| | |
|---|---|
| **Title** | 法的文書含意関係認識システムの曖昧性解消能力向上のための意味強化アプローチ |
| **Author(s)** | BUI, MINH QUAN |
| **Citation** | |
| **Issue Date** | 2023-09 |
| **Type** | Thesis or Dissertation |
| **Text version** | ETD |
| **URL** | http://hdl.handle.net/10119/18778 |
| **Rights** | |
| **Description** | Supervisor: NGUYEN, Minh Le, 先端科学技術研究科, 博士 |

Doctoral Dissertation

# Semantic Enhancement Approaches for Strengthening the Ambiguous Discrimination of Legal Text Entailment System

**Bui Minh Quan**

Supervisor: Nguyen Le Minh

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

September, 2023

# Abstract

Ambiguity poses a significant challenge within the legal domain, and the utilization of natural language processing (NLP) has emerged as a potential solution for resolving ambiguity in legal texts. This research proposes a semantic enhancement approach aimed at addressing the ambiguity prevalent in legal language. The approach entails leveraging external knowledge sources to enhance the accuracy and consistency of legal decision-making processes.

Furthermore, this study emphasizes the potential benefits associated with the semantic enhancement approach in the context of legal decision-making. These benefits include improved accuracy, enhanced consistency, and increased transparency within the decision-making process. The paper also acknowledges and discusses the inherent challenges and limitations associated with this approach. These challenges encompass the necessity for high-quality knowledge sources and the potential presence of bias and errors within the external knowledge utilized.

By introducing this semantic enhancement approach, the research aims to contribute to the field of legal language processing and facilitate more effective and reliable legal decision-making processes by mitigating the impact of ambiguity.

***Keywords***— Deep Learning, Large Language Model, Abstract Meaning Representation, Legal Domain, Transformer Model

# Acknowledgments

I would like to acknowledge the individuals and institutions who have contributed to the successful completion of my Doctor's dissertation. Firstly, I would like to express my heartfelt gratitude to my supervisor, Prof. Nguyen Le MInh, from Japan Advanced Institute of Science and Technology (JAIST), for providing invaluable guidance and support throughout the entire research process. His expertise and insights have been a constant source of inspiration to me.

I am also grateful to the faculty and staff of the School of Information Science for creating a stimulating and intellectually challenging academic environment that motivated me to strive for excellence.

I extend my sincere appreciation to my family for their unwavering support and encouragement throughout my academic journey. Their love and belief in my abilities gave me the strength and motivation to overcome obstacles.

Finally, I wish to acknowledge the members of the Nguyen lab and my friends whose enthusiasm, patience, and willingness to share their experiences were integral to the completion of this research. I am grateful to everyone who contributed to this dissertation and I hope that this research will serve as a valuable contribution to the academic community.

# Contents

# Chapter 1

# Introduction

The legal domain is characterized by its intricate language and complex terminology, making it a challenging area of study for natural language processing (NLP) techniques. The use of ambiguous language is particularly prevalent in legal documents, which can lead to various challenges in accurately processing and interpreting legal text. Ambiguity arises in various forms, including syntactic, semantic, and pragmatic ambiguity, where the intended meaning of a word or phrase may be unclear or open to multiple interpretations.

The challenge of ambiguity in legal NLP has significant implications, particularly in the areas of document classification, information extraction, and legal reasoning. Ambiguous language can hinder the ability of NLP models to accurately identify and extract relevant information from legal documents, leading to errors in document classification and retrieval. Additionally, the lack of clarity in legal language can impede the development of automated legal reasoning systems, which rely on accurate and consistent interpretation of legal text.

Despite these challenges, recent advancements in NLP have led to significant progress in addressing ambiguity in the legal domain. Techniques such as syntactic and semantic parsing, named entity recognition, and machine learning-based approaches have shown promising results in improving the accuracy of legal text analysis. However, there is still much work to be done in developing NLP systems that can accurately and consistently interpret the complexities of legal language.

Here 1.1 is and example of ambiguity in the legal domain. The sentence "The police helped the dog bite victim" is ambiguous and can be interpreted in different ways. Here are two possible interpretations:

1. The police helped the victim who was bitten by a dog.

2. The police helped the dog to bite the victim.

In the first interpretation, the police are assisting the victim who has been bitten by a dog, while in the second interpretation, the police are helping the dog to attack the victim.

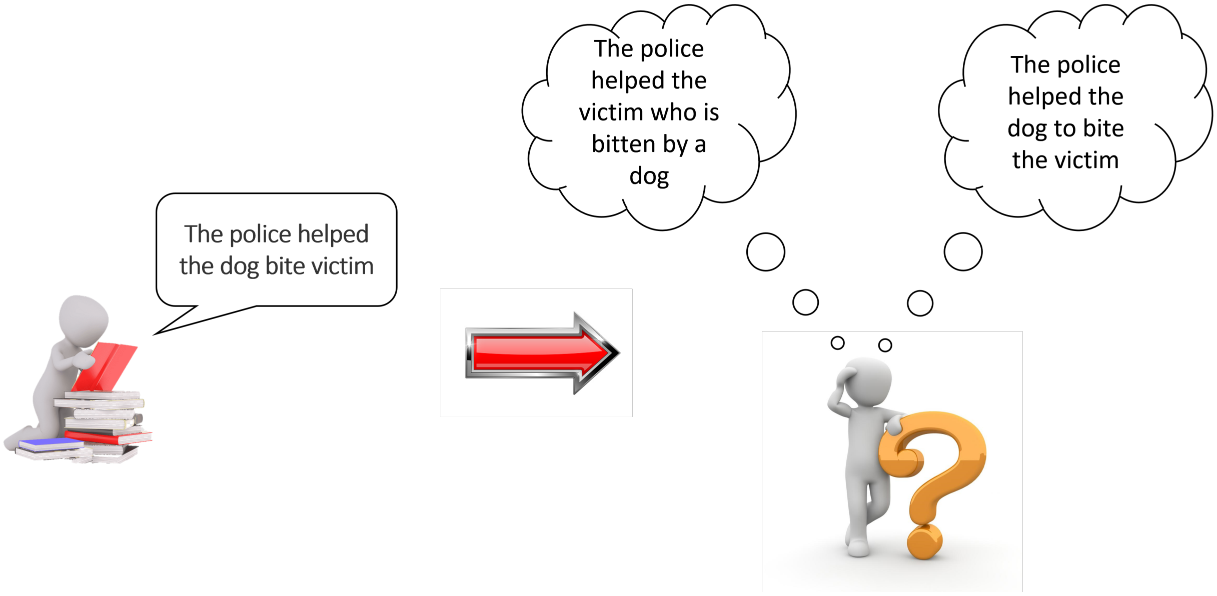

The police helped the dog bite victim
The police helped the victim who is bitten by a dog
The police helped the dog to bite the victim

Figure 1.1: An Example of Ambiguity in The Legal Domain

Without additional context, it is difficult to determine which interpretation is intended. Therefore, this sentence is an example of syntactic ambiguity, which arises due to the sentence structure and the different ways in which the words can be grouped and interpreted.

This sentence is an example of an ambiguous sentence, which can be difficult for both humans and machines to understand.

For humans, the ambiguity in the sentence can lead to confusion and misinterpretation, especially in a legal context where the meaning of language is crucial. In order to correctly interpret the sentence, the reader would need to rely on additional context and their own knowledge of the subject matter.

For machines, the ambiguity in the sentence presents a challenge for natural language processing (NLP) systems, which rely on accurately interpreting and understanding language. Without additional context, an NLP system may struggle to accurately classify or extract information from the sentence, leading to errors or inaccurate results.

Addressing ambiguity in language is an ongoing challenge for both humans and machines and requires a deep understanding of the context and the different ways in which language can be interpreted.

## 1.1 Contributions

Legal Text Entailment is a significant area of research in the field of natural language processing (NLP) and has gained increased attention in recent years. It refers to the process of determining whether a legal text implies or entails the truth of another legal text. Legal text entailment has critical implications for various legal applications, such as legal reasoning, case law analysis, and legal document summarization. The ambiguity and complexity of legal texts pose significant challenges for legal text entailment systems, requiring the development of sophisticated NLP techniques to accurately and efficiently identify entailment relations.

In the domain of natural language processing (NLP), two highly effective techniques are the Transformer [61] models and the BM25 algorithm [47]. These approaches have demonstrated considerable potential in a range of applications, including legal text entailment. Transformer models, a type of neural network model, have significantly transformed NLP by enabling superior performance in tasks such as language modeling, text generation, and machine translation. This success can be attributed to their utilization of self-attention mechanisms that effectively capture inter-word dependencies in a sentence. Thus, the Transformer models enable efficient and precise processing of natural language data.

On the other hand, the BM25 algorithm is a widely used information retrieval algorithm that is commonly used in search engines to rank documents based on their relevance to a given query. It is a probabilistic retrieval model that takes into account the frequency of query terms in a document and the inverse document frequency of those terms across the entire corpus, allowing for effective retrieval of relevant documents.

In the context of legal text entailment, both transformer models and the BM25 algorithm have shown significant potential in improving the accuracy and efficiency of entailment systems. Transformer models can be used to capture the complex semantic relationships between legal texts, allowing for more accurate identification of entailment relations. At the same time, the BM25 algorithm can be used to retrieve relevant legal documents based on their similarity to a given query, providing a useful tool for legal professionals to access and analyze relevant legal documents.

The primary objective of the present study is to introduce innovative and efficient methods for improving the efficacy of legal textual entailment systems, which are vital in the field of natural language processing (NLP). The study focuses on exploring the potential of Abstract Meaning Representation (AMR) [24]. The utilization of AMR involves encoding the meaning of legal texts in a structured and abstract format, which facilitates the precise comparison of semantic relationships. The integration of these approaches is expected to achieve cutting-edge performance in legal text entailment and offer useful resources for legal practitioners.

The rise of large language models, such as GPT-3 [8], Bloom [53], has transformed the way we interact with language and information. These models are based on the deep learning technique called transformer, which allows them to understand and generate natural language text with remarkable accuracy and fluency. Following are some state-of-the-art models recently:

1. GPT-3 [8], developed by OpenAI, is one of the most well-known large language models. It has 175 billion parameters and can generate human-like text in a variety of styles and tones. It has been used in a wide range of applications, including language translation, content creation, and chatbots.

2. The Bloom model [53], along with its several iterations, has been presented via the BigScience Workshop [1]. Drawing inspiration from existing open science endeavors, BigScience represents a collaborative effort by researchers to optimize their collective impact by pooling their time and resources. Although BLOOM's underlying structure closely resembles that of GPT-3, an auto-regressive model designed for predicting subsequent tokens, it has been trained on a broad range of languages, including 46 natural languages and 13 programming languages.

3. The present study, entitled "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer" [43] offers a comprehensive empirical investigation aimed at identifying the most effective transfer learning techniques. The resulting insights are subsequently leveraged to develop a novel model, known as the Text-To-Text Transfer Transformer (T5), which is implemented at scale. Additionally, they introduce an open-source pre-training dataset, namely the Colossal Clean Crawled Corpus (C4). T5, pre-trained on C4, exhibits exceptional performance on various NLP benchmarks while retaining sufficient flexibility for fine-tuning on a diverse array of downstream tasks. To facilitate the extension and reproduction of our findings, they provide the relevant code and pre-trained models, along with a user-friendly Colab Notebook to aid in implementation.

Despite their many applications and benefits, large language models such as GPT-3, Bloom, and T5 also raise important ethical and regulatory concerns. These models have the potential to perpetuate biases and discrimination in language processing, and there are questions about how they should be regulated and governed in the legal and other domains.

Given the remarkable capabilities and impact of large language models, the present study seeks to extend their application to the legal domain. To this end, we undertake an investigation employing several variants of such models to determine their efficacy in this context. Specifically, we explore the extent to which these models can accurately process legal language and yield relevant insights for legal practitioners.

## 1.2 Thesis Organization

As we can see in Figure 1.2, The following is the proposed organization of this thesis:

1. Abstract: This section provides a brief overview of the research objectives, methods, and contributions.

2. Introduction: This chapter introduces the research problem and provides an overview of the current state of the art in legal text entailment. It also presents the research questions, objectives, and significance of the study.

[1] https://bigscience.huggingface.co/

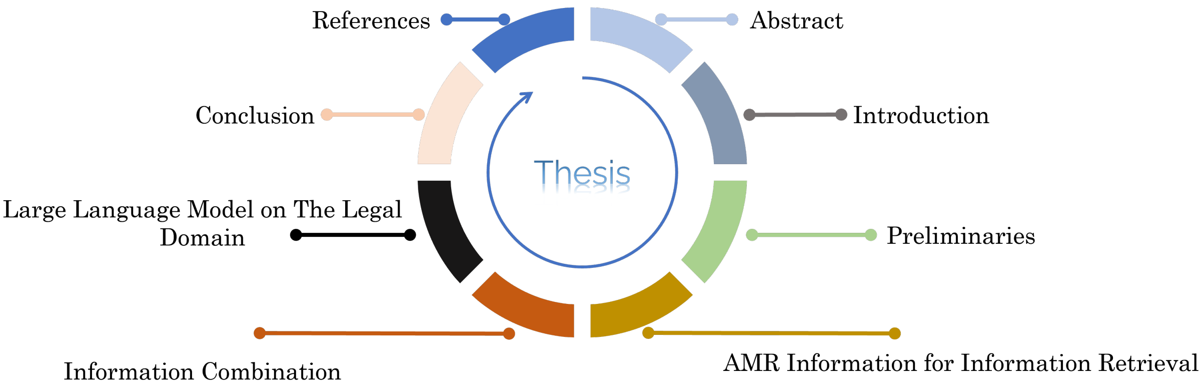


Figure 1.2: Thesis Organization

3. Preliminaries: This section discusses the fundamental concepts and theories related to natural language processing and legal text entailment, including the Transformer models and BM25 algorithm and several previous works in the legal domain.

4. AMR Information for Information Retrieval: This chapter investigates the efficacy of employing Abstract Meaning Representation (AMR) to eliminate extraneous data from a given query and candidates, thereby augmenting the information retrieval system's performance. The presented approach's methodology, experimental outcomes, and performance analysis are expounded.

5. Information Combination: This chapter explores the potential of integrating the BM25 algorithm, Transformer models, and Abstract Meaning Representation (AMR) to enhance the performance of legal textual entailment systems. The BM25 algorithm, widely employed in information retrieval, ranks documents based on their relevance to a given query. In parallel, Transformer models, known for their cutting-edge natural language processing capabilities, including text classification and entailment, offer promising opportunities for improving system performance. By combining these methodologies, the aim is to leverage their respective strengths and achieve superior results in legal textual entailment.

6. Influence of Large Language Model on The Legal Domain: This chapter examines the application of extensive language models in the context of the legal domain and their effectiveness in legal text entailment systems.

7. Conclusion: This chapter provides a summary of the research findings, including a discussion of the research questions, objectives, and contributions. It also presents the limitations of the study and directions for future research.

# Chapter 2

# Preliminaries

The following chapter aims to provide a comprehensive and in-depth discussion of the Transformer models, BM25 algorithm, and Abstract Meaning Representation (AMR), and how these approaches are currently performing in the legal domain. The chapter begins by introducing the basic concepts and principles behind these techniques and their applications in natural language processing.

## 2.1 Lexical Matching

Lexical matching is a fundamental technique in natural language processing and information retrieval that involves comparing words or phrases for similarity or overlap. The idea behind lexical matching is to identify words or phrases that are semantically related or have similar meanings. There are several types of lexical matching algorithms, each with its own strengths and weaknesses.

One of the simplest and most commonly used lexical matching algorithms is exact matching, which compares words or phrases to see if they are an exact match. Exact matching is useful when searching for specific phrases or when there is a clear answer to a question. However, it can be limited in cases where there is variability in word choice or spelling.

The term frequency-inverse document frequency (TF-IDF) algorithm is a lexical matching approach that has been extensively employed for several decades. It determines the relevance of a document by considering the frequency of occurrence of each term within the document, as well as the inverse document frequency of the term. TF-IDF operates on the principle that terms appearing frequently within a document but infrequently across the entire collection are more likely to hold greater importance and provide more informative content for that specific document. This algorithm is widely utilized in search engines and other information retrieval systems to prioritize and rank documents based on their pertinence to a user's query.

A lexical matching algorithm known as the Okapi BM25 algorithm is considered an alternative form of the TF-IDF algorithm. BM25 incorporates various factors such as document length, average document length within the collection, and the frequency of query terms in both the document and the collection. Extensive research has demonstrated that BM25 [48] is a resilient and efficient algorithm, particularly suitable for information retrieval tasks that require a balance between precision and recall.

The equations for TF-IDF and BM25 are as follows:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{ij}} \tag{2.1}$$

$$idf(w) = log\frac{N}{df_t} \tag{2.2}$$

$$tf - idf = tf_{i,j} \cdot idf(w) \tag{2.3}$$

$$BM25_{score(D,Q)} = \sum_{i=1}^{n} IDF(qi) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{avgdl})} \tag{2.4}$$

where:

- $tf_{i,j}$: number of occurrences of i in j
- $idf(w)$: number of documents containing i
- $N$: total number of documents
- $|D|$: is the length of the document D in words
- $avgdl$: is the average document length in the text collection from which documents are drawn
- $k_1$ : 1.2 by default in ElasticSearch [1]
- $b$: 0.75 by default in ElasticSearch

In the subsequent examples, we aim to demonstrate the efficacy of lexical matching within the legal domain. Here is an example provided by Figure 1.1:

- Query: The police helped the dog bite victim
- Document 1: The police helped the victim who is bitten by a dog
- Document 2: The police helped the dog to bite the victim

After applying tokenization to Document 1, and Document 2, we can calculate the TF of each word in the Query using Equation 2.1 as follows:

- TF("The",D1) = 2/11
- TF("police",D1) = 1/11

[1] https://www.elastic.co/

- TF("helped",D1) = 1/11
- TF("the",D1) = 2/11
- TF("dog",D1) = 1/11
- TF("bite",D1) = 0/11
- TF("victim",D1) = 1/11
- TF("The",D2) = 3/9
- TF("police",D2) = 1/9
- TF("helped",D2) = 1/9
- TF("the",D2) = 3/9
- TF("dog",D2) = 1/9
- TF("bite",D2) = 1/9
- TF("victim",D2) = 1/9

The IDF using Equation 2.2:

- IDF("The") = log(2/2) = 0
- IDF("police") = log(2/2) = 0
- IDF("helped") = log(2/2) = 0
- IDF("the") = log(2/2) = 0
- IDF("dog") = log(2/2)= 0
- IDF("bite") = log(2/1) = 0.3
- IDF("victim") = log(2/2) = 0

and TF-IDF score using Equation 2.3:

- TF-IDF("the",D1) = 2/11 · 0
- TF-IDF("police",D1) = 1/11 · 0
- TF-IDF("helped",D1) = 1/11 · 0
- TF-IDF("the",D1) = 2/11 · 0
- TF-IDF("dog",D1) = 1/11 · 0

- TF-IDF("bite",D1) $= 0/11 \cdot 0.3$
- TF-IDF("victim",D1) $= 1/11 \cdot 0$
- TF-IDF("the",D2) $= 3/9 \cdot 0$
- TF-IDF("police",D2) $= 1/9 \cdot 0$
- TF-IDF("helped",D2) $= 1/9 \cdot 0$
- TF-IDF("the",D2) $= 3/9 \cdot 0$
- TF-IDF("dog",D2) $= 1/9 \cdot 0$
- TF-IDF("bite",D2) $= 1/9 \cdot 0.3 = 0.03$
- TF-IDF("victim",D2) $= 1/9 \cdot 0$

Finally, the TF-IDF score that Document 1 has a score of zero, whereas Document 2 has a score of 0.03. This implies that the second document is the most suitable match for the Query at hand, albeit lacking in semantic congruity.

Using BM25 algorithm in the Equation 2.4:

- $BM25_{score}$("the",D1) $= 0$
- $BM25_{score}$("police",D1) $= 0$
- $BM25_{score}$("helped",D1) $= 0$
- $BM25_{score}$("the",D1) $= 0$
- $BM25_{score}$("dog",D1) $= 0$
- $BM25_{score}$("bite",D1) $= 0$
- $BM25_{score}$("victim",D1) $= 0$
- $BM25_{score}$("the",D2) $= 0$
- $BM25_{score}$("police",D2) $= 0$
- $BM25_{score}$("helped",D2) $= 0$
- $BM25_{score}$("the",D2) $= 0$
- $BM25_{score}$("dog",D2) $= 0$
- $BM25_{score}$("bite",D2) $= 0.3 \cdot \frac{(1/9)\cdot(1.2+1)}{(1/9)+1.2\cdot(1-0.75+0.75\cdot\frac{7}{20})} = 0.34$
- $BM25_{score}$("victim",D2) $= 0$

As demonstrated in this example, it is evident that even the most advanced lexical matching approaches are capable of producing erroneous outcomes.

## 2.2 Semantic Matching

### 2.2.1 Transformer Model

The progress made in natural language processing (NLP) is primarily attributed to the remarkable advancements in neural network-based models. Within this domain, the transformer architecture has emerged as a prominent and successful methodology for a wide range of NLP tasks. The introduction of the transformer model, initially described in the influential paper "Attention is All You Need" by Vaswani et al. [61], has garnered substantial recognition and adoption within both industry and academia.

Compared to earlier neural network architectures such as convolutional neural networks (CNNs) [36] and long short-term memory (LSTM) networks [16], the transformer model possesses several distinct advantages. In this paper, we will discuss these advantages in detail and examine the impact of the transformer model on the field of NLP.

One of the primary advantages of the transformer model is its ability to effectively process long sequences of text. Traditional neural network architectures such as CNNs and LSTMs often struggle with processing long sequences due to the limitations of sequential processing. In contrast, the transformer model employs a self-attention mechanism that allows it to process all positions in a sequence simultaneously, thus avoiding the limitations of sequential processing. This results in more efficient processing of longer sequences, making the transformer model ideal for tasks that involve lengthy input sequences.

An additional notable benefit offered by the transformer model lies in its capacity to acquire contextualized word representations. Unlike conventional bag-of-words methods, the transformer model considers the contextual information surrounding a word to produce a representation that captures its meaning within its specific context. This is achieved through the utilization of self-attention mechanisms, enabling the model to selectively attend to different segments of the input sequence. The capability to learn contextualized representations has demonstrated substantial enhancements in the transformer model's performance across a diverse range of NLP tasks.

The transformer model has emerged as a dominant architecture in natural language processing (NLP) research, and its variants have significantly impacted various NLP domains. Among these, BERT, RoBERTa, ALBERT, and Legal-BERT have gained considerable attention in the legal domain. These variants of the transformer model have been specifically designed to improve the performance of NLP tasks in the legal domain. In this paper, we will discuss these variants of the transformer model and their impact on NLP in the legal domain. The following are typical variations of transformer model:

1. BERT, which stands for Bidirectional Encoder Representations from Transformers, is a transformer-based model that was initially proposed by Devlin et al. (2018) [13]. This model has established itself as a benchmark in the field of natural language processing (NLP) by attaining state-of-the-art performance across multiple NLP tasks such as natural language inference,

question answering, and sentiment analysis. The effectiveness of BERT can be attributed to its pre-training methodology, which enables it to acquire contextualized word representations. This capability has been empirically demonstrated to significantly enhance BERT's performance on various NLP tasks within the legal domain.

2. RoBERTa, which stands for Robustly Optimized BERT Approach, is a variant of BERT proposed by Liu et al. (2019) [29]. This variant expands upon the pre-training methodology employed by BERT, incorporating further optimizations and modifications to both the pre-training and fine-tuning procedures. RoBERTa has demonstrated exceptional performance on numerous NLP benchmarks, including the General Language Understanding Evaluation (GLUE) benchmark. In the legal field, RoBERTa has been successfully applied to tasks such as legal case retrieval and legal document classification.

3. BART, short for Bidirectional and Auto-regressive Transformers, is a sequence-to-sequence model introduced by Lewis et al. (2019) [26], which has demonstrated state-of-the-art performance across various natural language processing (NLP) tasks. BART combines the bidirectional training approach of the transformer model with the auto-regressive training approach of the decoder within the model. This unique architectural design empowers the model to generate high-quality summaries, facilitate language translation, and accomplish other NLP tasks with exceptional proficiency. In this manuscript, we will extensively discuss the BART model, its underlying architecture, and its substantial impact on the field of NLP.

4. The Generative Pretrained Transformer (GPT) [8], one of the notable deep learning models, has emerged as a remarkably proficient language model capable of generating coherent and natural text. Built upon the Transformer architecture, which has demonstrated remarkable effectiveness across various natural language processing (NLP) tasks such as language modeling, text classification, and machine translation, the GPT model employs a self-supervised learning paradigm. By training on extensive amounts of unstructured textual data, the GPT model acquires an understanding of the underlying patterns and structure of language. Consequently, the model can be fine-tuned on specific downstream tasks using relatively limited training data, leading to state-of-the-art performance across a broad spectrum of tasks.

5. Legal-BERT, an innovative transformer-based model tailored for the legal domain, was developed by Chen et al. (2020) [10] and underwent training using an extensive corpus of legal documents. This domain-specific variant, Legal-BERT, has demonstrated exceptional performance by achieving state-of-the-art results across various legal natural language processing (NLP) tasks, encompassing tasks like legal case classification and legal contract analysis.

### 2.2.2 Transformer Model for Textual Entailment

The task of textual entailment, also known as natural language inference, has received considerable attention in the field of natural language processing (NLP) in recent years. It involves determining the semantic relationship between two pieces of text, namely the premise and the hypothesis. The premise refers to a given statement, while the hypothesis is a new statement that can either be entailed, contradicted, or be neutral with respect to the premise. The challenge lies in capturing the subtle and nuanced differences in meaning that may exist between the two statements, which may involve complex reasoning and inference.

To address this challenge, various machine learning models have been developed, ranging from rule-based systems to neural network-based approaches. In recent years, Transformer-based models, such as BERT, RoBERTa, and GPT, have emerged as the state-of-the-art models for various NLP tasks, including textual entailment. These models leverage the self-attention mechanism to capture the contextual relationships between words in a sentence, enabling them to learn highly expressive representations of the input text.

As an illustration in Figure 2.1, the Cross-Encoder model has been proposed as a means of evaluating the semantic similarity between two input sentences. This model involves the utilization of a Transformer network, which is a type of deep learning architecture that has shown great efficacy in various NLP tasks. To obtain a similarity score for a given sentence pair, the Cross-Encoder employs the Transformer to simultaneously process both sentences as a single input. The Transformer network subsequently outputs a value that ranges between 0 and 1, reflecting the degree of similarity between the input sentences. A value of 1 indicates that the two sentences are semantically identical, whereas a value of 0 indicates no similarity between the two. This mechanism of parallel processing of two sentences (premise and hypothesis) allows for a more nuanced and accurate evaluation of semantic similarity, thereby enabling more effective use of NLP models in various downstream applications such as text classification, machine translation, and information retrieval.

As can be seen in Figure 2.1, Bi-Encoders has emerged as a promising class of models that are capable of generating sentence embeddings. Specifically, given a sentence input, a Bi-Encoder utilizes a neural network to generate a corresponding embedding that captures the semantic information of the input. To this end, the input premise and hypothesis are passed independently to a transformer model, which is a powerful deep learning architecture used in various NLP applications. This results in the generation of two distinct sentence embeddings u (for premise) and v (hypothesis), respectively. To evaluate the similarity between these embeddings, cosine similarity is utilized as a metric. Cosine similarity measures the degree of similarity between two vectors by calculating the cosine of the angle between them. The resulting score ranges between 0 and 1, with a score of 1 indicating that the two embeddings are identical, and a score of 0 indicating that they are completely dissimilar. Another metric for measuring the similarity between two vectors is Euclidean distance. Euclidean distance is a fundamental measure in mathematics and statistics that can be

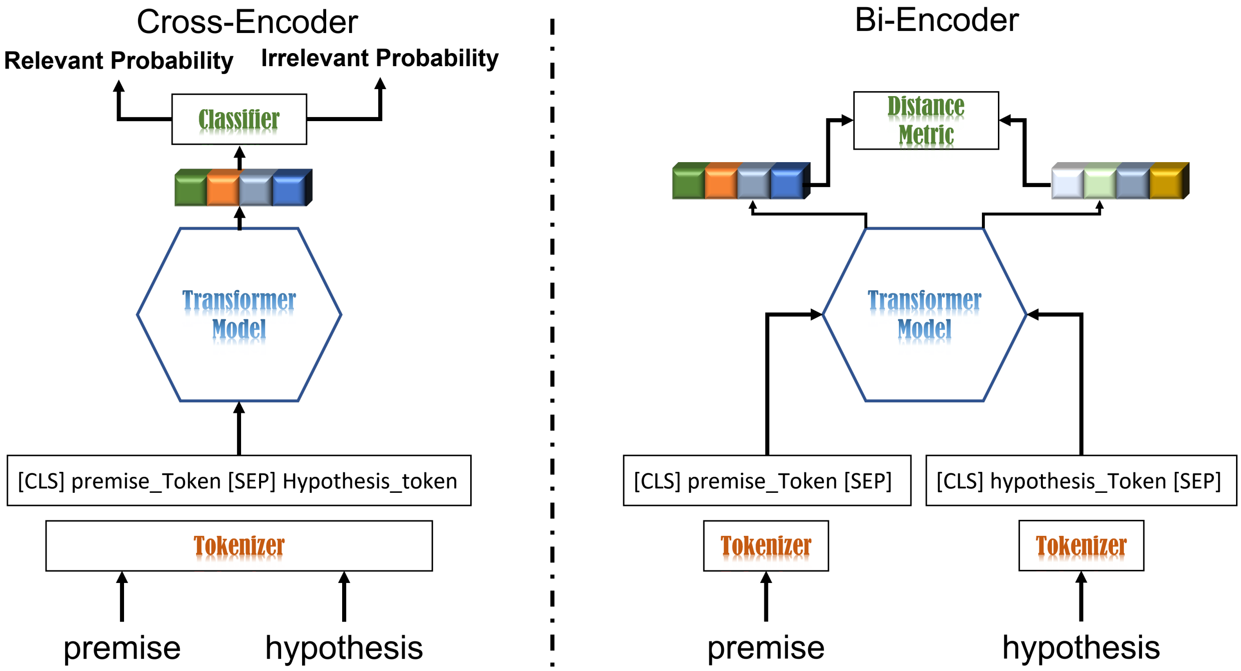


Figure 2.1: Cross-Encoder vs Bi-Encoder

used to compare the similarity between two tensors in machine learning and data science. A tensor is a multi-dimensional array that represents a set of data points or features. Euclidean distance is a metric that calculates the straight-line distance between two tensors, providing a measure of their similarity or dissimilarity in a high-dimensional space.

