Universiteit
Leiden

Master Computer Science

Improve Answer Retrieval in the Legal Domain with Structured Inputs for Cross-Encoder Re-rankers

Name:	Zihui Yang
Student ID:	S3061418
Date:	[19/07/2023]
Specialisation:	Advanced Computing and Systems
1st Supervisor:	Suzan Verberne
2nd Supervisor:	Zhaochun Ren
Additional Supervisor:	Arian Askari

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

This thesis examines the answer retrieval task in the legal community question answering (CQA) system, with the goal of identifying relevant answers from a repository of certified legal practitioners in response to a given question. Our approach uses a two-stage retrieval pipeline that first applies lexical similarity functions such as BM25, followed by a re-ranking of results using a fine-tuned Cross-Encoder.

To improve the performance of the Cross-Encoder, we propose injecting query tags and splitter tokens into the input to create a structured input format as follows:

[CLS] <Query_subject> [S] <Query_description> [D] <Query_tags splitted by “;”> [T] [SEP] <Answer(passage)> [SEP]

where [S], [D] and [T] are new tokens added to the tokenizer, referred to as “Subject”, “Description”, and “Category Tags” of the query, respectively.

Using this Structured Input format, we fine-tune the Cross-Encoder to produce our proposed SI-Reranker. To look into the effectiveness of different parts of the injection, experiments were conducted with different variations of the SI-Reranker, collectively referred to as the “SI-Reranker Family”. Evaluation results indicate that the injection of query category tags yielded the greatest improvement.

Keywords Answer Retrieval; Two-stage Retrieval; Cross-Encoder Re-ranker; Legal Information Retrieval; Community Question Answering

Contents

1 Introduction 1

2 Background 3
2.1 Community Question Answering (CQA) 3
2.2 Legal Information Retrieval (LIR) 4
2.3 Cross-Encoder Rankers 5
2.4 Evaluation Methods 6
2.4.1 Metrics 6
2.4.2 The Text REtrieval Conference (TREC) 7
2.5 Related Work 8

3 Methods 9
3.1 Baseline 1: BM25 9
3.2 Baseline 2: Zero-shot Re-ranker 10
3.3 Fine-tuned Re-ranker 11
3.4 Proposed Model: SI-Reranker 11

4 Experiments 12
4.1 Dataset 12
4.2 Evaluation on Strict or Lenient Relevance Label 14
4.3 First Stage Retrieval with Similarity Functions 14
4.3.1 Workflow 15
4.3.2 Other similarity functions 15
4.4 Re-rank with Cross-Encoders 15
4.4.1 Zero-shot Re-ranker 16
4.4.2 Fine-tuned Re-ranker 16
4.5 Re-rank with SI-Reranker Family 16
4.5.1 SI-Reranker VS. SI-Reranker without Splitter tokens 16
4.5.2 Ablation Study on the Structured Input 17
4.6 Other Experiments 17

5 Results 18
5.1 First Stage Retrieval with Similarity Functions 18
5.2 Re-rank with Cross-Encoders and SI-Reranker Family 18
5.3 Ablation Study on the Structured Input 19
5.4 Other Experiments 20
5.4.1 Question De-duplication 20

5.4.2 Re-rank with Strict or Lenient Relevance Label 22

6 Discussion 24

6.1 Limitations of Term-based Retrieval Models 24

6.2 Effectiveness of Different Injections in SI-Reranker 25

6.3 Impact of Strict or Lenient Relevance Label 25

6.4 Limitations in Data Processing 25

6.5 Specifities of Legal Retrieval 27

6.6 Generalization to Other Domains 27

7 Conclusion 29

7.1 Answers to Research Questions 29

7.2 Future Work 30

Bibliography 32

Chapter 1

Introduction

Community Question Answering (CQA) systems have emerged as an effective way for individuals to seek information and expertise from a community of users. This approach has also brought benefits to the legal domain, where users can raise legal queries and conveniently receive professional suggestions from certified lawyers online instead of meeting an attorney in person. However, CQA systems are not exempt from issues such as question starvation, where users seeking information on a particular topic may experience extended response times or a significant number of questions may remain unanswered for a prolonged period due to the phrasing of duplicated questions in differently ways [1]. To address this problem, information retrieval (IR) models that leverage existing corpora of relevant content can provide users with immediate answers while they await a response from a real lawyer. By identifying and retrieving the questions or answers that closely align with the user's inquiry, we can fulfil their information needs and improve both user satisfaction and response time.

Question retrieval and answer retrieval are two distinct approaches to CQA information retrieval. Question retrieval models, such as the one proposed by Othman *et al.* [2], aim to retrieve historical questions that are semantically equivalent to the queried ones, assuming that the answers to the similar questions should also be relevant to the new query. On the other hand, a typical example of answer retrieval approaches could be the work of Zhou *et al.* [3], who trained a re-ranking model to predict the best answer. The underlying assumption is that if a model can effectively identify the best answer, it can also find the most similar previous best answer in real-world applications.

In recent years, semantics-based text representation methods from Natural Language Processing (NLP) have helped improve semantic search and enable users to obtain more relevant results [4]. Pre-trained transformer models such as BERT [5] have outperformed many term-based models with promising results due to their superior ability in context processing. They can also be used as underlying models for Cross-Encoders (readers are referred to Section 2.3 for more information) to handle *(query, answer)* input pairs for re-ranking [6].

In the standard Cross-Encoder input pair, the question typically requires both a subject and a description to adequately represent its content. Therefore, each query and answer pair in the input should be formatted as Equation 1.1.

$$< Query\_subject > < Query\_description > < Answer\ (passage) > \tag{1.1}$$

However, it is worthwhile to further consider the inclusion of additional details such as category tags and new splitter tokens to better identify the input structure, which may serve to provide more comprehensive information during the retrieval process.

This thesis focuses on the legal CQA answer retrieval task. We propose a model that uses structured information in the input to improve the performance of matching new queries with existing answers in the legal CQA system. Our model follows a two-stage approach: first, it retrieves relevant answers from a repository of certified legal practitioners using lexical similarity functions such as BM25; second, the retrieved results are re-ranked with a fine-tuned Cross-Encoder that uses different structured inputs. This approach has the potential to make a contribution to the field of legal IR and CQA.

Our work is conducted on a dataset from a legal CQA forum categorised under the topic of “bankruptcy”. Within this context, we address three primary research questions (RQs). Firstly, we evaluate first-stage retrieval models under both strict and lenient relevance label setups with the RQ1:

> **RQ1** How effective are term-based retrieval models in legal answer retrieval?

Subsequently, we assess the performance of transformer-based re-rankers with RQ2:

> **RQ2** How effective is a zero-shot Cross-Encoder re-ranker in retrieving legal answers, and to what extent does fine-tuning enhance its effectiveness?

Building upon the findings of the previous two RQs, we introduce a method called **SI-Reranker**, which involves fine-tuning a pre-trained Cross-Encoder with structured input, to address the RQ3:

> **RQ3** To what extent can we further improve the performance of a Cross-Encoder re-ranker by injecting additional information into the input? Which part(s) of the injections are more effective in legal answer retrieval?

This thesis is organized as follows. Chapter 2 provides the necessary preliminary domain knowledge. Chapter 3 presents the methodologies including our proposed SI-Reranker. Chapter 4 describes the experimental settings, including zero-shot and fine-tuned cross-encoder re-rankers, as well as our SI-Reranker family. The experiment results are shown in Chapter 5 and discussed in Chapter 6. Finally, Chapter 7 provides an overall conclusion of this thesis.

Chapter 2

Background

2.1 Community Question Answering (CQA)

CQA platforms, such as Quora and Stack Overflow, have gained widespread popularity due to their effectiveness in providing quick and reliable answers to users' queries. These platforms host large repositories of community-generated questions and answers, making them valuable sources of information for users seeking knowledge in various domains. To improve the retrieval accuracy of relevant answers, numerous novel models with different focuses have been proposed in the field of CQA research.

Question Retrieval Question retrieval can serve as one of the entry points, as for a new query, the existing questions are considered to be the gateway to the possible matching answers in the archive [7]. The primary challenge in question retrieval is the problem of word dissimilarity between similar questions, as the same meaning can be expressed with completely different words [2]. To address this issue, researchers have explored various techniques, such as leveraging word embeddings and semantic similarity measures, to enhance the retrieval of relevant questions. For example, Cao *et al.* [8] give a context to exploiting category information of questions for a better retrieval, while Zhang *et al.* [7] highlights the instinctive heterogeneity of questions and answers, who also takes the answer quality into account to provide more reliable and informative answers to users.

Answer Retrieval Compared to question retrieval, answer retrieval is a more direct way to find the most relevant answer(s) given a user's query, but it also faces its own set of challenges. Unlike questions, which are typically concise and well-defined, answers in CQA systems may contain additional information, opinions, or explanations that are not explicitly present in the corresponding questions, which presents a vocabulary gap [9]. This variability and ambiguity make it trickier for answer retrieval tasks to accurately identify and retrieve relevant answers, in addition to the intrinsic challenges for question retrieval tasks.

Furthermore, the focus of answer retrieval tasks may vary depending on the field of application. For example, Zanibbi *et al.* [10] introduced "math retrieval" to find answers to mathematical questions among posed answers, which leverages both math notation and text to improve the quality of retrieval results. Another example could be in [11], Yang *et* *al.* studied semantic and context features, beyond traditional text matching features, for retrieving answers to non-factoid queries under a learning to rank framework.

