NTU IR Final Project Report

Team_3

r09922A25 廖哲偉
r12922218 吳文心
r12922171 梁軒豪

2024 Fall

1 Introduction

The objective of the final project is to develop an advanced retrieval system capable of addressing real-life problems by providing relevant legal articles from Taiwan's legislative framework. The training data set comprises 1,175 questions, each paired with a varying number of associated legal articles. The system has to answer the 200 test questions by retrieving the relevant articles from a corpus of 200 codices.

2 Method

2.1 Data pre-processing

2.1.1 Traditional Approach

For the data pre-processing of traditional approach, we directly concatenate each article name (`law_name`) with its associated documents, including the law content and provided train data.

The process begins by unzipping the law files and extracting the law names (e.g., "XX法第XX條之X") along with their corresponding content. These are stored in a dictionary, where the keys represent the law names, and the values are their content.

Next, we process the training data provided by TA. For each entry, the `label` is used to identify the relevant provisions. The `title` and `question` of the entry are concatenated and appended to the content of the corresponding provisions.

After preparing all the necessary data, the Chinese word segmentation tool, Jieba, is used to tokenize the content. Finally, the law names are concatenated with their tokenized content to form the input for subsequent sparse retrieval tasks.

We also experimented with another data pre-processing approach, which involved concatenating the law content and training data separately with the law names. However, we did not find an effective normalization method for the retrieved results from this approach.

2.1.2 LLM-based approach

Each element of the test answers is represented as a combination of a law name and an article number. Accordingly, individual articles extracted from the unzipped law files are stored as a dictionary object with three keys: `law_name`, `article_number`, and `content` and are saved in a `jsonl` file.

The processed articles were transformed into LangChain `Document` objects while the `page_content` containing `law_name`, `article_number`, and `content` and also put `law_name`, `article_number` in the their `metadata` to facilitate retrieving theses critical components.

2.2 Traditional Approach

We experimented with a series of traditional approaches, including query expansion (WordNet), sparse retrieval (BM25) and dense retrieval (LegalBERT).

We first applied the Chinese word segmentation tool, Jieba, to tokenize the query for improved processing of Chinese text. Subsequently, WordNet is employed to identify synonyms of the query terms, enabling query expansion. An initial retrieval was then performed using BM25, limiting the result set to 200 documents. Documents with relevance score lower than the average were filtered out. Finally, we used LegalBERT to compute the embeddings for the query and the documents, evaluating their relevance using consine similarity. The the relevance score were then used to rerank the initial retrieved document set. Extract the law, with the top 20 documents output as the final result.

2.3 LLM assisted Zero-Shot Retrieval

The contextual understanding capability of a large language model (LLM) is well recognized. We first tried zero-shot prompting TAME (TAiwan Mixtures of Experts), a `Llma-3-70B` model fine-tuned on a large corpus of high quality multidisciplinary traditional mandarin data[1, 2]. However, the preliminary results are visibly not satisfied.

Given our limited experience with fine-tuning an LLM at this scale and constraints in computational resources, we adopted an alternative approach. Leveraging the LangChain package, we harnessed the capabilities of LLMs by embedding the legal codices and indexing them in a vector store using the Facebook AI Similarity Search (`FAISS`) library[3]. For our prototype baseline, we utilized the built-in `similarity_search()` function to perform zero-shot retrieval, providing a foundation for iterative refinement and optimization of our methodology. Our zero-shot retrieval is illustrated in Figure 1.

