidipati, N. (2015). Context- and content-aware embeddings for query rewriting in sponsored search. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 383–392.

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., and Poon, H. (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthcare*, 3(1).

Hammami, E., Akermi, I., Faiz, R., and Boughanem, M. (2019). Deep learning for french legal data categorization. In *Model and Data Engineering: 9th International Conference, MEDI 2019, Toulouse, France, October 28–31, 2019, Proceedings 9*, pages 96–105. Springer.

He, H., Zhang, H., and Roth, D. (2022). Rethinking with retrieval: Faithful large language model inference. *arXiv preprint arXiv:2301.00303*.

Hearst, M., Dumais, S., Osuna, E., Platt, J., and Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28.

Hoang, L., Bui, T., Nguyen, C., and Nguyen, L.-M. (2023). Aiepu at alqac 2023: Deep learning methods for legal information retrieval and question answering. In *2023 15th International Conference on Knowledge and Systems Engineering (KSE)*, pages 1–6. IEEE.

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

Hohfeld, W. N. (1913). Some fundamental legal conceptions as applied in judicial reasoning. *Yale Lj*, 23:16.

Huang, J., Gu, S. S., Hou, L., Wu, Y., Wang, X., Yu, H., and Han, J. (2022). Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. (2023a). Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jiang, S., Wang, Y., and Wang, Y. (2023b). Selfevolve: A code evolution framework via large language models. *arXiv preprint arXiv:2306.02907*.

Kim, K., Hong, K., and Rhim, Y.-Y. (2017). Lstm based legal text representation learning.

Kim, M., Rabelo, J., and Goebel, R. (2021a). Bm25 and transformer-based legal information extraction and entailment. In *Proceedings of the COLIEE Workshop in ICAIL*.

Kim, M.-Y., Rabelo, J., and Goebel., R. (2021b). Bm25 and transformer-based legal information extraction and entailment. In *Proceedings of the COLIEE Workshop in ICAIL*.

Kim, M.-Y., Rabelo, J., Goebel, R., Yoshioka, M., Kano, Y., and Satoh, K. (2022). Coliee 2022 summary: Methods for legal document retrieval and entailment. In *JSAI International Symposium on Artificial Intelligence*, pages 51–67. Springer.

Kim, M.-Y., Rabelo, J., Goebel, R., Yoshioka, M., Kano, Y., and Satoh, K. (2023). Coliee 2022 summary: Methods for legal document retrieval and entailment. In *New Frontiers in Artificial Intelligence: JSAI-isAI 2022 Workshop, JURISIN 2022, and JSAI 2022 International Session, Kyoto, Japan, June 12–17, 2022, Revised Selected Papers*, pages 51–67. Springer.

Kreutzer, J., Khadivi, S., Matusov, E., and Riezler, S. (2018). Can neural machine translation be improved with user feedback? In Bangalore, S., Chu-Carroll, J., and Li, Y., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 92–105, New Orleans - Louisiana. Association for Computational Linguistics.

Kusner, M. J., Sun, Y., Kolkin, N. I., and Weinberger, K. Q. (2015). From word embeddings to document distances. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 957–966. JMLR.org.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2019). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Li, M., Peng, B., Galley, M., Gao, J., and Zhang, Z. (2023). Self-checker: Plug-and-play modules for fact-checking with large language models. *arXiv preprint arXiv:2305.14623*.

Lin, M., Huang, S., and Shao, H. (2022a). Rethinking attention: an attempting on revaluing attention weight with disjunctive union of longest uncommon subsequence for legal queries answering. In *Sixteenth international workshop on Juris-informatics (JURISIN)*.

Lin, M., Huang, S., and Shao, H. (2022b). Rethinking attention: An attempting on revaluing attention weight with disjunctive union of longest uncommon subsequence for legal queries answering. i. *Proceedings of the Sixteenth International Workshop on Juris-informatics (JURISIN 2022)*.

Liu, P., Qiu, X., and Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019a). Roberta: A robustly optimized BERT pretraining approach. *CoRR*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., et al. (2024). Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mitra, A., Del Corro, L., Mahajan, S., Codas, A., Simoes, C., Agarwal, S., Chen, X., Razdaibiedina, A., Jones, E., Aggarwal, K., et al. (2023). Orca 2: Teaching small language models how to reason. *arXiv preprint arXiv:2311.11045*.

Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S., Scao, T. L., Bari, M. S., Shen, S., Yong, Z.-X., Schoelkopf, H., et al. (2022). Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Nalisnick, E., Mitra, B., Craswell, N., and Caruana, R. (2016). Improving document ranking with dual word embeddings. In *Proceedings of the 25th international conference companion on world wide web*, pages 83–84.