Finding Experts Another important aspect of IR in CQA is the identification of expert users who can provide high-quality answers. To reduce response time and improve the likelihood of receiving satisfactory answers, models such as in [12] have been developed to predict and invite expert users to answer questions that match their areas of expertise. By leveraging user expertise, these models aim to enhance the overall quality of answers and user satisfaction. Considering that it is impossible for the average individual to possess comprehensive knowledge in all areas, expertise retrieval is particularly significant in specialized professional fields [13] including the legal domain, as has been proposed in [14].

Generative Models In recent years, generative models have emerged as a popular solution for fluent, natural language question answering, which represents a powerful transition from single-turn to multi-turn QA [15]. This shift means a departure from providing isolated answers to individual queries, towards engaging in a more dynamic and interactive dialogue with the user, where the system can ask for clarification and return answers that fit the situation better. These models, such as ChatGPT and Microsoft's Bing are capable of generating personalized natural language responses based on full sentences or even paragraphs of input. For QA in the open domain, large pre-trained generative models have demonstrated potential without the need for external knowledge, as proposed by Roberts *et al.* [16]. Izacard *et al.* [17] further enhanced their performance by retrieving support passages first and concatenating them with the question as input to a generative model.

The focus of this thesis will be on **answer retrieval** task that can directly find the most relevant answer(s) from the corpus in the **legal domain**. We aim to contribute to the improvement of legal CQA retrieval systems and enhance the retrieval quality for users seeking answers to their questions.

2.2 Legal Information Retrieval (LIR)

This thesis falls within the broad field of Legal Information Retrieval (LIR). As such, it employs the use of Information Retrieval (IR) techniques to accurately identify and retrieve the most relevant information in response to a given query [18]. However, LIR is distinct from IR in other domains due to the specific nature of legal language and the requirement for precision and relevance.

Charactistics For general IR, semantic understanding and sensitivity to context are two of the main characteristics [19]. For the legal domain which uses a technical language that combines common terminologies with domain-specific terms [4], LIR also places emphasis on the document structure, heterogeneity, legal terminology, legal hierarchy, temporal aspects, etc., as noted by Opijnen and Santos in [20] .

Challenges Due to the intrinsic features mentioned above, LIR also faces unique challenges. One of the main challenges is the length and complexity of legal documents, which can make it difficult to accurately assess their relevance to a given query. In addition, different relevance factors, such as document type, recency, law area, usability, annotated, and credibility, may play a role in LIR compared to open-domain search. This is noted in the work of Wiggers *et al.* [21], who conducted a study on domain relevance by legal professionals.

In addition, there are also related tasks that present their own challenges in the legal domain. For example, legal information extraction and entailment [22], and the development of a framework for assessing various approaches to finding relevant prior cases and statutes based on a specific situation [23] are two important tasks that can impact the effectiveness of LIR [4]. Other open issues include knowledge modeling, legal cases' variety, legal interpretation, etc.

Methods Sansone and Sperlí [4] identified three commonly used methodologies used for LIR: natural language processing based techniques, legal ontology based techniques, and deep learning based techniques. In our work, we drew inspiration from the release of Legal-BERT [24], which adapted the BERT model in two ways: one is fine-tuning BERT directly; another is pre-training BERT from scratch on legal data. It is indicated by the legal-BERT authors that further exploration of pre-trained models on additional legal datasets is a promising avenue for future research.

2.3 Cross-Encoder Rankers

A Cross-Encoder is a specialized transformer model that excels in sentence pair scoring and classification tasks [25]. This model operates by concatenating two sequences and sending them into a pre-trained transformer model, such as BERT or RoBERTa, to generate a score for the sentence pair [26]. These pre-trained transformer models have been extensively trained on vast amounts of text data to acquire general language representations. As such, when used as the underlying model for a Cross-Encoder, they offer a robust starting point for the model to execute its task. By harnessing the pre-existing knowledge and capabilities of a pre-trained transformer and fine-tuning on a relevance ranking dataset, the Cross-Encoder is highly effective in answer re-ranking tasks.

As illustrated in Figure 2.1, a Cross-Encoder differs from a Bi-Encoder in its input and output. While a Bi-Encoder produces a sentence embedding for each sentence separately and then computing the similarity, a Cross-Encoder takes a pair of sentences as its input. After processing through the transformer network, the output of the Cross-Encoder is a value between 0 and 1, representing the similarity degree between the two input sentences.

The performance of a Cross-Encoder can be further enhanced by fine-tuning it on specific data. One such example is "`cross-encoder/ms-marco-MiniLM`", which has been trained on the MS Marco Passage Ranking task[1] and is well-suited for information retrieval. This

[1] `https://www.sbert.net/docs/pretrained-models/ce-msmarco.html`

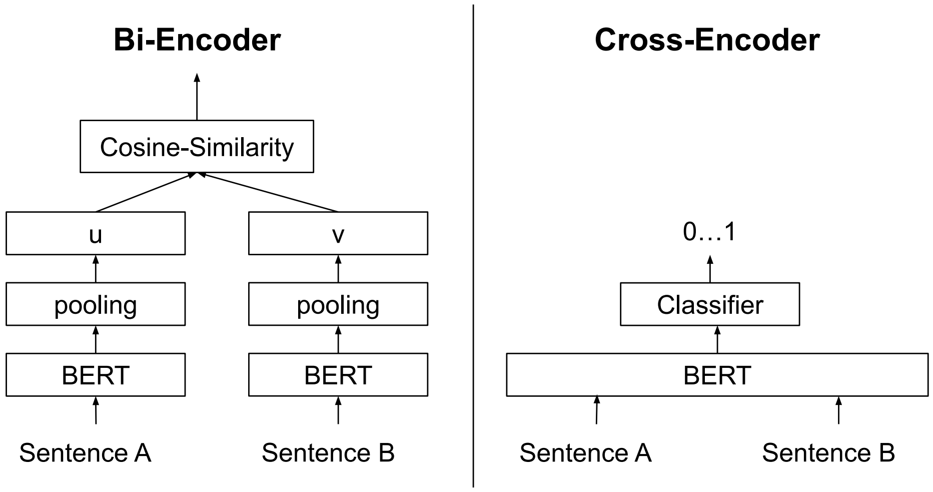

Figure 2.1: Bi-Encoder vs. Cross-Encoder [25]

model employs `MiniLM` [27] as its underlying model, which is a smaller and faster variant of the BERT model that has been trained to achieve comparable performance to BERT on a range of NLP tasks. When given a query, the model can encode the query along with all possible passages and then rank the answer passages in descending order of relevance.

2.4 Evaluation Methods

2.4.1 Metrics

Evaluation metrics provide a quantitative measure of how well a system is able to retrieve relevant information in response to a user's query [28]. In CQA research, some standard evaluation metrics are used for IR [29], among which we apply Precision, Recall, MAP, nDCG and MRR in our study.

Precision and **Recall** are two fundamental evaluation metrics. Precision is the fraction of relevant instances among the retrieved instances, while Recall is the fraction of relevant instances that have been retrieved over the total amount of relevant instances, as computed in the Equation 2.1 and 2.2 based on the retrieval results.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2.1}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2.2}$$

Mean Average Precision (**MAP**), Mean Reciprocal Rank (**MRR)** and Normalized Discounted Cumulative Gain (**nDCG**) are used to evaluate the quality of ranked retrieval results. **MAP** is calculated by taking the mean of the average precision for N queries, as in Equation

2.3. n_i is the number of results for query i, *P(k)* is the precision at rank k, *Rel(k)* is an indicator function that equals 1 if the result at rank k is relevant and 0 otherwise, and K is the number of results to consider.

$$\text{MAP} = \frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{k=1}^{n_i} P(k)\times \text{Rel}(k)}{\min(n_i, K)} \tag{2.3}$$

MRR is calculated by taking the mean of the reciprocal ranks of the first relevant result for N queries, as in Equation 2.4, where $rank_i$ is the rank of the first relevant result for query i.

$$\text{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\text{rank}_i} \tag{2.4}$$

nDCG is calculated by normalizing the discounted cumulative gain (DCG) values for a set of queries, as in Equation 2.5. n is the number of results, rel_i is the relevance score of result at rank i; and IDCG is the Ideal Discounted Cumulative Gain, which is the maximum possible DCG value for a given set of results, where $|REL|$ is the number of relevant results.

$$\text{nDCG} = \frac{\text{DCG}}{\text{IDCG}},\ \text{where DCG} = \sum_{i=1}^{n}\frac{\text{rel}_i}{\log_2(i+1)},\ \text{IDCG} = \sum_{i=1}^{|REL|}\frac{\text{rel}_i}{\log_2(i+1)} \tag{2.5}$$

2.4.2 The Text REtrieval Conference (TREC)

The Text REtrieval Conference (TREC)[2] is a widely recognized evaluation framework in the field of IR, which offers researchers a platform to assess the effectiveness of their retrieval algorithms using large-scale, standardized datasets. As part of its evaluation process, TREC produces standard *qrels* files in the following format which contain relevance judgments for a collection of queries and documents.:

$$< \texttt{question\_id} \quad \texttt{0} \quad \texttt{answser\_id} \quad \texttt{relevance\_label} > \tag{2.6}$$

These judgments serve as the ground truth for evaluating the performance of retrieval systems by comparing their ranked lists of documents against the provided relevance judgments [30].

trec_eval One common tool for evaluating retrieval systems using TREC *qrels* files is *trec_eval*. This tool computes a wide range of evaluation metrics, including Precision, Recall, and MAP, among others. These metrics provide a comprehensive view of a system's retrieval performance and can be used to compare different systems or algorithms [28].