By utilizing Legal-BERT to generate embeddings of three sentences depicted in Figure 1.1, one can quantify the cosine similarity or Euclidean distance between the query and the ambiguous sentence. As illustrated in Table 2.1, the results demonstrate that the cosine similarity and Euclidean distance between the query and sentence2 (representing the incorrect meaning) are higher than the corresponding values between the query and sentence1 (representing the correct meaning).

In the cross-encoder approach, we employed a fine-tuned Distil-BERT model trained on the MNLI (Textual Entailment dataset) to derive the probabilities of Entailment and Contradiction for a given query and two candidate sentences. The obtained results, as depicted in Table 2.1, reveal that the entailment probability for query-sentence2 (with incorrect meaning) is higher. This finding suggests that even language models fine-tuned on extensive datasets like MNLI are prone to making erroneous judgments.

| Bi-encoder | | |
|---|---|---|
| | Cosine | Euclidean |
| Query-Sentence1 | 0.8141 | 9.3773 |
| Query-Sentence2 | **0.8940** | **7.6523** |
| Cross-Encoder | | |
| | Entailment | Contradiction |
| Query-Sentence1 | 0.9799 | 0.0201 |
| Query-Sentence2 | **0.9861** | **0.0139** |

Table 2.1: Results Using Bi-Encoders and Cross-Encoders on Example from Figure.1.1

## 2.3 Related Work

### 2.3.1 Legal Information Retrieval

In 20202, Westermann et al. [65] contributed significantly to the evaluation, submitting three runs that showcased their unique methodology. Their approach involved the selection of the top 10 candidate paragraphs based on a sentence similarity score computed using a universal sentence encoder. Subsequently, they applied a Support Vector Machine (SVM) model, utilizing the vector formed between the base case and candidate case representations in TF-IDF format. Notably, the authors also submitted additional runs that augmented their base approach. In these runs, they trained a TF-IDF vectorizer on all available texts, including test samples, while excluding certain anomalous samples from the training set. By incorporating these variations, the cyber team sought to explore different avenues for enhancing their system's performance and effectiveness.

Mandal et al. [32] based their submissions on a combination of techniques, including the filtered bag-of-ngrams (FiBONG) approach and BM25, as utilized in task 1. In their first run, they employed the BM25 algorithm on a FiBONG representation of the case documents. For the second run, they utilized the FiBONG representation alongside a different scoring function. Specifically, they employed a modified version of BM25, where the new Inverse Document Frequency (IDF) term was multiplied by a standardized and normalized value of the collection frequency. Finally, in their third run, they represented the candidate paragraphs and base judgements using centroids of word embeddings. To measure similarity, they employed the cosine distance metric. Notably, the word embeddings were derived from Law2Vec8, a specialized embedding model tailored to legal text analysis. By leveraging these diverse techniques and embedding representations, the iiest team aimed to explore different aspects of their models' performance and leverage the unique characteristics of legal text for improved results.

Alberts et al. [1] conducted three runs in their study, employing an Xgboost classifier with various features as input. These features included the NLI (Natural Language Inference) probability obtained from bert-nli, the similarity between the entailed fragment and paragraphs based on fine-tuned BERT (bert-base-uncased), and the BM25 similarity between the entailed fragment and paragraphs. Additionally, the authors submitted runs incorporating other features as input, such as n-grams, BM25, NLI, and similarity features derived from fine-tuned ROBERTA and BERT models trained on EUR-LEX, which encompasses a vast collection of sentences from EU legal documents. By exploring different combinations of features, the tax-i team aimed to assess their impact on the performance of their system and identify the most effective features for the task at hand.

Shao et al. [56] adopted a comprehensive approach, conducting three runs that revolved around fine-tuning the BERT (uncased-base) model in a sentence pair classification task. To handle text length limitations, the team employed symmetric truncation when the total input tokens exceeded the limit of 512. In the second run, they introduced asymmetrical truncation, limiting the tokens of the decision fragment to 128 and only truncating the tokens in the candidate paragraph if the total length of the text pair exceeded 512 tokens. In their final run, the authors extracted the output vector of the fully-connected layer from the two previous models, resulting in a 4-dimensional feature representation. Additionally, they calculated BM25 scores, contributing a 1-dimensional feature. Two additional features, namely the position ID and length of the paragraph, were incorporated, resulting in a total of 7-dimensional features. These features were then used as input for a RankSVM model, enabling the team to rank the paragraphs effectively and make informed classifications.

Hudzina et al. [17] developed a two-stage approach consisting of similarity features-based ranking followed by Random Forest binary classification. The team ranked paragraphs based on a combined criterion that considered the cosine similarity coefficients obtained using different sentence vectorizers, including n-grams, universal sentence encoder, averaged glove embeddings, and topic modeling probability scores. The likelihood of a relevant paragraph falling within the top K paragraphs was estimated using training data. Subsequently, for a specific likelihood value, similarity features were computed on the top K paragraphs and supplied as input to a random forest classifier. This approach allowed the TR team to leverage the ranking of paragraphs based on similarity features, providing an effective means of classification.

Rabelo et al. [40] contributed three runs to the evaluation, employing transformer-based techniques in their methodology. They generated features by fine-tuning a pre-trained BERT model on text entailment using the provided training dataset. The score produced in this task, along with two transformer-based models fine-tuned on a generic entailment dataset, were used as features. Furthermore, the team applied zero-shot techniques by utilizing BERT fine-tuned for paraphrase detection. To augment their training data, they employed data augmentation techniques based on back translation. Ultimately, the generated features were fed into a Random Forest classifier. By leveraging the power of transformer models and incorporating data augmentation, the UA team aimed to improve the performance and robustness of their system.

In 2021, Schilder et al. [55], operating under the team name TR, conducted an investigation using hand-crafted similarity features and implemented a classical random forest classifier. To gauge the similarity between each paragraph within the noticed case and the decision fragment in the query, they employed a combination of n-gram vectors, universal sentence encoder vectors, and averaged word embedding vectors. By calculating the similarity scores, they were able to identify the most similar k paragraphs, upon which they proceeded to train a random forest classifier.

Kim et al. [20], known as the UA team, pursued a different approach by utilizing BERT, a state-of-the-art language model, pre-trained on a large general-purpose dataset. To adapt BERT to their task, they fine-tuned the model using the provided training dataset. In situations where the tokenization process exceeded the maximum limit of 512 tokens, they devised an additional transformer-based model to generate a summary of the input text. Subsequently, they subjected the summary and the original text pair to further processing. Given that the input text frequently contained French language segments, the team incorporated a simple language detection model based on a naive Bayesian filter to filter out these fragments. To optimize the output, they set limits on the maximum number of outputs allowed per case during the post-processing stage, while also imposing a minimum score threshold to minimize false positives.

Li et al. [27], operating as the siat team, proposed an innovative approach involving a pre-training task on BERT (BERT-base-uncased) with dynamic N-gram masking. This novel approach enabled them to develop a specialized BERT model enriched with legal knowledge, aptly named BERTLegal. The process involved employing N-gram masking to generate masked inputs for what they referred to as "masked language model" targets. Notably, the length of each n-gram mask was randomly chosen from a pool consisting of 1, 2, and 3. To augment their dataset, they employed data augmentation techniques and incorporated a Fast Gradient method into their methodology.

Rosa et al. [49] pursued an alternative path by leveraging the power of monoT5-zero-shot, monoT5, and DeBERTa. Additionally, they conducted an ensemble evaluation of their monoT5 and DeBERTa models to capitalize on their collective strengths. Notably, the monoT5-zero-shot model emerged as a prominent choice, as it represented a sequence-to-sequence adaptation of the widely recognized T5 model.

### 2.3.2 Legal Yes/No Question Answering

In 2021, The HUKB team's approach involved leveraging a BERT-based Information Retrieval (IR) system in conjunction with the Indri framework to facilitate the IR module. To derive the final results, the team employed a comparative analysis of the output generated by each system. Notably, they devised a novel article database comprising two distinct types. The first type entailed expanding the detailed information by incorporating the content of referred articles. On the other hand, the second type employed text-splitting techniques to describe individual judicial decisions in a more granular manner. By meticulously configuring three runs, the team aimed to assess the performance and effectiveness of their approach.

Similarly, Nguyen et al. [33] embarked on their research journey with a systematic approach, conducting three runs to thoroughly explore their proposed methodology. Central to their investigation was the utilization of BERT-based IR models, which integrated multiple BERT models to enhance the generation of results. A key aspect of their methodology involved the creation of a comprehensive training dataset containing relevant articles. To accomplish this, they employed a sliding window technique to select the most relevant portions of the articles. Through this process, they aimed to train their models on highly informative and pertinent data. Among their submissions, the best-performing run was identified as CrossLMultiLThreshlod. Notably, this run harnessed the power of an ensemble approach, combining outputs from three distinct systems and selecting the highest result among them. By meticulously designing their experimental setup and leveraging the strength of ensemble modeling, the JNLP team sought to optimize their results and drive the advancement of their research.

Wehnert et al. [63] embarked on their research endeavors by employing a diverse array of BERT models coupled with various data enrichment techniques across their three runs. Through careful experimentation, they aimed to explore the impact of different approaches on the performance of their system. Among their runs, OvGU run1 emerged as the most promising. This run incorporated the utilization of sentence-BERT embeddings, a powerful technique for capturing semantic information, combined with the traditional TF-IDF methodology. To enrich their training data, the OvGU team incorporated metadata, relevant web data associated with the articles, and pertinent queries extracted from the training data itself. By incorporating these additional sources of information, they sought to enhance the comprehensiveness and effectiveness of their models, ultimately leading to improved results.

Schilder et al. [55] contributed to the evaluation by using Word Mover's Distance (WMD) approach to calculate the similarity between queries and articles. By leveraging this technique, the TR team aimed to capture the semantic similarity and relevance between textual elements, enabling a more accurate retrieval of relevant information.

Kim et al. [20] adopted a more conventional approach by utilizing ordinary IR modules to generate results. The best-performing run was identified as BM25, which harnessed the power of the BM25 algorithm as the IR module. Through careful configuration and experimentation, the UA team aimed to optimize the retrieval of relevant information, contributing to the overall effectiveness of their system.

# Chapter 3

# AMR Information for Textual Entailment System

## 3.1 Introduction

Abstract Meaning Representation (AMR) [24] encodes the semantic interpretation of a sentence using a directed acyclic graph structure. In this representation, concepts or entities are denoted as nodes, while the relationships between them are represented by edges. The AMR graph effectively captures diverse linguistic phenomena, encompassing coreference, negation, and modality.

AMR has found applications in various natural language processing (NLP) tasks, including machine translation, question answering, and semantic parsing. In the present study, we aim to introduce the utilization of AMR in textual entailment systems.

## 3.2 Abstract Meaning Representation (AMR)

Abstract Meaning Representation (AMR) is a linguistic formalism and a method of representing the meaning of natural language sentences in a structured form. The need for a structured representation of the meaning of language has been recognized in linguistics for many years, but the development of AMR as a specific approach can be traced back to the early 2010s.

The goal of AMR is to provide a unified and consistent way of representing the meaning of sentences, regardless of their surface form. This is achieved by representing the meaning of a sentence as a directed acyclic graph (DAG), where the nodes in the graph represent concepts or entities, and the edges represent the relationships between them.

The nodes in an AMR graph are labeled with concepts, which are typically abstract and can be interpreted in a variety of ways depending on the context. For example, the concept of "person" can refer to an individual, a group of people, or a fictional character, depending on the sentence being analyzed.

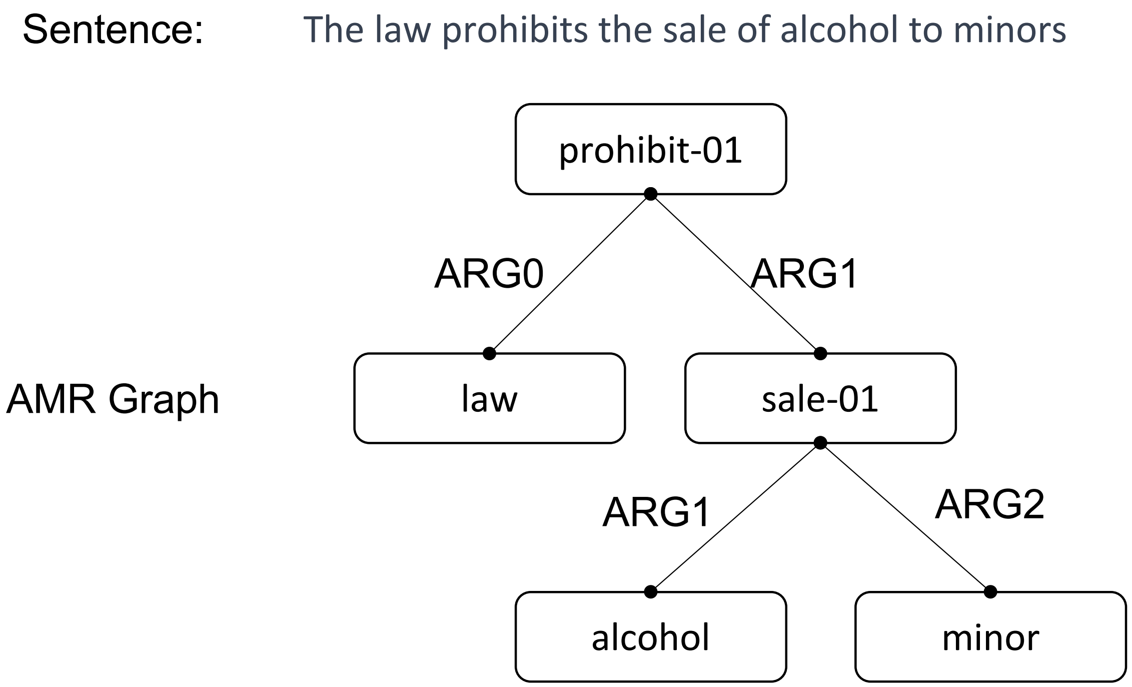


Figure 3.1: Example of Abstract Meaning Representation (AMR)

The edges in an AMR graph represent the relationships between the concepts in the sentence, such as the subject-verb-object relationship in a simple sentence. The edges can also represent more complex relationships, such as coreference, negation, and modality. Each edge has a label that describes the relationship between the nodes. For example, the label "ARG0" is used to indicate the agent of an action, while the label "ARG1" is used to indicate the patient or theme of an action.

We can see a given sentence and corresponding AMR graph in Figure 3.1. In this example, "prohibit-01" is the main predicate and has two arguments:

1. ARG0: the law which is responsible for the prohibition

2. ARG1: the action that is prohibited, in this case, the sale of alcohol

The sub-predicate "sale-01" is a modifier of "prohibit-01" and has two arguments:

1. ARG1: the type of thing being sold, in this case, alcohol

2. ARG2: subject to whom alcohol is sold (minor)

AMR has found many applications in NLP, including parsing, machine translation, question answering, summarization, and sentiment analysis [3] [58] [19]. Specifically, AMR parsers use machine learning models and linguistic resources to generate a graph that represents the meaning of the input text. This graph can then be used to identify the most important concepts and relationships in the text, making it a powerful tool for tasks such as question answering and summarization.

Moreover, AMR can also be used for machine translation, where it can help to align the meaning of the source and target text, resulting in more accurate translations. Additionally, AMR can be useful in information retrieval by capturing the nuances and complexities of natural language expressions, making it possible to identify the overall similarity of sentences or documents.

## 3.3 Methodology

### 3.3.1 Legal Case Document Similarity Using AMR

In this work, AMR uses to remove redundant words from documents, which intends to improve the performance of information retrieval algorithms such as BM25. The present analysis pertains to the sentence, "The law prohibits the sale of alcohol to minors" as displayed in Figure 3.1. Through the application of Abstract Meaning Representation (AMR), it is observed that semantic association between the terms "alcohol" and "minor" is established via the verb "sale-01", and the phrase "sale alcohol minor" is further connected to "law" through "prohibit-01". Hence, it may be deduced that certain non-essential terms, including "the", "of", and "to," can be disregarded.

Considering a more complex example in the legal domain:

*”The jurisprudence established that a leave to appeal proceeding was a preliminary step to a hearing on the merits, and was a lower hurdle for the applicant for leave to meet since the case did not have to be proven”*

The logical triples produced by the Spring parser [5] can depict the AMR as follows:

establish-01 :ARG0 jurisprudence,
establish-01 :ARG1 and,
and :op1 step-01,
step-01 :ARG1 leave-16,
leave-16 :ARG2 proceeding-02,
proceeding-02 :ARG1 appeal-01,
step-01 :ARG2 hearing-02,
hearing-02 :ARG2 merit-01,
step-01 :mod preliminary,
and :op2 hurdle-01,
hurdle-01 :ARG1 apply-01,
apply-01 :ARG0 person
apply-01 :ARG1 leave-16,
leave-16 :ARG2 meet-03,
meet-03 :ARG0 person,
cause-01 :ARG1 hurdle-01,
cause-01 :ARG0 obligate-01,
obligate-01 :polarity -,
obligate-01 :ARG2 prove-01,
prove-01 :ARG1 case-03,
have-degree-91 :ARG1 hurdle-01,
have-degree-91 :ARG2 low-04,
low-04 :ARG1 hurdle-01,
have-degree-91 :ARG3 more

The AMR allows for the elimination of extraneous details within a given sentence, resulting in a reduced set of remaining information, which is a set of nodes as follows:

- establish
- jurisprudence

- step
- hearing
- hurdle
- ...

and we have some roles nodes defined by AMR, such as :

- have-degree-91
- have-org-role-91
- have-concession-91

and we use Spacy[1] Part-of-speech (POS) tagging to detect useful nodes in AMR and remove redundant nodes as follows:

1. Use an AMR parser to generate an AMR graph for the input sentence or document.
2. Use Spacy to perform POS tagging on the text associated with each node in the AMR graph.
3. Identify the nodes in the AMR graph that correspond to nouns, verbs, adjectives, adverbs, and other useful parts of speech based on their POS tags.
4. Remove any nodes in the AMR graph that do not correspond to useful parts of speech or are redundant based on their semantic relationships to other nodes in the graph.
5. Utilize the simplified nodes to assess the degree of similarity between legal documents.

Based on the premise that the AMR eliminates redundant information and retains solely the most pertinent details, it is anticipated that the exclusion of non-essential words from the output may enhance the efficacy of the BM25 algorithm.

### 3.3.2 Legal Case Document Similarity Using Legal-BERT

Legal-BERT [10] is a language model that has been developed specifically to address the unique challenges of processing legal language. It is based on the BERT architecture, which has become a popular choice for a variety of natural language processing tasks due to its ability to capture contextual information and produce high-quality language representations. However, legal language is known to be particularly complex, with a unique vocabulary, syntax, and structure that can be difficult for standard language models to understand.

[1] https://spacy.io/

To address these challenges, the same as previous work such as SciBERT [4], BioBERT [25], Legal-BERT was trained on a large corpus of legal texts, including case law, statutes, and regulations. This training data was carefully curated to ensure that the model would be able to learn the nuances of legal language and the specific rules and structures of legal documents. The resulting model has been shown to be effective in a variety of legal NLP tasks, such as legal document classification, legal question answering, and legal language modeling.

One of the key features of Legal-BERT is its ability to capture the meaning of legal terms and phrases in context. Legal language is often filled with technical terms and legal jargon that can be difficult for non-experts to understand, and Legal-BERT is able to take into account the broader context in which these terms are used to understand their meaning better. This makes it a valuable tool for lawyers, legal researchers, and anyone working with legal documents who need to analyze and understand legal language.

The Legal-BERT model has been recognized as a noteworthy progression in the domain of legal NLP and holds immense potential to transform the approach employed by legal experts for comprehending and analyzing legal language. Its capacity to effectively comprehend the subtleties of legal terminology and generate superior-quality language representations renders it a potent instrument for a diverse range of legal applications. In the present study, we incorporate the Legal-BERT model in conjunction with the BM25 technique to facilitate information retrieval.

### 3.3.3 Information Ensemble

Ensemble methods can also be used in deep learning to improve the performance and robustness of models.

One approach to ensemble learning in deep learning is to use a technique called model averaging, where multiple models are trained with different initialization or hyperparameters, and their predictions are averaged to obtain the final output. Model averaging can help to reduce the impact of overfitting and increase the stability of the model.

Another approach to ensemble learning in deep learning is to use a technique called bagging [14] [9], where multiple models are trained on different subsets of the training data, with each subset sampled with replacement. The outputs of the models are then combined to obtain the final output. Bagging can help to reduce the variance of the model and improve its generalization performance.

A third approach to ensemble learning in deep learning is to use a technique called boosting [9], where multiple weak models are trained sequentially, with each new model trained on the examples that were misclassified by the previous model. The final output is then obtained by combining the outputs of all the models. Boosting can help to reduce the bias of the model and improve its accuracy.

Drawing on prior research, the present study utilizes a weighted averaging ensemble methodology to amalgamate semantic information derived from Legal-BERT [10] and lexical information garnered from the BM25 algorithm. The combination of the semantic and lexical information can be achieved

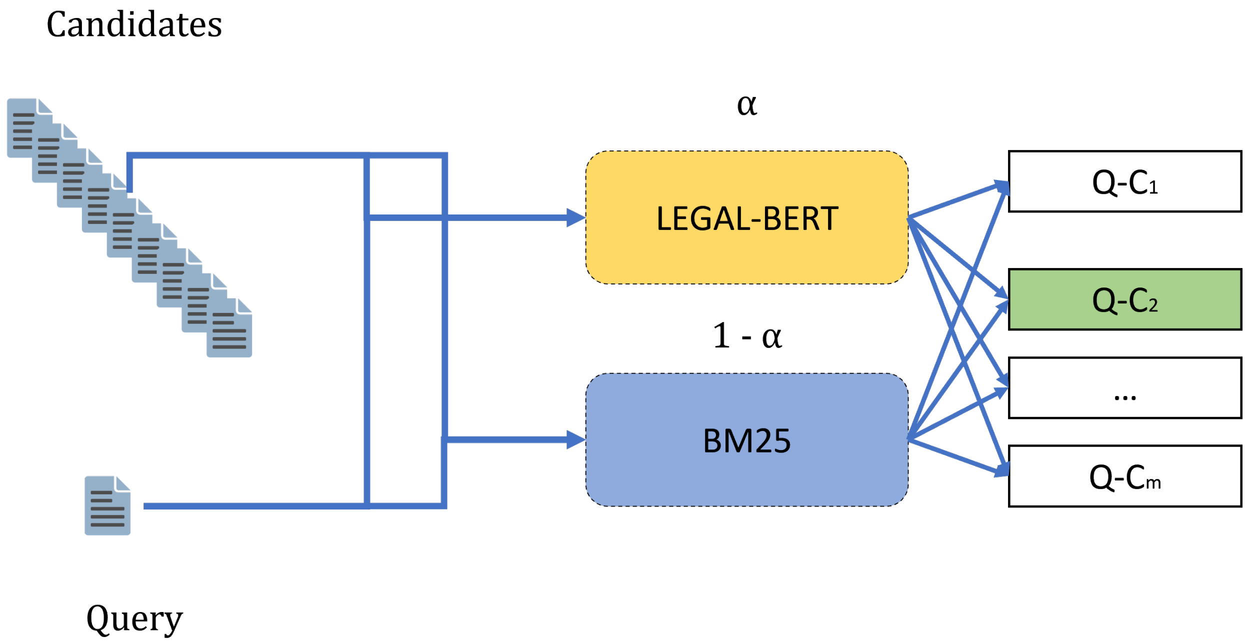


Figure 3.2: Average Ensemble for Legal Case Document Similarity

by Formula 3.1.

$$Ensemble_{Score} = (Semantic_{Score} * \alpha) + (Lexical_{Score} * (1 - \alpha)) \tag{3.1}$$

The choice of weights $\alpha$ depends on the desired emphasis placed on each score and should be selected in a manner that ensures their sum is equal to one.

Despite the remarkable performance of the transformer model, Legal-BERT and other transformer variants, such as BERT [13], RoBERTa [29], ALBERT [23], exhibits a limitation in handling lengthy sentences ($>$ 512 tokens). When conducting predictions with a specific query Q and corresponding candidate C, if the combined length of Q and C exceeds 512 tokens, the sentence segmentation technique implemented in spacy3[2] will be utilized to generate a set of segmented sentences, denoted as Sc = c1, c2, c3, ..., cN. The $Semantic_{Score}$ can be computed by Equation 3.2:

$$Semantic_{Score} = \frac{\sum_{i=1}^{N} O_i}{N} \tag{3.2}$$

Let $N$ represent the overall count of sentences resulting from the application of sentence segmentation. The semantic score $O_i$ corresponds to the candidate sentence $c_i$, and the corresponding query sentence is obtained through the implementation of LEGAL-BERT utilizing the cross-encoder approach.

[2] https://spacy.io/

### 3.3.4 Experimental Settings

#### Task Definition

To assess the resilience of our proposed methodology, we employed the second task of the Conference on Legal Information Extraction and Entailment (COLIEE) competition as a means of validation. This task revolves around the identification of a paragraph extracted from a corpus of existing cases that may offer insights into a decision made in a new case. As depicted in Figure 3.2, the Information Retrieval (IR) system aims to categorize paragraphs in set R as either relevant or entailing, relative to a given statement or decision Q. The presence of both entailing and non-entailing paragraphs within the text poses a challenge, as some paragraphs in R offer pertinent information while others do not.

During the training phase, each sample consists of a query, a set of candidate paragraphs, and a label indicating whether the candidate paragraph is a positive (entailment) or negative (non-entailment) example.

Table 3.1 displays the analysis of the training and test data used in the experiment. The training set comprises 525 queries, which are associated with 18,740 candidate paragraphs, giving an average of 35.6 candidate paragraphs per query. In contrast, the test set contains 100 queries, with 3,278 candidate paragraphs, averaging 31.8 candidate paragraphs per query.

In terms of the number of samples, the training set consists of 599 positive and 18,141 negative samples, while the test set has 118 positive and 3,160 negative samples. These samples were used to train the model to classify candidate paragraphs as either entailment or non-entailment paragraphs with respect to the given query.

The table also provides information on the length of queries and candidate paragraphs, both in terms of their average and maximum lengths. The average length of queries in the training and test sets were 43.1 and 38.06 tokens, respectively, with the maximum length being 133 and 130 tokens, respectively. The candidate paragraphs in the training and test sets had similar average lengths of 138.5 and 141.0 tokens, respectively, but differed in their maximum lengths, with the training set containing candidate paragraphs up to 3,795 tokens in length, while the test set contained candidate paragraphs up to 1,640 tokens in length.

#### Evaluation Metric

In order to assess the performance of an information retrieval (IR) system, the competition organizers have introduced three metrics, namely Precision, Recall, and F1. In this particular competition, a micro-average approach was utilized, whereby the metrics were calculated on a per-query basis and then averaged. Consequently, all the evaluation metrics have undergone certain modifications specific to this competition. The specific details of these three metrics are described in Formula 3.3 3.4 3.5.

| | Train | Test |
|---|---|---|
| #Query | 525 | 100 |
| #Candidate | 18740 | 3278 |
| #Candidate/Query | 35.6 | 31.8 |
| #Positive Sample | 599 | 118 |
| #Negative Sample | 18141 | 3160 |
| Query Average Length (tokens[1]) | 43.1 | 38.06 |
| Query max Length (tokens[1]) | 133 | 130 |
| Candidate Average Length (tokens[1]) | 138.5 | 141.0 |
| Candidate Max Length (tokens[1]) | 3795 | 1640 |

Table 3.1: Task 2 COLIEE 2022 Analysis

$$Precision = \frac{the\ number\ of\ correctly\ retrieved\, paragraphs\ for\ each\ query}{the\ number\ of\ retrieved\, paragraphs\ for\ each\ query} \tag{3.3}$$

$$Recall = \frac{the\ number\ of\ correctly\ retrieved\, paragraphs\ for\ each\ query}{the\ number\ of\ correctly\ retrieved\, paragraphs\ for\ each\ query} \tag{3.4}$$

$$F1 = \frac{2\ \cdot\ Precision\ \cdot\ Recall}{Precision\ +\ Recall} \tag{3.5}$$

### Model Settings

The presented Table 3.2 outlines the hyperparameters utilized in a particular machine learning model. The Legal-BERT was trained for three epochs with a maximum sequence length of 512. The AdamW optimizer was used with a learning rate of 2e-5, beta1 of 0.9, beta2 of 0.999, and epsilon of 1e-8. The per-device batch size was set to 32 for both training and evaluation.

| | |
|---|---|
| Max Sequence Length | 512 |
| Learning Rate | 2e-5 |
| Number of training epochs | 3 |
| adam_beta1 | 0.9 |
| adam_beta2 | 0.999 |
| adam_epsilon | 1e-8 |
| per_device_train_batch_size | 32 |
| per_device_eval_batch_size | 32 |
| optimizer | AdamW |

Table 3.2: Legal-BERT Model Configuration

In prediction phase, we have 3 settings as follows:

1. As we can see in Figure 3.3, the combination of LEGAL-BERT and BM25 was utilized,

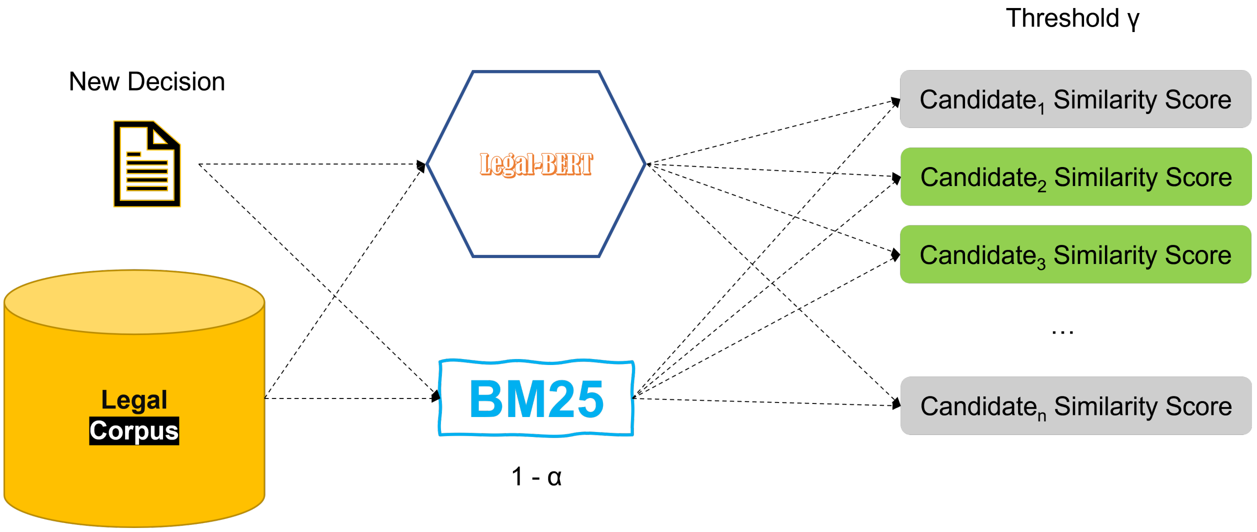

Threshold γ
New Decision
Legal-BERT
Legal Corpus
BM25
1 - α
Candidate$_1$ Similarity Score
Candidate$_2$ Similarity Score
Candidate$_3$ Similarity Score
...
Candidate$_n$ Similarity Score

Figure 3.3: Relevant Candidate Selection Using Ensemble Approach

whereby the values of $\alpha$ were set to 0.9 and 0.1 for semantic and lexical scores, respectively. Furthermore, the threshold was established to be the top 1 score candidate subtracted by 0.003.