Nguyen, C., Le, N.-K., Nguyen, D.-H., Nguyen, P., and Nguyen, L.-M. (2022). A legal information retrieval system for statute law. In *Asian Conference on Intelligent Information and Database Systems*, pages 370–382. Springer.

Nguyen, C., Luu, S. T., Tran, T., Trieu, A., Dang, A., Nguyen, D., Nguyen, H., Pham, T., Pham, T., Vo, T.-T., et al. (2023a). A summary of the alqac 2023 competition. In *2023 15th International Conference on Knowledge and Systems Engineering (KSE)*, pages 1–6. IEEE.

Nguyen, C. and Nguyen, L.-M. (2023). Employing label models on chatgpt answers improves legal text entailment performance. *Lecture Notes in Artificial Intelligence*.

Nguyen, C. and Nguyen, L.-M. (2024). Employing label models on chatgpt answers improves legal text entailment performance. *New Frontiers in Artificial Intelligence*.

Nguyen, C., Nguyen, P., Tran, T., Nguyen, D., Trieu, A., Pham, T., Dang, A., and Nguyen, L.-M. (2023b). Captain at coliee 2023: Efficient methods for legal information retrieval and entailment tasks. *Proceedings of COLIEE 2023 workshop: Competition on Legal Information Extraction/Entailment (COLIEE 2023)*.

Nguyen, H.-T., Nguyen, P. M., Vuong, T.-H.-Y., Bui, Q. M., Nguyen, C. M., Dang, B. T., Tran, V., Nguyen, M. L., and Satoh., K. (2021a). Jnlp team: Deep learning approaches for legal processing tasks in coliee 2021. In *Proceedings of the COLIEE Workshop in ICAIL*.

Nguyen, H.-T., Nguyen, P. M., Vuong, T.-H.-Y., Bui, Q. M., Nguyen, C. M., Dang, B. T., Tran, V., Nguyen, M. L., and Satoh, K. (2021b). Jnlp team: Deep learning approaches for legal processing tasks in coliee 2021. *arXiv preprint arXiv:2106.13405*.

Nguyen, H.-T., Toni, F., Stathis, K., and Satoh, K. (2023c). Beyond logic programming for legal reasoning. *arXiv preprint arXiv:2306.16632*.

Nguyen, H.-T., Tran, V., Nguyen, P. M., Vuong, T.-H.-Y., Bui, Q. M., Nguyen, C. M., Dang, B. T., Nguyen, M. L., and Satoh, K. (2021c). Paralaw nets–cross-lingual sentence-level pretraining for legal text processing. *Proceedings of the COLIEE Workshop in ICAIL (2021)*.

Nguyen, H.-T., Vuong, H.-Y. T., Nguyen, P. M., Dang, B. T., Bui, Q. M., Vu, S. T., Nguyen, C. M., Tran, V., Satoh, K., and Nguyen, M. L. (2020). Jnlp team: Deep learning for legal processing in coliee 2020. *COLIEE 2020*.

Nogueira, R., Jiang, Z., Pradeep, R., and Lin, J. (2020). Document ranking with a pretrained sequence-to-sequence model. In Cohn, T., He, Y., and Liu, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online. Association for Computational Linguistics.

OpenAI (2023). Gpt-4 technical report.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., and Lowe, R. (2022). Training language models to follow instructions with human feedback. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Paul, D., Ismayilzada, M., Peyrard, M., Borges, B., Bosselut, A., West, R., and Faltings, B. (2023). Refiner: Reasoning feedback on intermediate representations. *arXiv preprint arXiv:2304.01904*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Prokofyev, R., Tonon, A., Luggen, M., Vouilloz, L., Difallah, D. E., and Cudré-Mauroux, P. (2015). Sanaphor: Ontology-based coreference resolution. In *The Semantic Web-ISWC 2015: 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I 14*, pages 458–473. Springer.

Rabelo, J., Goebel, R., Kim, M.-Y., Kano, Y., Yoshioka, M., and Satoh, K. (2022a). Overview and discussion of the competition on legal information extraction/entailment (coliee) 2021. *The Review of Socionetwork Strategies*, 16(1):111–133.

Rabelo, J., Kim, M.-Y., and Goebel, R. (2022b). Semantic-based classification of relevant case law. In *JSAI International Symposium on Artificial Intelligence*, pages 84–95. Springer.

Rabelo, J., Kim, M.-Y., and Goebel, R. (2023). Transformer-based legal information extraction. In *Workshop of the tenth competition on legal information extraction/entailment (COLIEE'2023) in the 19th international conference on artificial intelligence and law (ICAIL)*.