[2] https://trec.nist.gov/data/qrels_eng/

2.5 Related Work

As mentioned in Chapter 1, Zhou *et al.* [3] employed a two-stage approach to identify the best answer for evaluating the performance of the entire ranking system when dealing with an answer retrieval task in CQA. This is not the only instance of combining the first retrieval and re-ranking in CQA research. Chen *et al.* [31] generalized a hybrid model of lexical first-stage retrieval and deep pre-trained model, and its performance indicates that these two models can be complementary to each other when retrieving from different sets.

However, when using a two-stage retrieval pipeline for answer retrieval in legal CQA, which model is suitable for re-ranking? Many studies, such as [32, 33], have introduced a Cross-Encoder for passage retrieval, which is quite suitable for our re-ranking. Furthermore, Askari *et al.* [34] proposed injecting the first-stage retrieval score as extra information into the input of the Cross-Encoder re-ranker, which has been proven to be effective. Izacard *et al.* [17] also used added tokens to concatenate the inputs to the re-ranking BERT model, but so far, such splitter tokens injection or structured inputs have not been applied in a Cross-Encoder re-ranker for legal answer retrieval.

Chapter 3

Methods

In this chapter, we present our method for a question answering retrieval pipeline [35], which is primarily comprised of first-stage retrieval and re-ranking, as illustrated in Figure 3.1. Upon receiving a user's query, the first-stage retrieval returns a set of initial results from the repository. These results are then re-ranked in the second step to improve their relevance and accuracy. We establish a baseline during each stage: for the first-stage retrieval, we have the BM25 baseline introduced in Section 3.1, and for the re-ranker, we use a pre-trained Cross-Encoder as the zero-shot baseline in Section 3.2. Building on this, we develop a fine-tuned re-ranker in Section 3.3 and our proposed method SI-Reranker in Section 3.4.

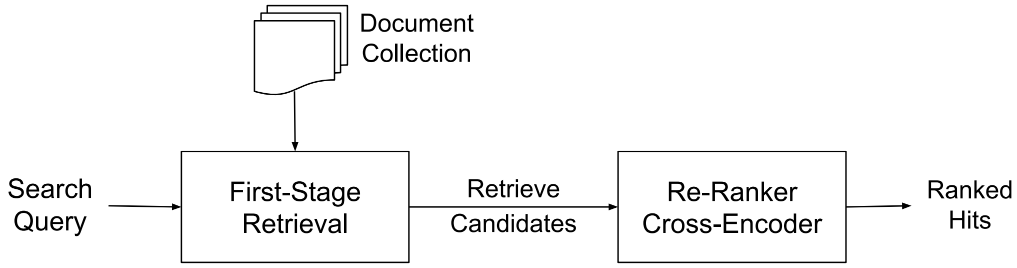

Figure 3.1: Retrieve & Re-Rank Pipeline

3.1 Baseline 1: BM25

The primary objective of the first-stage retrieval is to provide a more precise context for identifying the most relevant answer. To achieve this, term-based models are typically utilized in the first-stage retrieval process. We use BM25, a commonly used ranking function that efficiently retrieves a set of documents from the full document collection based on word overlap [36]. It is a bag-of-words retrieval function that ranks the *top-k* lexical relevance score between queries and documents. Before the emergence of transformer-based models, BM25 has been popular for decades and is still widely used in both academia and industry [31].

BM25 calculates the relevance score between a query and a document based on term frequencies and document lengths. It takes into account term frequency, document length normalization, and document frequency as follows:

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i,D)\cdot(k_1+1)}{f(q_i,D)+k_1\cdot(1-b+b\cdot\frac{|D|}{\text{avgdl}})} \tag{3.1}$$

where D is a document, Q is a query, $f(q_i,D)$ is the frequency of term q_i in document D, $|D|$ is the length of document D, $avgdl$ is the average document length in the collection, and k_1 and b are free parameters [37]. IDF (inverse document frequency) is the weight of the query term q_i computed by Equation 3.2, where N is the total number of documents in the collection, and $n(q_i)$ is the number of documents containing q_i.

$$\text{IDF}(q_i) = \log\frac{N-n(q_i)+0.5}{n(q_i)+0.5} \tag{3.2}$$

Besides BM25, we also employed two additional retrieval models, namely a Probabilistic Language Model with Dirichlet smoothing (LM Dirichlet) and Divergence From Randomness (DFR) for comparative analysis of their effectiveness. LM Dirichlet is a probabilistic retrieval model that estimates the likelihood of generating a query from a document. It incorporates Dirichlet priors to smooth the maximum likelihood estimate of query generation probability using the collection language model [38]. DFR is another probabilistic retrieval model that measures the divergence of the within-document term frequency from its expected value under a random process. It uses this divergence to estimate the informativeness of a term and combines these estimates to rank documents [39]. All of these three similarity functions are built-in in ElasticSearch[1], which makes it convenient to conduct comparison experiments.

3.2 Baseline 2: Zero-shot Re-ranker

After obtaining the first-stage retrieval ranking results, the Cross-Encoder re-ranker is used for re-ranking. We use the pre-trained model "`cross-encoder/ms-marco-MiniLM-L-12-v2`"[2] as the zero-shot re-rankering baseline, which is available on the HuggingFace library [40]. Given a query, the model passes the query and each candidate answer from the first-stage retrieval simultaneously to a transformer network, which will return a relevance score between 0 and 1 for each answer. All relevance scores of the answers concerning this query are then sorted in decreasing order.

When using this zero-shot re-ranker, we employ an **original input format** as follows:

$$\textbf{[CLS]} < \textit{Query\_subject} > < \textit{Query\_description} > \textbf{[SEP]} < \textit{Answer (passage)} > \textbf{[SEP]} \tag{3.3}$$

where [CLS] and [SEP] are special tokens that indicate the start and end of a segment respectively. The query subject and description are concatenated and separated by a space.

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

[2] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2

3.3 Fine-tuned Re-ranker

Fine-tuning is a process of adapting a pre-trained model to a downstream task by adjusting its parameters on a new, labelled dataset [41]. In our case, given that the zero-shot re-ranker was trained on MS MARCO passage ranking dataset, which is not legal-specific, it is necessary to fine-tune the model with legal domain data to improve its performance in this specific field.

This fine-tuning process involves training the zero-shot re-ranker on pairs of queries and answers from a legal CQA dataset, which uses the same input format as Equation 3.3. The fine-tuned re-ranker is expected to produce higher relevance scores for correct answers and lower relevance scores for incorrect answers, and can be evaluated in the same way as the zero-shot re-ranker by passing a query and its candidate answers to the model and sorting the answers based on their relevance scores.

3.4 Proposed Model: SI-Reranker

Considering the feasibility of more injection into the input, based on the fine-tuned re-ranker and the original input format, we propose a new structured input format to fine-tune the zero-shot re-ranker. This format is designed to enrich the retrieval context by introducing query tags and adding new splitter tokens to the tokenizer. These tokens are [S], [D], and [T], which stand for “Subject”, “Description”, and “Category tags” of the query, respectively.

As a result, the input format will be constructed in the following way:

$$\textbf{[CLS]} < \textit{Query\_subject} > \textbf{[S]} < \textit{Query\_description} > \textbf{[D]} < \boldsymbol{\textit{Query\_tags}} \text{ splitted by ``;''} > \textbf{[T]} \; \textbf{[SEP]} < \textit{Answer (passage)} > \textbf{[SEP]} \tag{3.4}$$

The motivation behind these additional injections is to provide the re-ranker with more knowledge about the query, such as identifying an overview of the topic of the legal question with its detailed category tags. By structuring the input in this way, we can also better represent the different components of the query and highlight their importance to the re-ranker.

With this **S**tructured **I**nput, we fine-tune the zero-shot re-ranker and get our **SI-Reranker**. This fine-tuning process involves training the zero-shot re-ranker on pairs of queries and answers, using the structured input format described above. To look into the effectiveness of each injection part, we conducted experiments with different SI-Reranker variations, collectively referred to as the “SI-Reranker Family”, which will be detailed explained in Section 4.5.

Chapter 4

Experiments

In this chapter, We designed our experiments in accordance with the methods proposed in Section 3.

1. First-stage retrieval with term-based models. For each question, the top 1000 answers were retrieved using ElasticSearch with BM25 and other similarity functions.
2. Re-ranking with three different Cross-Encoder models:
   (a) The pre-trained Cross-Encoder re-ranker, referred to as the **Zero-shot Re-ranker** (see Section 3.2), was evaluated.
   (b) The pre-trained model was subsequently fine-tuned with legal data using the original input format to produce the **Fine-tuned Re-ranker** (see Section 3.3).
   (c) The pre-trained model was fine-tuned using our proposed method, resulting in the **SI-Reranker** Family (see Section 3.4).