2. In the second experiment, Legal-BERT was coupled with BM25, where the sentences were initially converted into an AMR graph, and spacy POS tagging was used to extract the most significant parts of speech, such as nouns, verbs, adjectives, and others. The BM25 algorithm was applied to these sets of POS tagging, while the coefficient $\alpha$ was set to 0.9 and 0.1 for semantic and lexical scores, respectively and threshold $= 0.004$.

3. The synthesis of results was achieved through the specification of $\alpha = 1$ (indicating the absence of an ensemble) and threshold values solely for semantic scores. As detailed in Figure 3.4, the selection of top N semantic-based relevant paragraphs were executed, with the optimized value of N being 2, as deduced from the development set results. For lexical-based candidates, the top M candidates were chosen without a threshold, while the optimized value of M was determined to be 2. Subsequently, the intersection of M and N was conducted to obtain the ultimate set of candidate paragraphs. In the event that the outcome following the intersection procedure was null, the top 1 semantic score was selected as the relevant document.

### 3.3.5 Experimental Results

Upon analysis of Table 3.3, it is evident that *monot5-ensemble* achieved the highest f1 score of 0.6783 using a large language model, closely followed by Intersection (AMR) with a score of 0.6694. These two models also exhibit relatively high precision and recall values, indicating their efficacy in identifying true positives.

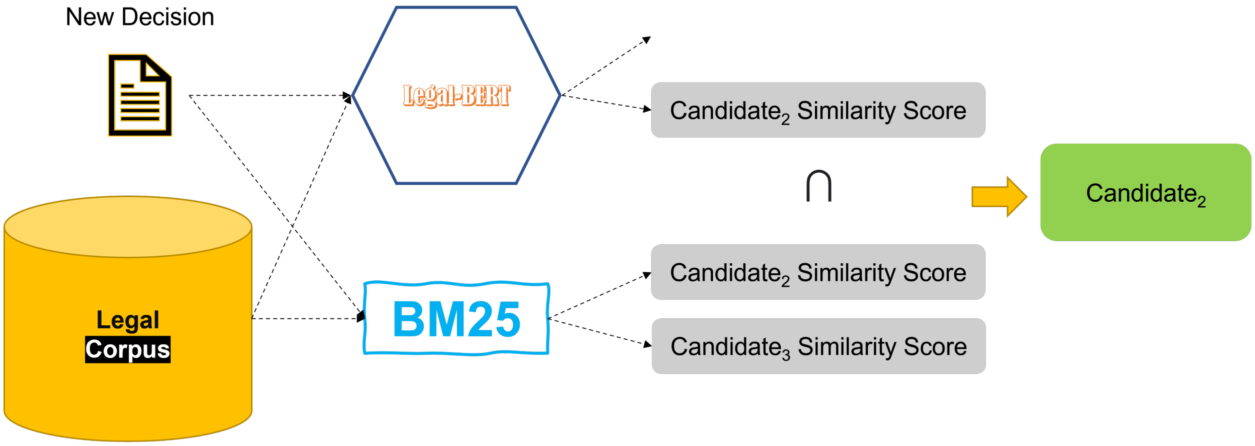

New Decision
Legal Corpus
Legal-BERT
BM25
Candidate$_2$ Similarity Score
∩
Candidate$_2$ Similarity Score
Candidate$_3$ Similarity Score
Candidate$_2$

Figure 3.4: Relevant Candidate Selection Using Intersection Approach

| | F1 | Precision | Recall |
|---|---|---|---|
| monot5-ensemble.txt | 0.6783 | 0.6964 | 0.6610 |
| **Intersection (AMR) (Ours)** | **0.6694** | **0.6532** | **0.6864** |
| Intersection (W/O AMR) (Ours) | 0.6638 | 0.6667 | 0.6610 |
| BM25 + Legal-BERT (Ours) | 0.6612 | 0.6452 | 0.6780 |
| BM25(AMR) + Legal-BERT (Ours) | 0.6452 | 0.6154 | 0.6780 |
| Legal-BERT (Ours) | 0.6431 | 0.5985 | 0.6949 |
| BM25 (Ours) | 0.5164 | 0.5 | 0.5339 |
| BM25 (AMR) (Ours) | 0.4891 | 0.5045 | 0.4746 |
| BM25 (AMR) (W/O Spacy)(Ours) | 0.4848 | 0.4955 | 0.4745 |
| bm25EF[35] | 0.3204 | 0.1980 | 0.8390 |

Table 3.3: Results on COLIEE 2022 Task2 test Set

The third-best performing model is *BM25 + Legal-BERT* with an f1 score of 0.6612, which is very close to the second-best model. However, this model exhibits lower precision and recall values, implying that it may generate more false positives or false negatives.

Models *BM25(AMR) + Legal-BERT* and *Intersection (W/O AMR)* achieved comparable f1 scores of 0.6452 and 0.6638, respectively. However, the former has lower precision, while the latter exhibits lower recall, suggesting that the models may possess divergent strengths and weaknesses.

*Legal-BERT* achieved an f1 score of 0.6431, which is relatively close to the aforementioned models. Nonetheless, its precision value is much lower than the other models, implying that it may generate more false positives.

Conversely, the last three results from BM25, *BM25*, *BM25 (AMR)*, and *BM25 (AMR) (W/O Spacy)*, exhibited significantly lower f1 scores than the other models, which indicates less accuracy.

## 3.4 Chapter Conclusion

The experimental findings have brought to light certain issues in our research, namely:

1. Are there any other potential applications of AMR in information retrieval systems?

2. As indicated in Table 3.1, the candidate length may extend up to 3795 tokens due to the limited capacity of the transformer model, which allows a maximum length of 512 tokens.

3. There is an imbalance between the number of positive and negative samples in the training and test sets.

In the upcoming chapter, we aim to present our approach to addressing these challenges and achieving state-of-the-art (SOTA) results on task2 COLIEE 2021 and COLIEE 2022.

# Chapter 4

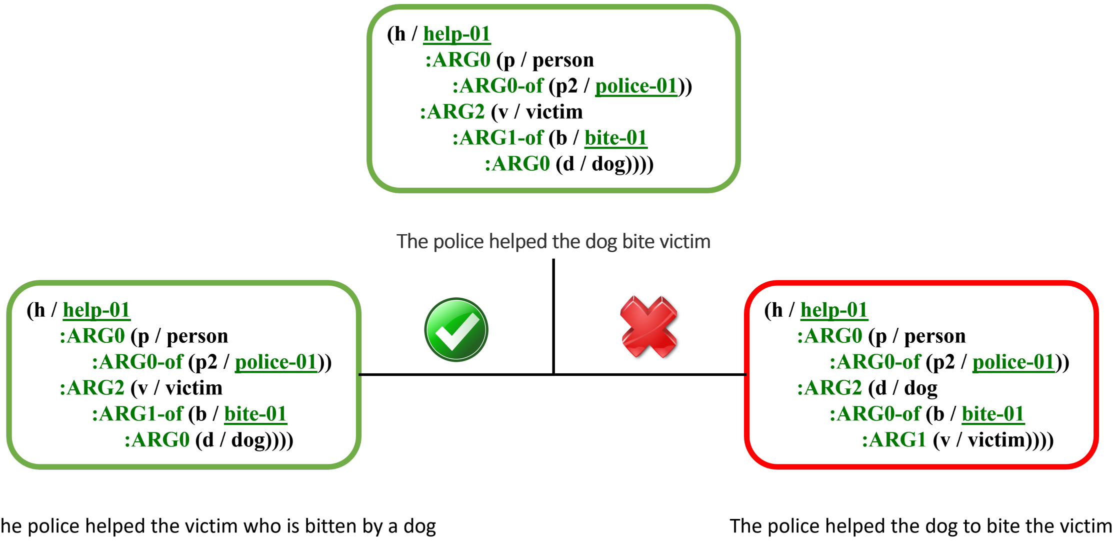
# Structural Information from AMR and Information Combination

## 4.1 Introduction

In the 1970s, Luhn introduced the first algorithm of Information Retrieval (IR) called *term frequency* (TF) weights [30]. This method calculated the occurrence of words within a document and was later complemented by Jonse's work, which introduced the concept of Inverse Document Frequency (IDF) [30]. The introduction of IDF was based on the assumption that less common words are associated with more specific concepts that are more important for IR. In 1973, Salton and Yang proposed a method to combine TF and IDF, which effectively ranks documents using Okapi BM25 (BM standing for best matching) [50]. BM25 and its variants, such as ATIRE BM25 [60], BM25L [31], BM25+ [31], BM25T [31], generate TF-IDF score that is commonly used in document retrieval.

Despite their usefulness, these algorithms have been found to be inadequate when used in the legal domain. This is because there is a multiplicity of meanings associated with words or situations, depending on the perspective of the person. As shown in Figure 1.1, ambiguity arises when a sentence has more than one interpretation, making it difficult even for humans to determine the correct meaning. Such examples pose a significant challenge to the TF-IDF algorithm due to the presence of numerous overlapping words, including "the," "police," "dog," and others. This weakness of the algorithm is further highlighted by the results of the Competition on Legal Information Extraction/Entailment (COLIEE) competition, where methods using word overlapping are unable to compete with modern approaches.

The past few years have witnessed a significant increase in the performance of Deep Learning (DL), attributed to advancements in hardware and DL architectures. The advent of the pre-trained model, BERT [13], in 2018 revolutionized Natural Language Processing (NLP) and resulted in remarkable breakthroughs in various language-based tasks, including Information Retrieval (IR), sentiment analysis, name entity recognition, and question answering. Unlike GloVe [38] representation,



```
(h / help-01
    :ARG0 (p / person
        :ARG0-of (p2 / police-01))
    :ARG2 (v / victim
        :ARG1-of (b / bite-01
            :ARG0 (d / dog))))
```

The police helped the dog bite victim

```
(h / help-01
    :ARG0 (p / person
        :ARG0-of (p2 / police-01))
    :ARG2 (v / victim
        :ARG1-of (b / bite-01
            :ARG0 (d / dog))))
```

The police helped the victim who is bitten by a dog

```
(h / help-01
    :ARG0 (p / person
        :ARG0-of (p2 / police-01))
    :ARG2 (d / dog
        :ARG0-of (b / bite-01
            :ARG1 (v / victim))))
```

The police helped the dog to bite the victim

Figure 4.1: AMR representation for resolving ambiguity sentence

which assigns a constant vector to a word regardless of its context, BERT generates contextualized word embeddings, which can vary for the same word depending on the context to which it belongs. Consequently, several pre-trained models such as RoBERTa [29] and ALBERT [23] were developed with modifications in the architecture and training data, but based on the transformer model [61] and trained on a large corpus. Extensive experimentation has shown that the use of a transformer-like model can achieve state-of-the-art (SOTA) results in downstream tasks.

For instance, the JNLP team fine-tuned the BERT model on silver data and combined it with BM25 to capture semantic and lexical features, respectively. The combination of these features played an integral part in the team's high-rank achievement in the IR task in the COLIEE 2020 [34] and 2021 [33] competitions. Therefore, the combination of lexical and semantic information, using TF-IDF and a transformer-like model, respectively, plays a crucial role in developing an outstanding system, as demonstrated in various research works.

Numerous studies have investigated and enhanced various aspects of the approach whereby an increase in the amount of information available to an Information Retrieval (IR) system results in improved performance. However, the semantic information obtained from the transformer model and the lexical information obtained from the BM25 algorithm are the only factors that have been considered to date. This research introduces a novel approach that incorporates semantic enhancement information generated by Abstract Meaning Representation (AMR) into the IR system.

AMR is a rooted, labeled graph that presents information in a manner that is easy for humans to comprehend. Its primary significance is that it abstracts away the meaning of a given sentence. Different sentences that convey the same basic meaning will have identical AMR graphs, which clearly display information such as subject, object, verb, and time without any ambiguity or vagueness. Specifically, an AMR representation comprises nodes that represent concepts and edges that represent semantic relations between pairs of concepts. Furthermore, words with functional meanings are ignored to eliminate ambiguity caused by syntax or articulateness, enabling the focus to be solely on factual meaning.

AMR has been proven to be useful in many areas, including Biomedical Event Extraction [45], multi-document summarization [28], paraphrase detection [18], and more. The correct interpretation of a sentence can be determined with certainty using AMR representation, as illustrated in Figure 4.1. For instance, the definition of ARG2 for **help-01** is *"benefactive, secondary agent (when separate from arg1)"*. Consequently, it is evident that in this context, ARG2 of **help-01** must refer to the **victim**. The correct interpretation of the sentence presented in Figure 3.1 can be determined by analyzing the AMR graph.

This study is premised on the assumption that incorporating more semantic information into the textual entailment system can improve its performance. The study aims to achieve the following objectives:

1. Demonstrate that the addition of Abstract Meaning Representation (AMR) using **triplet** features can enhance the performance of the information retrieval (IR) system, leading to state-of-the-art (SOTA) results on task 2 of the Competition on Legal Information Extraction/Entailment (COLIEE) in both 2021 and 2022.

2. Present an effective approach for integrating different types of information to improve candidate ranking.

3. Provide viable techniques for addressing the constraints imposed by data and length limitations of the transformer model.

## 4.2 Methodology

Drawing on prior research, we compute the similarity score between a given query-candidate paragraph pair by integrating the transformer model for semantic information and BM25 for lexical information. Our proposed framework, however, incorporates an extra module, known as the **Semantic Enhancement Module**, which is designed to augment the ranking of candidates with additional useful information. As depicted in Figure 4.2, we employ three distinct modules to capture varied information and integrate them cohesively to enhance performance.

AMR encompasses not only the fundamental meaning but also the contextual information represented by concepts, which is elucidated in Section 4.4.1. This information has the potential to improve semantic understanding and candidate ranking. Moreover, to tackle the 512 tokens limitation of transformer models, we split the documents into passages of a suitable length. AMR parsers are well-equipped to handle these passages since they are trained on short passages.

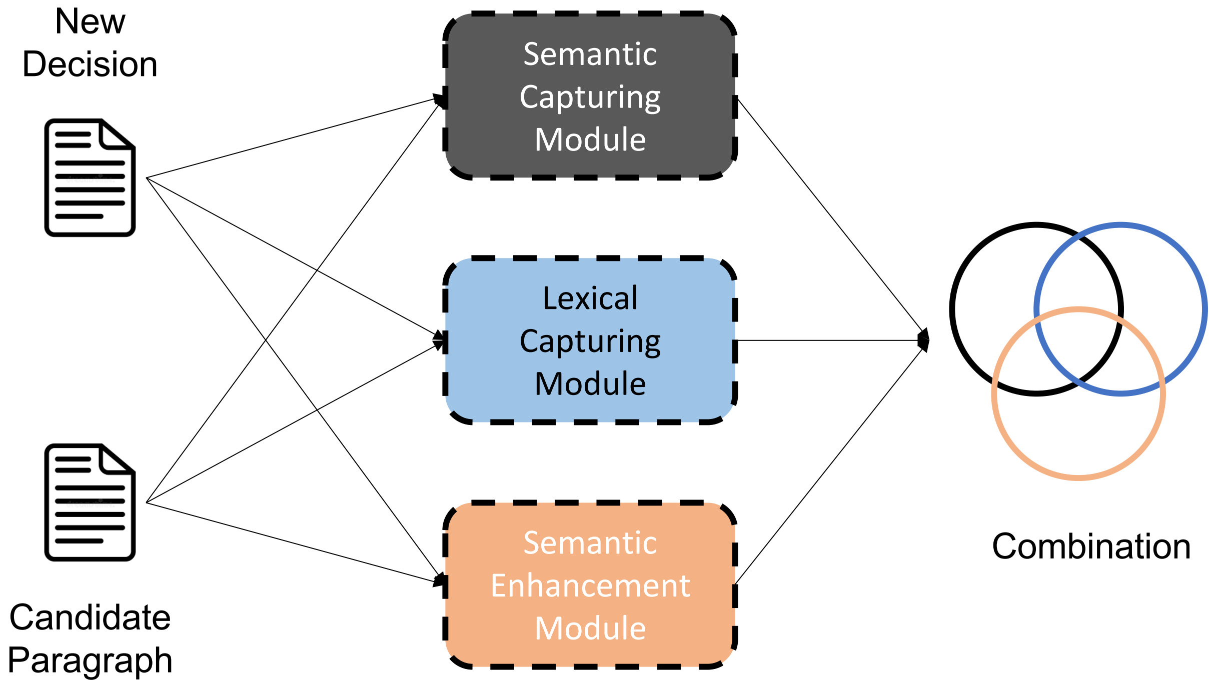


Figure 4.2: High-Level Look of Information Combination Using Semantic Enhancement Module

### 4.2.1 Structural Information From AMR

In accordance with prior research [39], the utilization of sole nodes from the Abstract Meaning Representation (AMR) graph presents inherent challenges. As depicted in Figure 4.3, the shared node set between the erroneous and correct sentences becomes apparent. This observation serves to exemplify the diminished efficacy of the BM25 algorithm when applied to datasets within the legal domain.

In this context, triplets refer to the explicit representation of the syntactic relationships between nodes in an AMR graph. They offer a more comprehensive understanding of the sentence structure

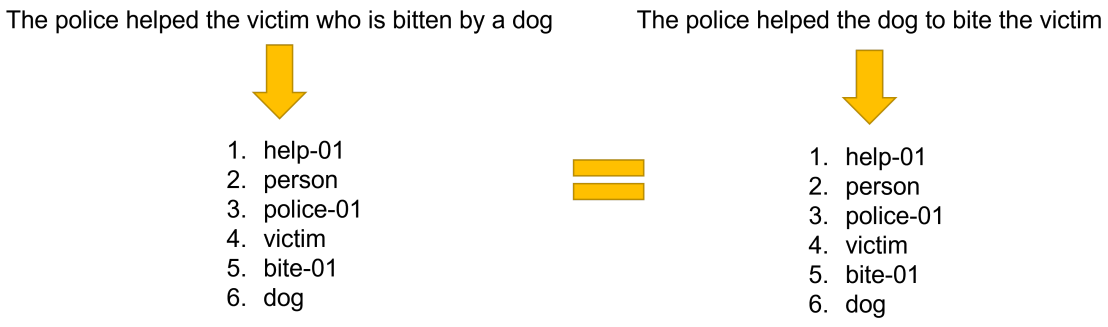


Figure 4.3: Ambiguity During Using AMR Nodes for Ranking Document

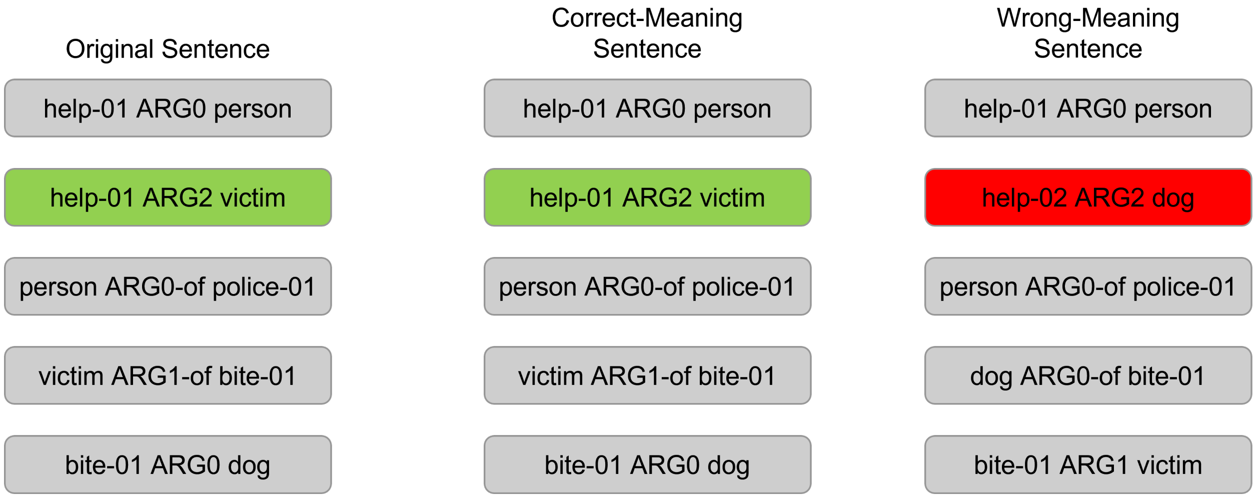


Figure 4.4: Document Ranking Using Triplets of AMR Graph

and facilitate a more nuanced interpretation of meaning. By incorporating triplets into AMR, researchers have gained a deeper insight into the complex interdependencies between words and phrases within a sentence.

One of the primary advantages of utilizing triplets in AMR is the ability to capture more fine-grained semantic relationships. With the inclusion of relation labels, the AMR graph becomes enriched with valuable information about the connections between concepts. This allows for a more precise representation of semantic roles, dependencies, and other intricate linguistic phenomena.

Furthermore, the use of triplets enhances the robustness and interpretability of the AMR representation. By explicitly indicating the head-dependent relationships, it becomes easier to discern the hierarchical structure of a sentence and determine the roles played by different elements. This can aid in tasks such as information extraction, question answering, and machine translation, where a deeper understanding of the sentence structure is crucial for accurate results.

As depicted in Figure 4.4, a visual comparison highlights the disparity between the correct-meaning sentence and the wrong-meaning sentence. It becomes evident that the "police" are intended to assist the "victim" rather than the "dog". The present study encompasses numerous experiments aimed at substantiating the hypothesis that employing triplet-level representations surpasses the efficacy of node-level representations.

### 4.2.2 Lexical Features

The development of effective information retrieval (IR) systems is essential for efficiently handling the vast amount of legal data that has accumulated over the years. In this context, the BM25 algorithms have been widely used in the legal domain to improve the IR system's performance. These variants have been proven effective in addressing the unique characteristics of legal texts, including complex sentence structures, legal terminologies, and extensive use of citation links. However, the performance of these variants can vary depending on the specific legal task and dataset. Therefore, it is crucial to evaluate and compare the effectiveness of these variants in the legal domain. This paper presents a comparative study [60] of:

1. BM25 Okapi
2. BM25+
3. BM25L

for legal text retrieval, aiming to identify the most suitable variant for a given legal task and dataset. The results comparison between these variants can be found in Section 4.3.2.

### 4.2.3 Semantic Features

There is a vast array of pre-trained models available for capturing latent semantic information within text. This study employs several transformer model variants to extract semantic features, namely:

1. BERT [13]
2. RoBERTA [29]
3. Legal-BERT, which is a pre-trained model specifically trained on a sizable legal corpus [10]

Prior research has shown the importance of domain adaptation in achieving optimal performance in downstream tasks. While BERT and RoBERTa are commonly used pre-trained models for open-domain tasks, Legal-BERThas demonstrated effectiveness in the legal domain. However, the quality of fine-tuning data is also a crucial factor in achieving optimal performance. In the following section, we discuss how we created training data for transformer models and explore methods for capturing semantic features.

#### Cross Validation and down-sampling

Table 3.1 presents an analysis of the limitations of the training data for Task 2 of COLIEE 2022. The shortage of training data, with only 525 examples in total, and the imbalanced labels, with 599 positive and 18,141 negative labels, pose significant challenges for deep learning models. The importance of large training datasets in deep learning is well-established, as demonstrated by the success of ImageNet [22] and GPT-3 [8], among others. When the training dataset is limited, deep learning models often struggle to extract useful information, resulting in poor performance.

Moreover, the class imbalance in the COLIEE 2022 Task 2 training dataset, with a positive-negative ratio of nearly 1:30, may bias the classifier toward predicting the dominant class. This issue can result in the model not learning enough from positive examples as compared to negative examples. To address this issue, we employed cross-validation and down-sampling techniques.

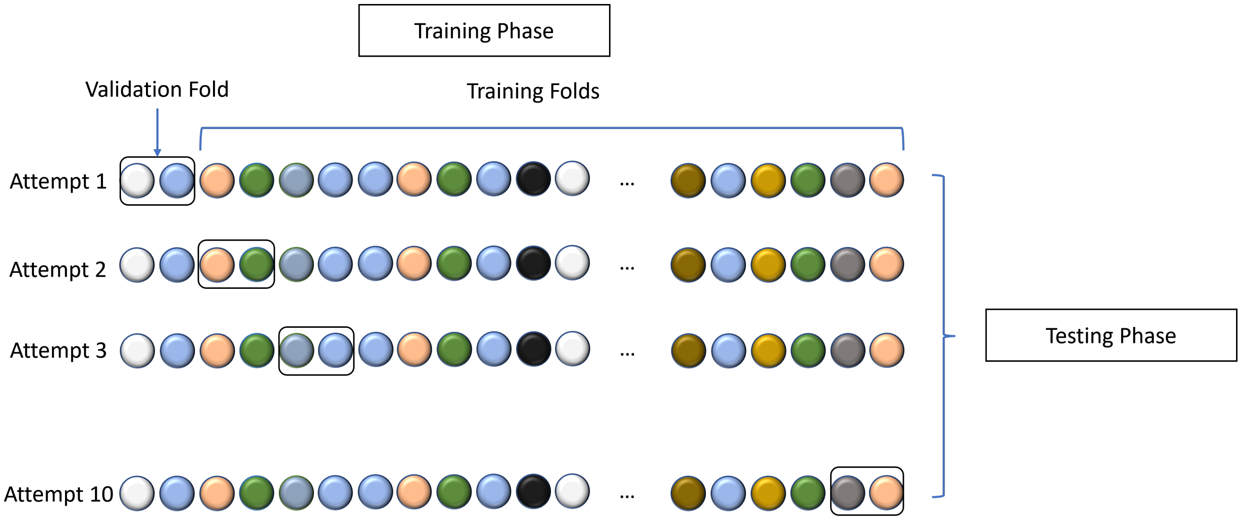


Figure 4.5: Cross Validation for Mining The Best Training Set

Down-sampling the dataset by a factor of 4, 5, 6, or 7 improved the class balance to 1 positive to 7, 6, 5, or 4 negatives, respectively.

Cross-validation is a widely-used technique in machine learning to evaluate the model's performance by resampling the limited set of data. However, with a small amount of training data, such as in the COLIEE competition, over-fitting may occur, making cross-validation necessary. To avoid over-fitting, we utilized k-fold cross-validation, which helped us identify which parts of the training data are useful for the transformer model. Figure 4.5 illustrates our approach. For the given set of training dataset $D$ given by the organizer, we separate it into 10 folds $F = \{f_1, f_2, ..., f_{10}\}$. For each fold:

1. Take this fold as a validation set.

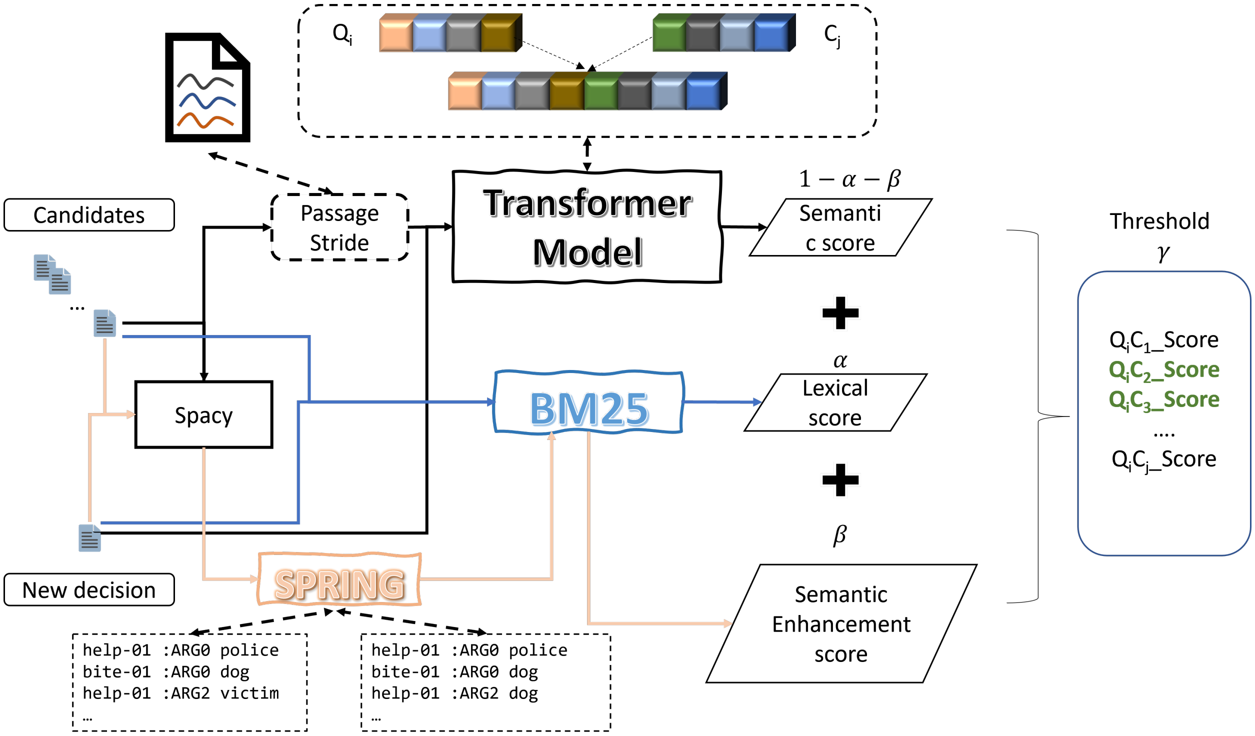
2. Assign the remaining group as a training set.

