SCaLe-QU: Sri Lankan case law embeddings for legal QA.

JAYAWARDENA, L., WIRATUNGA, N., ABEYRATNE, R., MARTIN, K., NKISI-ORJI, I. and WEERASINGHE, R.

2024

SCaLe-QA: Sri lankan Case Law Embeddings for Legal QA*

Lasal Jayawardena[1,2,†], Nirmalie Wiratunga[1,†], Ramitha Abeyratne[1], Kyle Martin[1], Ikechukwu Nkisi-Orji[1] and Ruvan Weerasinghe[2]

[1]Robert Gordon University, Aberdeen, United Kingdom
[2]Informatics Institute of Technology, Sri Lanka

Abstract

SCaLe-QA is a foundational system developed for Sri Lankan Legal Question Answering (LQA) by leveraging domain-specific embeddings derived from Supreme Court cases. The system is tailored to capture the unique linguistic and structural characteristics of Sri Lankan law through fine-tuned embeddings. While Case-Based Reasoning (CBR) will be integrated into the question-answering framework, it is primarily set for future development and evaluation. Currently, SCaLe-QA employs semantic chunking, tokenization, and BM25-based ranking to generate context-driven triplets from unlabeled corpora. In addition, an angle-optimised contrastive learning framework is applied to enhance retrieval accuracy. Preliminary results indicate promise, establishing SCaLe-QA as a significant step toward robust AI applications in the Sri Lankan legal domain.

Keywords

text embeddings, legal AI, RAG, CBR, legal question answering, retrieval

1. Introduction

The increasing complexity of legal texts, particularly within the Sri Lankan judicial system, poses significant challenges for the development of effective Legal Question Answering (LQA) systems. Legal documents are characterised by specialised vocabulary, intricate syntactic structures, and context-dependent semantics, making the task of automated question answering both demanding and essential. The ability to accurately and efficiently answer legal questions is critical, as it enhances access to legal information, supports legal research, and facilitates informed decision-making processes for legal professionals, researchers, and the general public [1].

Recent advancements in Natural Language Processing (NLP) and Machine Learning (ML) have spurred the development of sophisticated LQA systems that leverage deep learning techniques to process and understand legal texts effectively. These advancements have been well-documented, highlighting the importance of domain-specific datasets and models tailored to the unique characteristics of legal texts [2]. The development of domain-specific embeddings is crucial for enhancing the performance of LQA systems. [3] emphasise the necessity of sentence embeddings tailored to the legal domain, given the specialised vocabulary and unique semantic interpretations found in legal texts

Retrieval Augmented Generation (RAG) has emerged as a powerful approach in enhancing the performance and reliability of LQA systems. RAG combines the strengths of retrieval-based methods with generative models, allowing for more accurate and contextually relevant responses to legal queries. This approach involves retrieving relevant documents or passages from a large corpus of legal texts and then using this retrieved information to augment the generation process [4]. Furthermore, the integration of case-based reasoning (CBR) systems with specialised embeddings as investigated in [5] has been shown to improve the performance of such systems compared to typical information retrieval techniques [6] in the legal context.

SICSA REALLM Workshop 2024

[†] Corresponding author.

l.jayawardena@rgu.ac.uk (L. Jayawardena); n.wiratunga@rgu.ac.uk (N. Wiratunga); r.abeyratne@rgu.ac.uk (R. Abeyratne); k.martin3@rgu.ac.uk (K. Martin); i.nkisi-orji@rgu.ac.uk (I. Nkisi-Orji); ruvan.w@iit.ac.lk (R. Weerasinghe)
0009-0002-7100-6015 (L. Jayawardena); 0000-0003-4040-2496 (N. Wiratunga); 0009-0008-5582-8311 (R. Abeyratne); 0000-0003-0941-3111 (K. Martin); 0000-0001-9734-9978 (I. Nkisi-Orji); 0000-0002-1392-7791 (R. Weerasinghe)

In this contribution, we explore the impact of tuning domain-specific embeddings for legal contexts, focusing on how these embeddings can be utilised to transform triplets from unprocessed legal documents into structured representations. Our aim is to improve retrieval accuracy within Retrieval Augmented Generation (RAG) systems, which is crucial for the effectiveness of Legal Question Answering (LQA) systems. By enhancing these embeddings, we aim to significantly boost the performance and reliability of LQA systems tailored to the legal domain, particularly in the Sri Lankan legal space.

