A Question-Answering System for Vietnamese Public Administrative Services

Anh Pham Duy*
Huong Le Thanh*
anh.pd183481@sis.hust.edu.vn
huonglt@soict.hust.edu.vn
Hanoi University of Science and Technology
Hanoi, Vietnam

Abstract

In the realm of legal question-answering (QA) systems, information retrieval (IR) plays a pivotal role. Despite thorough research in numerous languages, the Vietnamese research community has shown limited interest in legal information retrieval, particularly in the context of public administrative services. In this paper, we propose the development of a QA system tailored to the Vietnamese language, specifically focusing on the domain of public administrative services. Our system provides legal-based responses, and it is built upon a combination of retrieval and re-ranking techniques. We employ both lexical-based and semantic-based retrieval models and integrate them to create the final model. Our research shows that the system outperforms existing models in retrieving public administrative information and answering questions related to Vietnamese legal documents.

CCS Concepts: • **Artificial Intelligence** → **Natural Language Processing**; • **Information Systems** → **Information Retrieval**.

Keywords: Question-Answering, Information Retrieval, Public Administrative Services, Vietnamese

ACM Reference Format:
Anh Pham Duy and Huong Le Thanh. 2023. A Question-Answering System for Vietnamese Public Administrative Services. In *Proceedings of The 12th International Symposium on Information and Communication Technology (SoICT 2023).* ACM, New York, NY, USA, 8 pages. https://doi.org/3628797.3628965

*Both authors contributed equally to this research.

SoICT 2023, Dec. 07–08, 2023, Ho Chi Minh city, Vietnam

ACM ISBN 979-8-4007-0891-6/xx/xx…$15.00
https://doi.org/3628797.3628965

1 Introduction

The development of Large Language Models (LLMs) has led to some progress in automatic question-answering systems. The AI chatbot ChatGPT[1] can answer questions across multiple domains, but it frequently provides inaccurate information when addressing specific domains like Vietnamese law. This is where information-seeking methods can come into play, bridging the gap between what people inquire about and the intricate legal documents they require.

Legal documents are known for being long, complex, and filled with many details. The complexity of legal language and its various topics can make it difficult to answer legal questions.

This article focuses on building a legal QA system, in which the answer is a related passage in the law document. The contribution of this paper is as follow. First, we construct a legal dataset consisting of 785,996 passages from legal documents, accompanied with 4,547 QA pairs belongs to the public service domain. Second, we build a QA system specialized in the public service domain. The system has several components that function across stages, including Understanding, Retrieval, re-ranking, and Ensemble. We conduct experiments on a legal dataset to evaluate our system's efficiency.

The rest of this paper is organized as followed: Section 2 discusses some related works to our research. Section 3 describes the process of generating our dataset. Our QA system is represented in Section 4. Experimental results with our legal dataset are analyzed in Section 5. Finally, Section 6 concludes the paper and proposes some future works.

2 Related Works

Currently, English legal information retrieval has gained attention with events like COLIEE[2] and JURIX[3]. However, Vietnamese legal information retrieval is limited, particularly in public administrative services. **Information Retrieval (IR)** has improved greatly over the years, thanks to technology and AI. Within the spectrum of IR, two major categories are prominent: non-neural and neural approaches.

[1] https://chat.openai.com/
[2] https://sites.ualberta.ca/~rabelo/COLIEE2022/
[3] http://jurix.nl/conferences/

Non-neural approaches involve methods that do not utilize neural networks, focusing on relevance determination based on term frequencies in questions and documents. These methods often assess relevance through lexical overlap, frequency, and statistical characteristics of terms.

Notable among non-neural methods are BM25 and TF-IDF. BM25 [15] is derived from probabilistic information retrieval models, gauging the importance of terms in questions and documents. TF-IDF [16] evaluates term significance in documents by considering their occurrence frequency and inverse statistics across different documents.

While non-neural methods may not excel in solving semantic search challenges, they provide fundamental concepts and support for modern information retrieval models. The combination of non-neural and neural methods holds promise in developing robust retrieval systems.

Neural approaches emerged to address limitations in non-neural methods regarding semantic understanding. Instead of relying on fixed features, these approaches propose efficient neural architectures and training methods to extract various abstraction levels of semantics from data, expanding system capabilities.