3. Train a transformer model on the training and validation set and evaluate the performance on the test set.

4. Retain the evaluation score and move to the next attempt.

### Document Stride Prediction

Transformer models have emerged as one of the most powerful deep learning (DL) architectures in recent years. These models take text as input and generate large vectors or arrays, which are then further processed. The high-dimensional similarity derived from these vectors has been demonstrated to be highly effective, as evidenced by the performance of transformer models on sentence similarity tasks such as the Stanford Natural Language Inference (SNLI) corpus [7]. The



```
help-01 :ARG0 police
bite-01 :ARG0 dog
help-01 :ARG2 victim
…
```

```
help-01 :ARG0 police
bite-01 :ARG0 dog
help-01 :ARG2 dog
…
```

Figure 4.6: Details of Information Enhancement Framework

accuracy of frameworks-based transformer models on such tasks can be over 90%. In our overall system, the transformer block (as shown in Figure 4.6) utilizes a cross-encoder approach to determine sentence similarity, as follows:

1. The query and possible candidate are simultaneously fed into the transformer model, which outputs a score between 0 and 1 that indicates the degree of relevance of the document to the query.

2. The process is repeated for other candidates.

3. The system selects candidates with the highest scores as the most relevant ones.

In order to emphasize the significance of this section, we will discuss the advantages and disadvantages of transformer-based models. In Vaswani's [61] publication, "Attention is all you need," it is suggested that attention can increase the training speed and performance of deep learning models. Consider the sentence "The police helped the victim who is bitten by a dog." For humans, it is a simple question to determine the meaning of the word "who" - is it referring to the police, the victim, or the dog? However, for machines, this is a complex task. To represent this sentence, it must first be tokenized into words and then converted into a vector of numerical values. As illustrated in Figure 4.7, the attention mechanism facilitates the word "who" in obtaining information from other words, which can lead to a more informative representation of this word. When there is appropriate

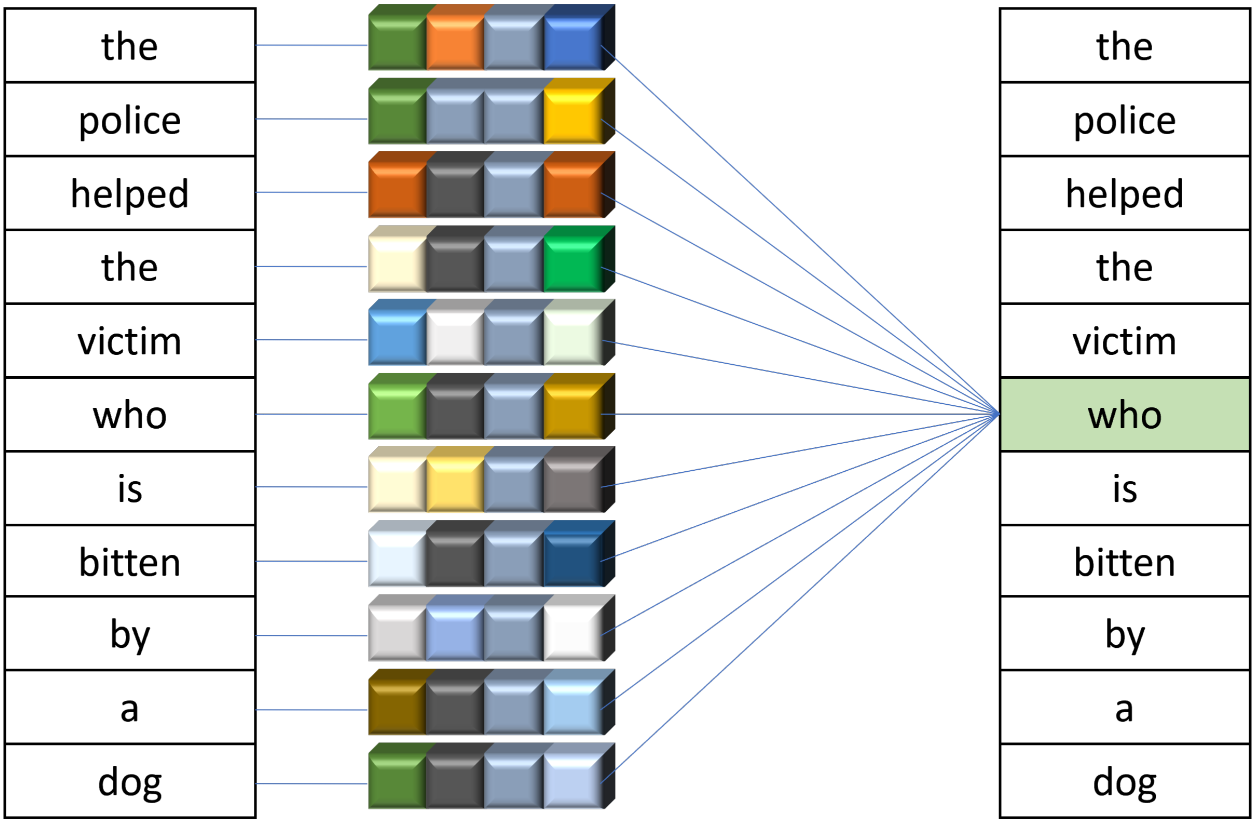


Figure 4.7: Self-attention Algorithm Illustration

contextual information for a word, the attention algorithm can generate optimal representations for each word.

In contrast, when the context is too lengthy, each word's representation suffers due to the abundance of information it receives from other words, such as a context comprising 5000 words. Consequently, transformer models can only process input sequences with a maximum length of 512 tokens, which poses a significant drawback since contemporary documents are considerably longer. As evident from the analysis of COLIEE 2022 presented in Table 3.1, many candidates surpass the length threshold of 512 tokens, and using transformer models incompetently can result in poor performance.

To address this issue, we leveraged sentence segmentation facilitated by Spacy[1] to split long candidates $C_j$ into sentences $S = s_1, s_2, ..., s_K$ and employed Algorithm 1 to extract a set of passages $P = p_1, p_2, ..., p_M$. The similarity score for the pair $Q_i$-$C_j$ was calculated by Equation 3.2.

Here, $N$ refers to the total number of passages, and $O_i$ denotes the similarity score of $Q_i$ and the n-th passage of the corresponding candidate obtained through fine-tuning the transformer model on the data described in Section 4.2.3.

[1] https://spacy.io/

**Algorithm 1** Context Striding

```
Require: $S$, query, tokenizer
P ⇐ {}
i ⇐ 0
while $i < |S|$ do
    $j ⇐ i$
    while $|(tokenizer(query, P[i:j]))| < 512$ do
        if $j < |S|$ then
            $j ⇐ j + 1$
        else$j = |S|$
            break
        end if
    end while
    if $j = |S|$ then
        P.insert($S[i : j - 1]$)          ▷$S[i : j - 1]$ is a passage from i index to j-1 index
        $i ⇐ i + 1$
        break
    else
        P.insert($S[i : j - 1]$)          ▷$S[i : j - 1]$ is a passage from i index to j-1 index
        $i ⇐ i + 1$
    end if
end while
```

### 4.2.4 Semantic Enhancement Information

The objective of this section is to discuss obtaining useful information from Abstract Meaning Representation (AMR) to enhance an Information Retrieval (IR) system, with the quality of AMR significantly impacting the parser choice. Therefore, it is crucial to select a high-quality AMR parser. In this study, we utilized SPRING, a state-of-the-art (SOTA) AMR parser developed by Bevilacqua in 2021 [24]. However, our observations revealed that SPRING is incapable of handling excessively long paragraphs due to the parser's backbone being a transformer model. Additionally, due to the complexity of AMR, the availability of data is limited. Furthermore, data annotation necessitates comprehension of linguistics, typically for short sentences. To achieve better performance, query Qi and candidate Cj are segmented into sentences using Spacy. Each sentence is subsequently parsed into AMR logical triples, and once the parsing process is completed, an AMR graph for the query and candidate can be obtained as a set of logical triples.

Considering the sample from Chapter 3, *"the jurisprudence established that a leave to appeal proceeding was a preliminary step to a hearing on the merits, and was a lower hurdle for the applicant for leave to meet since the case did not have to be proven"*

The logical triples produced by the Spring parser [5] can depict the AMR as follows:

establish-01 :ARG0 jurisprudence,
establish-01 :ARG1 and,
and :op1 step-01,
step-01 :ARG1 leave-16,
leave-16 :ARG2 proceeding-02,
proceeding-02 :ARG1 appeal-01,
step-01 :ARG2 hearing-02,
hearing-02 :ARG2 merit-01,
step-01 :mod preliminary,
and :op2 hurdle-01,
hurdle-01 :ARG1 apply-01,
apply-01 :ARG0 person
apply-01 :ARG1 leave-16,
leave-16 :ARG2 meet-03,
meet-03 :ARG0 person,
cause-01 :ARG1 hurdle-01,
cause-01 :ARG0 obligate-01,
obligate-01 :polarity -,
obligate-01 :ARG2 prove-01,
prove-01 :ARG1 case-03,
have-degree-91 :ARG1 hurdle-01,
have-degree-91 :ARG2 low-04,
low-04 :ARG1 hurdle-01,
have-degree-91 :ARG3 more

Based on our observations, it has been found that several triples in the query and candidate graphs have a low quality which makes it challenging to rank the candidates accurately. For instance, triples like *have-degree-91 :ARG1 hurdle-01*, *have-degree-91 :ARG2 low*, and *have-degree-91 :ARG3 more* are frequently encountered in the graphs, representing the degree of intensity. We propose that triples consisting of a verb, subject, and object contain more significance for ranking the documents. To mitigate the effect of frequently occurring tokens that lack semantic value, such as *have-degree-91* and *more*, we applied three variants of BM25, which are discussed in Section 4.2.2, to the logical triples of queries and candidates to extract semantic information through concepts and edges. Unlike the approach we used in Chapter 3, we keep all the related information (edge of AMR), and the concept definition such as *leave-16* instead of *leave*.

### 4.2.5 Information Combination

In a number of previous studies [34] [33], the coefficient $\alpha$ has been utilized to integrate lexical and semantic scores. In the current investigation, we incorporate an additional semantic feature derived from AMR and introduce a new coefficient $\beta$ to unify the semantic, lexical, and semantic enhancement factors into a single score. As demonstrated in Figure 4.6, once we obtain the semantic score $S_{s_{ij}}$, lexical score $S_{l_{ij}}$, and semantic enhancement score $S_{se_{ij}}$ for the given query $i$ and candidate $j$, we compute the combination score using Equation 4.1:

$$S_{ij_{combination}} = \alpha \cdot S_{l_{ij}} + \beta \cdot S_{se_{ij}} + (1 - \alpha - \beta) \cdot S_{s_{ij}} \tag{4.1}$$

Determining the optimal values for $\alpha$ and $\beta$ is a challenging task, as selecting appropriate values can significantly improve the effectiveness of the information retrieval system. Additionally, given a query, it is necessary to identify which candidate is the entailment paragraph among a set of candidates. To manage the number of candidates chosen as entailment paragraphs, a threshold $\gamma$ is employed. If a candidate's $S_{combination}$ score equals or exceeds $\gamma$, the candidate is designated as an entailment paragraph for the given query. To address this issue, we present Algorithm 2 to determine the optimal values of $\alpha$ and $\beta$, as well as the threshold $\gamma$.

**Algorithm 2** Coefficient Mining

```
Require: $S_S$, $S_L$, $S_ST$, $\alpha$ ,$\beta$ ,$\gamma$
$results \Leftarrow \{\}$
$\alpha \Leftarrow 0$
while $\alpha < 1$ do
    $\beta \Leftarrow 0$
    while $1 - \alpha - \beta > 0$ do
        $\beta \Leftarrow \beta + 0.5$
        $\gamma \Leftarrow 0.0000075$
        while $\gamma < 0.001$ do
            $\gamma \Leftarrow \gamma + 0.000005$
            $Score \Leftarrow$ evaluate($S_S, S_L, S_ST, \alpha, \beta, \gamma$)        ▷Recal, Precision, F1
            $results$.insert($\alpha, \beta, \gamma, Score$)
        end while
    end while
     $\alpha \Leftarrow \alpha + 0.5$
end while
```

| | Train | Test |
|---|---|---|
| #Query | 425 | 100 |
| #Candidate | 15216 | 3524 |
| #Candidate/Query | 35.7 | 34.9 |
| #Positive Sample | 499 | 117 |
| #Negative Sample | 14717 | 3407 |
| Query Average Length (tokens[1]) | 44.1 | 38.9 |
| Query max Length (tokens[1]) | 133 | 133 |
| Candidate Average Length (tokens[1]) | 138.5 | 138.5 |
| Candidate Max Length (tokens[1]) | 3440 | 3795 |

[a] Tokenized by Transformer's Tokenizer.

Table 4.1: Task 2 COLIEE 2021 Analysis

## 4.3 Experiment Results

### 4.3.1 Evaluation Dataset

For evaluating our proposed method, we use a dataset of Task2 COLIEE 2022 and COLIEE 2021 as evaluations task. The data analysis of Task2 COLIEE 2022 can be seen in Table 3.1. Table 4.1 presents the statistics for task 2 COLIEE 2021. The dataset contains two sets, Train and Test, and the statistics are provided for each set. The dataset consists of queries and candidates, where each query is associated with a set of candidates, and the goal is to rank the candidates based on their relevance to the query.

The Train set contains 425 queries and 15,216 candidates, resulting in an average of 35.7 candidates per query. The Test set contains 100 queries and 3,524 candidates, resulting in an average of 34.9 candidates per query. The number of positive samples (relevant candidates) is 499 for the Train set and 117 for the Test set, while the number of negative samples (irrelevant candidates) is 14,717 for the Train set and 3,407 for the Test set.

The average length of the queries in the Train set is 44.1 tokens, while the average length of the queries in the Test set is 38.9 tokens. The maximum length of queries is 133 tokens in both sets. The average length of candidates is 138.5 tokens in both sets, while the maximum length of candidates is 3,440 tokens in the Train set and 3,795 tokens in the Test set.

The statistics show that the Train and Test sets have a similar number of queries and candidates per query. The number of positive samples is relatively small compared to the number of negative samples, indicating that the dataset is imbalanced. The average and maximum lengths of queries and candidates are similar in both sets, suggesting that the sets have a similar distribution of query and candidate lengths. These statistics can be used to understand the characteristics of the dataset and design models for the task.

### 4.3.2 Lexical Similarity Results

In this section, we present a comparative analysis of the IR system performance using three variants of the BM25 retrieval model, namely BM25 Okapi, BM25+, and BM25L. We anticipated that a desirable outcome would be an increase in the F1 score. Initially, we set the threshold $\gamma$ to the lowest possible value, resulting in the retrieval of only one candidate as an entailment paragraph for each query, and the Precision score was high as expected. As the threshold $\gamma$ increased (allowing the retrieval of more than one candidate for each query), the F1 score increased until reaching a peak at $\gamma = 0.0002$ and then started to drop when $\gamma$ exceeded 0.002. As shown in Figure 4.8, BM25 Okapi and BM25+ exhibit promising results, achieving F1 scores of **51.14%** and **49.4**, respectively. Conversely, BM25L performed poorly on the COLIEE test set and is not used in the combination phase.

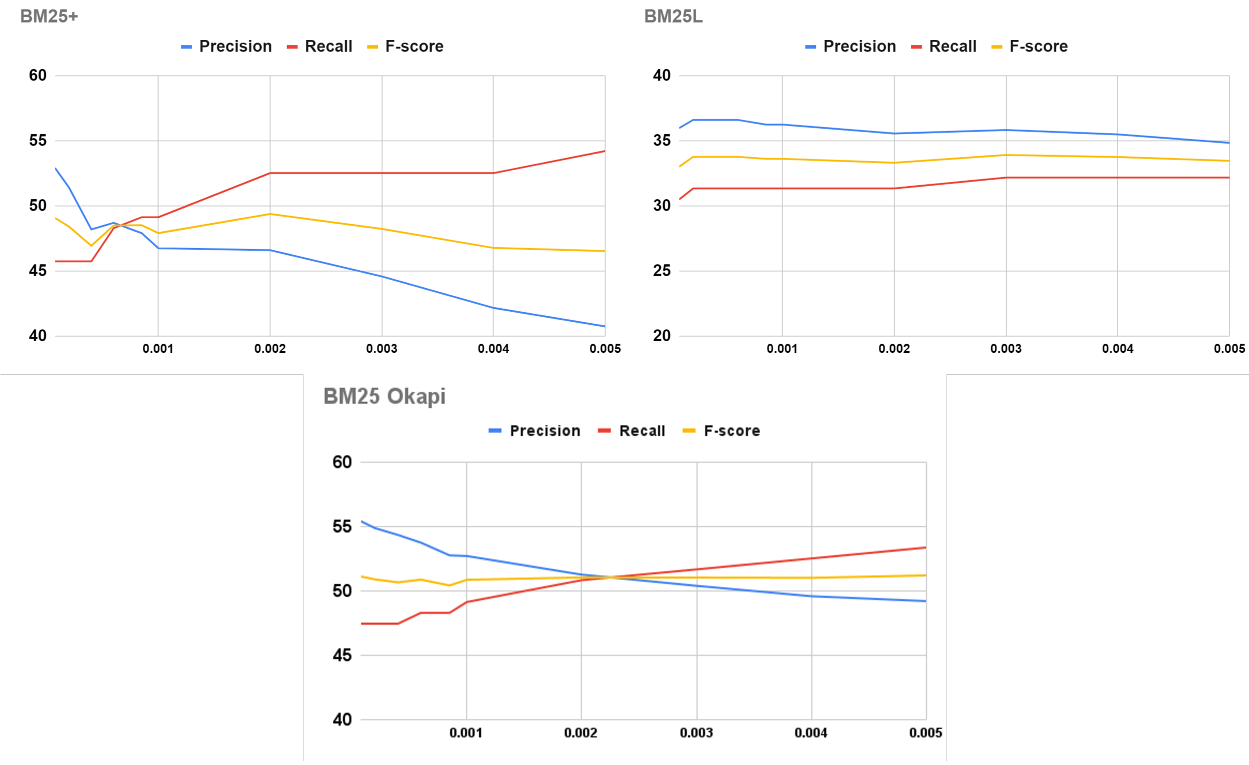


Figure 4.8: Results of BM25 Variants on Test Set of Task 2 COLIEE 2022

### 4.3.3 Semantic Information From AMR Results

To capture the semantic information between the query's AMR representation and the candidate's AMR representation, we employed different variants of the BM25 algorithm. We adopted the same settings and threshold $\gamma$ as used in the experiments described in Section 3.2. Our analysis, presented in Figure 4.9, indicates that this approach is not effective in the legal domain, as evidenced by the F-score remaining under 40% across all thresholds. Conversely, we observed that BM25 Okapi and BM25+ perform well in COIEE2022, with an overall F1 score exceeding 50%.

### 4.3.4 Semantic Similarity Results

To ensure consistency and fairness across models, we employ the configuration presented in Table 3.2 for all transformer model variants. In this study, we utilize the data collected in Section 4.2.3 to train three transformer models (BERT base, RoBERTa base, and LEGAL-BERT) for capturing semantic features. We vary the ratio between positive and negative samples during training (i.e., 1:4, 1:5, 1:6, 1:7, 1:8), and evaluate each model using Precision@1, Recall@1, and F1@1 metrics (evaluating on the top 1 retrieved paragraph using Equations 3.3, 3.4, and 3.5) on different folds. We select models based on their performance on the top 1 retrieved paragraph, as precision tends to drop with increasing threshold $\gamma$. This approach aims to limit the search space and identify the

| Ratio 1:4 | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Attempt | 0 | | | 1 | | | 2 | | | 3 | | | 4 | | |
| Metric | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F |
| bert-base-uncased | 52.00 | 44.07 | 47.71 | 59.00 | 50.00 | 54.13 | 53.00 | 44.92 | 48.62 | 28.0 | 23.7 | 25.7 | 29.0 | 24.6 | 26.6 |
| RoBERTa | 28.00 | 23.73 | 25.69 | 54.00 | 45.76 | 49.54 | 43.00 | 36.44 | 39.45 | 49.0 | 41.5 | 45.0 | 46.0 | 39.0 | 42.2 |
| LEGAL_BERT | 62.00 | 52.54 | 56.88 | 63.00 | 53.39 | 57.80 | 63.00 | 53.39 | 57.80 | 66.0 | 55.9 | 60.6 | 73.0 | 61.9 | 67.0 |
| Ratio 1:5 | | | | | | | | | | | | | | | |
| bert-base-uncased | 51.00 | 43.22 | 46.79 | 54.00 | 45.76 | 49.54 | 57.00 | 48.31 | 52.29 | 51.0 | 43.2 | 46.8 | 54.0 | 45.8 | 49.5 |
| RoBERTa | 11.00 | 9.32 | 10.09 | 61.00 | 51.69 | 55.96 | 46.00 | 38.98 | 42.20 | 65.0 | 55.0 | 59.6 | 53.0 | 44.9 | 48.6 |
| LEGAL_BERT | 57.00 | 48.31 | 52.29 | 57.00 | 48.31 | 52.29 | 64.00 | 54.24 | 58.72 | 52.0 | 44.1 | 47.7 | 64.0 | 54.2 | 58.7 |
| Ratio 1:6 | | | | | | | | | | | | | | | |
| bert-base-uncased | 41.00 | 34.75 | 37.61 | 29.00 | 24.58 | 26.61 | 44.00 | 37.29 | 40.37 | 32.0 | 27.1 | 29.4 | 44.0 | 37.3 | 40.4 |
| RoBERTa | 51.00 | 43.22 | 46.79 | 56.00 | 47.46 | 51.38 | 57.00 | 48.31 | 52.29 | 58.0 | 49.2 | 53.2 | 33.0 | 28.0 | 30.3 |
| LEGAL_BERT | 68.00 | 57.63 | 62.39 | 69.00 | 58.47 | 63.30 | 72.00 | 61.02 | 66.06 | 65.0 | 55.1 | 59.6 | 71.0 | 60.2 | 65.1 |
| Ratio 1:7 | | | | | | | | | | | | | | | |
| bert-base-uncased | 36.00 | 30.51 | 33.03 | 41.00 | 34.75 | 37.61 | 33.00 | 27.97 | 30.28 | 40.0 | 33.9 | 36.7 | 30.0 | 25.4 | 27.5 |
| RoBERTa | 53.00 | 44.92 | 48.62 | 48.00 | 40.68 | 44.04 | 55.00 | 46.61 | 50.46 | 65.0 | 55.0 | 59.6 | 54.0 | 45.8 | 49.5 |
| LEGAL_BERT | 69.00 | 58.47 | 63.30 | 72.00 | 61.02 | 66.06 | 70.00 | 59.32 | 64.22 | 51.0 | 43.2 | 46.8 | 70.0 | 59.3 | 64.2 |
| Ratio 8 | | | | | | | | | | | | | | | |
| bert-base-uncased | 40.00 | 33.90 | 36.70 | 42.00 | 35.59 | 38.53 | 39.00 | 33.05 | 35.78 | 58.0 | 49.2 | 53.2 | 48.0 | 40.7 | 44.0 |
| RoBERTa | 58.00 | 49.15 | 53.21 | 57.00 | 48.31 | 52.29 | 56.00 | 47.46 | 51.38 | 62.0 | 52.5 | 56.9 | 57.0 | 48.3 | 52.3 |
| LEGAL_BERT | 70.00 | 59.32 | 64.22 | 62.00 | 52.54 | 56.88 | 67.00 | 56.78 | 61.47 | 34.0 | 28.8 | 31.2 | 63.0 | 53.4 | 57.8 |

Table 4.2: Cross Validation and Ratio Selection Results on Precision@1, Recall@1 and F1@1(1)

| Ratio 1:4 | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Attempt | | 5 | | | 6 | | | 7 | | | 8 | | | 9 | |
| Metric | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F |
| bert-base-uncased | 32.00 | 27.12 | 29.36 | 34.0 | 28.8 | 31.2 | 39.00 | 33.05 | 35.78 | 41.0 | 34.8 | 37.6 | 37.0 | 31.4 | 34.0 |
| RoBERTa | 59.00 | 50.00 | 54.13 | 43.0 | 36.4 | 39.5 | 64.00 | 54.24 | 58.72 | 66.0 | 55.9 | 60.6 | 47.0 | 39.8 | 43.1 |
| LEGAL_BERT | 63.00 | 53.39 | 57.80 | 72.0 | 61.0 | 66.1 | 62.00 | 52.54 | 56.88 | 66.0 | 55.9 | 60.6 | 71.0 | 60.2 | 65.1 |
| Ratio 1:5 | | | | | | | | | | | | | | | |
| bert-base-uncased | 52.00 | 44.07 | 47.71 | 17.0 | 14.4 | 15.6 | 51.00 | 43.22 | 46.79 | 34.0 | 28.8 | 31.2 | 55.0 | 46.6 | 50.5 |
| RoBERTa | 48.00 | 40.68 | 44.04 | 56.0 | 47.5 | 51.4 | 63.00 | 53.39 | 57.80 | 44.0 | 37.2 | 40.4 | 60.0 | 50.8 | 55.0 |
| LEGAL_BERT | 69.00 | 58.47 | 63.30 | 62.0 | 52.5 | 56.9 | 64.00 | 54.24 | 58.72 | 54.0 | 45.8 | 49.5 | 62.0 | 52.5 | 56.9 |
| Ratio 1:6 | | | | | | | | | | | | | | | |
| bert-base-uncased | 39.00 | 33.05 | 35.78 | 47.0 | 39.8 | 43.1 | 38.00 | 32.20 | 34.86 | 41.0 | 34.8 | 37.6 | 37.0 | 31.4 | 34.0 |
| RoBERTa | 54.00 | 45.76 | 49.54 | 50.0 | 42.4 | 45.9 | 46.00 | 38.98 | 42.20 | 55.0 | 46.6 | 50.5 | 41.0 | 34.8 | 37.6 |
| LEGAL_BERT | 67.00 | 56.78 | 61.47 | 69.0 | 58.5 | 63.3 | 68.00 | 57.63 | 62.39 | 65.0 | 55.1 | 59.6 | 68.0 | 57.6 | 62.4 |
| Ratio 1:7 | | | | | | | | | | | | | | | |
| bert-base-uncased | 49.00 | 41.53 | 44.95 | 40.0 | 33.9 | 36.7 | 40.00 | 33.90 | 36.70 | 54.0 | 45.8 | 49.5 | 59.0 | 50.0 | 54.1 |
| RoBERTa | 60.00 | 50.85 | 55.05 | 59.0 | 50.0 | 54.1 | 56.00 | 47.46 | 51.38 | 62.0 | 52.5 | 56.9 | 65.0 | 55.1 | 59.6 |
| LEGAL_BERT | 63.00 | 53.39 | 57.80 | 72.0 | 61.0 | 66.0 | 69.00 | 58.47 | 63.30 | 76.0 | 64.4 | 69.7 | 70.0 | 59.3 | 64.2 |
| Ratio 8 | | | | | | | | | | | | | | | |
| bert-base-uncased | 46.00 | 38.98 | 42.20 | 51.0 | 43.2 | 46.7 | 41.00 | 34.75 | 37.61 | 42.0 | 35.6 | 38.5 | 52.0 | 44.1 | 47.7 |
| RoBERTa | 65.00 | 55.08 | 59.63 | 49.0 | 41.5 | 45.0 | 52.00 | 44.07 | 47.71 | 55.0 | 46.6 | 50.5 | 68.0 | 57.6 | 62.4 |
| LEGAL_BERT | 58.00 | 49.15 | 53.21 | 71.0 | 60.1 | 65.1 | 73.00 | 61.86 | 66.97 | 69.0 | 58.5 | 63.3 | 67.0 | 56.8 | 61.5 |

Table 4.3: Cross Validation and Ratio Selection Results on Precision@1, Recall@1 and F1@1(2)

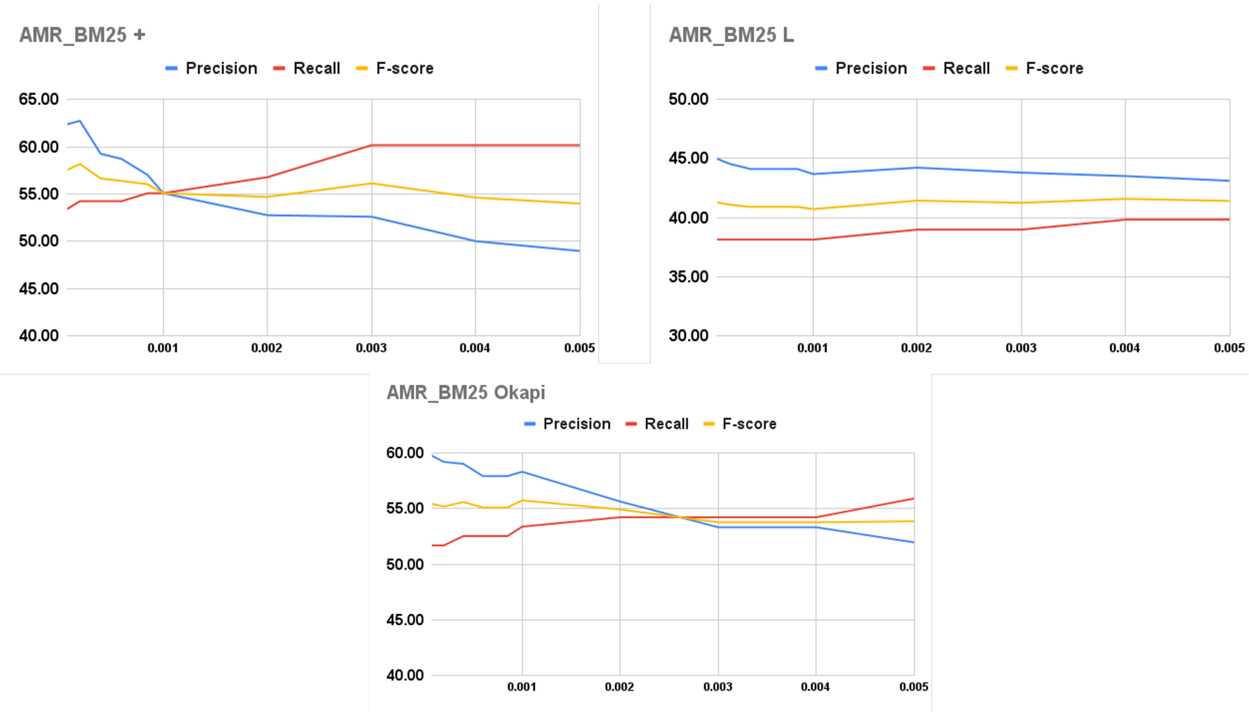


Figure 4.9: Results of BM25 variants on Test Set of Task 2 COLIEE 2022 Using Features of AMR

most effective model variants for the combination phase.