2.4 Refinement of LLM-Based Approach

2.4.1 Training Reranker

To enhance retrieval performance, we sought to fine-tune a reranker using the training data associated with our zero-shot retriever. Positive samples were constructed by pairing the questions in the training set with each of its corresponding articles respectively. Leveraging the zero-shot retriever, we retrieved the top 10 articles per query, filtered out the correct responses, and designated the remaining articles as negative examples. These curated examples were then used to train a

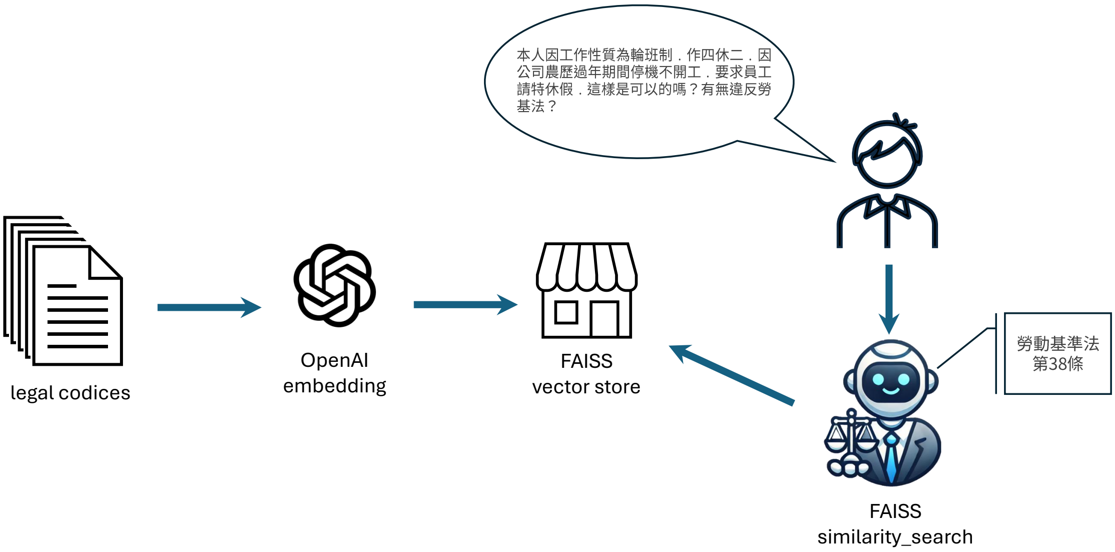
本人因工作性質為輪班制．作四休二．因公司農歷過年期間停機不開工．要求員工請特休假．這樣是可以的嗎？有無違反勞基法？
legal codices
OpenAI
embedding
FAISS
vector store
勞動基準法
第38條
FAISS
similarity_search

Figure 1: Illustration of the zero-short approach

new embedding model, aimed at refining the reranking process by minimizing the `CosineSimilarityLoss`. It is illustrated in Figure 2.

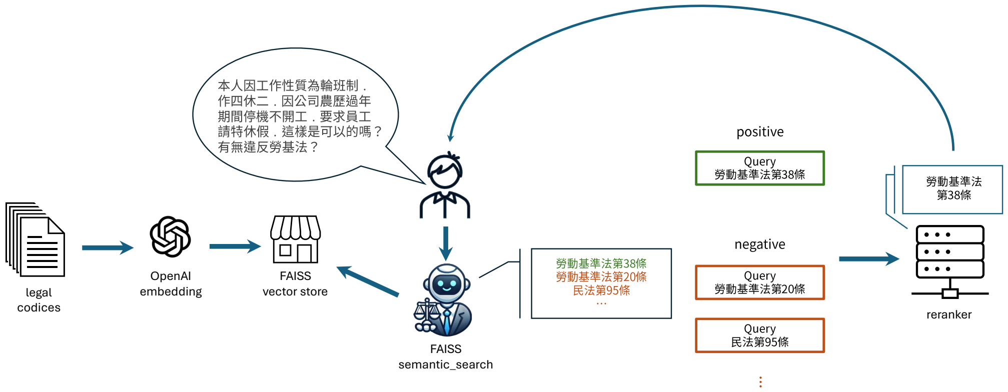

Figure 2: Using positive and negative examples training reranker

2.4.2 Testing Embedding Alternative

We tested and compared two different embedding models, `text-embedding-ada-002` and `text-embedding-3-large` in our zero-shot retrieval setting to evaluate the effects of embedding models in the present task.

3 Experiment & Discussion

3.1 Traditional Approach

Two experiments were conducted for traditional approaches.