Before BERT [2], systems mainly relied on pre-trained embedding models like Word2Vec [8] and GloVe [12]. Researchers like Palangi (2016) [10] utilized LSTM networks for semantic sentence embeddings through calculating sentence similarity. Shen [17] proposed a CNN-based semantic contextual model, compressing words into low-dimensional vectors. Pang [11] introduced DeepRank, simulating human ranking processes.

Transformer models, especially BERT, have achieved remarkable success in various NLP tasks, including question-answering and text-matching tasks, thanks to their cross-encoder and bi-encoder approaches. Moreover, multilingual BERT models such as mBERT and XLM-R have shown promising results in multilingual information retrieval. These advancements have opened up new opportunities in the field of NLP.

Transformer cross-encoder utilizes cross-attention to encode both question and document simultaneously, providing a binary output representing the similarity between input pairs. Birch [18], a hybrid method, combines lexical-based retrieval with a cross-encoder for effective information extraction. Cross-encoder's dual-encoding mechanism understands question and document semantics, yielding confident predictions of relevance. Dense Passage Retrieval (DPR) [5] utilizes a cross-encoder for open questions, achieving high multilingual information retrieval performance.

Transformer bi-encoder is more resource-efficient than cross-encoder, independently encoding question and document to generate embedding vectors, assessed through similarity metrics. SBERT [14] is such a transformer bi-encoder, which employs SiameseBERT for meaningful sentence embeddings, achieving notable results in QA and IR tasks.

Gao et al.[3] proposed SimCSE, a pre-training model that focuses on learning sentence embeddings by comparing sentence pairs, using a simple yet effective contrastive loss to evaluate similarity. It maximizes similarity between different encodings of the same sentence while increasing distances between embeddings of different sentences. Their experiments show that the SimCSE is effective in the sentence retrieval task.

In summary, legal information retrieval faces complexities that non-neural and neural methods aim to address. The fusion of both approaches holds promise in building robust and efficient retrieval systems.

3 Dataset Construction

In order to facilitate the development of a public administration QA system, we constructed two datasets: (i) a dataset containing legal documents related to public administration services, and (ii) a set of QA pairs related to administrative law.

The legal documents were collected from two reliable sources: "Law library[4]" and "National public service portal[5]". We employed the Beautiful Soup[6] library to gather and extract information from various types of documents, such as Laws, Decrees, Circulars, Joint Circulars, Resolutions, Ordinances, Decisions, and the Constitution.

The "National public service portal" also contains frequently asked questions and corresponding answers about national public services. The QA pairs from this website were collected for creating our QA dataset. The processes to generate the datasets from the above websites are introduced below.

3.1 Data Preprocessing

During the dataset construction, multiple data processing steps were undertaken to ensure the completeness, accuracy, and coherence of information within the dataset. The data processing steps are as follows:

1. Handling missing data: After collecting legal documents and QA pairs from the above websites, a thorough examination of the data was conducted to identify instances containing missing or incomplete information. The missing or incomplete instances can be questions without an answer or incomplete legal documents. Data samples with missing information were excluded from the dataset.

2. Word Tokenization: Since Vietnamese is a monosyllable language, Vietnamese text should be tokenized before further analysis. The Pyvi[7] library was utilized for accurate and efficient tokenization of Vietnamese text.

[4] https://thuvienphapluat.vn/
[5] https://dichvucong.gov.vn/p/home/dvc-cau-hoi-pho-bien.html
[6] https://pypi.org/project/bs4/
[7] https://pypi.org/project/pyvi/

3. Building a Legal Text Repository: Legal documents were collected from relevant sources and combined into a repository known as the legal document dataset. To ensure optimal performance of the QA system and prevent overloading with lengthy data, the legal documents were segmented into passages whose length was less than 256 words. If a provision in a legal document was less than 256 words, it would be kept as it was. Otherwise, it was further split into passages of less than 256 words. Sentences were kept intact during this splitting process. Each passage was attached with the title of the corresponding legal document and stored in the database for the retrieval process.

4. Generating the QA Dataset: For each QA pair collected from the "National public service portal" website, we mapped it with the corresponding provision and the legal document based on the provision's ID and the legal document's ID in the answer. The set of QA pairs accompanied by the provision's ID and the legal document's ID is used for training and testing the QA system.