According to prior research, such as SciBERT [4] and BioBERT [25], transformer models trained on a specific domain exhibit superior performance compared to those trained on the general domain. As demonstrated in Table 4.2 and 4.3, the BERT base model, which was trained on the general domain, underperforms on the COLIEE dataset (with the best F1 score of 54.1% and most attempts scoring below 40%). On the other hand, the Roberta base model outperforms BERT, achieving the best F1 score of 62.4%. However, Legal-BERTshows better performance overall, with almost all attempts achieving an F1@1 score of over 60%, and the best-performing version being the one with a positive-negative ratio of 1:7 on the 8th attempt, achieving nearly 70% on F1@1.

### 4.3.5 Information Combination Results

As shown in Sections 4.3.2, 4.3.4, and 4.3.3, it is evident that using only semantic, lexical, or semantic information from AMR cannot yield satisfactory results, and the best F1 score achieved is 69.7% using Legal-BERT(attempt 8 with a positive-negative ratio of 1:7). However, this variant does not perform well in the combination phase, as explained in Section 4.4.2. In Table 4.2 and 4.3, we present the most promising Legal-BERTmodels that were fine-tuned with positive-negative ratios of 1:4 and 1:8, as these models outperform other models at the combination phase (with several models exceeding 70% on F1). For the lexical and semantic enhancement scores, we use the

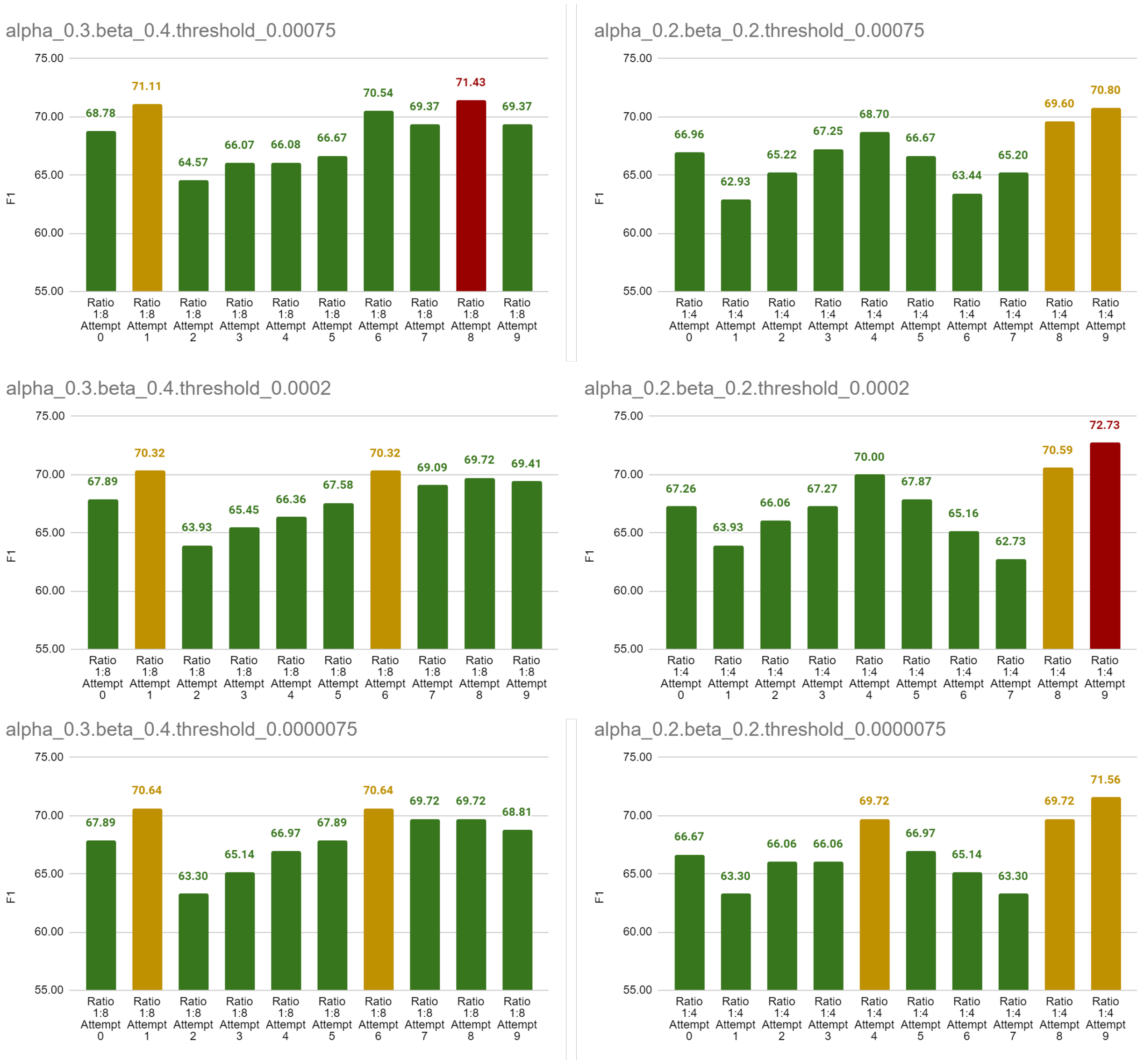


Figure 4.10: Results of Combination of Semantic, Lexical, and Semantic Enhancement Score on Task 2 COLIEE 2022

| Team | File | F1 |
|---|---|---|
|  | **Knowledge Enhancement Framework (alpha_0.2.beta_0.2.threshold_0.0002)** | **72.73** |
|  | **Knowledge Enhancement Framework (alpha_0.3.beta_0.4.threshold_0.00075)** | **72.15** |
| NM | monot5-ensemble.txt | 67.83 |
| NM | monot5-3b.txt | 67.57 |
| JNLP | run2_bert_amr_remove_reduntdant_filter.txt | 66.94 |
| JNLP | run3_bert_BM25.txt | 66.12 |
| jljy | run2_task2.txt | 65.14 |
| jljy | run3_task2.txt | 65.14 |
| JNLP | run1_bert_amr_remove_reduntdant.txt | 64.52 |
| jljy | run1_task2.txt | 63.3 |
| NM | monot5-base.txt | 63.25 |
| UA | res_score-0.95_max-1.txt | 54.46 |
| UA | res_score-0.5_max-1.txt | 53.63 |
| UA | res_score-0.95_max-5.txt | 41.21 |
| nigam | bm25EF.txt | 32.04 |
| nigam | bm25BC.txt | 21.04 |

Table 4.4: Results of Knowledge Enhancement Framework on task 2 of COLIEE 2022

| Only Legal-BERT | alpha | Beta | gamma | Combination Result |
|---|---|---|---|---|
| 0.5161 | 0.5 | 0.25 | 0.0001 | 0.66359 |
| 0.5990 | 0.2 | 0.05 | 0.0001 | 0.7032 |
| 0.5714 | 0.25 | 0.1 | 0.0003 | 0.6879 |
| 0.5898 | 0.2 | 0.1 | 0.0001 | 0.7090 |
| 0.6728 | 0 | 0.15 | 0.0003 | 0.7239 |
| 0.6728 | 0.3 | 0.05 | 0.0001 | 0.7281 |

Table 4.5: Results of Using Coefficient Mining on DevSet

BM25 Okapi variant, starting with $\alpha = 0$, $\beta = 0$, and ascending by Algorithm 2. We use a small dev set to identify the most reasonable set of $\alpha$, $\beta$, and $\gamma$.

Based on the results presented in Table 4.5, it is apparent that the parameter $\gamma$ displays variability in the range of 0.0001 to 0.0003. Conversely, the parameters $\alpha$ and $\beta$ exhibit instability, oscillating between 0 and 1. To evaluate the test set, we establish constant values for both $\alpha$ and $\beta$ as specified in Table 4.6.

As shown in Fig. 4.10, the optimal F1 score is obtained by setting $\alpha = 0.2$, $\beta = 0.2$, and 0.6 for the lexical, semantic, and semantic enhancement scores, respectively. By employing these coefficients along with a threshold of $\gamma = 0.0001$, we achieve state-of-the-art results of **72.73%** in F1 using Legal-BERTfine-tuned on the 9th attempt of the cross-validation method with a negative-positive ratio of 1:4. The second-best result is obtained with an F1 score of 71.43% by using $\alpha = 0.3$

| $\alpha$ | $\beta$ | $\gamma$ | gamma |
|---|---|---|---|
| 0.1 | 0.1 | 0.0001 | 0.0001 |
| 0.1 | 0.2 | 0.0001 | 0.0001 |
| 0.2 | 0.1 | 0.0001 | 0.0001 |
| 0.2 | 0.2 | 0.0001 | 0.0001 |
| 0.2 | 0.3 | 0.0001 | 0.0003 |
| 0.3 | 0.4 | 0.0001 | 0.0001 |

Table 4.6: Results of Using Coefficient Mining on DevSet

and $\beta = 0.4$ and fine-tuning the 8th attempt of the cross-validation method with a negative-positive ratio of 1:8.

As depicted in Table 4.4, the Task2 of COLIEE 2022 exhibits intense competition, with teams being separated by less than 1%. For instance, the NM team secured the first position with an F1 score of 67.83%, while the JNLP team claimed the second position with an F1 score less than 0.89% behind the NM team. Nonetheless, our study demonstrates that incorporating semantic information from AMR led to an improved correlation with experimental results. Specifically, our proposed method achieved the best results with 78.4% precision, 67.8% recall, and 72.73% F1 score, outperforming the NM team by 4.9% on F1 score. Furthermore, as illustrated in Fig. 11, multiple attempts exceeded 68% on F1 score, further affirming the effectiveness of our proposed method in enhancing the performance of the IR system for the textual entailment task.

To demonstrate the robustness of our proposed framework, we conducted experiments on Task 2 of the COLIEE 2021 test set. We employed the same settings for LEGAL-BERT, BM25 (Okapi), and AMR parser and applied cross-validation and down-sampling approaches for generating training and development sets. The challenge in COLIEE 2021 is the limited number of training data compared to COLIEE 2022. As illustrated in Table 4.1, only 499 positive samples were available, which hindered the efficiency of fine-tuning LEGAL-BERT. Nevertheless, our semantic enhancement framework achieved state-of-the-art results compared to the first-place team in COLIEE 2021. As presented in Table 4.7, our framework achieved a **72.15%** F1 score, outperforming the result of the NM team by **3.03%**.

## 4.4 Discussion

### 4.4.1 Advantage of AMR

In this study, our goal is to develop an IR system that can comprehend the meaning of legal documents. To achieve this, we need a formalism that can distinguish between documents that convey the same meaning and those that modify the underlying meaning. For example, in task 2 of COLIEE 2022, the query paragraphs are relatively short, averaging 43 tokens (as shown in Table 3.1), with the longest being 133 tokens. On the other hand, the candidate paragraphs have an

| Team | File | F1 |
|---|---|---|
| | **Semantic Enhancement Framework (alpha_0.2.beta_0.2.threshold_0.0004)** | **72.15** |
| NM | Run_task2_DebertaT5.txt | 69.12 |
| NM | Run_task2_monoT5.txt | 66.10 |
| NM | Run_task2_Deberta.txt | 63.39 |
| UA | UA_reg_pp.txt | 62.74 |
| JNLP | JNLP.task2.BM25Supporting_Denoising.txt | 61.16 |
| JNLP | JNLP.task2.BM25Supporting_Denoising_Finetune.txt | 60.91 |
| UA | UA_def_pp.txt | 58.75 |
| JNLP | JNLP.task2.NFSP_BM25.txt | 58.68 |
| siat | siatCLS_result-task2.txt | 58.60 |
| DSSIR | run_test_bm25.txt | 58.06 |
| siat | siatFGM_result-task2.txt | 56.70 |
| UA | UA_loose_pp.txt | 56.03 |
| TR | task2_TR.txt | 54.38 |
| DSSIR | run_test_bm25_dpr.txt | 51.61 |
| DSSIR | run_test_dpr.txt | 51.61 |
| MAN01 | [MAN01] task2 run1.txt | 50.69 |
| MAN01 | [MAN01] task2 run0.txt | 25.00 |

Table 4.7: Results of Knowledge Enhancement Framework on task 2 of COLIEE 2021

average length of 138 tokens and can be as long as 3700 tokens. This results in a high overlap rate between the query and candidate paragraphs, leading to similar scores for all candidates, making ranking challenging. Therefore, a better ranking system will require more information to address this issue.

We found that AMR representation contains helpful information to rank candidates. First of all, AMR has clear concept nodes. For example, the verb *hit* has the following concepts:

1. hit-01 - "strike"

2. hit-02 - "reach, encounter"

3. hit-03 - "go to, turn to"

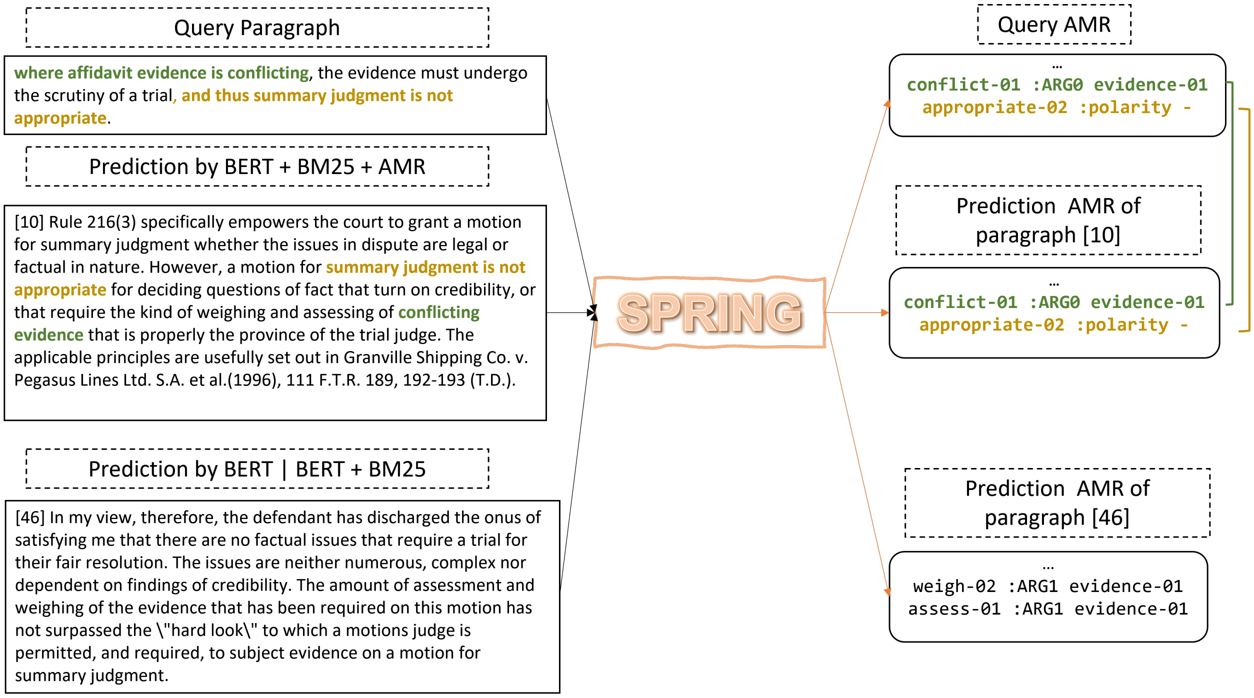
4. hit-06 - "murder"

5. hit-07 - "earn points (on exam/competition)"

There will be different concepts based on different contexts. Secondly, the unambiguous relation between concepts. For the given concept *hit-06*, the corresponding relations are:

- ARG0: assassin, agent agent

- ARG1: ARG1: person assassinated patient


Query Paragraph
Prediction by BERT + BM25 + AMR
Prediction by BERT | BERT + BM25
SPRING
Query AMR
Prediction AMR of paragraph [10]
Prediction AMR of paragraph [46]

where affidavit evidence is conflicting, the evidence must undergo the scrutiny of a trial, and thus summary judgment is not appropriate.

[10] Rule 216(3) specifically empowers the court to grant a motion for summary judgment whether the issues in dispute are legal or factual in nature. However, a motion for summary judgment is not appropriate for deciding questions of fact that turn on credibility, or that require the kind of weighing and assessing of conflicting evidence that is properly the province of the trial judge. The applicable principles are usefully set out in Granville Shipping Co. v. Pegasus Lines Ltd. S.A. et al.(1996), 111 F.T.R. 189, 192-193 (T.D.).

[46] In my view, therefore, the defendant has discharged the onus of satisfying me that there are no factual issues that require a trial for their fair resolution. The issues are neither numerous, complex nor dependent on findings of credibility. The amount of assessment and weighing of the evidence that has been required on this motion has not surpassed the \"hard look\" to which a motions judge is permitted, and required, to subject evidence on a motion for summary judgment.

```
…
conflict-01 :ARG0 evidence-01
appropriate-02 :polarity -
```

```
…
conflict-01 :ARG0 evidence-01
appropriate-02 :polarity -
```

```
…
weigh-02 :ARG1 evidence-01
assess-01 :ARG1 evidence-01
```

Figure 4.11: Example of AMR representation enhances the performance of IR system

and the concept *hit-07*:

- ARG0: athlete, scorer
- ARG1: cognate object, points
- ARG2: test/game

In terms of concepts and relations, *hit-06* usually appears to be in the legal domain, and *hit-07* seems to be in the news domain.

As depicted in Fig. 4.11, this scenario illustrates one of several cases where utilizing AMR representation can aid the IR system in distinguishing the correct entailment paragraph more effectively than relying solely on BERT or a combination of BERT and BM25. Specifically, the query paragraph comprises the phrase *"where affidavit evidence is conflicting"*, and our proposed approach predicts paragraph [10], which contains *"conflicting evidence"*. Since both phrases share the same underlying meaning that *evidence is in conflict*, their AMR must be identical, such as the logical triple: **(conflict-01 :ARG0 evidence-01)**. Moreover, both the query paragraph and paragraph [10] contain *"and thus summary judgment is not appropriate."* and *"summary judgment is not appropriate"*, respectively, indicating the same *is not appropriate* sentiment, which can be represented in their AMR as the same triple **(appropriate-02 :polarity -)**. In the absence of this information, BERT alone or in combination with BM25 selects paragraph [46] as the entailment paragraph, given its several overlapping words with the query paragraph such as: *trial*, *evidence*, *summary*, and *judgment*.

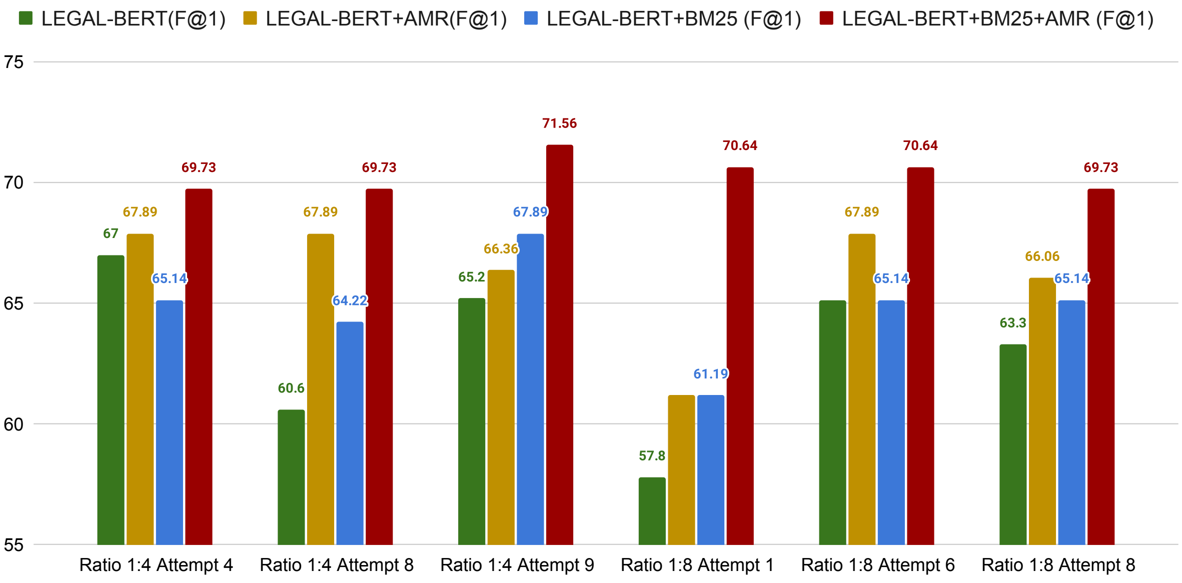


Figure 4.12: F1@1 comparison between Legal-BERTand other Combination Variants

Moreover, the threshold $\gamma$ can potentially increase the system's ability to identify more accurate cases. However, its methodological significance is not apparent. In Fig. 4.12, we aim to demonstrate the significance of our proposed method by comparing the F1@1 scores between Legal-BERTand other knowledge enhancement variants, namely Legal-BERT+ AMR, Legal-BERT+ BM25, and Legal-BERT+ BM25 + AMR.

Our experiments show that combining only AMR or lexical information does not result in a significant increase in performance. Furthermore, integrating only BM25 scores leads to a decrease in performance. For example, in the experiment where BM25 was combined with Legal-BERT(Ratio 1:4 Attempt 4), the F1@1 score decreased by nearly 2% (as shown in Fig. 4.12). Conversely, the best results are consistently achieved when all three sources of information are combined. Noteworthy experiments include:

- Ratio 1:8 Attempt 1 improved Legal-BERTby **12.74%**
- Ratio 1:4 Attempt 8 improved Legal-BERTby **9.13%**
- Ratio 1:8 Attempt 9 improved Legal-BERTby **6.36%**

In contrast to the outcomes presented in Chapter 3 (refer to Figure 4.13), the use of triplets in BM25 Okapi demonstrates superior performance when compared to the usage of only nodes of Abstract Meaning Representation (AMR) by nearly 7% on the F-score metric.

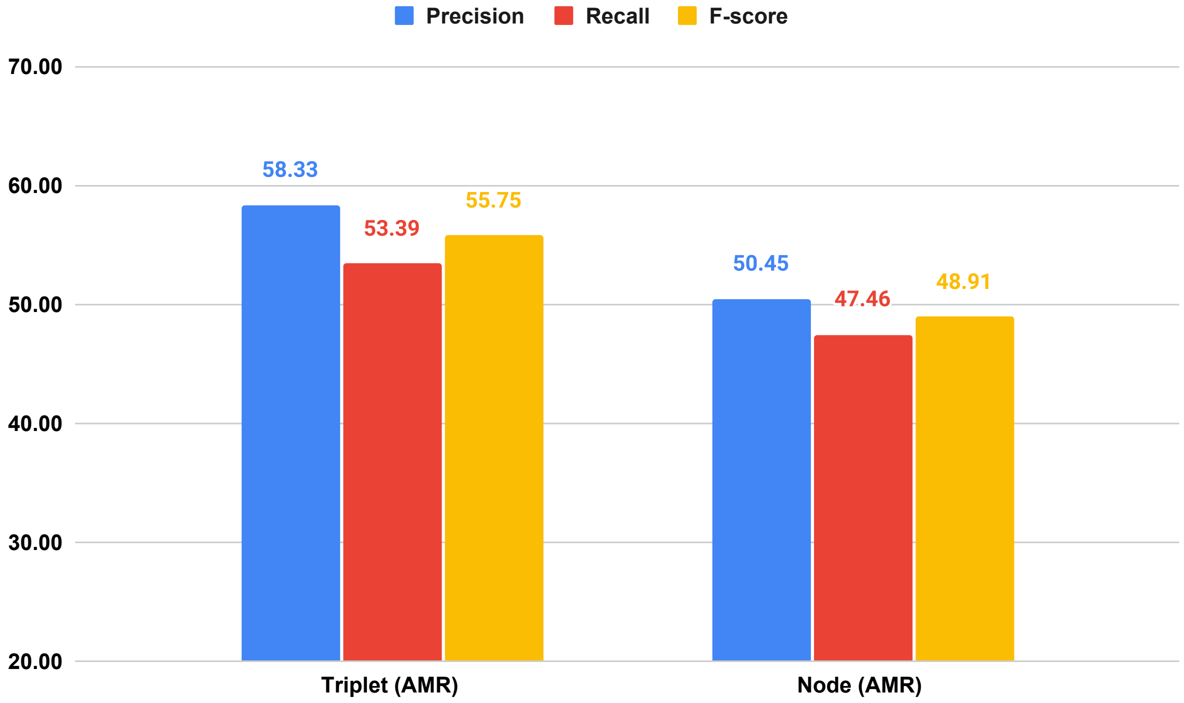


Figure 4.13: BM25 Okapi Performs on Different Information From AMR

### 4.4.2 Error Analysis

In our study, we observed that the use of BM25 can affect the outcome in certain cases, as exemplified by case 545th in the test set (for the dataset, contact COLIEE organizer). In this case, we examined a total of 30 candidates and found that candidate [19] was an entailment paragraph. We conducted experiments using LEGAL-BERT, BM25, and semantic information from AMR to obtain similarity scores. The heatmap in Fig. 4.14 indicates that candidates [19] and [21] have similar semantic and AMR semantic enhancement scores. However, a wrong prediction was made in the combination phase because candidate [21]'s BM25 score was much higher than candidate [19]'s.

We attempted to modify the values of $\alpha$, $\beta$, and threshold $\gamma$ to address this issue. However, such adjustments can affect other correct cases where the candidates are very similar, and even a small modification can lead to incorrect predictions. This is why the highest performance to capture semantic information was achieved by Legal-BERT fine-tuned on attempt 8th with a positive-negative ratio of 1:7 (as shown in Table 4.2 and 4.3), but this model did not show significant improvement in the combination phase.