Rabelo, J., Kim, M.-Y., Goebel, R., Yoshioka, M., Kano, Y., and Satoh, K. (2021). Coliee 2020: methods for legal document retrieval and entailment. In *New Frontiers in Artificial Intelligence: JSAI-isAI 2020 Workshops, JURISIN, LENLS 2020 Workshops, Virtual Event, November 15–17, 2020, Revised Selected Papers 12*, pages 196–210. Springer.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Ratner, A. J., De Sa, C. M., Wu, S., Selsam, D., and Ré, C. (2016). Data programming: Creating large training sets, quickly. *Advances in neural information processing systems*, 29.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Satoh, K., Asai, K., Kogawa, T., Kubota, M., Nakamura, M., Nishigai, Y., Shirakawa, K., and Takano, C. (2010). Proleg: an implementation of the presupposed ultimate fact theory of japanese civil code by prolog technology. In *JSAI international symposium on artificial intelligence*, pages 153–164. Springer.

Scheurer, J., Campos, J. A., Korbak, T., Chan, J. S., Chen, A., Cho, K., and Perez, E. (2023). Training language models with language feedback at scale. *arXiv preprint arXiv:2303.16755*.

Schick, T., Dwivedi-Yu, J., Jiang, Z., Petroni, F., Lewis, P., Izacard, G., You, Q., Nalmpantis, C., Grave, E., and Riedel, S. (2022). Peer: A collaborative language model. *arXiv preprint arXiv:2208.11663*.

Schilder, F., Chinnappa, D., Madan, K., Harmouche, J., Vold, A., Bretz, H., , and Hudzina., J. (2021a). A pentapus grapples with legal reasoning. In *Proceedings of the COLIEE Workshop in ICAIL*.

Schilder, F., Chinnappa, D., Madan, K., Harmouche, J., Vold, A., Bretz, H., and Hudzina, J. (2021b). A pentapus grapples with legal reasoning. In *Proceedings of the COLIEE Workshop in International Conference on Artificial Intelligence and Law (ICAIL)*.

Shao, Y., Mao, J., Liu, Y., Ma, W., Satoh, K., Zhang, M., and Ma, S. (2020). Bert-pli: Modeling paragraph-level interactions for legal case retrieval. In *IJCAI*, pages 3501–3507.

Shu, C., Han, J., Liu, F., Shareghi, E., and Collier, N. (2023). POSQA: Probe the world models of LLMs with size comparisons. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Sienčnik, S. K. (2015). Adapting word2vec to named entity recognition. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 239–243.

Silveira, R., Fernandes, C., Neto, J. A. M., Furtado, V., and Pimentel Filho, J. E. (2021). Topic modelling of legal documents via legal-bert. *Proceedings http://ceur-ws org ISSN*, 1613:0073.

Sleimi, A., Sannier, N., Sabetzadeh, M., Briand, L., Ceci, M., and Dann, J. (2021). An automated framework for the extraction of semantic legal metadata from legal texts. *Empirical Software Engineering*, 26:1–50.

Soria, C., Bartolini, R., Lenci, A., Montemagni, S., and Pirrelli, V. (2007). Automatic extraction of semantics in law documents. In *Proceedings of the V Legislative XML workshop*, pages 253–266. European Press Academic Publishing.

Strohman, T., Metzler, D., Turtle, H., and Croft, W. (2005). Indri: A language-model based search engine for complex queries. *Information Retrieval - IR*.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023a). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023b). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Tran, V., Le Nguyen, M., Tojo, S., and Satoh, K. (2020). Encoded summarization: summarizing documents into continuous vector space for legal case retrieval. *Artificial Intelligence and Law*, 28:441–467.

Tuggener, D., Von Däniken, P., Peetz, T., and Cieliebak, M. (2020). Ledgar: A large-scale multi-label corpus for text classification of legal provisions in contracts. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1235–1241.

Van Opijnen, M. and Santos, C. (2017). On the concept of relevance in legal information retrieval. *Artificial Intelligence and Law*, 25:65–87.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Vuong, T.-H.-Y., Nguyen, H.-L., Nguyen, T.-M., Nguyen, H.-T., Nguyen, T.-B., and Nguyen, H.-T. (2024). Nowj at coliee 2023: Multi-task and ensemble approaches in legal information processing. *The Review of Socionetwork Strategies*, pages 1–21.

Wehnert, S., Kutty, L., and De Luca, E. W. (2022). Using textbook knowledge for statute retrieval and entailment classification. In *JSAI International Symposium on Artificial Intelligence*, pages 125–137. Springer.