3.2 Dataset Statistics

The above data collection process resulted in a corpus of 24,911 legal documents, containing 312,061 provisions, divided into 785,996 passages. The number of legal documents is notably more significant, approximately three times, compared to the datasets in previous papers (i.e., [6], [13]) for Vietnamese legal document retrieval. This poses a significant challenge for the information retrieval task, as real-world legal document repositories are often extensive. Table 1 describes the provision lengths in our original legal document dataset, with most provisions having fewer than 256 words, aligning with the PhoBERT model's input limitations. However, there are some longer provisions that need to be further split into smaller passages to ensure compatibility with the model's requirements.

Table 1. Distribution of Provision Lengths in the Original Legal Document Dataset

Length	Quantity	Percent
<100	203443	39.59%
101 - 256	156233	30.40%
257 - 512	87437	17.02%
513+	66727	12.99%

The collected QA dataset contains 4,547 pairs, with 4,000 pairs allocated for training and 547 for testing. Within this dataset, there are 2,668 distinct provisions, accounting for 0.85% of the total provisions in the legal document repository. Most QA pairs consist of fewer than 100 words, making them suitable for BERT-based models. Each question in the QA dataset may be associated with 1 to 7 relevant passages, with 98% having 1 to 2 related passages, as detailed in Table 2.

Table 2. The Number of Passages Related to each Question

No. related passages	1	2	3	4	5	7
No. questions	4036	432	57	18	3	1

4 The System Architecture

Traditional information retrieval models often use BM25 for lexical-based search. However, this approach often omits semantically related results. To solve this problem, we propose a solution that combines both lexical-based retrieval and semantic-based retrieval. Our proposed system architecture for the QA system is shown in Figure 1.

This architecture is an ensemble of two models: lexical-based retrieval and semantic-based retrieval. The BM25+ is used to retrieve relevant passages at the lexical level. Initially, the input text is tokenized using the Pyvi library, followed by the removal of stopwords to focus the model on important words. The goal of the lexical-based retrieval is to obtain a list of passages that have a high lexical-level similarity with the input question. These passages are then passed through a bi-encoder and then through a cross-encoder to re-rank the output of BM25+ based on their semantic meaning.

The SimCSE[3] and FAISS[4] are used at the semantic level. The SimCSE is used to encode sentences and passages, whereas FAISS is used for semantic search. The retrieval results from the semantic-based model are re-ranked using BM25+ and a cross-encoder. The final prediction is a result of combining the re-ranking steps from both the lexical-based module and the semantic-based one. Among all of the above-mentioned tasks, the SimCSE plays a key role in the success of the QA system. Our method of using the SimCSE in the QA task will be introduced next.

4.1 Training the SimCSE model for Representing Vietnamese Legal Text

The SimCSE model is highly effective in sentence representation tasks. It is based on the idea of contrastive learning, creating representations in such a way that similar data pairs have embeddings close to each other in the embedding space, while dissimilar data pairs have embeddings far apart. This helps the model learn a discriminative structure of the data.

Assuming we have $D = \left\{\left(x_i, x_i^+\right)\right\}_i$, where x_i and x_i^+ form similar pairs, let h_i and h_i^+ be the representations of x_i and x_i^+, then the loss function is defined as:

$$\mathfrak{L}_i = -\log \frac{e^{sim(h_i, h_i^+)/\tau}}{e^{sim(h_i, h_i^+)/\tau} + \sum_N^{j=1} e^{sim(h_i, h_j)/\tau}} \quad (1)$$

Here, $sim(x, y)$ represents the cosine similarity between two embedding vectors x and y, and τ is the temperature parameter, which adjusts the sensitivity of the loss function to negative samples. When the temperature is high, the loss function reduces the penalty for negative samples, making the embedding vectors closer together in the embedding