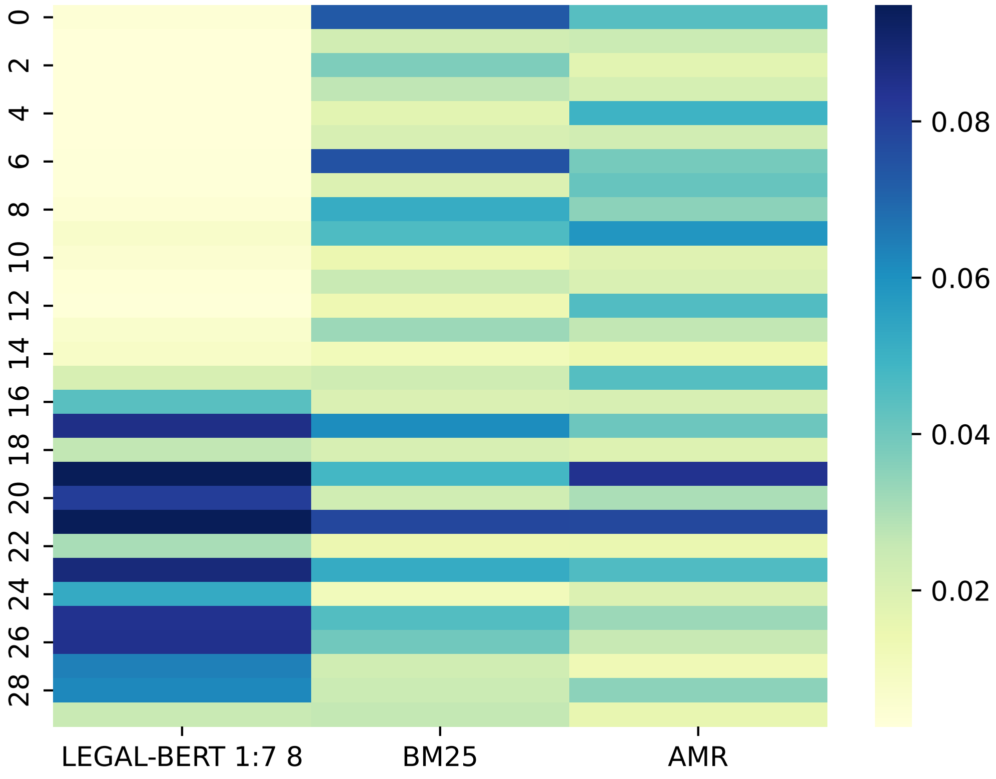


Figure 4.14: Output Score Provided by LEGAL-BERT, BM25, and AMR

### 4.4.3 Limitation of AMR Parser

AMR annotation is a linguistically complex graph, which presents a challenge due to its requirement for annotators with linguistic knowledge. This challenge results in a scarcity of AMR data, with most AMR datasets being for open domains such as the *The Little Prince corpus* (1562 samples) and AMR Corpora released by LDC[2] (59,255 samples). In the Bio domain, the Bio AMR corpus provides 6,952 samples. Notably, there are no AMR datasets available for the legal domain. As shown in Table 4.2 and 4.3, domain adaptation is crucial, as LEGAL-BERT outperforms the BERT base model by nearly 20% on the F1@1 score. Although SPRING is one of the most powerful AMR parsers for open domains, achieving an 84.3% Smatch Score, many errors occur in the parser's prediction on the legal domain without domain adaptation. For example, given the statement:

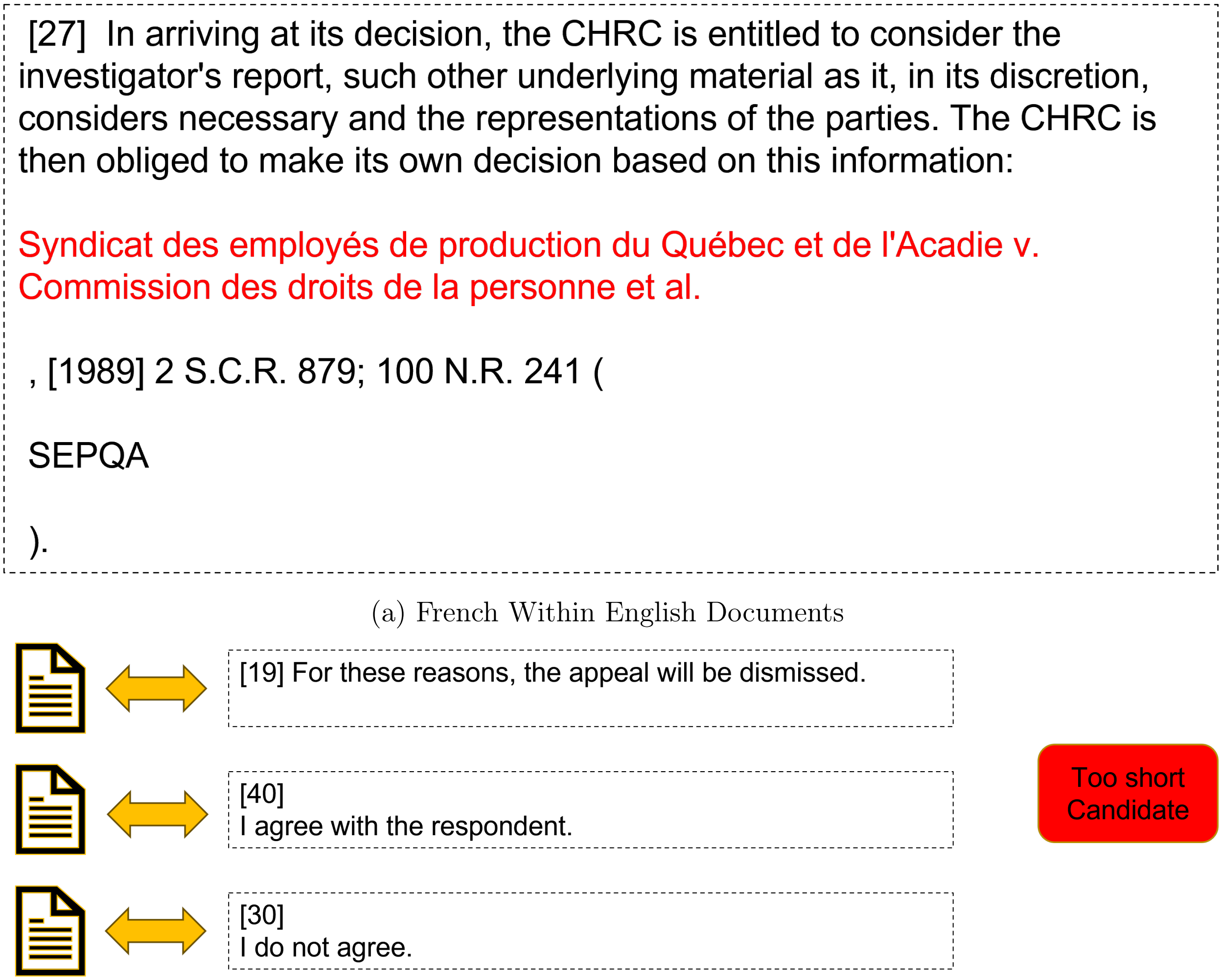
*"That is, the case law under section 55.1(1)(a) of the Canada Pension Plan is quite clear: These provisions are mandatory, and the division of pensionable earnings is to be the rule and not the exception."*

The logical output triples by SPRING are as follows:

clear-06 :ARG0 law<br>
law :mod case<br>
law :location section<br>
section :mod section<br>
section :mod and<br>
**and :op1 1**<br>
**and :op2 et-cetera**<br>
**plan :part section**<br>
plan :wiki "Canada_Pension_Plan"<br>
plan :name name<br>
name :op1 "Canada"<br>
name :op2 "Pension"<br>
name :op3 "Plan"<br>
clear-06 :ARG1 and<br>
and :op1 mandatory<br>
mandatory :domain provision<br>
provision :mod this<br>
and :op2 contrast-01<br>
**contrast-01 :ARG1 equal-01**<br>
**equal-01 :ARG1 divide-02**<br>
divide-02 :ARG1 earn-01<br>
pension-01 :ARG1 earn-01<br>
possible-01 :ARG1 pension-01<br>
**contrast-01 :ARG2 equal-01**<br>
**equal-01 :polarity -**<br>
**equal-01 :ARG1 divide-02**<br>
**equal-01 :ARG2 except-01**<br>
**except-01 :ARG2 divide-02**<br>
clear-06 :degree quite

The logical triples extracted from the given statement reveal several errors, including incorrect assignment of **and :op1 1** instead of **55.1(1)(a)**, and an incorrect value **et-cetera** for **and :op2**. Moreover, the incorrect concept **equal-01** is assigned, which is not relevant in this context. The parser-based transformer model struggles to recognize the sentence structure and is unable to handle long documents. To improve the performance of our proposed framework, it is necessary to adapt the AMR parser to the legal domain and develop alternative approaches.

[2] https://www.ldc.upenn.edu/



[27] In arriving at its decision, the CHRC is entitled to consider the investigator's report, such other underlying material as it, in its discretion, considers necessary and the representations of the parties. The CHRC is then obliged to make its own decision based on this information:

Syndicat des employés de production du Québec et de l'Acadie v. Commission des droits de la personne et al.

, [1989] 2 S.C.R. 879; 100 N.R. 241 (

SEPQA

).

(a) French Within English Documents

[19] For these reasons, the appeal will be dismissed.

[40]
I agree with the respondent.

[30]
I do not agree.

(b) Too Short Candidate Examples

Figure 4.15: Training Data Limitation

### 4.4.4 Training Data Limitation

One limitation of the training data is the presence of French text within the English training corpus 4.15a. This mixed-language content poses challenges for models trained specifically in English, as it introduces noise and potential confusion during the learning process. The inclusion of French text may affect the model's ability to accurately understand and generate English-language content.

Another limitation is the presence of overly short documents in the training data 4.15b. Short documents often lack the contextual depth necessary for comprehensive language modeling. As a result, the model may struggle to capture long-range dependencies and nuanced linguistic patterns, potentially leading to suboptimal performance in tasks that require a thorough understanding of the text.

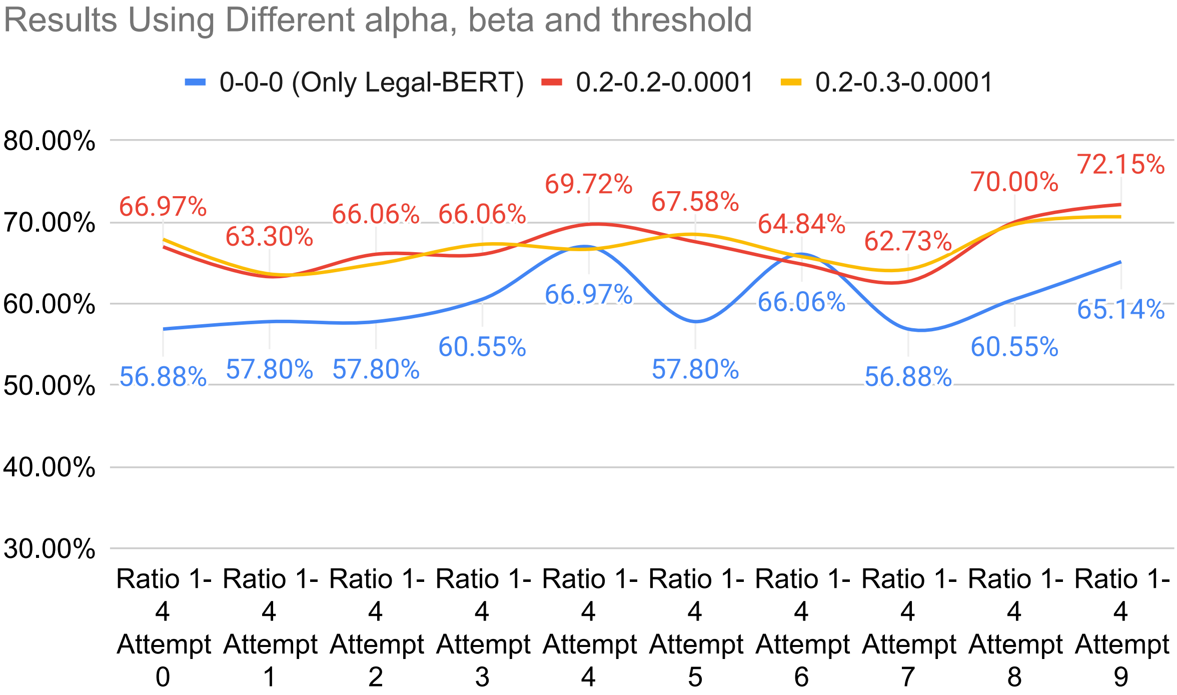
These limitations highlight the need for careful preprocessing and curation of the training data to mitigate the impact of mixed-language content and ensure an appropriate distribution of document lengths. Additionally, expanding the training data with more extensive and diverse text sources could help address these limitations and enhance the model's language understanding capabilities.

The utilization of an information combination framework holds significant promise in various fields where multiple sources of information need to be considered. This framework aims to mitigate the limitations of relying on a single source of information, which can lead to biased or incomplete outcomes. By incorporating multiple information sources, such as data from different sensors, modalities, or perspectives, the system becomes more robust and capable of generating superior results.

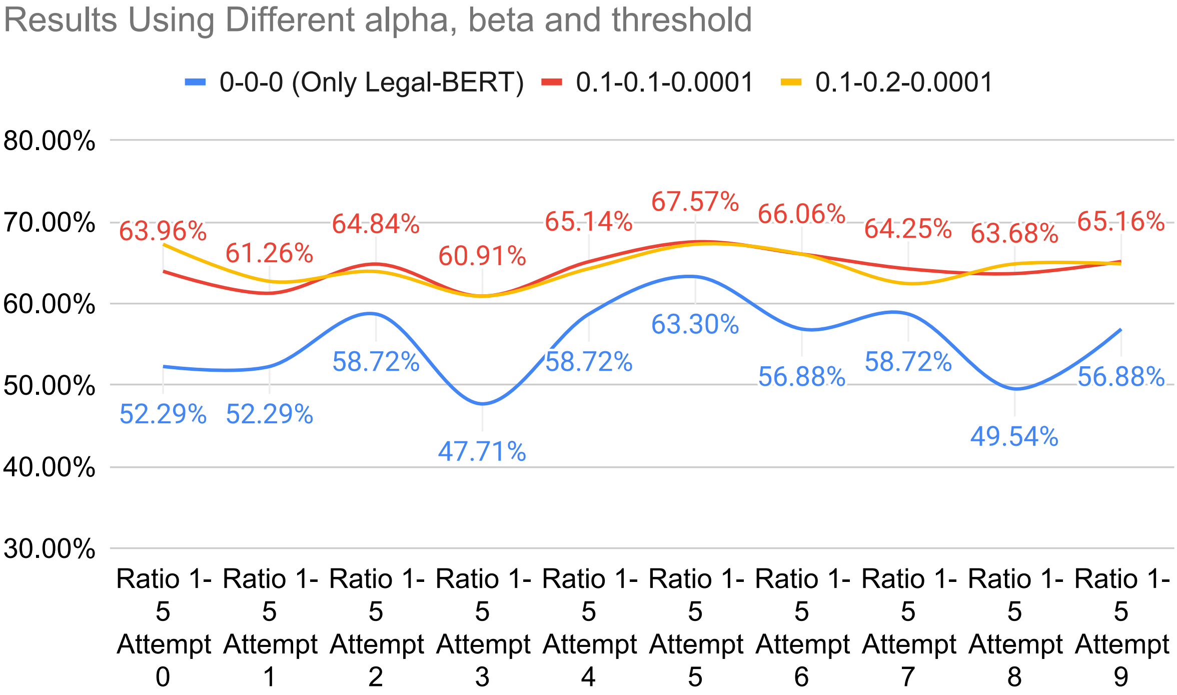
Traditionally, systems heavily relying on a single source of information may suffer from biases and limitations inherent to that specific source. These biases can arise from factors such as data collection methods, measurement errors, or subjective interpretations. Moreover, relying on a single information source may lead to incomplete or distorted conclusions, as it fails to account for the comprehensive and diverse aspects of the problem domain.

In contrast, an information combination framework integrates multiple sources of information, harnessing their collective power to make informed decisions and generate more accurate and reliable results. By incorporating diverse perspectives, the framework enables a more comprehensive understanding of the problem at hand, minimizing the risk of bias and incompleteness.

One of the key advantages of an information combination framework is its ability to capture complementary information from different sources. Each source may provide unique insights or details that are not apparent from the others alone. By combining these diverse pieces of information, the framework can synthesize a more holistic representation of the problem, resulting in improved decision-making and analysis.

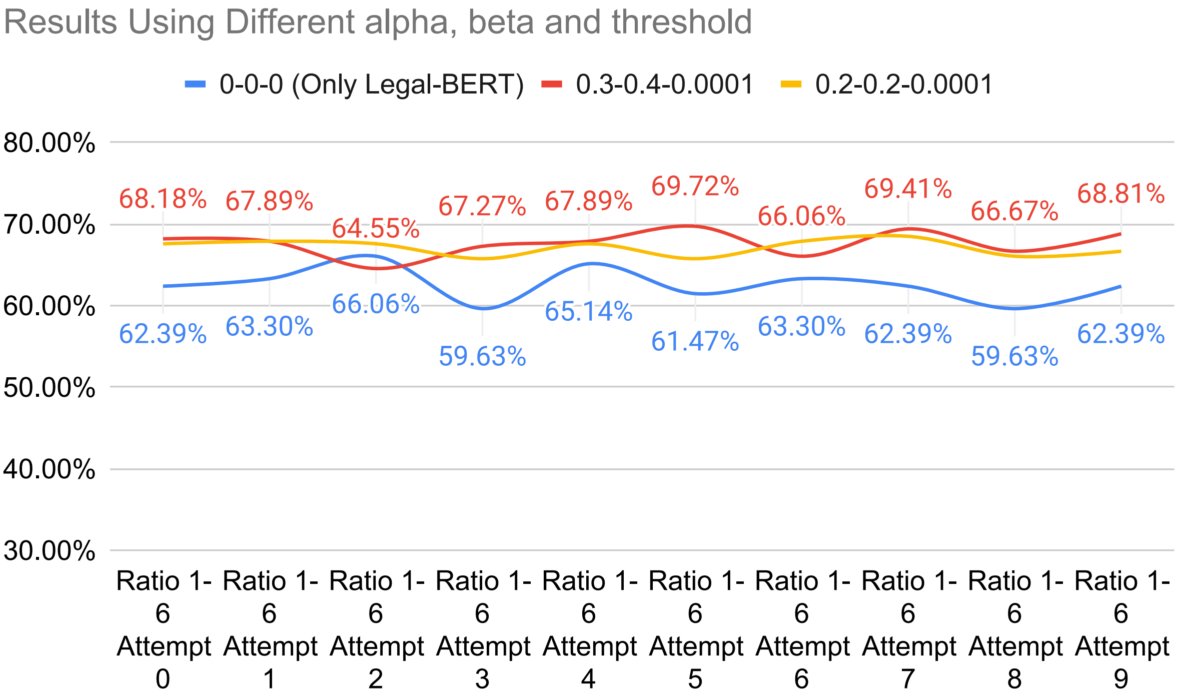


(a) Ratio 1:4

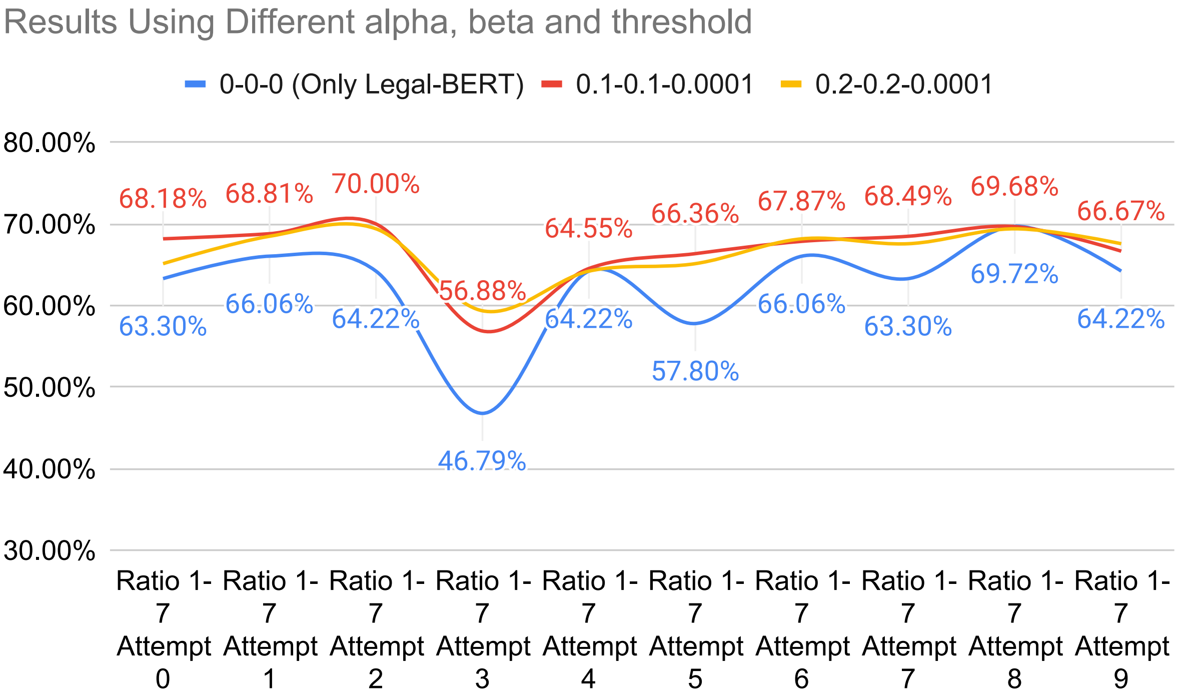


(b) Ratio 1:5

Figure 4.16: Results Comparision Between Information Combination Framework and Legal-BERT Only (1)

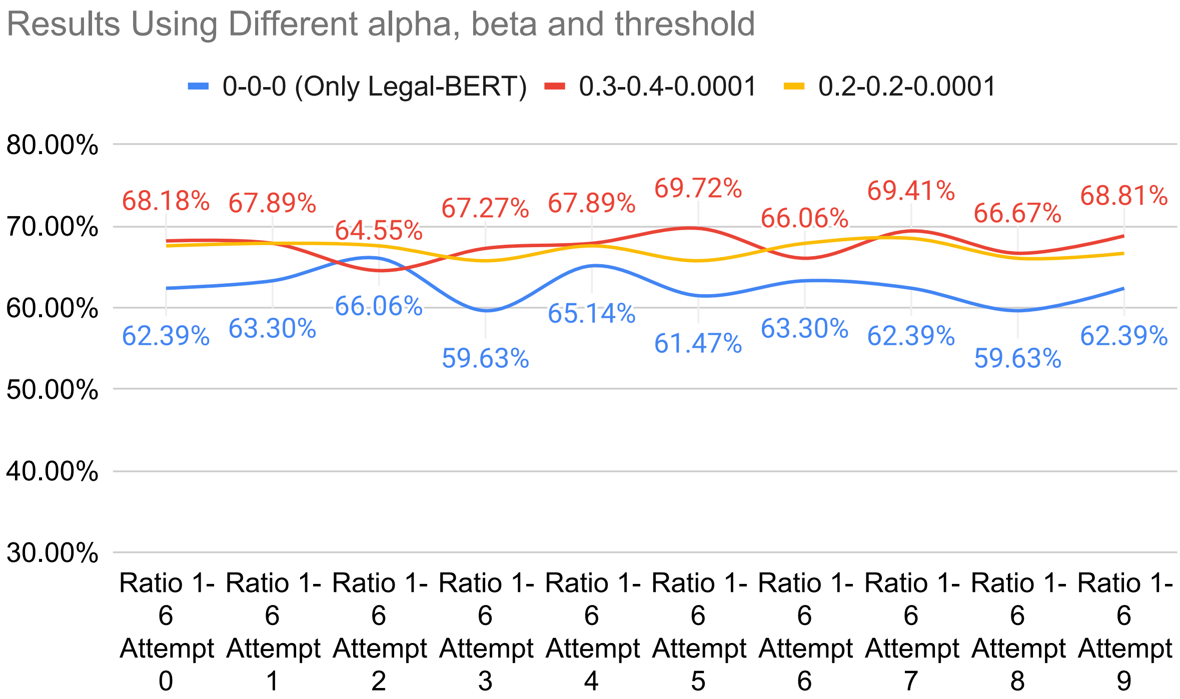


(a) Ratio 1:6



(b) Ratio 1:7

Figure 4.17: Results Comparision Between Information Combination Framework and Legal-BERT Only (2)



(a) Ratio 1:8

Figure 4.18: Results Comparision Between Information Combination Framework and Legal-BERT Only (3)

## 4.5 Summary

Our study focuses on the difficult task of textual entailment. In contrast to previous studies that rely on only the transformer model and different versions of BM25 to identify correct entailment documents, we introduce additional valuable features of an AMR representation to aid the system in better document classification. Furthermore, we employ various advanced techniques to achieve optimal performance in a data-scarce environment. Our experimental results and error analysis demonstrate that our proposed method substantially improves the performance of the textual entailment system and achieves state-of-the-art results on task 2 COLIEE 2020 and COLIEE 2021.

## 4.6 Chapter Conclusion

In this chapter, we have explored an additional utilization of Abstract Meaning Representation (AMR) for information retrieval and have attained state-of-the-art results. However, there exist opportunities to enhance the system's efficacy, including:

1. the refinement of AMR parsing performance due to the presence of erroneous concepts and relationships in the output of the SPRING parser (refer to Section 4.4.3)

2. Additionally, numerous transformer models exhibit exceptional performance in various domains, thus, Legal-BERT may be substituted by more potent models.

In our initial approach, we introduced a curriculum learning strategy to enhance the efficacy of the AMR parser. As depicted in Figure 4.19, we have established various curriculum learning criteria to facilitate the parser's training from simpler to more complex samples.

Nonetheless, the utilization of curriculum learning to enhance the performance of the AMR parser is limited due to the intricate structure of the AMR (see also Figure 4.20). Furthermore, a significant constraint in applying AMR to the legal domain is the absence of AMR datasets for legal language, thereby restricting the feasibility of fine-tuning and impeding a fair assessment of the AMR parser's performance in this domain.

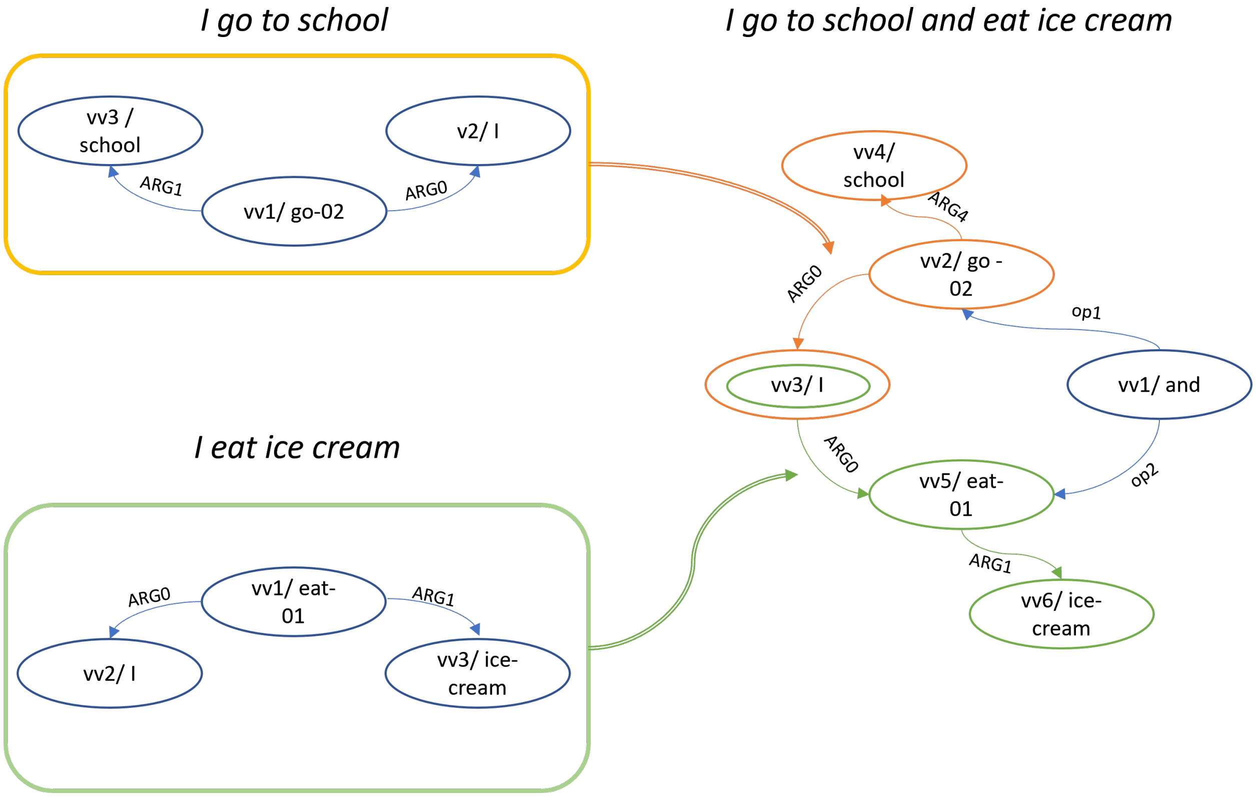


Figure 4.19: Illustrationg of Curriculum Learning for Training AMR parser

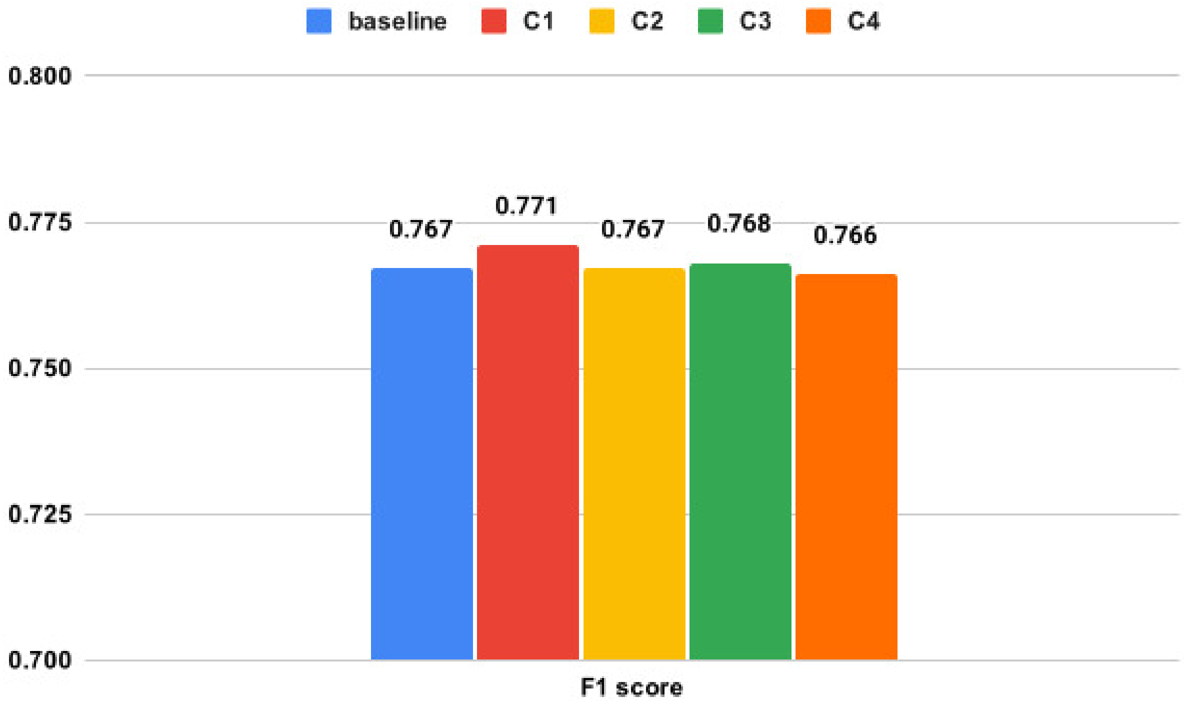


Figure 4.20: Performance Comparison Between Original AMR Parser and AMR Parser Using Curriculum Learning (C Stand For Criteia)

# Chapter 5

# Influence of Large Language Model on the Legal Domain

## 5.1 Introduction

### 5.1.1 Large Language Model

As illustrated in Figure 5.1, the evolution of transformer models has been one of the most significant advances in natural language processing (NLP) in recent years. These models are designed to understand the complex structures and meanings of natural language, enabling machines to perform tasks such as language translation, text summarization, and sentiment analysis.

The first transformer model was introduced in 2017 by Vaswani et al. [61] in their paper "Attention Is All You Need." This model, known as the Transformer, marked a significant departure from previous NLP models, which relied on recurrent neural networks (RNNs) or convolutional neural networks (CNNs) [36]. Instead, the Transformer used self-attention mechanisms to process sequences of words, allowing it to capture long-range dependencies and context information more effectively.

Since then, the field of NLP has seen a rapid proliferation of transformer models, each with its own set of innovations and improvements. One of the most significant developments has been the emergence of encoder-only models, such as BERT [13], RoBERTa [29], ELECTRA [12], DeBERTa [15], and ALBERT [23], etc. These models use only the encoder component of the Transformer, which is responsible for processing input sequences and generating contextualized representations of each word. By training on massive amounts of data, these models have achieved state-of-the-art performance on a wide range of NLP tasks, including language understanding and question answering.