Wehnert, S., Kutty, L., and De Luca, E. W. (2023). Using textbook knowledge for statute retrieval and entailment classification. In *New Frontiers in Artificial Intelligence: JSAI-isAI 2022 Workshop, JURISIN 2022, and JSAI 2022 International Session, Kyoto, Japan, June 12–17, 2022, Revised Selected Papers*, pages 125–137. Springer.

Wehnert, S., Sudhi, V., Dureja, S., Kutty, L., Shahania, S., and De Luca, E. W. (2021a). Legal norm retrieval with variations of the bert model combined with tf-idf vectorization. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, ICAIL '21, page 285–294, New York, NY, USA. Association for Computing Machinery.

Wehnert, S., Sudhi, V., Dureja, S., Kutty, L., Shahania, S., and De Luca, E. W. (2021b). Legal norm retrieval with variations of the bert model combined with tf-idf vectorization. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pages 285–294.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Weng, Y., Zhu, M., Xia, F., Li, B., He, S., Liu, S., Sun, B., Liu, K., and Zhao, J. (2022). Large language models are better reasoners with self-verification. *arXiv preprint arXiv:2212.09561*.

Weston, J. and Watkins, C. (1998). Multi-class support vector machines. Technical report, Citeseer.

Wu, R., Chen, S.-E., Zhang, J., and Chu, X. (2022). Learning hyper label model for programmatic weak supervision. In *The Eleventh International Conference on Learning Representations*.

Xiao, C., Zhong, H., Guo, Z., Tu, C., Liu, Z., Sun, M., Feng, Y., Han, X., Hu, Z., Wang, H., et al. (2018). Cail2018: A large-scale legal dataset for judgment prediction. *arXiv preprint arXiv:1807.02478*.

Xie, Y., Kawaguchi, K., Zhao, Y., Zhao, J. X., Kan, M.-Y., He, J., and Xie, M. (2024). Self-evaluation guided beam search for reasoning. *Advances in Neural Information Processing Systems*, 36.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2020). mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Yang, K., Deng, J., and Chen, D. (2022a). Generating natural language proofs with verifier-guided search. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 89–105, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yang, K., Tian, Y., Peng, N., and Klein, D. (2022b). Re3: Generating longer stories with recursive reprompting and revision. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4393–4479, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yoshioka, M. and Aoki, Y. (2021). Bert-based ensemble methods for information retrieval and legal textual entailment in coliee statute law task. *Proceedings of the Eigth International Competition on Legal Information Extraction/Entailment (COLIEE 2021)*, pages 78–83.

Yoshioka, M. and Aoki, Y. (2023). Hukb at coliee 2023 statute law task. *Proceedings of COLIEE 2023 workshop: Competition on Legal Information Extraction/Entailment (COLIEE 2023)*.

Yoshioka, M., Aoki, Y., and Suzuki, Y. (2021a). Bert-based ensemble methods with data augmentation for legal textual entailment in coliee statute law task. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pages 278–284.

Yoshioka, M., Kano, Y., Kiyota, N., and Satoh, K. (2018). Overview of japanese statute law retrieval and entailment task at coliee-2018. In *Twelfth international workshop on Juris-informatics (JURISIN 2018)*.

Yoshioka, M., Suzuki, Y., , Aoki, Y., Libin, Shahania, Saijal, Luca, D., and W., E. (2021b). Bert-based ensemble methods for information retrieval and legal textual entailment in coliee statute law task. In *Proceedings of the COLIEE Workshop in ICAIL*.

Yoshioka, M., Suzuki, Y., and Aoki, Y. (2022). Hukb at the coliee 2022 statute law task. In *JSAI International Symposium on Artificial Intelligence*, pages 109–124. Springer.

Yu, W., Zhang, Z., Liang, Z., Jiang, M., and Sabharwal, A. (2023). Improving language models via plug-and-play retrieval feedback. *arXiv preprint arXiv:2305.14002*.

Zhang, J., Hsieh, C.-Y., Yu, Y., Zhang, C., and Ratner, A. (2022). A survey on programmatic weak supervision. *arXiv preprint arXiv:2202.05433*.

Zhang, J., Song, L., and Ratner, A. (2023a). Leveraging instance features for label aggregation in programmatic weak supervision. In *International Conference on Artificial Intelligence and Statistics*, pages 157–171. PMLR.

Zhang, K., Li, Z., Li, J., Li, G., and Jin, Z. (2023b). Self-edit: Fault-aware code editor for code generation. *arXiv preprint arXiv:2305.04087*.