Evaluations were performed using *trec_eval* tool at each step of the process. We report MAP, Precision@k (k=1,5,10,20), Recall@k (k=10,100), nDCG@k (k=1,5,10,1000) and MRR@10 as evaluation metrics.

4.1 Dataset

Experiments were conducted on a recently published dataset gathered from the AVVO legal online forum. The dataset contains the subset from the forum related to "bankruptcy" for California[1] in the period of January 2008 to July 2021, which contains 9,846 questions and 33,670 answers. All answers in the dataset are from certificated lawyers. Table 4.1 provides a sample of the original data, including the fields "question subject", "question description", "question tags" and "answer passage". The average number of tags per question is 9.07.

[1] `https://www.avvo.com/topics/bankruptcy`

Question	Subject	*Are small business assets exempt under personal chapter 7 bankruptcy petition in CA state*
	Description	*My husband and myself own and are the only employees of a small automobile repair shop and because of a decrease in our business and high personal debt, we are forced to file Chapter 7 bankruptcy. I have been told that since our business is a sole proprietorship that we are one and the same. The business really isn't worth anything, since my husband is the one that does all the repairs and some of his customers have been coming to him for over 20 years. We buy our parts as needed per car from parts houses. I was wondering if the bankruptcy trustee could force us to close our business and if he does, can we just reopen in the same location with a different name. Also how soon could we reopen. I am really concerned because this is our only source of income to pay our secured tax debts.*
	Tags	[*'Sole proprietorship', 'Chapter 7 bankruptcy for businesses', 'Business assets', 'Liquidating business assets', 'Bankruptcy', 'Chapter 7 bankruptcy', 'Bankruptcy petition', 'Bankruptcy trustee', 'Bankruptcy documents', 'Bankruptcy exemptions', 'Debt', 'Bankruptcy liquidation', 'Bankruptcy and debt', 'Renting a house or apartment', 'Business'*]
Answer		*theoretically, a bankruptcy trustee could close your business and try to sell it. your tools are at risk unless they can be claimed as exempt , . there are rather generous exemptions in california. you'd have to work out a deal with your landlord for renting your place of business. if your business has no value without you, i can't imagine a trustee liquidating it. be sure to protect your business assets with exemptions to the fullest extent possible .*

Table 4.1: An Example of the Orignal Data

Question De-duplication We paid particular attention to the removal of duplicate questions during data pre-processing. To identify these duplicates, we employed a computation of lexical similarity. However, performing this computation across the entire dataset can be time-consuming, potentially taking several days. To address this challenge, we decided to use a two-step approach: first, we used K-means clustering [42] to cluster the questions into different groups; then, we computed the lexical similarities based on Levenshtein distance [43] within each cluster.

The original questions, each composed of the question subject and description, were divided into several clusters. Within each cluster, we used *theFuzz* library [44] to compute the lexical similarity score between each question and all other questions. Questions with scores greater than 90% were identified as duplicates. The longest question among all questions sharing the same comparison pair, including the comparison pair itself, was chosen as the representative question. The answers to the duplicated questions were then assigned to the representative question. All questions designated as representative or similar to representative questions would be excluded from subsequent comparisons.

To determine the optimal number of clusters, we conducted a comparison using 200 and

	Train Set		**Test Set**	
	Question Num	Answer Num	Question Num	Answer Num
Before De-duplication	7,877	26,107	1,969	7,563
After De-duplication	7,842		1,960	

Table 4.2: The Number of Question and Answer in Train and Test Set Before and After Question De-duplication.

2000 clusters for the first-stage retrieval, with the average number of questions in each cluster being around 50 and 5, respectively, given that the total number of questions was 9,846. Results in Section 5.4.1 indicate that 200 clusters could be a better option, which was then adopted in the re-ranking experiments.

After removing approximately 50 duplicated questions, the updated dataset was split into an 80% train set and a 20% test set based on question time sorting, using the time of the earliest answer to each question as the question time. The number of questions and answers before and after question de-duplication is presented in Table 4.2.

4.2 Evaluation on Strict or Lenient Relevance Label

For evaluation, we used the *trec_eval* tool as introduced in Section 2.4.2. As required, *qrel* files containing relevance labels were created. In our experiment, we considered two different relevance label settings separately: strict relevance and lenient relevance.

Strict Relevance The answer will be assigned a label "1" if it is selected as the most useful by the question poster or if receives "lawyer agree" votes from more than three certified lawyers, which is different from "helpful" upvotes given by other users. Otherwise, the answer will be labelled as "0".

Lenient Relevance In contrast to strict relevance, the lenient relevance label of an answer is determined based on the number of upvotes it received from lawyers, as follows:

- Label 0: "Not relevant", applicable to answers getting 0 upvote.
- Label 1: "Somehow relevant", applicable to answers getting fewer than 3 upvotes.
- Label 2: "Relevant", applicable to answers getting 3 or higher upvotes.
- Label 3: Applicable to answers marked as "best answer" by the person who asked the question, regardless of the number of upvotes.

4.3 First Stage Retrieval with Similarity Functions

For the first-stage retrieval, we utilized ElasticSearch, an open-source full-text search engine with three different similarity functions including BM25, LM Dirichlet and DFR. We adopted the BM25 parameters with the original built-in settings of $k_1 = 1.2$ and $b = 0.75$. The parameters for both LM Dirichlet and DFR were also set to their respective default values, as stated on the official website under "Similarity Module"[2].

It is important to note that while these term-based models are unsupervised methods and do not require a train/test set split, such a split was still applied to facilitate comparison of the results with those of supervised methods in subsequent re-ranking.

4.3.1 Workflow

In the case of BM25, the following steps were conducted.

1. We produced the *qrels* files by following the standard format, where the relevance label was either **strict(0,1)** or **lenient(0,1,2,3)**. Both cases were treated separately.
2. Two separate indexes were created with ElasticSearch for the train and test sets. Only the query contents consisting of <*Query Subject*> + <*Query Description*> and the answer passages were indexed as text into the repository.
3. Given a question from the train set, all answers were retrieved from the train set using BM25 (the same process was applied to the test set).
4. The top 1000 answers were kept and then evaluated on the train set using *trec_eval* tool and *qrels* files (the same process was applied to the test set).

4.3.2 Other similarity functions

The steps involved in the similarity functions of LM Dirichlet and DFR are identical to those for BM25, except for the similarity function setting.

4.4 Re-rank with Cross-Encoders

Re-ranking with Cross-Encoders comprises training and evaluation. We use the `CrossEncoder` package of the `sentence_transformer` library and the `Pytorch` library. In the `CrossEncoder` class, the loss function for strict relevance labels combines a Sigmoid layer and Binary Cross Entropy Loss, taking advantage of the log-sum-exp trick for numerical stability [45]. For lenient relevance labels, a loss function that computes the cross-entropy loss between the predicted output and true labels is used [46], which is suitable for multi-class classification problems where classes are mutually exclusive.

[2] Default values in ElasticSearch similarity module .https://www.elastic.co/guide/en/elasticsearch/reference/7.6/index-modules-similarity.html

Training Settings For training, we used the example of fine-tuning a Cross-Encoder.[3] The network is trained as a binary classification task, where a label of "1" indicates relevance and a label of "0" indicates irrelevance for a given <query, passage> pair. A positive-to-negative ratio of 1:999 is used, where for each positive sample (label 1), 999 negative samples (label 0) are included in the training setup. The default learning rate of 2×10^{-5} is employed for all Cross-Encoder layers and the batch size is set to 32. The training epoch is set to 1.

4.4.1 Zero-shot Re-ranker

For the zero-shot re-ranker, we directly evaluated the pre-trained model "`cross-encoder/ms-marco-MiniLM-L-12-v2`" on the test set using the original input format in Equation 3.3, which only contains query subject and description, as well as answer passage.

4.4.2 Fine-tuned Re-ranker

When implementing the fine-tuning experiments, we first fine-tuned the pre-trained model "`cross-encoder/ms-marco-MiniLM-L-12-v2`" on the train set, and then evaluated this fine-tuned re-ranker on the test set. Both steps were conducted using the original input configuration as in Equation 3.3 also.

4.5 Re-rank with SI-Reranker Family

In our experiments, we carefully followed the established settings and parameters for the SI-Reranker family as outlined in 4.4, which ensured that our results were consistent and comparable with Zero-shot Re-ranker and Fine-tuned Re-ranker.

4.5.1 SI-Reranker VS. SI-Reranker without Splitter tokens

SI-Reranker In our experiments with the SI-Reranker, we incorporated query tags and splitter tokens as additional components in the input, along with the original question and answer content. We fine-tuned the zero-shot re-ranker on the training set using a fully structured input format, as outlined in Equation 3.4, which included both splitter tokens and query tags. The evaluation was conducted on the test set using the same input structure.

SI-Reranker w/o Splitter Tokens We also explored a variant setting in which we only added query tags to the original input format excluding self-defined splitter tokens. The evaluation input was identical to that used during training. By comparing the performance of these two settings - with and without splitter tokens - we were able to assess the impact of incorporating splitter tokens into our proposed structured input format.