2. Finetuning Methodology

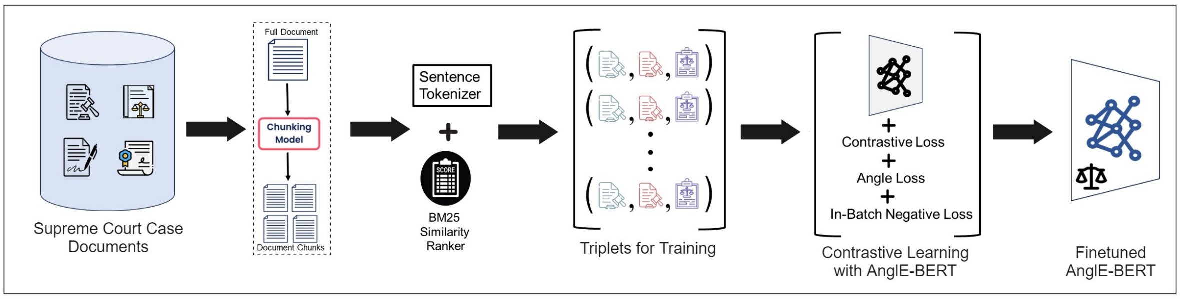


Figure 1: Workflow for Finetuning Process

2.1. Data Source

The dataset used for this study consists of the officially reported Supreme Court judgments from Sri Lanka, spanning from 2009 to 2024. In total, 1541 documents were scraped from the official Supreme Court website[1]. The documents covered a wide area of the Sri Lankan legal context such as:

- **Appeals**: Includes general appeals such as standard appeals, civil appeals, and specific appeals related to legal provisions or leave to appeal applications.
- **Applications**: Divided into constitutional applications, fundamental rights applications, and other various legal procedure applications.
- **Civil Cases**: Encompasses general civil matters, including commercial and procedural cases, as well as specific civil cases like divorce, testamentary, and land disputes.
- **Criminal Cases**: All cases related to criminal law.
- **Constitutional Matters**: Includes cases dealing with constitutional law, references, and specific declarations under the Constitution.
- **Commercial High Court Cases**: Covers commercial disputes handled by the Commercial High Court.
- **Other**: A variety of other case types, including contempt of court, election-related matters, and writs of certiorari and prohibition.

Most of these documents were directly text-parsable, while others required additional processing. The non-parsable documents were fed into an OCR model using Adobe [2], and the resulting text was manually corrected to ensure accurate extraction. This cleaned and corrected dataset served as the primary data source for the subsequent stages of this research.

[1] https://www.supremecourt.lk/

[2] https://www.adobe.com/in/acrobat/online/ocr-pdf.html

2.2. Document to Sentence Segmentation

In this work, we followed the document chunking strategies from the Open Australian Legal Question-Answering (ALQA) dataset[3], which is based on legal question answering for Australian law. The document chunking strategy served as a preprocessing step, assisting the sentence tokenizer in breaking segments into sentences. More importantly, it established the foundation for creating the testing framework necessary for the embedding fine-tuning process, which will be discussed later. We employed the semantic chunking model provided by the SemChunk library[4], which was particularly useful for handling legal documents. We maintained consistency with the Australian Legal QA dataset by setting the chunk size to 384 tokens, as tokenised according to the tiktoken tokenizer for GPT-4[5]. Upon manual inspection, the chunk size was deemed appropriate, and it did not negatively impact sentence integrity.

To gain insights into the dataset, after preprocessing, we performed some sentence-level visualisation as shown in Figure 2. The visualisation highlights that the dataset includes very long documents, with some exceeding 1500 sentences. These visualisations emphasise the significant variability in both the size of documents and the length of sentences, which presents unique challenges for processing and analysing legal texts.