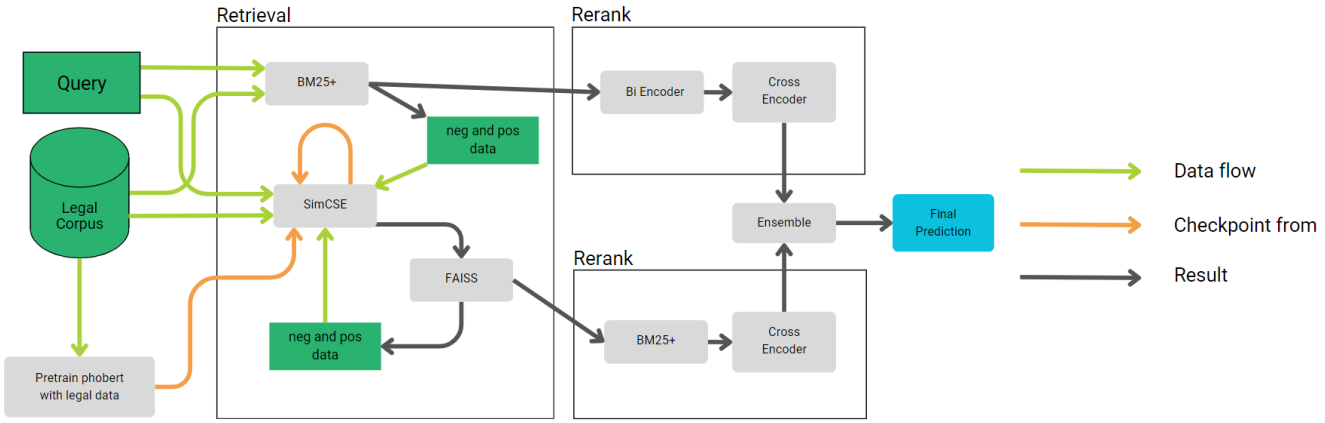

Figure 1. Architecture of our Public Service QA System

space. Conversely, when the temperature is low, the loss function increases the penalty for negative samples, making their embedding vectors farther apart in the embedding space.

The numerator in the loss function measures the similarity between embedding vectors of similar pairs (h_i, h_i^+), while the denominator aggregates the similarity of h_i with other h_j in the same batch or the dissimilarity of h_i. By minimizing this loss function, SimCSE encourages high similarity scores for positive pairs and low similarity scores for negative pairs, effectively representing sentences semantically. This training process is iterated several times to create a well-performing retrieval model (multi-stage).

Applying the SimCSE to our QA task, we first pre-trained it with our legal passage dataset, then fine-tuned it with positive and negative samples of QA pairs. To pre-trained the SimCSE, the unsupervised version of the SimCSE is employed to pre-train the PhoBERT [9] to learn sentence representations from Vietnamese legal documents. The loss function used in this stage is the Multiple Negatives Ranking Loss, which is based on the idea of contrastive learning. The model learns to create similar representations for semantically related sentences and different representations for sentences with unrelated meanings.

The Multiple Negatives Ranking Loss works by taking a batch of sentences and calculating the similarity between all pairs of representations. For each sentence, there is a positive pair, where two sentences share the same meaning or label. The remaining pairs serve as negative samples, involving sentences with different labels or meanings. This loss function aims to maximize the similarity between positive pairs while minimizing the similarity between negative pairs. It is particularly useful when only positive data is available, such as similar sentence pairs, duplicate questions, or translations. It does not require any negative samples or labels since it generates them automatically within the batch. However, having a large and diverse batch size is essential as it affects the quality and difficulty of the negative pairs. The formula for this lost function is in Equation 1.

The next stage involves fine-tuning the SimCSE with positive and negative samples of QA pairs. Through the lexical-based passage retrieval using BM25+, a labeled dataset is created, consisting of positive pairs (question-passage pairs relevant in the QA training dataset) and negative pairs (top-5 question-passage pairs returned by BM25+ but irrelevant to the question). In the first round, the newly generated data from BM25+ is used as the training data for the SimCSE, with the loss function updated as follows:

$$\mathfrak{L}_i = -\log \frac{e^{sim(h_i,h_i^+)/\tau}}{\sum_N^{j=1}(e^{sim(h_i,h_j^+)/\tau} + e^{sim(h_i,h_j^-)/\tau})} \quad (2)$$

Here, instead of using only similar pairs (x_i, x_i^+), we extend it to a triplet (x_i, x_i^+, x_i^-), with x_i being the question, x_i^+ being the relevant passage, and x_i^- being an irrelevant passage retrieved by BM25+. These irrelevant passages have high lexical similarity but low semantic similarity to the question. In the second round, we use the model from the first round to create a new dataset with positive and negative samples similar to the first round. The negative samples in this round have higher semantic similarity to the question compared to the previous round. In the third round, a smaller amount of data derived from the second round is used for training, and the negative samples are quite semantically related to the initial question.