[3] `https://github.com/UKPLab/sentence-transformers/blob/master/examples/training/ms_marco/train_cross-encoder_scratch.py`

4.5.2 Ablation Study on the Structured Input

To look into the influence of different components of the fully structured input, after obtaining the SI-Reranker, we did an ablation study to evaluate this fine-tuned model using various variants of the fully structured input. These **inputs** vary according to the **scenario index**:

0 - Fully structured input, i.e., the input includes query subject, query description, query tags and all added splitter tokens

1 - Based on scenario 0, but without all added splitter tokens

2 - Based on scenario 0, but without token [T]

3 - Based on scenario 0, but without token [T] and query tags

4 - Based on scenario 0, but without token [D]

5 - Based on scenario 0, but without token [D] and query description

6 - Based on scenario 0, but without token [S]

7 - Based on scenario 0, but without token [S] and query subject

By systematically removing different elements from the input and evaluating the model's performance in each scenario, we were able to gain a deeper understanding of how each component contributes to the overall effectiveness of the SI-Reranker.

4.6 Other Experiments

To enhance the comprehensiveness of our analysis, we also implemented the following additional experiments:

1. Evaluating the impact of removing duplicated questions on the performance of the first-stage retrieval. As outlined in 4.1, lexical duplicated questions were identified and removed during the dataset pre-processing. To assess the effectiveness of this step, we conducted a set of control experiments, comparing the retrieval performance under three different settings. In the first setting, all duplicates were retained, while the second and third settings involved removing duplicates with a lexical similarity threshold of 90% and K-Means applied, considering the total number of the questions, 2000 and 200 clusters settings were adopted respectively.

2. Re-ranking using Cross-Encoder with both strict and lenient relevance labels. The pre-trained model we used is under the binary relevance setting, which is not compatible with our lenient relevance label system that ranges from "0" to "3". Therefore, we adapted our setup by assigning the label "1" to any question-answer pair that had some degree of relevance, and the label "0" to those that were of no relevance at all.

Chapter 5

Results

In this chapter, we present all the evaluation results of the experiments designed in Chapter 4, including first-stage retrieval, re-ranking with three different kinds of Cross-Encoders using varied structured inputs, and two additional preliminary experiments.

5.1 First Stage Retrieval with Similarity Functions

After the first stage retrieval using ElasticSearch, the evaluation results for the strict relevance labels are presented in Table 5.1, while those for the lenient relevance labels are in Table 5.2 (MRR metric is not included because it is not applicable when using non-binary relevance labels). As intuitively shown, the BM25 model obtained much better results than LM Dirichlet (LMD) and DFR. Therefore, we decided to select BM25 as our baseline model.

A comparison of the two tables also reveals that the performance of lenient relevance is superior to that of strict relevance on both train and test sets. However, despite the significant difference, our findings indicate that using lenient relevance results in a reduction in the effectiveness of Cross-Encoders. Consequently, we maintain the use of strict relevance for the following Cross-Encoder experiments. A more detailed discussion is provided in Section 5.4.2, where readers are referred to for further information. What's more, the Recall@1000 scores for both the train and test sets are relatively low, which suggests that the answer retrieval task could be pretty challenging for the legal CQA.

5.2 Re-rank with Cross-Encoders and SI-Reranker Family

In the series of Cross-Encoder experiments involving fine-tuning and re-ranking, we first compared those where the evaluation input structures are consistent with the fine-tuning inputs. The main results on the **test set** using strict relevance labels are presented in Table 5.3. The Zero-shot and Fine-tuned model all adopt the initial input format without either query tags or added tokens, while SI-Reranker employs a fully-structured input comprising both tags and splitter tokens. Besides, an SI-Reranker variant model is fine-tuned and evaluated using input query with tags but without splitter tokens.

Data Set	Train Set			Test Set		
Model Name	BM25	LMD	DFR	BM25	LMD	DFR
Recall@10	**.1059**	.0798	.0702	**.1923**	.1526	.1077
Recall@1000	**.3271**	.3169	.2570	**.5416**	.5397	.4507
nDCG@10	**.0748**	.0532	.0491	**.1365**	.0953	.0724
MRR@10	**.2614**	.1845	.1721	**.3450**	.2589	.2131

Table 5.1: Evaluation Results of First Stage Retrieval (Strict Relevance)

Data Set	Train Set			Test Set		
Model Name	BM25	LMD	DFR	BM25	LMD	DFR
Recall@10	**.1339**	.1014	.0856	**.2036**	.1627	.1146
Recall@1000	**.4236**	.4123	.3230	**.6170**	.6236	.5089
nDCG@10	**.1168**	.0825	.0752	**.1899**	.1398	.1055

Table 5.2: Evaluation Results of First Stage Retrieval (Lenient Relevance)

As shown, the performance of the zero-shot model is even inferior to that of BM25. However, after fine-tuning, the performance scores of the other three models more than doubled. Especially, by incorporating additional splitter tokens and query tags, SI-Reranker demonstrates a 10% improvement across most metrics compared to Fine-Tuned re-ranker.

Interestingly, the variant of the SI-Reranker that lacks splitter tokens performs even better. In fact, among all experiments conducted on the SI-Reranker family, this particular variant achieves the highest level of performance. This suggests that the absence of splitter tokens may not have as significant an impact on the model's efficacy as previously thought.

5.3 Ablation Study on the Structured Input

In addition to the cases where the structure of the evaluation inputs aligns with the fine-tuning inputs, we also investigated the ablation study about optimal fine-tuning strategies and input structures to maximize the effectiveness of Cross-Encoder re-rankers.

Table 5.4 presents an additional comparison of the eight different evaluation scenarios for the fully-structured SI-Reranker, which has been fine-tuned using injections of both query tags and splitter tokens.

In general, Evaluation 5 exhibits a stark disparity in comparison to other scenarios, resulting in the lowest level of effectiveness. This can be due to the fact that, in this case, the evaluation input lacks both the token [D] and the query description. These elements are likely to convey the most crucial information about a query, and their absence may significantly impact the model's effectiveness.

On the other hand, Evaluation 4, which lacks the token [D] in its evaluation input, displays superior performance across a majority of metrics, implying that the role played by the

Model Name	Zero-Shot	Fine-Tuned	SI-Reranker (Proposed Method)	SI-Reranker w/o splitter tokens
MAP	.1089	.2364	.2695	**.2732**†
P@1	.0807	.1819	.2090	**.2156**†
P@5	.0362	.0798	.0895	**.0917**†
P@10	.0214	.0470	.0524	**.0528**†
P@20	.0135	.0260	.0288	**.0289**†
Recall@10	.1733	.3813	.4277	**.4309**†
Recall@100	.3413	.4958	.5243	**.5237**†
nDCG@1	.0807	.1819	.2090	**.2156**†
nDCG@5	.1142	.2546	.2906	**.2957**†
nDCG@10	.1231	.2744	.3114	**.3150**†
nDCG@1000	.1847	.3059	.3360	**.3390**†
MRR@10	.3074	.6296	.7038	**.7067**†

Table 5.3: Evaluation Results of Re-rankers on the Test Set. † denotes a statistically significant improvement of **SI-Reranker w/o splitter tokens** over the **Fine-Tuned** Cross-Encoder.

Index	Evaluation Input Structure	Recall@10	nDC@10	MRR@10
0	Base (Fully Structured)	.4277	.3114	**.7038**
1	Base removing Added Splitter Tokens	.4257	.3104	.7031
2	Base removing [T]	.4284	.3116	.7010
3	Base removing [T] & All Query Tags	.4054	.2924	.6572
4	Base removing [D]	**.4299**	**.3139**	.7017
5	Base removing [D] & Query Description	.3147	.2220	.5281
6	Base removing [S]	.4297	.3137	.7019
7	Base removing [S] & Query Subject	.3789	.2756	.6476

Table 5.4: Results of the Ablation Study on Structured Input Components (Strict Relevance)

token [D] in distinguishing between query elements may not be significantly helpful to facilitate the model's overall performance.

5.4 Other Experiments

5.4.1 Question De-duplication

During the first-stage retrieval with similarity functions, data pre-processing involves the removal of duplicated questions. Table 5.5 gives 3 groups of examples of the identified duplicates. As shown, each pair of examples are actually identical questions. In CQA forums, it is common for users to repeatedly post the same question if they experience long wait times, system delays, or other issues. Identifying these duplicates can reduce the unnecessary burden of data processing in subsequent stages.