- The first experiment focused on pure sparse retrieval using BM25. We retrieved the top-10 documents, normalized their scores, and retained only those with scores exceeding 0.8, with the threshold determined based on the results.

- The second experiment combined query expansion, sparse retrieval and dense retrieval to evaluate the effectiveness of an integrated approach, as described in 2.1.1.

3.2 LLM-assisted Approach

3.2.1 Zero-shot retrieval

The legal codices were split by LangChain's `RecursiveCharacterTextSplitter` with chunk size 300 and chunk overlap 50. The document chunks were embedded using OpenAI's `text-embedding-3-large`. The generated vectors were indexed and managed using `FAISS` library. Zero-shot semantic search was conducted utilizing the library's built-in `similarity_search()` function and the top-2 to top-5 results were submissions to Kaggle for evaluation. For embedding alternative evaluation, we simply replaced `text-embedding-3-large` with `text-embedding-ada-002` and retained the rest of the configurations.

3.2.2 Training Reranker

In the initial retrieval phase, we experimented with two different retrievers. The first was the `similarity_search()` function from the `FAISS` library, previously utilized in our zero-shot retrieval. The second was a customized `semantic_search()` function. This approach performed an initial search for $k \times 2$ results using an ensemble retriever that combined sparse (BM25) and dense (`FAISS.as_retriever()`) retrieval methods, weighted at [0.3, 0.7], respectively. The retrieved candidates were then reranked using the `CrossEncoder` from the `Sentence-Transformers` library (pre-trained model: `ms-marco-MiniLM-L-6-v2`), producing the final top-k results.

Training examples were constructed based on the results of the initial retrieval, with positive examples labeled as 1 and negative examples as 0. We fine-tuned the `paraphrase-multilingual-mpnet-base-v2` model from `Sentence-Transformers` as our reranker. The model was trained and evaluated over 5 and 10 epochs. The final submission consisted of the top-3 results, as our zero-shot experiments demonstrated better performance under this configuration on public score.

3.3 Results & Discussion

As shown in Table 1, the performance of pure sparse retrieval (BM25) is 1.5 to 2 times better than the integrated approach, which combines query expansion, sparse retrieval (BM25) and dense retrieval (LegalBERT). Upon examining the outputs of both approaches, we observed that the both initial retrieval and re-ranking stages in the integrated approach are erroneous, leading to error propagation and further compounding inaccuracies. These accumulated errors within the retriever components ultimately result in the integrated approach performing worse than pure sparse retrieval (BM25).

Table 1: Comparison of different methods on Public F1 scores

	Private F1 scores	Public F1 scores
Pure BM25	0.13062	0.12429
integrated approach	0.06392	0.08586

As shown in Table 2, the best public score of LLM-based approach is the top-3 zero-shot retrieval with F1 score 0.18062. However, on the contrary, the top-3 zero-shot retrieval had the lowest score among the top-k comparison in private score and the best private score is the top-2 zero-shot retrieval with F1 score 0.19236. Interestingly, all the LLM-based approaches scored higher on the private sector(Table 3).

Table 2: Comparison of different methods on Public F1 scores

	Zero-Shot Retrieval	Train Reranker (5) Built-In Search	Train Reranker (5) Custom Search	Train Reranker (10) Custom Search	Alternative Embedding Built-In Search
top-2	0.15175	0.06589	0.10130	0.08837	0.10819
top-3	0.18062				
top-4	0.17897				
top-5	0.16750				

Table 3: Comparison of different methods on Private F1 scores

	Zero-Shot Retrieval	Train Reranker (5) Built-In Search	Train Reranker (5) Custom Search	Train Reranker (10) Custom Search	Alternative Embedding Built-In Search
top-2	0.19236	0.05933	0.10708	0.09948	0.11744
top-3	0.18055				
top-4	0.18535				
top-5	0.18324				

Our zero-shot retrieval outperforms traditional BM25 and dense retrieval. It demonstrates the semantic richness in LLM's embeddings could be easily harnessed to capture the nuanced contextual relationships between queries and legal articles without additional training and is more effective in retrieval than the traditional methods which rely on lexical matching (BM25) or pre-trained vector representations (dense retrieval).