4.2 Re-ranking Stage

The re-ranking stage is applied to both lexical-based and semantic-based retrieval blocks. We utilize semantic models to re-rank the output of BM25+. These semantic models consist of a bi-encoder and a cross-encoder. The best-performing SimCSE model from the previous stage serves as the sentence

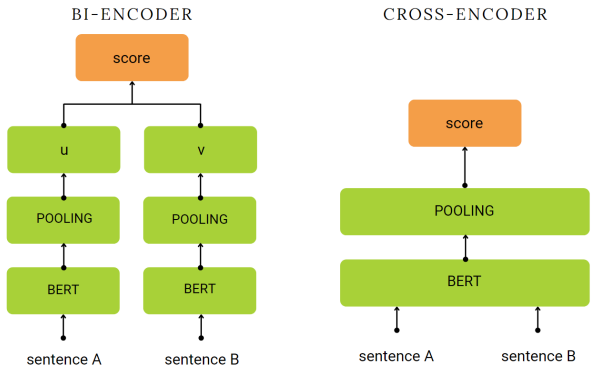

Figure 2. Bi-Encoder and Cross-Encoder

embedding model. The question, question-passage pairs, and the returned passages are passed through the embedding model, and their cosine similarity (Equation 3) is calculated for re-ranking.

$$\text{cosine-similarity} = \frac{u \cdot v}{\|u\| \, \|v\|} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2} \sqrt{\sum_{i=1}^{n} v_i^2}} \quad (3)$$

In the semantic-searching block, as the retrieved results are already optimized semantically, we further enhance the results by applying a lexical-based retriever followed by a cross-encoder. For the lexical-based retriever, several lexical-based comparison methods are employed, including BM25+. We experiment with various term lengths using n-grams (we define a gram as a word in Vietnamese).

A cross-encoder is a type of model trained to evaluate the similarity between a pair of input sentences, typically two text strings. Unlike bi-encoders, cross-encoders take both sentences into the Transformer network simultaneously, rather than passing them separately. This allows cross-encoders to leverage context information from both sentences to create better representations. Cross-encoders outperform bi-encoders like SimCSE, Condenser, etc., making them suitable for addressing the weaknesses of the aforementioned bi-encoder models. However, it computes much slower than the BM25 and the bi-encoder. Therefore, the cross-encoder is used as the final re-ranking stage for both the lexical-based and the semantic-based retrieval blocks. To train the cross-encoder, we use pairs of question and passage from the QA pairs dataset. Negative samples are generated by randomly sampling passages unrelated to the question.

4.3 Ensemble Model

As previously mentioned, our retrieval system comprises two distinct blocks: lexical-based and semantic-based retrievals. The final stage of the system involves synthesizing results from these two blocks, including the re-ranking process. The combined score from these stages can be expressed as (1-α)*lexical_score + α*semantic_score, where α can be predefined. We propose two methods for calculating the final score, as shown in Equations 4 and 5:

$$\text{ensemble-score} = \sqrt{\frac{score_1^2 + score_2^2 + \ldots + score_n^2}{n}} \quad (4)$$

$$\text{ensemble-score} = \sqrt[n]{score_1 * score_2 * \ldots * score_n} \quad (5)$$

Here, $score_i$ represents the score calculated by model i, and all scores are normalized within the range (0, 1) to ensure fairness. The proposed formulas (4, 5) ensure that the scores for each block are in the same range, facilitating straightforward combinations. The final result is the top k_2 most relevant passages, determined based on the aggregated score, selected from the best k_1 passages from each block.

5 Experiments

5.1 Experimental Setup

The entire model was implemented on the Google Colaboratory Pro platform, utilizing a Tesla T4 GPU with 25GB of RAM, an Intel(R) Xeon(R) dual-core 2.30GHz CPU, and a server supported by an NVIDIA TITAN RTX GPU with 62GB of RAM, and an AMD Ryzen 9 3900X 12-core CPU, although it was less frequently used. Google Colaboratory is highly suitable for research and training deep learning models.

The models were implemented using the Python programming language and leveraged libraries such as Sentence Transformer and Huggingface. Both libraries are widely used in language models and deep learning research.

5.2 Evaluation

Our system's performance was evaluated using four measures, which included MAP@k, NDCG@k, Recall@k, and F2@k, with k representing the number of top selected results. The MAP and NDCG [7] were employed to assess the quality of the ranking system. Recall determines the percentage of relevant items retrieved from the total number of relevant items in the dataset, while the F2-measure represents a weighted harmonic mean of precision and recall, taking into account both coverage (recall) and accuracy (precision).