Cluster id	Question id	Question Subject + Description
89	7727	*On form Schedule F of a chapter 7 bankruptcy petition, what does ”set off” mean On Schedule F of Ch 7, Does “set off” mean the same as “charged off”? I looked on my credit report for the words “set off” and it is not anywhere, however I do see the date the company “charged off my debt. Is that what I put under “set off”? On Schedule F of Ch 7, Does what does “set off” mean? Does it mean “Charged off”?*
	7714	*On Schedule F of Ch 7 , Does “set off” mean the same as “charged off”? ch 7 bankruptcy On Schedule F of Ch 7 bankruptcy, Does “set off” mean the same as “charged off”? I looked on my credit report for the words “set off” and it is not anywhere, however I do see the date the company “charged off my debt. Is that what I put under “set off”?*
198	389	*Application to Adjust Status (I-485) denied due to J1 requirement. Today my I-485 was denied because I did not apply for a J1 two-year waiver. Four weeks ago I applied for a J1 waiver with the Department of State because my country does not require me to stay two years. It’s been four weeks and the Department of State still does not record of my case. The processing time is 6-8 weeks. In the mean time, I am requested to leave the US. Should I wait four more weeks and request a motion to reopen/reconsider my I-485, or should I leave the US immediately? Thank you.*
	5833	*I-485 DENIED! due to J1 requirement. Today my I-485 was denied because I did not apply for a J1 two-year waiver. Four weeks ago I applied for a J1 waiver with the Department of State because my country does not require me to stay two years. It’s been four weeks and the Department of State still does not record of my case. The processing time is 6-8 weeks. In the mean time, I am requested to leave the US. Should I wait four more weeks and request a motion to reopen/reconsider my I-485, or should I leave the US immediately? Thank you.*
113	8213	*Should my boyfriend & i Get married even tho he has a lot of debt will collectors come after me 4 his debt?or file 4 bankruptcy? I am a 24 year old woman, he is a 24 year old man, he needs medical coverage and my work will cover him only if we are married. Problem is he is buried in about 10,000 of debt with collectors coming after him, he wants to file-bankruptcy soon, but in the mean time he needs medical coverage. Should we get married to get coverage? or should he file bankruptcy first? will collectors coming after me for his preexisting debt? can i buy a house if we are married, & he files bankruptcy we have 1 child together. thanks.*
	8214	*Should my boyfriend and i Get married even though he has a lot of debt, will the collectors come after me for his debt? I am a 24 year old woman, he is a 24 year old man, he needs medical coverage and my work will cover him only if we are married. Problem is he is buried in about 10,000 of debt with collectors coming after him, he will be filing bankruptcy soon, but in the mean time he needs medical coverage. Should we get married to get coverage? or should he file bankruptcy first? will collectors coming after me for his preexisting debt. we have 1 child together. thanks.*

Table 5.5: Examples of Identified Duplicated Questions

Data Set	Train Set			Test Set		
Model Name	Base[1]	2000 Clusters[2]	200 Clusters[3]	Base[1]	2000 Clusters[2]	200 Clusters[3]
Recall@10	.1057	.1057	**.1059**	.1911	.1912	**.1923**
nDCG@10	.0747	**.0748**	**.0748**	.1355	.1358	**.1365**
MRR@10	.2609	.2613	**.2614**	.3440	.3442	**.3450**

[1] *Base*: Keeping duplicated queries
[2] *2000 Clusters*: Removing duplicated queries using K-means with 2000 clusters
[3] *200 Clusters*: Removing duplicated queries using K-means with 200 clusters

Table 5.6: Evaluation Results of First-stage Retrieval Using BM25 with and without Duplicated Queries (**Strict Relevance**)

To prove the validity of this step, the results of the first stage retrievers, both with and without the removal of duplicated queries, are presented in Tables 5.6 and 5.7. The "*Base*" condition retains all duplicated queries, while the "*2000 Clusters*" and "*200 Clusters*" conditions correspond to the removal of duplicated queries based on a lexical similarity threshold of 90%, applying two different number of cluster settings with K-Means.

According to the results, after removing duplicate questions, cases with strict relevance demonstrate improved performance, while those with lenient relevance show some decline in both the train and test sets. Theoretically, de-duplication can optimize performance by reducing noise and increasing the efficiency of data processing [47]. By eliminating unnecessary information, models can more accurately identify patterns and make predictions based on the most relevant data.

This unexpected conclusion in our case may be attributed to the limited size of the data set. When working with a small sample size, it is possible that the data may not accurately represent the population, leading to anomalous results. Increasing the size of the data set can improve the reliability and validity of the conclusions drawn from the study.

Furthermore, these two tables also indicate that using 200 clusters is a more effective option than using 2000 clusters, especially when using strict relevance labels. Given that the total number of questions in the entire data set is only approximately 10,000, 2000 clusters provide limited space for clustering more-similar questions. Additionally, considering the computational time, we have chosen to use 200 clusters as the standard experimental setting.

5.4.2 Re-rank with Strict or Lenient Relevance Label

In Section 5.1, it is observed that lenient relevance outperformed strict relevance during the first-stage retrieval. However, after fine-tuning Cross-Encoders, lenient relevance yielded much lower results than strict relevance, as presented in Table 5.8. This can be attributed to the lenient training setting, where labels "1", "2" and "3" are merged as label "1", inevitably introducing noise. As a result, many non-relevant answers under strict set-

Data Set	Train Set			Test Set		
Model Name	Base[1]	2000 Clusters[2]	200 Clusters[3]	Base[1]	2000 Clusters[2]	200 Clusters[3]
Recall@10	**.1421**	.1340	.1339	.2035	.2023	**.2036**
nDCG@10	**.1247**	.1168	.1168	**.1902**	.1889	.1899
MRR@10	.2609	.2612	**.2614**	0.344	.3442	**.3449**

[1] *Base*: Keeping duplicated queries
[2] *2000 Clusters*: Removing duplicated queries using K-means with 2000 clusters
[3] *200 Clusters*: Removing duplicated queries using K-means with 200 clusters

Table 5.7: Evaluation Results of First-stage Retrieval Using BM25 with and without Duplicated Queries (**Lenient Relevance**)

Relevance Label	Strict	Lenient
Recall@10	**.4277**	.1050
nDCG@10	**.3114**	.0668
MRR@10	**.7038**	.1017

Table 5.8: Evaluation Results of Training the Fully-structured SI-Reranker with Strict and Lenient Relevance Labels

tings are considered relevant during the lenient training, leading to an overestimation of relevant answers and adversely affecting the effectiveness of the model. Therefore, strict relevance labels were finally chosen to be used in re-ranking experiments, while a better lenient training setting is left for future work. In fact, even with term-based retrieval models, many of the top-performing systems employ boolean settings for finer-grained control [48].

Chapter 6

Discussion

In this chapter, we will delve into a more comprehensive analysis based on the experiment results presented in Chapter 5. Our discussion will encompass the constraints of the retrieval models, the impact of the different parts of the input injections in our proposed SI-Reranker, the impact of strict or lenient relevance labels, the limits during data processing, the specificities of LIR, and some generalization of our study to other domains.

6.1 Limitations of Term-based Retrieval Models

Table 5.1 and 5.2 show the effectiveness of the term-based retrieval models. It is indicated that the model's effectiveness is limited across all metrics. This limitation may be attributed to several factors. For instance, the term-based models rely on statistical properties of terms, while in the practice of answer retrieval, it is often observed that there is minimal overlap between the terms present in a question and those in its corresponding answer [49]. Such term-based models also do not take into account the context in which the keywords appear or their relationships with other words in the text, and may struggle to capture the underlying meaning of queries and answer passages.

What's more, they may be less effective in handling complex queries or those containing multiple concepts. In the context of legal domain retrieval, it is crucial to deeply process the content rather than rigidly matching words, as legal documents often contain complex concepts and terminologies that require more profound processing before a reliable retrieval.

For example, consider the question in the Table 4.1. The query contains multiple concepts and legal terminologies, such as "small business assets exempt" and "chapter 7 bankruptcy petition". A term-based retrieval model may not be able to accurately capture the relationships between these concepts and may retrieve documents that rigidly match the query terms without considering their contextual meaning. In contrast, a transformer-based retrieval model that can further process the deep linking between the terms in the query and answers would be better equipped to handle such complex queries. For instance, it could identify that the query is asking about the exemption of small business assets under a personal chapter 7 bankruptcy petition in California state and retrieve relevant documents that address this specific issue.

6.2 Effectiveness of Different Injections in SI-Reranker

In our proposed structured input, the purpose of the added splitter tokens is to serve as separators between the different components of a query, facilitating the model's ability to distinguish them. However, the results in Table 5.3 between the columns of SI-Reranker and SI-Reranker without added tokens imply that treating the query components separately may not be necessary. Instead, after removing the splitter tokens, analyzing the whole context as a combination of all elements has the potential to enhance the model's performance. The outperformance of Evaluation 4 in Table 5.4 may be attributed to the same reason: processing the content as an integrated entity is more beneficial for the model's performance than isolating its components.

In contrast, the injection of query tags greatly improves the performance of re-ranking. As shown in Table 5.4, Evaluation 3, where the query tags are removed from the evaluation input, presents the third poorest performance, next to removing the query description and query subject. It can be concluded that query tags serve as useful information injections for the Cross-Encoder when dealing with legal answer retrieval.

So far, it seems the existence of tokens has an adverse impact on the model, however, considering the tokens are consisted of [S], [D] and [T], which represent different components of the query, it is necessary to conduct further qualitative analysis to validate which specific token or tokens are possible for this negative impact. Besides, in our implementation, all query tags are included in the injection. However, given that the input length to a Cross-Encoder is limited, when the input query or answer is too long, a choice must be made between the injection tags. This necessitates further analysis to determine the optimal selection among the query tags. The work of Askari *et al.* [34], where a visual tool was employed to use different colors to indicate word-level attribution values, represents a promising approach to these issues.

6.3 Impact of Strict or Lenient Relevance Label

From Table 5.1 and Table 5.2, the performance of strict relevance labels in first-stage retrieval is not as satisfactory as that of lenient relevance labels. Under the lenient relevance setting, more documents are counted as relevant, resulting in a Recall@1000 of over 0.6. This provides a broader context for re-ranking, where more possible best answers could be retrieved. However, as discussed in Section 5.4.2, our current training strategy creates an obstacle to the appliance of lenient relevance labels in re-ranking. One possible solution to this problem could be to apply different relevance labels at different stages of the process. For example, lenient relevance could be used in the first-stage retrieval to achieve higher recall, while strict relevance could be used in re-ranking to improve performance.