Another important class of transformer models is the encoder-decoder architecture, exemplified by models like BART [26], T5 [43], mT5 [67], T0 [52], and Flan T5 [11], etc. These models use both

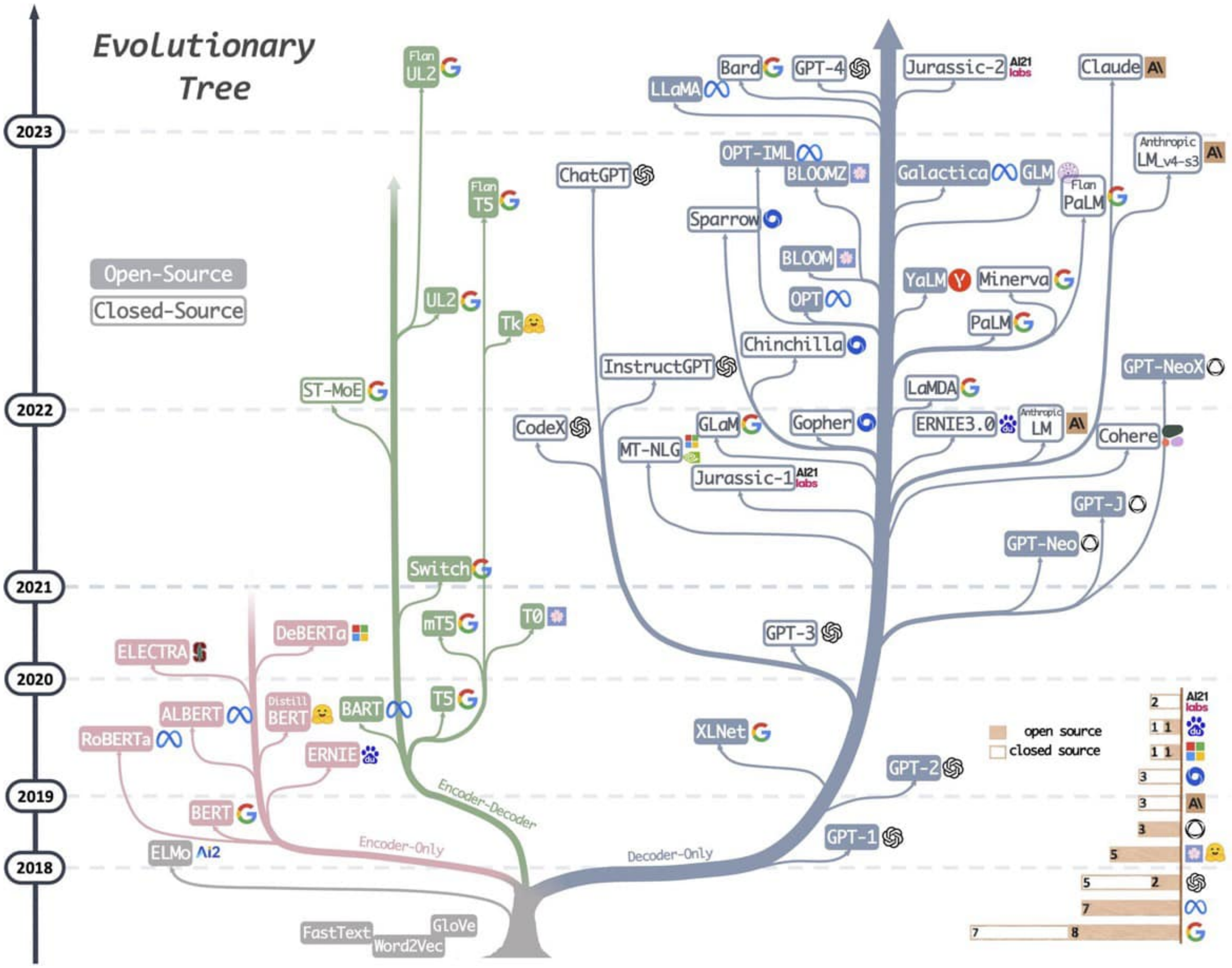


Figure 5.1: Evolutionary Tree of Pre-trained Model [68]

encoder and decoder components to perform tasks such as language translation and text generation. Encoder-decoder models have been particularly successful in multilingual settings, where they can be trained on parallel corpora in multiple languages to enable seamless translation between them.

Finally, there are the decoder-only transformer models, such as XLNet [69], GPT-1 [41], GPT-2 [42], GPT-3 [8], GPT-Neo [6], GPT-J [62], Instruction GPT [37], Chat-GPT, etc. These models use only the decoder component of the Transformer, which is responsible for generating output sequences based on input context. Decoder-only models have been particularly successful in text-generation tasks, such as story writing and poetry generation.

In various domains, including the legal field, pre-trained language models (PLMs) such as BERT [13] have demonstrated their effectiveness in recent years. The common approach for using PLMs is the "pre-training and fine-tuning" learning paradigm, which involves fine-tuning the PLM to adapt it to specific downstream tasks. These downstream tasks may include textual entailment [40, 70], among others.

Large language models (LLMs) have a high number of parameters, ranging from several billion to several hundred billion, such as the T5-xxl [43] with 11B-parameters and GPT3 with 175B-parameters. Researchers have found that LLMs possess unique abilities, such as the ability to follow instructions [51], compared to medium-sized language models. By fine-tuning LLMs on multi-task datasets with natural language instructions, LLMs can perform well on unseen tasks described in the form of instructions [37]. Consequently, various studies have examined LLMs' performance in zero-shot settings of downstream tasks, including textual entailment [64].

### 5.1.2 Application of Large Language Model

According to the Figure 5.2, zero-shot and few-shot learning are two emerging areas in NLP that aim to address the problem of building robust NLP models with limited or no training data. In traditional machine learning, models are trained on large amounts of labeled data to learn how to make accurate predictions. However, in many real-world scenarios, there may not be enough labeled data available to train a high-performing model.

Zero-shot learning [44] [66] is a type of transfer learning that allows a model to make predictions about new or unseen classes without any training data for those classes. In other words, a zero-shot model can generalize to new classes it has never seen before by using its understanding of the relationships between known classes. For example, a zero-shot NLP model can be trained on a corpus of news articles about politics and then make accurate predictions about the sentiment of tweets about sports without ever having seen any sports-related data.

Few-shot learning [59] [57], on the other hand, involves training a model on a small amount of labeled data for new or unseen classes, typically much smaller than the amount required for traditional supervised learning. Few-shot learning aims to leverage the knowledge learned from previously seen classes to quickly adapt to new classes with limited data. For example, a few-shot NLP model can be trained on a small set of labeled data for a new language and then be able to

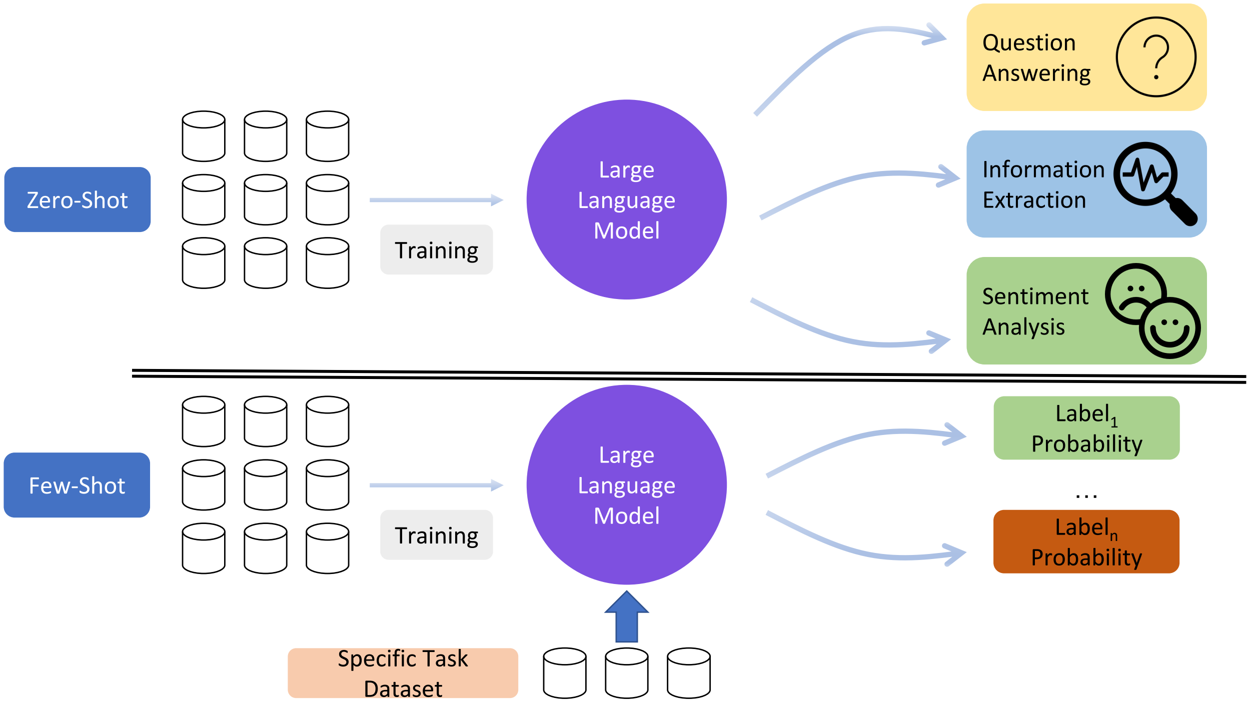


Figure 5.2: Illustration of Zero-Shot and Few-Shot Method

accurately translate new text in that language with very little additional training.

Training large language models is a complex and technically demanding process that requires careful attention to various hyperparameters and settings. To ensure the success of the training process, it is crucial to provide instructions that guide the model's learning and fine-tune its performance.

The aim of this chapter is to evaluate the decision-making ability of LLMs, in particular their performance in the legal domain. This evaluation will involve assessing the effectiveness of few-shot using instruction and zero-shot methods in enabling LLMs to make accurate decisions for the example illustrated in Figure 1.1.

## 5.2 Methodology

### 5.2.1 Prompt (Instruction) Tuning For Yes/No Question Answering

As illustrated in Figure 5.3,to perform prompt tuning, we follow three following steps:

1. Prompt Collecting

2. Zero-Shot Prediction

3. Label Extraction

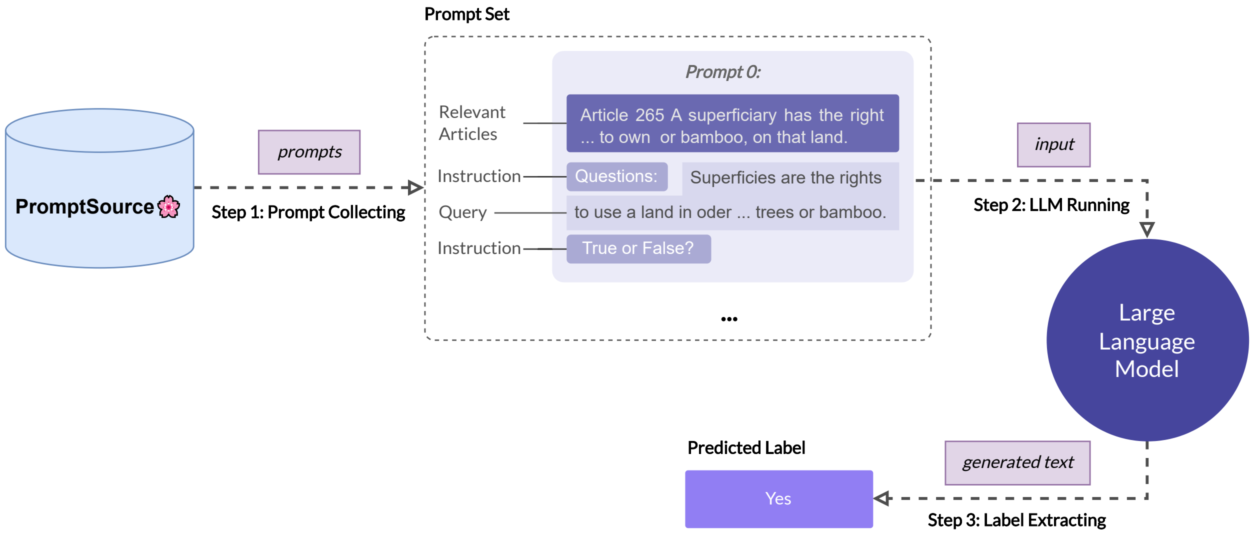


Figure 5.3: High-Level Look of our Prompt Tuning

### Prompt (Instruction) Collecting

In recent years, the use of large language models (LLMs) has become increasingly popular in natural language processing (NLP) research, particularly for text classification and generation tasks. However, a critical aspect of their success in such applications is the selection of an appropriate prompt or instruction. Previous studies have highlighted the importance of prompt engineering in enhancing the performance of LLMs, particularly in zero-shot learning settings, where the models are required to perform a task for which they have not been explicitly trained [21, 46].

To address this issue, we first constructed a collection of prompts to facilitate the legal textual entailment task. To do so, we gathered all the prompts from the General Language Understanding Evaluation (GLUE) tasks available in the PromptSource library [2], which is a collection of NLP benchmarks designed to evaluate the performance of LLMs on a range of tasks. We then converted these prompts to JavaScript Object Notation (JSON) format, as shown in Listing. 5.1, which provides a structured and standardized representation of the prompts.

After processing the prompts, we obtained a set of 56 instructions that could serve as input for the LLMs. These prompts covered a range of semantic and syntactic structures, including negation, disjunction, and conjunction. We expect that this set of prompts will aid in enhancing the performance of LLMs in legal textual entailment tasks, providing a valuable resource for future research in this area. Furthermore, the methodology we used to construct this set of prompts could be adapted to other domains, enabling the generation of domain-specific prompts for a range of NLP applications.

```
[
  {
    "id": 0,
    "label": [
      "True",
      "False"
    ],
    "prompt": "<relevant_articles>
               Question: <query> True or
               False?"
   },
  ...
]
```

Listing 5.1: Prompt Example

### Zero-Shot Prediction

To generate the outputs for each sample in the test set, we employ the prompts gathered in the previous phase to construct the input for the machine learning model. This input is designed to provide the necessary information and context for the model to make informed predictions. Once the input is created, the model processes it and generates the corresponding outputs, which we use to evaluate the performance of the model on the test set. This process involves iteratively testing and refining the model until it achieves the desired level of accuracy and generalization to new data.

### Label Extraction

Recent advances in large language models have demonstrated their remarkable ability to generate high-quality and coherent text. However, one significant challenge that remains is the tendency of these models to generate redundant information. This issue can lead to longer and less concise text, which may negatively impact the readability and coherence of the generated output. Additionally, the presence of redundant information can increase the computational cost and storage requirements of these models, further complicating their practical implementation. Therefore, developing strategies to mitigate the generation of redundant information in large language models is an important research problem that needs to be addressed. For example:

- Question: What is the penalty for driving under the influence of alcohol in California?
- Answer: In California, driving under the influence of alcohol is a serious offense that can result in a range of penalties, including fines, license suspension, mandatory DUI education, and even jail time. The severity of the penalties can depend on several factors, including the driver's blood alcohol concentration (BAC), prior DUI convictions, and whether the DUI resulted in an accident or injury. For a first-time DUI offense in California, the penalties can include a fine of up to $1,000, a six-month license suspension, mandatory attendance at a DUI education program, and up to six months in jail. Repeat offenders and those convicted

> of DUI causing injury or death can face more severe penalties, including longer jail sentences, larger fines, and longer license suspensions. Overall, it is crucial to avoid driving under the influence of alcohol in California, as the penalties can be severe and have a significant impact on your life.

In this example, the answer is excessively long and contains more information than what is required to answer the question. While some of the information provided may be relevant and useful, the answer is too long and may be difficult to understand. A more concise and direct answer that provides only the required information would be more effective and easier to comprehend. In this example, the answer is excessively long and contains more information than what is required to answer the question. While some of the information provided may be relevant and useful, the answer is too long and may be difficult to understand. A more concise and direct answer that provides only the required information would be more effective and easier to comprehend. This demonstrates that we need to extract useful information from the output to evaluate the performance of the LLMs.

### 5.2.2 Prompt Tuning Information Retrieval

We can get the ranking score of a given new decision and candidate paragraphs by following 3 steps:

1. Prompt Collecting.

2. Zero Shot Prediction.

3. Likely Correct Word's Probability Gathering.

For the first and second steps, we use the same approach as Section 5.2.1. However, we face some challenges of using LLMs for legal information retrieval as follows:

- Training Data Bias: Large language models are trained on vast amounts of text data, which can be biased toward certain demographics, languages, or topics. This can affect the model's ability to generalize to new data or make accurate predictions.

- Limited Contextual Understanding: The length limitation of large language models is primarily due to computational constraints. These models require a significant amount of computational power to process each input, and longer inputs require more computational resources. As a result, large language models are typically limited to processing inputs that are several hundred to a few thousand words or characters in length. Given the limited capacity of large language models (LLMs) to process large amounts of input data, it may be impractical to feed all available candidate paragraphs into the models for a given query. As a result, it is necessary to employ a final step **Likely Correct Word's Probability Gathering algorithm** to determine the most relevant candidate paragraphs for the query.

### Likely Correct Word's Probability Gathering

At a high level, LLMs generate the next word by predicting the probability of each possible word given the previous words in the text. It does this by using a technique called "autoregression," where the model predicts the probability distribution of the next word given the previous words.

To generate the next word, LLMs first process the input text using a process called tokenization, where the text is split into individual tokens, such as words or subwords. Each token is then mapped to a corresponding vector representation using an embedding matrix.

Once the input text is tokenized and embedded, GPT uses a transformer architecture to learn the relationships between the tokens in the text. The transformer consists of multiple layers of self-attention and feedforward neural networks, which allow the model to understand the context of the input text and generate the next words based on that context.

To generate the next word, LLMs take the embeddings of the previous words and pass them through the transformer to obtain a sequence of hidden states. It then uses a linear layer and a softmax activation function to calculate the probability distribution of each possible word given the previous words.

Finally, LLMs sample a word from the probability distribution using a process called beam search or greedy search which considers multiple possible next words and selects the one with the highest probability according to the model.

As illustrated in Figure 5.4, when generating text, large language models generate a large number of words that are consistent with the context and prior words in the input. However, in some cases, we may only be interested in the probability of specific words rather than generating a full sequence of words.

In the context of information retrieval, for instance, we may be interested in the probability of a small set of words that are relevant to a particular query or task. In such cases, generating a large amount of text is unnecessary, and we can instead extract the probability distribution of the desired words directly from the language model.

For example, in the case of the two words:

1. Yes

2. True

We can extract their probability distribution directly from the language model without generating a full sequence of text. These probabilities can then be used for ranking the candidates.

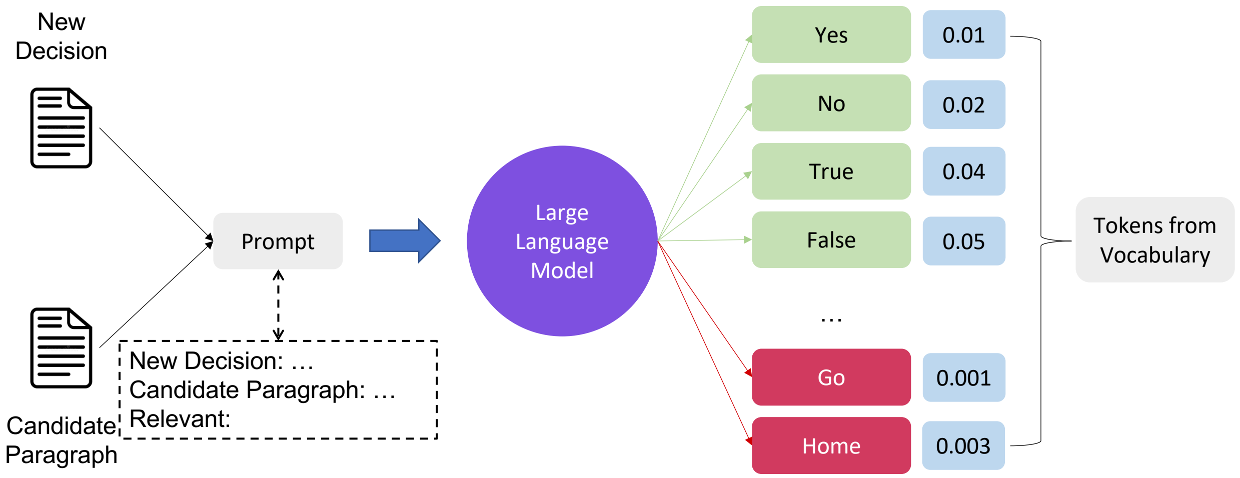


Figure 5.4: Illustration of How to Obtain the Ranking Score for Each Pair Query-Candidate

## 5.3 Experiment Settings

### 5.3.1 Evaluation Task

Similar to Chapters 2 and 4, the current study utilizes COLIEE task 2 as the primary evaluation task. Furthermore, the study employs the COLIEE task 4 to assess the impact of large language models on the legal domain.

#### COLIEE Task 4

The legal textual entailment task's fourth task entails identifying an entailment relationship denoted as Entails(S1, S2, ..., Sn , Q) or Entails(S1, S2, ..., Sn , not Q), where S1 through Sn are relevant articles retrieved through the initial phase. The task requires determining whether the retrieved articles entail either "Q" or "not Q" given the question Q. The resulting output is binary, represented as either "YES"("Q") or "NO"("not Q").

According to Table 5.1, the training set consists of 996 queries, with an equal number of candidate documents per query. The test set comprises 101 queries and candidate documents. In the training set, there are 505 positive samples and 491 negative samples, whereas in the test set, there are 49 positive samples and 52 negative samples. These samples serve as the labeled data for the model's training and testing phases.

The average length of queries in the training set is 48.1 tokens, which is slightly shorter than the average length of queries in the test set, which is 50.1 tokens. The maximum length of queries in the training set is 171 tokens, while in the test set, it is 97 tokens. This suggests that the queries in the training set may be more complex or varied than those in the test set.

The candidate documents in the training set have an average length of 128.2 tokens, which is considerably longer than the average length of candidates in the test set, which is 83.5 tokens. The maximum length of candidates in the training set is also much longer, at 912 tokens, compared to 294 tokens in the test set. This may indicate that the documents in the training set are more detailed or diverse than those in the test set.

| Description | Train | Test |
|---|---|---|
| #Query | 996 | 101 |
| #Candidate | 996 | 101 |
| #Positive Sample | 505 | 49 |
| #Negative Sample | 491 | 52 |
| Query Average Length (tokens[1]) | 48.1 | 50.1 |
| Query max Length (tokens[1]) | 171 | 97 |
| Candidate Average Length (tokens[1]) | 128.2 | 83.5 |
| Candidate Max Length (tokens[1]) | 912 | 294 |

[a] Tokenized by Legal-BERT's Tokenizer

Table 5.1: Task 4 COLIEE 2023 Analysis

### Evaluation Metric

This task will be evaluated based on the measure of accuracy as Equation 5.1 in confirming the yes/no question.

$$Recall = \frac{the\ number\ of\ correct\ queries}{the\ number\ of\ all\ queries} \tag{5.1}$$

## 5.3.2 Model Settings

### COLIEE Task 2

The LLMs model has emerged as a state-of-the-art language model due to its large size and exceptional performance on a wide range of natural language tasks. While smaller models such as BERT and RoBERTa have shown significant promise in legal information retrieval, the increased parameter size of LLMs may offer even greater potential for improving the efficiency and accuracy of legal research.

In the Masked Language Model, we utilized multiple variants of BERT models, including Legal-BERT and BERT Large, as well as two variants of XLM-RoBERTa models, namely the base and large versions. Additionally, two variants of Longformer models, namely the base and large versions, and two variants of DeBERTa models, namely the base and large versions, were also employed. To rank the candidates for all the masked language models, a cross-encoder approach was utilized.

For the causal language models, we utilized three variants of MonoT5 models, including the base, large, and 3B versions, as well as three variants of Flan models, including the large, xl, and XXL versions. In order to rank the candidates for the causal language models, we used the approach detailed in Section 5.2.2.

In the evaluation phase, we employed Task2 of COLIEE2023 as the benchmark dataset, and we used three performance metrics to evaluate the models, including precision (3.3), recall (3.4), and F1-score (3.5).

### COLIEE Task 4

In the context of the COLIEE 2023, the task aimed to identify whether a given statement is true or false based on a given question.

To accomplish this, seven different models were utilized, which are open source and publicly available. These models are:

1. flan-alpaca-xxl
2. flan-ul2
3. flan-t5-xxl
4. mt0-xxl
5. t0pp
6. mt0-xxl-mt
7. bloomz-7b1

The task was carried out by collecting a set of prompts, which were used to prompt the models to predict the training set. The prompts were chosen carefully to ensure they were diverse and covered a broad range of legal topics. These prompts were used sequentially to let the models make predictions on the training set.

The performance of the models was evaluated based on the accuracy of their predictions on the training set. The prompts with the best performance were selected, and the models were trained again with these prompts. Once the training was completed, the selected prompts were used to prompt the models to predict the test set.

The experimental procedure was replicated on Task 4 of COLEE in both the 2020 and 2021 editions, utilizing a restricted set of models, including:

1. flan-alpaca-xxl

2. flan-t5-xl
3. flan-t5-large

## 5.4 Experiment Results

### 5.4.1 COLIEE Task 2

Table 5.2 shows that the MonoT5 model with 3 billion parameters achieves the highest F1 score of 0.74, followed closely by the MonoT5 large and base models with F1 scores of 0.72. The MonoT5 models also exhibit higher precision and recall scores compared to other models.

The XLM-RoBERTa base model has the lowest F1 score of 0.55, indicating poor performance on the evaluation task. The legal-BERT model has an F1 score of 0.7, which is comparable to the MonoT5 models. The Longformer models and DeBERTa models have F1 scores ranging from 0.65 to 0.71, with the larger versions generally outperforming the base versions.

In terms of the number of parameters, the MonoT5 model with 3 billion parameters has the most, followed by the XLM-RoBERTa large model and the Longformer large model. The legal-BERT model has the smallest number of parameters among the models presented.

Overall, the results suggest that the MonoT5 models perform well on the evaluation task compared to other models, with the model with the most parameters achieving the highest F1 score.

#### Poor Performance of Pre-trained Model

Based on the results presented in Table 5.2, all variants of the flan-t5 model exhibited inadequate performance on the Task2 COLIEE 2023 test dataset. The training examples of the flan-t5 model are provided in List 5.2. Notably, the flan-t5 model is designed for classification and question-answering tasks and is not trained for text ranking. This lack of training in text ranking is the probable cause for the poor performance of all flan-t5 variants on the legal retrieval dataset, Task2 COLIEE 2023.

```
[
  {
    "id": 0,
    "text": "prism story of ci prism -lrb- story of ci -rrb- copyright 2010 rachel
   moschell published by rachel",
    "Domain": "Adventure"
   },
    {
    "id": 1,
    "question": "Weng earns $12 an hour for babysitting. Yesterday, she just did 50
   minutes of babysitting. How much did she earn?",
```

| Model | Paras | Precision | Recall | F1 |
|---|---|---|---|---|
| Seq2Seq Token Probability | | | | |
| monot5-3b-few-shot | 3B | **0.81** | **0.675** | **0.74** |
| monot5-large-few-shot | 770M | 0.79 | 0.66 | 0.72 |
| monot5-base-few-shot | 220M | 0.79 | 0.66 | 0.72 |
| monot5-3b-zero-shot | 3B | 0.77 | 0.64 | 0.7 |
| monot5-large-zero-shot | 770M | 0.77 | 0.64 | 0.7 |
| monot5-base-zeroshot | 220M | 0.72 | 0.6 | 0.66 |
| flan-t5-large-zero-shot | 780M | 0.11 | 0.09 | 0.1 |
| flan-t5-xl-zero-shot | 3B | 0.19 | 0.16 | 0.17 |
| flan-t5-xxl-zero-shot | 11B | 0.26 | 0.22 | 0.24 |
| Cross-Encoder | | | | |
| xlm-roberta-base | 270M | 0.6 | 0.5 | 0.55 |
| allenai/longformer-base-4096 | 190M | 0.71 | 0.59 | 0.65 |
| microsoft/deberta-v3-base | 185M | 0.74 | 0.62 | 0.67 |
| nlpaueb/legal-bert-base-uncased | 110M | **0.77** | **0.64** | **0.7** |
| xlm-roberta-large | 550M | 0.73 | 0.61 | 0.66 |
| allenai/longformer-large-4096 | 435M | **0.78** | **0.65** | **0.71** |
| microsoft/deberta-v3-large | 435M | **0.77** | **0.64** | **0.7** |
| bert-large-uncased | 335M | 0.74 | 0.62 | 0.67 |

Table 5.2: Results Comparison on Task 2 COLIEE 2023

| Description | Base Model | Params | Accuracy |
|---|---|---|---|
| Our Results | | | |
| flan-alpaca-xxl-zero-shot | t5-xxl | 11B | **0.80** |
| flan-alpaca-xxl-few-shot | t5-xxl | 11B | **0.79** |
| flan-ul2-zero-shot | ul2 | 20B | **0.75** |
| flan-t5-xxl-zero-shot | t5-xxl | 11B | **0.75** |
| mt0-xxl-zero-shot | mt5-xxl | 13B | **0.71** |
| t0pp-zero-shot | t5-xxl | 11B | 0.67 |
| mt0-xxl-mt-zero-shot | mt5-xxl | 13B | 0.64 |
| bloomz-7b1-zero-shot | bloom-7b1 | 7B | 0.59 |
| Other Teams | | | |
| KIS2 | - | - | 0.6931 |
| UA_V2 | - | - | 0.6634 |
| AMHR01 | - | - | 0.6535 |

Table 5.3: Results Comparison on Task 4 COLIEE 2023

```
    "answer": "Weng earns 12/60 = $<<12/60=0.2>>0.2 per minute. Working 50 minutes,
    she earned 0.2 x 50 = $<<0.2*50=10>>10. #### 10"
    },
  ...
]
```

Listing 5.2: Prompt Example

### 5.4.2 COLIEE Task 4

#### Results

According to Table 5.3, the model *flan-alpaca-xxl-zero-shot* achieved the highest accuracy, with a score of **0.8**. The model *flan-alpaca-xxl-few-shot* had the second-best result, with an accuracy of **0.79**. The third and fourth places were occupied by *flan-ul2-zero-shot* and *flan-t5-xxl-zero-shot*, respectively, both with the same accuracy of 0.7525. Finally, the model *mt0-xxl-zero-shot* achieved an accuracy of 0.7128, followed by "t0pp-zero-shot" with a score of 0.6732, and *mt0-xxl-mt-zero-shot* with a result of 0.6435. The model *bloomz-7b1-zero-shot* had the lowest accuracy among the "Our Results" models, with a score of 0.5940.

| Description | Base_Model | Params | Accuracy |
|---|---|---|---|
| Our Results | | | |
| flan-alpaca-xxl-zero-shot | t5 | 11B | **0.79** |
| flan-t5-xl-zero-shot | t5 | 3B | **0.74** |
| flan-t5-large-zero-shot | t5 | 780M | **0.68** |
| Other Teams | | | |
| KIS | - | - | 0.6789 |
| HUKB | - | - | 0.6697 |
| LLNTU | - | - | 0.6055 |

Table 5.4: Results Comparison on Task 4 COLIEE 2022

In 2023 test set, we conducted a comparison between the performance of our proposed method and that of other teams, including *KIS2*, *UA_ V2*, and *AMHR01.* The accuracy score of the *KIS2* model was found to be the highest among the other teams, with a value of 0.6931. However, our method achieved a much higher accuracy performance, with an increase of approximately 10% in accuracy compared to the *KIS2* model.