The experiments were carried out based on our QA dataset including 4,000 QA pairs for training and 547 QA pairs for testing. The system performance was evaluated using MAP@k, NDCG@k, Recall@k, and F2@k metrics, depending on the specific scenario defined.

1. Comparison of Information Retrieval Models

Table 3 illustrates our system's performance when using different models. Among these models, the SimCSE model with PhoBERT-base encoder, the SimCSE model with mBERT encoder, and the SimCSE model with XLM-RoBERTa encoder, exhibit the best performance across all Recall and MAP metrics at the top 20 and top 100. This indicates that sentence

representation-based models are a good choice to improve the performance of information retrieval systems.

Table 3. Comparison of Information Retrieval Models

Model	k = 20		k = 100	
	Recall	MAP	Recall	MAP
TF-IDF	0.5442	0.421	0.7308	0.2308
BM25	0.5635	0.2905	0.7206	0.2697
LM Dirichlet	0.4212	0.2002	0.6138	0.1779
LM Jelinek Mercer	0.5791	0.2901	0.7094	0.2610
$\text{SimCSE}_{\text{XLM-RoBERTa}}$	0.6436	0.3029	0.7316	0.2963
$\text{SimCSE}_{\text{mBERT}}$	0.6513	0.3427	0.7697	0.3313
$\text{SimCSE}_{\text{PhoBERT-base}}$	**0.7626**	**0.4405**	**0.8519**	**0.4350**

Among traditional retrieval models (TF-IDF and BM25) and language models (LM Dirichlet and LM Jelinek-Mercer [1]), there is not a significant difference. Traditional models even outperform language models such as LM Dirichlet and Jelinek-Mercer. This might be due to the way similarity scores are computed in language models, which do not fully utilize term frequency information as TF-IDF or BM25.

Table 3 demonstrates the significant effectiveness of the SimCSE model due to its specialized framework for enhancing semantic search in Vietnamese. Notably, the SimCSE model based on PhoBERT, a Vietnamese language model, outperforms across all Recall and MAP metrics at the top 20, top 50, and top 100, surpassing the multilingual models. This suggests that utilizing a pre-trained language model specifically designed for Vietnamese improves document retrieval performance more than using a multilingual model like Multilingual BERT or a multilingual variant of RoBERTa such as XLM-RoBERTa.

2. Comparison of Re-ranking Models

Table 4 presents the comparison results of passage re-ranking methods. In this context, re-ranking methods were applied to the top 100 results returned by the SimCSE model. Specifically, lexical-based re-ranking methods perform better in the re-ranking task since the search model is semantic, and the returned results are semantically similar to the question. Therefore, to improve the results, a lexical matching model is required, and BM25 and its variants excel in this role. The table compares different variations of BM25 with various n-grams. Here, the evaluation is not solely based on individual words but also on phrases (n-grams = 2, 3, 4). In the legal text language, there are instances of phrases like "vi_phạm hành_chính | *administrative violation*" or "an_toàn giao_thông | *traffic safety*". Clearly, when these wo- rds are adjacent in the question and, in related passages, they also appear together, they should carry a higher weight compared to passages containing these words separately. According to the experimental results in the table, when n-grams are equal to 1 or 2, BM25 performs better in passage ranking. In particular, BM25+ shows the best results in Recall metrics.

Table 4. Comparison of Re-ranking Models after Searching using SimCSE with k=100

Model	Recall@5	Recall@10	Recall@20
BM25	0,6307	0,7300	0,8293
BM25L	0.5771	0.7123	0.8253
BM25+	0.6404	0.7546	0.8459
ngrams=2	**0.6433**	**0.7603**	**0.8487**
ngrams=3	0.5999	0.6838	0.7780
ngrams=4	0.5605	0.6113	0.6872

Regarding the performance of re-ranking methods, Table 5 evaluates the re-ranking of the top 100 results returned by the BM25+ model. Since the search model here is a lexical-based model, re-ranking methods based on semantics have an advantage. In fact, re-ranking the top 20 results with these methods demonstrates remarkable effectiveness, outperforming conventional BM25+ search. All metrics, such as Recall, F2, MAP, and NDCG, are higher than BM25+, with Recall increasing by 18%. The cross-encoder yields lower results compared to cosine similarity using SimCSE embeddings, partly due to limitations in the language model's sequence length. PhoBERT can encode a maximum of 256 tokens, but the input to the cross-encoder combines the question and related text, which exceeds the 256-word limit, leading to inaccurate predictions. Another reason is that using the cross-encoder for re-ranking a large number of search results is time and resource-consuming. Therefore, we suggest using the cross-encoder only for re-ranking the top 20 results obtained from other ranking algorithms.