6.4 Limitations in Data Processing

As has been mentioned, legal documents and queries can often be complex, with multiple layers of meaning and interpretation. This can make it challenging to accurately assess the relevance of a given document to a particular query. In our case, some relevance judgments are based on the agreement upvotes from certified lawyers, while the “helpful” upvotes from non-lawyer users are not considered. However, there do exist some cases where non-lawyer users’ choices differ from those of lawyers, as shown in Table 6.1.

Question	Subject	*What is the best way to settle the past-due HOA fee after foreclosure?*
	Description	*The property was already foreclosed in 2010. The HOA had placed a lien against the property. I was told by the RE that the HOA lien would be wiped out after the bank foreclosed the property and I would not have to pay the past dues. Well, almost 2years later I got a letter from the collection company hired by HOA to pay $9000 (outrageously bloated amount!). I do not have $9000. I am devastated. My Q’s are: 1) can HOA still do this to me after I lost my house? 2) will the collection company settle for a lesser amount? and what would be a good ballpark % that I should ask for? is 10% worth a try? 3) should I hire a negotiator/ lawyer? Any tips and advise are greatly appreciated.*

Answer Index	**Upvotes**		**Answer Content**
	Lawyer	User	
1	*4*	*2*	*keep a good track of calls and communications and actions of the debt collector and see a consumer rights attorney . you may have case against them and not know it*
2	*3*	*1*	*obtain a credit report to see who claims this account receivable.*
3	*6*	*1*	*unfortunately , you are on the hook for your hoa dues and costs of collection for the period during which you owned the property . it is a personal obligation not wiped out by the foreclosure of a deed of trust . yes , you can negotiate a lesser sum . a bankruptcy would be an option to resolve it and your other debts . i too would recommend consultation with a bankruptcy or consumer rights attorney . it is likely such an attorney could negotiate a substantially better settlement than you would be able to on your own .*
4	*4*	*0*	*before i became a bankruptcy attorney , i represented hoas as a collection attorney . an hoa can be quite stubborn about delinquent hoa dues . the law in california is that you are liable for all dues the accrue until the property transfers to some else . for a $ 9000 debt , the hoa would probably want at least 50 % if you to make a lump sum offer . they might also consider a payment plan . debt settlement companies are nearly worthless , so i would consult a bankruptcy attorney about your bk and settlement options .*

Table 6.1: An Example of Non-Lawyer Users’ Choices against Lawyers’

One possible explanation for this discrepancy could be that an answer receives agreements from lawyers while other users find it not feasible. Alternatively, the real best answer may have been posted too close to the data collection time, preventing it from receiving enough upvotes from either lawyers or non-lawyer users. It is worth considering that while lawyers’ opinions are generally more reliable when dealing with serious and rigid legal matters, there may be situations where practical considerations come into play. An answer that is technically correct from a legal standpoint may not be practical in a real-world situation.

Under such conditions, the opinions of non-lawyer users could provide valuable insights and perspectives. These users may possess first-hand experience with the practical implications of legal issues and may be able to offer feedback on the feasibility of the proposed solutions. Therefore, it may be beneficial to incorporate the opinions of non-lawyer users when assessing the relevance of an answer to a query.

On the other hand, it is also possible that the opinions of non-lawyers should be excluded due to their lack of professional expertise in legal matters. Ultimately, the decision to incorporate or exclude non-lawyer upvotes would depend on various factors, including the specific characteristics of the legal domain.

6.5 Specifities of Legal Retrieval

Our study of answer retrieval in CQA focuses on the legal domain, which is inevitably constrained by its specificities. The complexity and nuanced meaning of legal documents and queries pose challenges in interpretation, and accurately assessing the relevance of answers to a query is a key challenge, as has been discussed in 6.4.

Furthermore, the selection of additional information for injection of the re-ranker input is another important consideration, which should be based on the relevance factors introduced in 2.2. For example, the dynamic nature of the legal documents presents a challenge. Laws and regulations can change over time, making it difficult to ensure that the answers retrieved from the repository are up-to-date and still applicable to a recently raised query.

Besides, document length is another factor not to be overlooked, as the input length to the Cross-Encoder is not infinite, if the query or the answer is too long, it may result in the truncation of important information and prevent it from being processed. This will undoubtedly influence the model's performance.

6.6 Generalization to Other Domains

The injection of additional useful information into the structured input of the Cross-Encoder for re-ranking can also play a positive role in the answer retrieval task in other domains, particularly those characterized by strong theoretical foundations and well-defined subfields. Examples of such domains include chemistry and medicine. For open-domain CQA platforms which have carefully defined and detailed categories for different fields, the injection of category tags can also be helpful in improving the retrieval.

However, there may exist certain domains in which symbolic formulas and abstract codes are integral components of the answer, such as on Stack Overflow. In such cases, if the query body primarily consists of text, relevant question retrieval may be a more effective solution than answer retrieval. Further research is needed to explore the potential benefits and limitations of using additional information injections in different domains and contexts.

Chapter 7

Conclusion

In this thesis, we undertake research on the answer retrieval task in the legal CQA. We propose an answer retrieval pipeline comprising two stages: first-stage retrieval with BM25 and then re-ranking with different kinds of fine-tuned Cross-Encoder, where the structured input is injected with category information and different splitter tokens for encoding query and answer components. We evaluate our methods on a legal CQA dataset and conduct a comprehensive analysis on the experimental results, comparing the effectiveness of term-based models and different fine-tuning models, finding that fine-tuning with domain-specific data can greatly improve retrieval performance. Furthermore, the injection of query category tags into the Cross-Encoder yields the most promising results. It is expected that the structured input with additional information to the Cross-Encoder will not only improve the answer retrieval in legal CQA but also prove effective in other domains as well.

7.1 Answers to Research Questions

RQ1 *How effective are term-based retrieval models in legal answer retrieval?*

Term-based retrieval models, such as BM25, rely on matching query terms with terms in the documents to retrieve relevant results. However, this approach may be less effective when handling complex queries or those containing multiple concepts, particularly in the legal domain where processing the deep meaning and linking under the context is crucial.

RQ2 *How effective is a zero-shot Cross-Encoder re-ranker in retrieving legal answers, and to what extent does fine-tuning enhance its effectiveness?*

Prior to fine-tuning, the performance of the zero-shot re-ranker is unsatisfactory as the pre-trained model can not capture heterogeneous semantic information and specific legal domain terms [4], models that are not specifically tailored to this domain may struggle to achieve high levels of performance.

However, fine-tuning with task-specific annotated data allows the model to better adapt to downstream tasks [50], in our case, it is legal domain answer retrieval. This can lead to more accurate and relevant results, as evidenced by the substantial increase in performance scores – more than double those of the zero-shot model.

RQ3 *To what extent can we further improve the performance of a Cross-Encoder re-ranker by injecting additional information into the input? Which part(s) of the injections are more effective in legal answer retrieval?*

The structured input could contribute to an over 10% improvement in the performance score compared to the fine-tuned model with the original input. Among all the components of the structured input, the injection of query tags has proven to be particularly effective in providing useful information and more precise context for legal answer retrieval.

7.2 Future Work

In light of the various aspects discussed in Chapter 6, including several limitations, moving forward, there are some directions for future work to consider.

Improving First-Stage Retrieval The first-stage retrieval amis to provide a good set of documents for re-ranking, but from Table 5.1, when using BM25, Recall@1000 on the test set is only 0.5416, meaning that nearly half of best answers are not retrieved. This limits the effectiveness of the subsequent re-ranking stage and may lead to missing relevant answers. Therefore, a possible direction for future work is to explore alternative methods to first-stage retrieval that can achieve **higher recall** of the candidate answers. For example, one could use deep neural methods [51] or hybrid methods that combine classical term-based models with semantic similarity [52].

Enhancing SI-Reranker The effectiveness of different injections in SI-Reranker has been explored in this study, revealing interesting findings regarding the impact of treating query components separately and the benefits of query tags. Future work could further investigate the optimal approach for injecting information into the SI-Reranker to improve its performance. Conducting qualitative analysis, similar to the work of Askari *et al.* [34], to attribute the impact of specific tags, specific tokens or combinations of these items would provide valuable insights. Additionally, exploring alternative injection strategies and evaluating their impact on re-ranking effectiveness could be a valuable direction for future research.

Exploring Relevance Labeling Strategies As has been discussed, the current training strategy presents challenges in utilizing lenient relevance labels effectively. Future work could investigate alternative strategies for incorporating lenient relevance labels in re-ranking.

Incorporating Legal Domain-specific Features The potential value of incorporating the opinions of non-lawyer users has been proposed when assessing the answer relevance to a query. Future work could focus on the methodologies to incorporate the feedback and upvotes from both lawyers and non-lawyers, considering the practical insights and professional expertise from both sides. Evaluating the impact and reliability of the comprehensive incorporation on the retrieval effectiveness could be essential for improving the performance of the entire answer retrieval model.

Furthermore, considering the temporal aspect of QA in the legal domain and continuously updating the repository with the latest legal developments would be crucial for ensuring the relevance and currency of retrieved answers. By embracing the unique characteristics of the legal domain, future research can provide more specialized and effective solutions for legal information retrieval applications.