The reranker manifested a better performance with our custom-built retriever than the built-in `similarity_search()` function. It is probably due to more the embeddings of our reranker are more aligned with custom-built retriever than the OpenAI embeddings.

Theoretically the design and training for the reranker should refine the retrieval results. However, our trained reranker did not perform as anticipated. Whether integrated with the `similarity_search()` function in the `FAISS` library or our custom-built retriever, it failed to yield any performance improvements. In fact, it exhibited significantly poorer results on both public and private test cases. This indicates that simple cosine similarity could not bring closer the embeddings generated by the models we adopt in our experiment. A more delicate ranking function or method discriminating the positive and negative examples is needed to overcome the embedding mismatch.

Our zero-shot retrieval using `text-embedding-3-large` outperformed `text-embedding-ada-002`, underscoring the critical role of embedding model capabilities in retrieval performance. While the smaller `text-embedding-ada-002`, released in 2022, is optimized for computational efficiency, the more advanced `text-embedding-3-large` (2024) demonstrates a significantly enhanced ability to capture the essential semantics of task-specific literature. This comparison highlights the substantial gains achievable with state-of-the-art embedding models in specialized retrieval tasks.

Compared with LLM-based approaches, our traditional approaches fail to demonstrate competitiveness. However, this might be due to the specific traditional approaches we selected, which did not perform well. Further experiments could be conducted to explore whether stronger traditional approaches exist.

The unsatisfied performance in our LLM-based approach could be owing to the limited information in the short context of the questions in train and test data. To expand the training context we could try Collecting real-world legal documents but it is time consuming, labor burdened and needs expertise guidance. We could also take advantages of LLMs'powerful capability of contextual understanding and reasoning by prompting the LLMs to do multi-hop reasoning and collect the responses to expand the training corpus. With sufficient query related information we are able to testify strategies other than pure retrieval such as article ID generation such as what we learned in our paper survey[4].

4 Conclusion

We once heard of cases where traditional methods outperformed AI models that had been extensively trained. With this project, we wanted to explore whether this legend holds true — can Large Language Models (LLMs) really be so easily surpassed? Based on our results, the traditional approach combination we selected was defeated. It may be due to our design upon our limited experience are not suitable for the current task. Perhaps in the future, we could experiment with other combinations, aiming to use classic methods to triumph over modern ones. Our experiments confirm that current commercial and open-source LLMs are not yet universally applicable in specialized domains. Beyond task-specific fine-tuning, enhancing the generalizability of LLMs for domain-specific applications remains a critical direction for future research.

5 Contribution

- 廖哲偉: manuscript editing, coding for ”LLM-assisted zero-shot” and ”Reranker Training”.
- 吳文心: manuscript editing and coding for ”Traditional Approaches”.
- 梁軒豪： manuscript editing and experiments for ”Alternative Embedding”.
- Gitbub (LLM-based): https://github.com/Eugene-Liao/IRIE2024_group3
- Github (traditional): https://github.com/luckyjp6/IRIE2024_group3_traditional

References

[1] Yen-Ting Lin and Yun-Nung Chen. Taiwan LLM: bridging the linguistic divide with a culturally aligned language model. *CoRR*, abs/2311.17487, 2023.

[2] Po-Heng Chen, Sijia Cheng, Wei-Lin Chen, Yen-Ting Lin, and Yun-Nung Chen. Measuring taiwanese mandarin language understanding. *CoRR*, abs/2403.20180, 2024.

[3] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library, 2024.

[4] Weicong Qin, Zelin Cao, Weijie Yu, Zihua Si, Sirui Chen, and Jun Xu. Explicitly integrating judgment prediction with legal document retrieval: A law-guided generative approach. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’24, page 2210–2220, New York, NY, USA, 2024. Association for Computing Machinery.