Table 5. Comparison of Re-ranking Models after Retrieving the Top-20 using BM25+ with k=100

Model	Recall	F2	MAP	NDCG
original	0.5635	0.2905	0.1456	0.2825
cosine-sim	0.7422	0.2036	0.4648	0.3008
cross-encoder	0.6637	0.2448	0.1657	0.2461

3. Impact of Data Augmentation Strategy

Table 6 describes each stage of the data augmentation process aimed at improving the performance of the passage retrieval model. Initially, the model was trained on a dataset consisting only of positive samples (question-passage pairs that are semantically similar). Negative samples were randomly selected from unrelated passages within the same batch, as reflected in the multiple negative ranking loss function. The model was then further improved by training on a dataset comprising both positive (initial) and "soft negative" samples, which are predictions that the BM25+ model got wrong. These "soft negative" samples had lexical similarity to the question but differed significantly in semantics. The

results show a 1% increase in Recall, an 8% increase in MAP, and a 7% increase in NDCG.

Table 6. Evaluate the Impact of the Data Augmentation Strategy with k=100 and k = 5

k	Strategy	Recall	MAP	NDCG
100	original	0.8443	0.3501	0.4803
	soft negative sampling	0.8519	0.4325	0.5489
	hard negative sampling	0.8781	0.5622	0.6519
5	original	0.5225	0.3729	0.4196
	soft negative sampling	0.5899	0.4479	0.4928
	hard negative sampling	0.6907	0.5819	0.6181

The model continued to be trained on "hard negative" data, which consisted of examples that the previous model misclassified. These were passages that had semantic similarity to the question but were not actually relevant, requiring the model to learn to reduce their similarity. The results show a 3% increase in Recall, with MAP and NDCG increasing by 13% and 11%, respectively.

For k = 100, Recall does not change significantly, but MAP and NDCG, which are ranking metrics, increase significantly, indicating that relevant passages are being pushed higher in the rankings compared to the original model. For k = 5, we observe a noticeable change in Recall, indicating that the number of relevant passages in the top 5 results of the retrieval model increases significantly through each stage of the data augmentation strategy.

The data augmentation strategy improved passage retrieval performance significantly, streamlining operations and enhancing productivity. We're excited to optimize it further and explore its potential to innovate and improve our search capabilities.

4. Evaluating the Performance of the Public Administrative Services QA System

In Table 7, you can see that the public administrative services system for answering questions was divided into two blocks. The first block, called BM25+, looked for relevant passages based on lexicon and sorted them by semantics. The second block, called SimCSE, first searched for passages based on semantics and then re-ranked them using BM25+. The system aimed for each block to cover different relevant passages to achieve the highest search performance when combined in the Ensemble Model.

In this comparison, we evaluated our system against a Vietnamese legal text information retrieval model in [13] that utilizes Sentence BERT, Condensor, and coCondensor models, in conjunction with Vietnamese language models such as PhoBERT and ViBERT. Our model outperformed Pham et al.'s model [13] in all evaluation metrics. This suggests that the Pham et al.'s model may not have been trained on a large dataset and may not have effectively utilized the advantages of passage re-ranking.

In the BM25+ block, after re-ranking, the model significantly outperforms using BM25+ alone. Specifically, the model was compared using both strategies (1) and (2) described in equations (4) and (5). It can be seen that strategy (1) yields better results in F2, MAP, and NDCG, allowing us to choose it for score aggregation.

In the SimCSE block, since the model retrieved a small top-k (k = 20), the re-ranking model only marginally improved the scores. However, it still demonstrated the effectiveness of the re-ranking model. Here, the model was also compared using two similar strategies, and strategy (2) yielded better results across all metrics, making it the choice for score aggregation.

The combined model gathered the best passages from both blocks and reorganized them. With a blend of lexical-based and semantic-based retrieval, the model significantly surpassed the individual component models. The test results demonstrated that the combined model performed better in all metrics, including Recall, MAP, and NDCG. This test partially evaluates the potential of the suggested government service QA system.

Regarding retrieval time for a single question, the SimCSE block took approximately 6.4 seconds, while the BM25+ block took approximately 9.5 seconds to provide question results. The combined model's retrieval time depended on the retrieval times of the component blocks and took approximately 15 seconds to deliver results to users.