Bibliography

[1] S. S. Shah, T. S. Bavaskar, S. S. Ukhale, R. A. Patil, and A. S. Kalyankar, "Answer ranking in community question answer (QA) system and questions recommendation," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pp. 1–6, 2018.

[2] N. Othman, R. Faiz, and K. Smaïli, "Enhancing question retrieval in community question answering using word embeddings," *Procedia Computer Science*, vol. 159, pp. 485–494, 2019. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 23rd International Conference KES2019.

[3] G. Zhou, Y. Zhou, T. He, and W. Wu, "Learning semantic representation with neural networks for community question answering retrieval," *Knowledge-Based Systems*, vol. 93, pp. 75–83, Feb. 2016.

[4] C. Sansone and G. Sperlí, "Legal information retrieval systems: State-of-the-art and open issues," *Information Systems*, vol. 106, p. 101967, 2022.

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

[6] R. Nogueira and K. Cho, "Passage re-ranking with bert," 2020.

[7] K. Zhang, W. Wu, H. Wu, Z. Li, and M. Zhou, "Question retrieval with high quality answers in community question answering," CIKM '14, (New York, NY, USA), p. 371–380, Association for Computing Machinery, 2014.

[8] X. Cao, G. Cong, B. Cui, and C. S. Jensen, "A generalized framework of exploring category information for question retrieval in community question answer archives," in *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, (New York, NY, USA), p. 201–210, Association for Computing Machinery, 2010.

[9] S. Momtazi and D. Klakow, "Bridging the vocabulary gap between questions and answer sentences," *Information Processing & Management*, vol. 51, no. 5, pp. 595–615, 2015.

[10] R. Zanibbi, D. W. Oard, A. Agarwal, and B. Mansouri, "Overview of ARQMath 2020: Clef lab on answer retrieval for questions on math," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, (Cham), pp. 169–193, Springer International Publishing, 2020.

[11] L. Yang, Q. Ai, D. Spina, R.-C. Chen, L. Pang, W. B. Croft, J. Guo, and F. Scholer, "Beyond factoid QA: Effective methods for non-factoid answer sentence retrieval," in *Advances in Information Retrieval*, (Cham), pp. 115–128, Springer International Publishing, 2016.

[12] F. Riahi, Z. Zolaktaf, M. Shafiei, and E. Milios, "Finding expert users in community question answering," in *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, (New York, NY, USA), p. 791–798, Association for Computing Machinery, 2012.

[13] K. Balog, Y. Fang, M. De Rijke, P. Serdyukov, L. Si, *et al.*, "Expertise retrieval," *Foundations and Trends® in Information Retrieval*, vol. 6, no. 2–3, pp. 127–256, 2012.

[14] A. Askari, S. Verberne, and G. Pasi, "Expert finding in legal community question answering," in *Advances in Information Retrieval*, vol. 13186 of *Lecture Notes in Computer Science*, (Cham), Springer, 2022.

[15] M. Zaib, W. E. Zhang, Q. Z. Sheng, A. Mahmood, and Y. Zhang, "Conversational Question Answering: A Survey," *arXiv e-prints*, p. arXiv:2106.00874, June 2021.

[16] A. Roberts, C. Raffel, and N. Shazeer, "How much knowledge can you pack into the parameters of a language model?," *arXiv e-prints*, pp. arXiv–2002, 2020.

[17] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," 2021.

[18] T. Bench-Capon, M. Araszkiewicz, K. Ashley, *et al.*, "A history of ai and law in 50 papers: 25 years of the international conference on ai and law," *Artificial Intelligence and Law*, vol. 20, pp. 215–319, 2012.

[19] B. Mitra and N. Craswell, "An introduction to neural information retrieval," *Foundations and Trends® in Information Retrieval*, vol. 13, no. 1, pp. 1–126, 2018.

[20] M. van Opijnen and C. Santos, "On the concept of relevance in legal information retrieval," *Artificial Intelligence and Law*, vol. 25, pp. 65–87, 2017.

[21] G. Wiggers, S. Verberne, G.-J. Zwenne, and W. Van Loon, "Exploration of domain relevance by legal professionals in information retrieval systems," *Legal Information Management*, vol. 22, no. 1, p. 49–67, 2022.

[22] *ICAIL '19: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, (New York, NY, USA), Association for Computing Machinery, 2019.

[23] P. Bhattacharya, K. Ghosh, S. Ghosh, A. Pal, P. Mehta, A. Bhattacharya, and P. Majumder, "FIRE 2019 AILA track: Artificial intelligence for legal assistance," in *Proceedings of the 11th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '19, (New York, NY, USA), p. 4–6, Association for Computing Machinery, 2019.

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

[25] "Cross-encoder." `https://www.sbert.net/examples/applications/cross-encoder/README.html`.

[26] F. Liu, Y. Jiao, J. Massiah, E. Yilmaz, and S. Havrylov, "Trans-encoder: Unsupervised sentence-pair modelling through self- and mutual-distillations," 2022.

[27] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," 2020.

[28] A. Baccini, S. Déjean, L. Lafage, and J. Mothe, "How many performance measures to evaluate information retrieval systems?," *Knowledge and Information Systems*, vol. 30, pp. 693–713, 2012.

[29] I. Srba and M. Bielikova, "A comprehensive survey and classification of approaches for community question answering," *ACM Trans. Web*, vol. 10, aug 2016.

[30] E. Voorhees and D. Harman, *TREC: Experiment and Evaluation in Information Retrieval*. 01 2005.

[31] T. Chen, M. Zhang, J. Lu, M. Bendersky, and M. Najork, "Out-of-domain semantics to the rescue! zero-shot hybrid retrieval models," in *Advances in Information Retrieval*, (Cham), pp. 95–110, Springer International Publishing, 2022.

[32] Y. Lu, Y. Liu, J. Liu, Y. Shi, Z. Huang, S. F. Y. Sun, H. Tian, H. Wu, S. Wang, D. Yin, and H. Wang, "Ernie-search: Bridging cross-encoder with dual-encoder via self on-the-fly distillation for dense passage retrieval," 2022.

[33] J. Kim, M. Kim, and S.-w. Hwang, "Collective relevance labeling for passage retrieval," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (Seattle, United States), pp. 4141–4147, Association for Computational Linguistics, July 2022.

[34] A. Askari, A. Abolghasemi, G. Pasi, W. Kraaij, and S. Verberne, "Injecting the BM25 score as text improves BERT-based re-rankers," in *Advances in Information Retrieval*, (Cham), pp. 66–83, Springer Nature Switzerland, 2023.

[35] Nils Reimers, "A pipeline for information retrieval / question answering retrieval." `https://github.com/UKPLab/sentence-transformers/tree/master/examples/applications/retrieve_rerank`.

[36] S. E. Robertson and S. Walker, "Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval," in *SIGIR'94*, pp. 232–241, Springer, 1994.

[37] "Okapi BM25." `https://en.wikipedia.org/wiki/Okapi_BM25`.

[38] M. Lease and E. Charniak, "A dirichlet-smoothed bigram model for retrieving spontaneous speech," in *Advances in Multilingual and Multimodal Information Retrieval*, (Berlin, Heidelberg), pp. 687–694, Springer Berlin Heidelberg, 2008.

[39] S. Clinchant and E. Gaussier, "Bridging language modeling and divergence from randomness models: A log-logistic model for ir," in *Advances in Information Retrieval Theory*, (Berlin, Heidelberg), pp. 54–65, Springer Berlin Heidelberg, 2009.

[40] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, *et al.*, "Huggingface's transformers: State-of-the-art natural language processing," 2020.

[41] Z. Liu, G. I. Winata, A. Madotto, and P. Fung, "Exploring fine-tuning techniques for pre-trained cross-lingual models via continual learning," 2020.

[42] J. MACQUEEN, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967*, pp. 281–297, 1967.

[43] V. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," in *Soviet Physics-Doklady*, vol. 10, pp. 707–710, 1966.

[44] "TheFuzz library for python." `https://github.com/seatgeek/thefuzz`.

[45] "BCEWithLogitsLoss - A loss function in Pytorch library." `https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html`.

[46] "CrossEntropyLoss - A loss function in Pytorch library." `https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html`.

[47] T. B. Hashimoto, M. D. Edwards, and D. K. Gifford, "Universal count correction for high-throughput sequencing," *PLoS computational biology*, vol. 10, no. 3, p. e1003494, 2014.

[48] S. Tellex, B. Katz, J. Lin, A. Fernandes, and G. Marton, "Quantitative evaluation of passage retrieval algorithms for question answering," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, (New York, NY, USA), p. 41–47, Association for Computing Machinery, 2003.

[49] A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal, "Bridging the lexical chasm: Statistical approaches to answer-finding," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, (New York, NY, USA), p. 192–199, Association for Computing Machinery, 2000.

[50] H. Wang, J. Li, H. Wu, E. Hovy, and Y. Sun, "Pre-trained language models and their applications," *Engineering*, 2022.

[51] J. Zhan, J. Mao, Y. Liu, M. Zhang, and S. Ma, "RepBERT: Contextualized text embeddings for first-stage retrieval," 2020.

[52] J. Guo, Y. Cai, Y. Fan, F. Sun, R. Zhang, and X. Cheng, “Semantic models for the first-stage retrieval: A comprehensive review,” *ACM Trans. Inf. Syst.*, vol. 40, mar 2022.