We also conducted a comparative analysis of the performance of LLMs in the 2022 test dataset and compared their performance to SOTA teams at the time. We used the flan model with 11B to test it against teams using other models with different parameter sizes.

Our results showed that the flan model with 11B parameters significantly outperformed other teams, achieving an accuracy of nearly 12% in Table 5.4. This result is impressive and underscores the importance of large-scale LLMs for NLP tasks. Additionally, the 3B and 780M parameter models also exhibited better performance compared to other teams, which highlights the importance of parameter tuning in LLMs.

In the 2021 test dataset, we also evaluated the performance of the flan model with 11B parameters against other teams and found that it outperformed all other teams by around 10% to 17%, according to Table 5.5. This indicates that the performance of the flan model is consistent across different test datasets and time periods. Furthermore, the 3B parameter variant also achieved an accuracy of 0.79%, which is a notable result considering its smaller parameter size.

### Influence of Prompt (Instruction) on the Performance of Large Language Models

Large Language Models (LLMs) are a particular class of language models that have gained increasing attention due to their superior performance in NLP tasks. These models are characterized by their massive size, containing billions of parameters that enable them to process complex sequences of text and generate highly accurate predictions.

| Description | Base_Model | Params | Accuracy |
|---|---|---|---|
| Our Results | | | |
| flan-alpaca-xxl-zero-shot | t5 | 11B | **0.80** |
| flan-t5-xl-zero-shot | t5 | 3B | **0.79** |
| flan-t5-large-zero-shot | t5 | 780M | 0.68 |
| Other Teams | | | |
| HUKB | - | - | 0.70 |
| UA | - | - | 0.67 |
| JNLP | - | - | 0.63 |

Table 5.5: Results Comparison on Task 4 COLIEE 2021

One of the critical aspects of training LLMs is the use of instruction or prompts that guide the model during the learning process [54]. These instructions are designed to provide the model with specific information or constraints about the task it is trying to learn and help it to generate more accurate predictions.

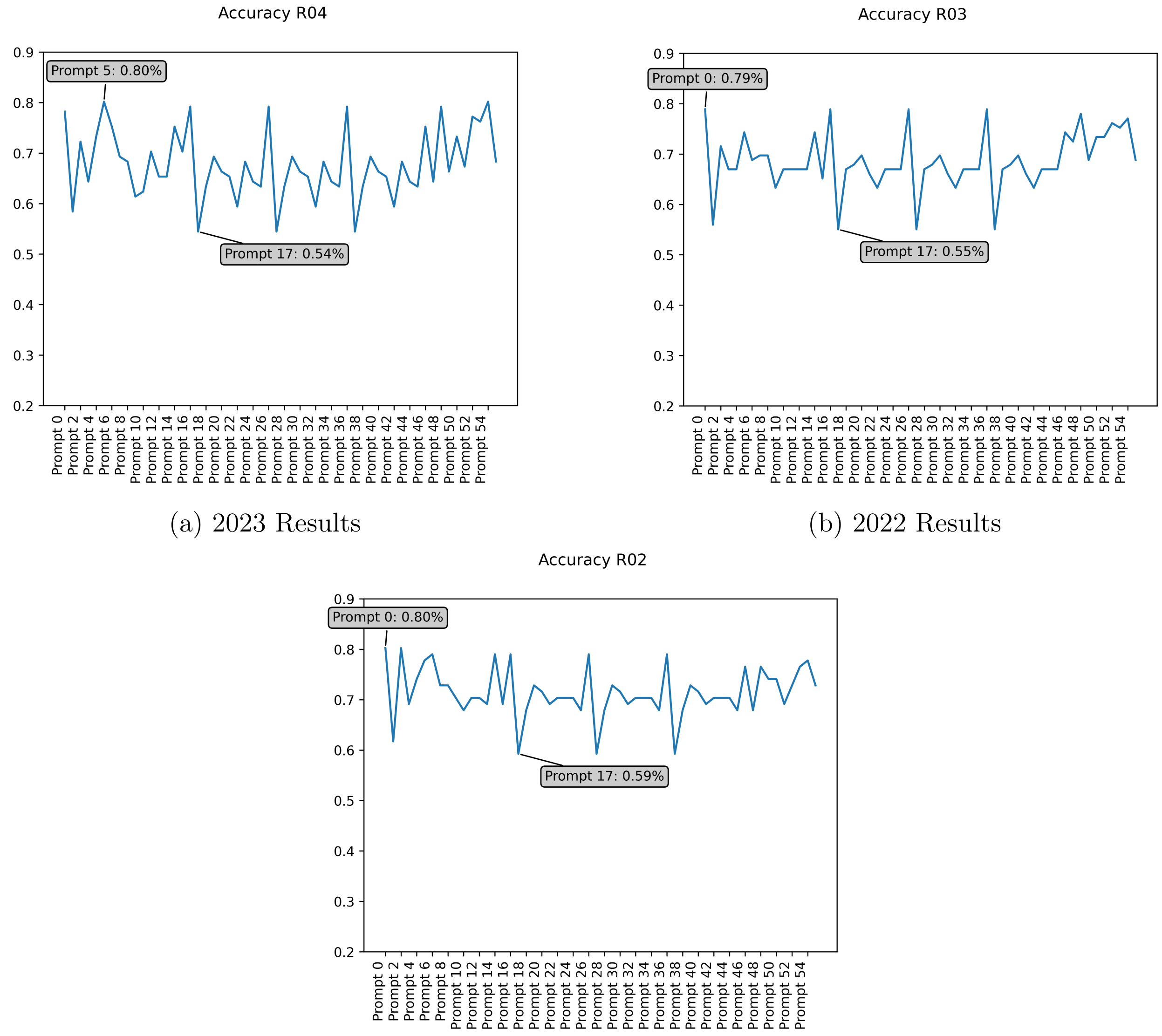
However, changing the instruction given to an LLM can have a significant impact on its performance, as it alters the context and the constraints under which the model is operating. For instance, using different instructions can lead to the model making incorrect predictions or producing outputs that are inconsistent with the task requirements.

Therefore, understanding how changes in the instruction given to an LLM can impact its performance is critical to the successful deployment of these models in real-world applications. In this regard, recent research has focused on investigating the effect of instruction on LLMs, and developing techniques to optimize their performance under different instruction settings.

In this work, we aim to explore the impact of instruction on the performance of large language models in prediction tasks. Specifically, we will investigate how different instructions affect the accuracy and robustness of LLMs, and identify the factors that contribute to these effects. Our findings will provide insights into how instruction can be used to optimize the performance of LLMs, and inform the development of better techniques for training and deploying these models.

The provided results show the performance of a large language model in predicting legal case outcomes based on different prompts. Each prompt represents a unique legal case scenario and the model's accuracy in predicting the outcomes varies accordingly.

In 2023, The provided results 5.5b represent a set of fifty-four prompts and their respective performance scores as evaluated by an automated language model. Each prompt was given a score between 0 and 1, with higher scores indicating better performance. The highest scoring prompt was *PROMPT 5*, with a score of **0.80**, while the lowest scoring prompt was *Prompt 17*, with a score of



(a) 2023 Results

(b) 2022 Results

(c) 2021 Results

Figure 5.5: Performance of Flan-t5-xxl on Task4 COLIEE Using Different Prompts

0.55.

In the 2022 test set, the highest accuracy score of 0.79 is achieved by the model using *Prompt 0*, while the lowest score of 0.55 is achieved by the model using *Prompt 17*, *Prompt 27*, and *Prompt 37*. It is interesting to note that some prompts, such as *Prompt 11*, *Prompt 12*, *Prompt 13*, *Prompt 23*, *Prompt 24*, and *Prompt 25*, resulted in the same accuracy score of 0.67, indicating that the model has the same level of difficulty in predicting outcomes for these prompts.

Furthermore, the results show that some prompts with similar contexts can have significantly different performance scores. For instance, *Prompt 46* and *Prompt 5* both involve a criminal case scenario, but the accuracy score for *Prompt 46* is higher at 0.74 compared to 0.69 for *Prompt 5*.

In 2021 test set, some prompts, such as *Prompt 0* and *Prompt 2*, have relatively high accuracy scores of **0.8**, suggesting that the model performs well on those particular prompts. Other prompts, such as *Prompt 1* and *Prompt 17*, have lower accuracy scores of around 0.6, suggesting that the model may struggle with those prompts. It is also notable that some prompts, such as *Prompt 4* and *Prompt 5*, have accuracy scores that are significantly higher than the others, at around 0.74 and 0.78 respectively. This could indicate that those prompts are particularly easy or well-suited to the model's capabilities.

### Few-shot vs Zero-shot

The given results indicate the performance of a few-shot model and a zero-shot model on a set of prompts. The results you provided show the accuracy of the alpaca-flan-xxl on each prompt, as evaluated by some metric. In the few-shot learning task, the model achieved an accuracy ranging from 0.693 to 0.792 on the prompts, while in the zero-shot learning task, the model achieved an accuracy ranging from 0.545 to 0.802 on the prompts. The few-shot model achieved an average accuracy of 0.75 across 56 prompts, while the zero-shot model achieved an average accuracy of 0.68 across the same prompts.

The results suggest that the few-shot model outperforms the zero-shot model, as it was able to achieve a higher average accuracy on the set of prompts.

## 5.5 Chapter Conclusion

Based on the experimental results, it can be concluded that using large language models is more effective compared to small-size models for natural language processing tasks. The experiments demonstrate that large language models have shown superior performance compared to small models, with more than a 10% improvement on task4 Coliee and a 4% improvement on task2 Coliee. These findings suggest that investing in the development and deployment of large language models could be a promising strategy for achieving better performance in natural language processing tasks, especially in legal textual entailment tasks.

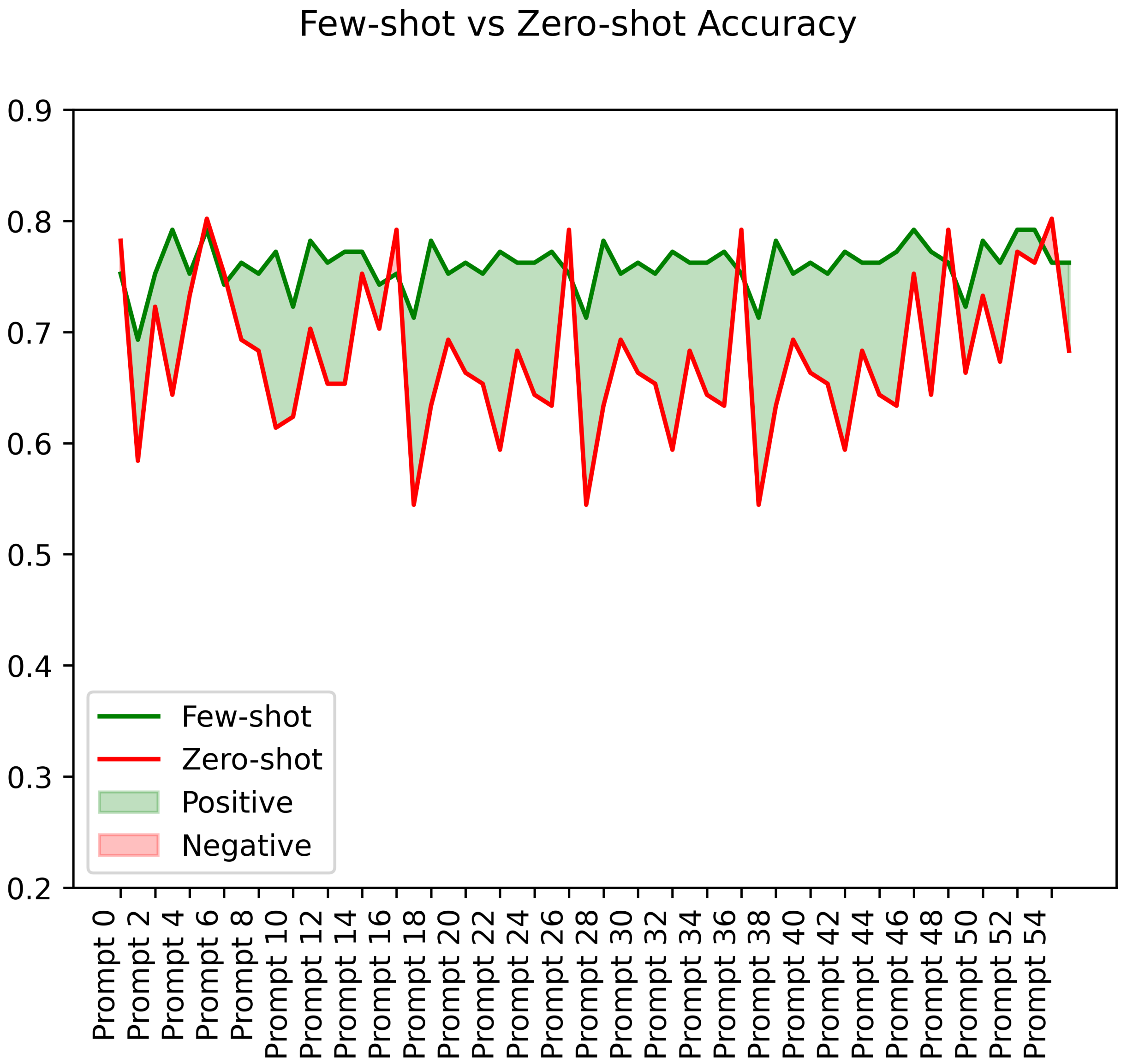


Figure 5.6: Performance Comparison between Zero-shot and Few-shot on task4 COLIEE 2023

# Bibliography

[1] Houda Alberts, Akin Ipek, Roderick Lucas, and Phillip Wozny. “COLIEE 2020: Legal information retrieval and entailment with legal embeddings and boosting”. In: *JSAI International Symposium on Artificial Intelligence*. Springer. 2020, pp. 211–225.

[2] Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. “PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 93–104. DOI: `10.18653/v1/2022.acl-demo.9`. URL: `https://aclanthology.org/2022.acl-demo.9`.

[3] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. “Abstract meaning representation for sembanking”. In: *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*. 2013, pp. 178–186.

[4] Iz Beltagy, Kyle Lo, and Arman Cohan. “SciBERT: A pretrained language model for scientific text”. In: *arXiv preprint arXiv:1903.10676* (2019).

[5] Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. “One SPRING to rule them both: Symmetric AMR semantic parsing and generation without a complex pipeline”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 14. 2021, pp. 12564–12573.

[6] Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. “Gpt-neox-

20b: An open-source autoregressive language model". In: *arXiv preprint arXiv:2204.06745* (2022).

[7] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. "A large annotated corpus for learning natural language inference". In: *arXiv preprint arXiv:1508.05326* (2015).

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

[9] Peter Bühlmann. *Bagging, boosting and ensemble methods.* Springer, 2012.

[10] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. "LEGAL-BERT: The muppets straight out of law school". In: *arXiv preprint arXiv:2010.02559* (2020).

[11] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. "Scaling instruction-finetuned language models". In: *arXiv preprint arXiv:2210.11416* (2022).

[12] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. "Electra: Pre-training text encoders as discriminators rather than generators". In: *arXiv preprint arXiv:2003.10555* (2020).

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[14] Mikel Galar, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.4 (2011), pp. 463–484.

[15] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. "Deberta: Decoding-enhanced bert with disentangled attention". In: *arXiv preprint arXiv:2006.03654* (2020).

[16] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[17] John Hudzina, Kanika Madan, Dhivya Chinnappa, Jinane Harmouche, Hiroko Bretz, Andrew Vold, and Frank Schilder. “Information extraction/entailment of common law and civil code”. In: *JSAI International Symposium on Artificial Intelligence*. Springer. 2020, pp. 254–268.

[18] Fuad Issa, Marco Damonte, Shay B Cohen, Xiaohui Yan, and Yi Chang. “Abstract meaning representation for paraphrase detection”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 2018, pp. 442–452.

[19] Xiaotong Jiang, Zhongqing Wang, and Guodong Zhou. “Semantic Simplification for Sentiment Classification”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 2022, pp. 11022–11032.

[20] MY Kim, J Rabelo, and R Goebel. “Bm25 and transformer-based legal information extraction and entailment”. In: *Proceedings of the COLIEE Workshop in ICAIL*. 2021.

[21] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. “Large language models are zero-shot reasoners”. In: *arXiv preprint arXiv:2205.11916* (2022).

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[23] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. “Albert: A lite bert for self-supervised learning of language representations”. In: *arXiv preprint arXiv:1909.11942* (2019).

[24] Irene Langkilde and Kevin Knight. “Generation that exploits corpus-based statistical knowledge”. In: *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*. 1998.

[25] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. “BioBERT: a pre-trained biomedical language representation model for biomedical text mining”. In: *Bioinformatics* 36.4 (2020), pp. 1234–1240.

[26] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension”. In: *arXiv preprint arXiv:1910.13461* (2019).

[27] J Li, X Zhao, J Liu, J Wen, and M Yang. "Siat@ coliee-2021: Combining statistics recall and semantic ranking for legal case retrieval and entailment". In: *Proceedings of the COLIEE Workshop in ICAIL*. 2021.

[28] Kexin Liao, Logan Lebanoff, and Fei Liu. "Abstract meaning representation for multi-document summarization". In: *arXiv preprint arXiv:1806.05655* (2018).

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

[30] Hans Peter Luhn. "The automatic creation of literature abstracts". In: *IBM Journal of research and development* 2.2 (1958), pp. 159–165.

[31] Yuanhua Lv and ChengXiang Zhai. "When documents are very long, bm25 fails!" In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 2011, pp. 1103–1104.

[32] A Mandal, S Ghosh, K Ghosh, and S Mandal. "Significance of textual representation in legal case retrieval and entailment". In: *COLIEE (2020)* (2020).

[33] Ha-Thanh Nguyen, Phuong Minh Nguyen, Thi-Hai-Yen Vuong, **Quan Minh Bui**, Chau Minh Nguyen, Binh Tran Dang, Vu Tran, Minh Le Nguyen, and Ken Satoh. "Jnlp team: Deep learning approaches for legal processing tasks in coliee 2021". In: 2021.

[34] Ha-Thanh Nguyen, Hai-Yen Thi Vuong, Phuong Minh Nguyen, Binh Tran Dang, **Quan Minh Bui**, Sinh Trong Vu, Chau Minh Nguyen, Vu Tran, Ken Satoh, and Minh Le Nguyen. "Jnlp team: Deep learning for legal processing in coliee 2020". In: 2020.

[35] Shubham Kumar Nigam, Navansh Goel, and Arnab Bhattacharya. "nigam@ COLIEE-22: Legal Case Retrieval and Entailment using Cascading of Lexical and Semantic-based models". In: *JSAI International Symposium on Artificial Intelligence*. Springer. 2022, pp. 96–108.

[36] Keiron O'Shea and Ryan Nash. "An introduction to convolutional neural networks". In: *arXiv preprint arXiv:1511.08458* (2015).

[37] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 27730–27744.

[38] Jeffrey Pennington, Richard Socher, and Christopher D Manning. “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[39] **Quan Minh Bui**, Chau Nguyen, Dinh-Truong Do, Nguyen-Khang Le, Dieu-Hien Nguyen, Thi-Thu-Trang Nguyen, Minh-Phuong Nguyen, and Minh Le Nguyen. “(Lecture Note)JNLP Team: Deep Learning Approaches for Tackling Long and Ambiguous Legal Documents in COLIEE 2022 (Accepted)”. In: (2022), pp. 68–83.

[40] Juliano Rabelo, Mi-Young Kim, and Randy Goebel. “The application of text entailment techniques in coliee 2020”. In: *New Frontiers in Artificial Intelligence: JSAI-isAI 2020 Workshops, JURISIN, LENLS 2020 Workshops, Virtual Event, November 15–17, 2020, Revised Selected Papers 12*. Springer. 2021, pp. 240–253.

[41] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. “Improving language understanding with unsupervised learning”. In: (2018).

[42] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. “Language models are unsupervised multitask learners”. In: *OpenAI blog* 1.8 (2019), p. 9.

[43] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. “Exploring the limits of transfer learning with a unified text-to-text transformer”. In: *The Journal of Machine Learning Research* 21.1 (2020), pp. 5485–5551.

[44] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. “Zero-shot text-to-image generation”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8821–8831.

[45] Sudha Rao, Daniel Marcu, Kevin Knight, and Hal Daumé III. “Biomedical event extraction using abstract meaning representation”. In: *BioNLP 2017*. 2017, pp. 126–135.

[46] Laria Reynolds and Kyle McDonell. “Prompt programming for large language models: Beyond the few-shot paradigm”. In: *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–7.

[47] Stephen Robertson, Hugo Zaragoza, et al. “The probabilistic relevance framework: BM25 and beyond”. In: *Foundations and Trends® in Information Retrieval* 3.4 (2009), pp. 333–389.

[48] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. "Okapi at TREC-3". In: *Nist Special Publication Sp* 109 (1995), p. 109.

[49] Guilherme Moraes Rosa, Ruan Chaves Rodrigues, Roberto de Alencar Lotufo, and Rodrigo Nogueira. "To tune or not to tune? zero-shot models for legal case entailment". In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*. 2021, pp. 295–300.

[50] Gerard Salton and Chung-Shu Yang. "On the specification of term values in automatic indexing". In: *Journal of documentation* (1973).

[51] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. "Multitask Prompted Training Enables Zero-Shot Task Generalization". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=9Vrb9D0WI4.

[52] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. "Multitask prompted training enables zero-shot task generalization". In: *arXiv preprint arXiv:2110.08207* (2021).

[53] Teven Le Scao, Angela Fan, and Christopher Akiki. "Bloom: A 176b-parameter open-access multilingual language model". In: *arXiv preprint arXiv:2211.05100* (2022).

[54] Teven Le Scao and Alexander M Rush. "How many data points is a prompt worth?" In: *arXiv preprint arXiv:2103.08493* (2021).

[55] Frank Schilder, Dhivya Chinnappa, Kanika Madan, Jinane Harmouche, Andrew Vold, Hiroko Bretz, and John Hudzina. "A pentapus grapples with legal reasoning". In: *Proceedings of the COLIEE Workshop in ICAIL*. 2021.

[56] Yunqiu Shao, Bulou Liu, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. "Thuir@ coliee-2020: Leveraging semantic understanding and exact matching for legal case retrieval and entailment". In: *arXiv preprint arXiv:2012.13102* (2020).

[57] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning". In: *Advances in neural information processing systems* 30 (2017).

[58] Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. "Semantic neural machine translation using AMR". In: *Transactions of the Association for Computational Linguistics* 7 (2019), pp. 19–31.

[59] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. "Learning to compare: Relation network for few-shot learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1199–1208.

[60] Andrew Trotman, Charles LA Clarke, Iadh Ounis, Shane Culpepper, Marc-Allen Cartright, and Shlomo Geva. "Open source information retrieval: a report on the SIGIR 2012 workshop". In: *ACM SIGIR Forum*. Vol. 46. 2. ACM New York, NY, USA. 2012, pp. 95–101.

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

[62] Ben Wang and Aran Komatsuzaki. *GPT-J-6B: A 6 billion parameter autoregressive language model*. 2021.

[63] Sabine Wehnert, Viju Sudhi, Shipra Dureja, Libin Kutty, Saijal Shahania, and Ernesto W De Luca. "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*. 2021, pp. 285–294.

[64] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. "Finetuned Language Models are Zero-Shot Learners". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=gEZrGCozdqR.

[65] Hannes Westermann, Jaromir Savelka, and Karim Benyekhlef. "Paragraph similarity scoring and fine-tuned BERT for legal information 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*. Springer. 2021, pp. 269–285.

[66] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. "Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly". In: *IEEE transactions on pattern analysis and machine intelligence* 41.9 (2018), pp. 2251–2265.

[67] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. “mT5: A massively multilingual pre-trained text-to-text transformer”. In: *arXiv preprint arXiv:2010.11934* (2020).

[68] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. “Harnessing the power of llms in practice: A survey on chatgpt and beyond”. In: *arXiv preprint arXiv:2304.13712* (2023).

[69] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. “Xlnet: Generalized autoregressive pretraining for language understanding”. In: *Advances in neural information processing systems* 32 (2019).

[70] Masaharu Yoshioka, Yasuhiro Aoki, and Youta Suzuki. “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*. 2021, pp. 278–284.

# Publications

## Journal Articles

[J1] **Quan Minh Bui** and Minh Le Nguyen. “Semantic Enhancement: Leverage Semantic Information From Abstract Meaning Representation To Improve Legal Textual Entailment Systems(Submitted)”. In: (2023).

[J2] **Quan Minh Bui**, Chau Nguyen, Dinh-Truong Do, Nguyen-Khang Le, Dieu-Hien Nguyen, Thi-Thu-Trang Nguyen, Minh-Phuong Nguyen, and Minh Le Nguyen. “(Lecture Note)JNLP Team: Deep Learning Approaches for Tackling Long and Ambiguous Legal Documents in COLIEE 2022 (Accepted)”. In: (2022), pp. 68–83.

[J3] Yen Thi-Hai Vuong, **Quan Minh Bui**, Ha-Thanh Nguyen, Thi-Thu-Trang Nguyen, Vu Tran, Xuan-Hieu Phan, Ken Satoh, and Le-Minh Nguyen. “SM-BERT-CR: a deep learning approach for case law retrieval with supporting model”. In: *Artificial Intelligence and Law* (2022), pp. 1–28.

# Conference and Workshop Papers

[C1] **Quan Minh Bui**, Dinh-Truong Do, Nguyen-Khang Le, Dieu-Hien Nguyen, Khac-Vu-Hiep Nguyen, Trang Pham Ngoc Anh, and Minh Nguyen Le. “JNLP @COLIEE-2023: Data Argumentation and Large Language Model for Legal Case Retrieval and Entailment (Accepted)”. In: 2023.

[C2] Chau Nguyen, **Quan Minh Bui**, Dinh-Truong Do, Nguyen-Khang Le, Dieu-Hien Nguyen, Thu-Trang Nguyen, Ha-Thanh Nguyen, Vu Tran, Le-Minh Nguyen, Ngoc-Cam Le, et al. “ALQAC 2022: A Summary of the Competition”. In: *2022 14th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE. 2022, pp. 1–5.

[C3] Ha-Thanh Nguyen, Minh-Phuong Nguyen, Thi-Hai-Yen Vuong, **Quan Minh Bui**, Minh-Chau Nguyen, Tran-Binh Dang, Vu Tran, Le-Minh Nguyen, and Ken Satoh. “Transformer-Based Approaches for Legal Text Processing: JNLP Team-COLIEE 2021”. In: vol. 16. 1. Springer, 2022, pp. 135–155.

[C4] Ha-Thanh Nguyen, Phuong Minh Nguyen, Thi-Hai-Yen Vuong, **Quan Minh Bui**, Chau Minh Nguyen, Binh Tran Dang, Vu Tran, Minh Le Nguyen, and Ken Satoh. “Jnlp team: Deep learning approaches for legal processing tasks in coliee 2021”. In: 2021.

[C5] Ha-Thanh Nguyen, Vu Tran, Tran-Binh Dang, **Quan Minh Bui**, Minh-Phuong Nguyen, and Le-Minh Nguyen. “HYDRA–Hyper Dependency Representation Attentions”. In: 2021.

[C6] Ha-Thanh Nguyen, Vu Tran, Phuong Minh Nguyen, Thi-Hai-Yen Vuong, **Quan Minh Bui**, Chau Minh Nguyen, Binh Tran Dang, Minh Le Nguyen, and Ken Satoh. “ParaLaw Nets–Cross-lingual Sentence-level Pretraining for Legal Text Processing”. In: 2021.

[C7] **Quan Minh Bui**, Vu Tran, Ha-Thanh Nguyen, Tran-Binh Dang, and Le-Minh Nguyen. “How Curriculum Learning Performs on AMR Parsing”. In: *2021 13th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE. 2021, pp. 1–6.

[C8] Nguyen Ha Thanh, Dang Tran Binh, **Quan Minh Bui**, and Nguyen Le Minh. “Evaluate and visualize legal embeddings for explanation purpose”. In: *2021 13th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE. 2021, pp. 1–6.

[C9] Nguyen Ha Thanh, **Quan Minh Bui**, Chau Nguyen, Tung Le, Nguyen Minh Phuong, Dang Tran Binh, Vuong Thi Hai Yen, Teeradaj Racharak, Nguyen Le Minh, Tran Duc Vu, et al. “A Summary of the ALQAC 2021 Competition”. In: *2021 13th international conference on knowledge and systems engineering (kse)*. IEEE. 2021, pp. 1–5.

[C10] Ha-Thanh Nguyen, Hai-Yen Thi Vuong, Phuong Minh Nguyen, Binh Tran Dang, **Quan Minh Bui**, Sinh Trong Vu, Chau Minh Nguyen, Vu Tran, Ken Satoh, and Minh Le Nguyen. “Jnlp team: Deep learning for legal processing in coliee 2020”. In: 2020.

# Awards

- COLIEE 2020 First Place: the best performance system on the Legal Case Retrieval and Statute Law Textual Entailment Task.
- COLIEE 2021 First Place: the best performance system on the Statute Law Question Answering Task.
- The paper titled "How Curriculum Learning Performs on AMR Parsing" was honored with the best presentation at KSE 2021 conference.
- The paper titled "Evaluate and Visualize Legal Embeddings for Explanation Purpose" was honored with the Runner-Up Student Paper Award at the KSE 2021 conference.
- COLIEE 2023 Frist Place: the best performance system on the Statute Law Question Answering Task.