In Table 8, we demonstrate the effectiveness of the top 5 passages in the public administrative services QA system compared to each retrieval block component. We evaluated each block with the best retrieval and ranking models. The BM25+ block with cosine ranking and weight calculation method (1) achieved a Recall of 0.7006. However, the SimCSE block with BM25+ ranking and weight calculation method (2) outperformed it with a Recall of 0.7602, which was close to the results of the retrieval model with k = 20. The combined model, which synthesizes two blocks with different functions (BM25+ for lexical-based search and SimCSE for semantic-based search) and cross-encoder score calculation, achieved the best results with a Recall of 0.7666. The other metrics, including F2 at 0.4955, MAP at 0.6513, and NDCG at 0.6918, were also high. However, there is still room for improvement, particularly in optimizing the re-ranking process. Our future research will focus on refining the retrieval and re-ranking approach to provide users with the most accurate answers in the QA system.

6 Conclusions

In this paper, we have introduced our approach to developing a QA system for Vietnamese public administrative services. We employ both lexical-based and semantic-based retrieval models and integrate them to create the final model. Our research shows that the system has achieved significant

Table 7. Evaluate the Performance of the Public Administrative Services QA system

Model		Recall	F2	MAP	NDCG	time
Pham et al.'s [13]		0.6265	0.1620	0.3044	0.3954	-
BM25+	Retriever: BM25+	0.6626	0.1799	0.0939	0.2929	-
	Ranker: cosine similarity (1)	0.8405	0.2344	0.5877	0.6630	9.46
	Ranker: cosine similarity (2)	0.8414	0.2317	0.5393	0.6280	9.10
SimCSE	Retriever: SimCSE	0.7968	0.2228	0.5706	0.6405	-
	Ranker: BM25+ score (1)	0.8362	0.2414	0.5609	0.6567	6.39
	Ranker: BM25+ score (2)	0.8653	0.2431	0.6264	0.7040	6.31
Ensemble	Our QA System	0.8895	0.1902	0.6281	0.7110	14.97

Table 8. Performance Evaluation of the System in Retrieving the Top 5 Passages

Model	Recall	F2	MAP	NDCG
BM25+ w/ BE	0.7006	0.4590	0.5890	0.6273
SimCSE w/ BM25+	0.7602	0.4949	0.6435	0.6845
Ensemble w/ CE	0.7666	0.4955	0.6513	0.6918

improvements in retrieving information from Vietnamese legal texts. It outperforms other models and demonstrates the potential of combining lexical-based and semantic-based retrieval methods. While there is still room for improvement, the system's accuracy in providing responses is promising for future advancements, emphasizing the importance of multi-stage information retrieval in this field.

To overcome the current limitations of our approach, future work could consider using multi-embedding techniques. By incorporating various types of embeddings, we can enhance the system's ability to capture a wider range of textual features, both in terms of lexicon and meaning.

References

[1] Arian Askari and Suzan Verberne. 2021. Combining Lexical and Neural Retrieval with Longformer-based Summarization for Effective Case Law Retrieval.. In *DESIRES*. 162–170.

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

[3] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821* (2021).

[4] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

[5] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[6] Phi Manh Kien, Ha-Thanh Nguyen, Ngo Xuan Bach, Vu Tran, Minh Le Nguyen, and Tu Minh Phuong. 2020. Answering Legal Questions by Learning Neural Attentive Text Representation. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 988–998. https://doi.org/10.18653/v1/2020.coling-main.86

[7] Raghavan P. Manning, C.D. and H. Schutze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge.

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

[9] Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. PhoBERT: Pre-trained language models for Vietnamese. *arXiv preprint arXiv:2003.00744* (2020).

[10] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24, 4 (2016), 694–707.

[11] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. 2017. Deeprank: A new deep architecture for relevance ranking in information retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 257–266.

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

[13] Nhat-Minh Pham, Ha-Thanh Nguyen, and Trong-Hop Do. 2022. Multi-stage Information Retrieval for Vietnamese Legal Texts. *arXiv preprint arXiv:2209.14494* (2022).

[14] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[15] Stephen Robertson, Hugo Zaragoza, and Michael Taylor. 2004. Simple BM25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. 42–49.

[16] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24, 5 (1988), 513–523.

[17] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*. 101–110.

[18] Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Applying BERT to document retrieval with birch. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*. 19–24.