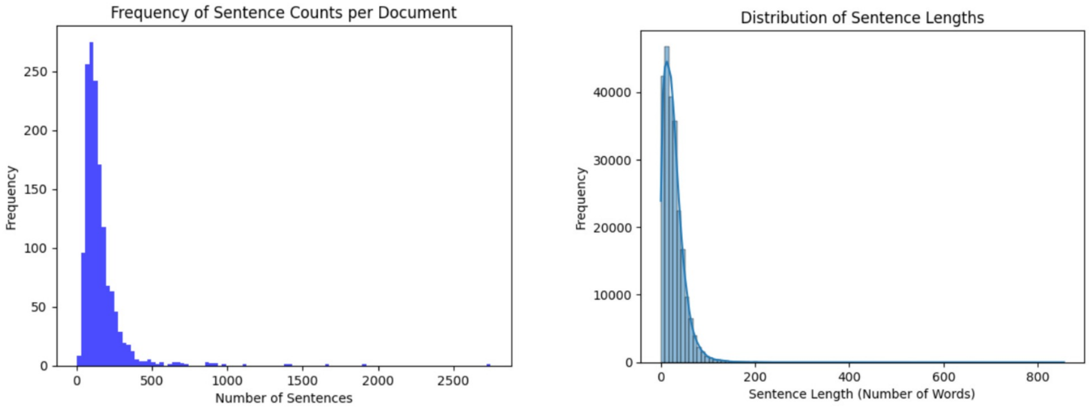


Figure 2: Sentence Level Visualisation of the Dataset

2.3. Triplet Creation

Each sentence after sentence segmentation is treated as individual units for subsequent processing to create triplets. The BM25 algorithm, a robust ranking function used in information retrieval [7], is applied to rank these sentences based on their relevance to each other within each case document. This method leverages the lexical knowledge embedded in the text as weak supervision, which has been shown to be a strong baseline for fine-tuning text embeddings, as discussed in [8]. The BM25 algorithm was implemented using the `rank_bm25` library.[6], and the algorithmic breakdown for the preprocessing will be shown below.

Given a sentence S in a document D, the BM25 score for another sentence S' in the same document is calculated using the following adaptation of the BM25 formula:

$$\text{score}(S', D) = \sum_{i=1}^{n} \text{IDF}(t_i) \cdot \frac{f(t_i, S') \cdot (k_1 + 1)}{f(t_i, S') + k_1 \cdot \left(1 - b + b \cdot \frac{|S'|}{\text{avgdl}}\right)} \tag{1}$$

[3] https://huggingface.co/datasets/umarbutler/open-australian-legal-qa

[4] https://github.com/umarbutler/semchunk

[5] https://github.com/openai/tiktoken

[6] https://github.com/dorianbrown/rank_bm25

t_i are the terms in sentence S, whilst $f(t_i, S')$ is the frequency of term t_i in sentence S', where $|S'|$ is the length of the sentence and avgdl is the average length of sentences in the document. k_1 and b are hyperparameters (typically $k_1 = 1.5$ and $b = 0.75$), and $\mathrm{IDF}(W_i)$ is the Inverse Document Frequency of term t_i.

In this approach, for each sentence S in a document, we rank all other sentences S' in the same document using the BM25 algorithm. The most similar sentence (i.e., the one with the highest BM25 score) is selected as the positive or like sample X_L. To select a negative sample or unlike sample X_U, we randomly choose one out of the top five least similar sentences (i.e., those with the lowest BM25 scores). This strategy helps avoid the issue of a very dissimilar sentence being repeatedly included in the triplets, which could make the triplets less informative for training. The number of least unlike sentences considered is obtained through empirical experimentation.

The sentences identified through this process are then denoted as follows:

- X_a: The anchor sentence, which is the sentence we are evaluating within the document.
- X_L: The positive sample, the sentence most like the anchor.
- X_U: The negative sample, a sentence sampled from a pool of N unlike (least similar to) the anchor.

These notations will be used in the subsequent embedding fine-tuning process.

2.4. Embedding Finetuning

A contrastive compound loss function was designed to optimise the distances within the embedding space between triplets (X_a, X_L, X_U), where X_a is the anchor sample, X_L is the positive sample (similar or like the anchor), and X_U is the negative sample (dissimilar or unlike the anchor). The compound loss function is inspired by the methodologies in [9], where angle optimisation is combined with the cosine objective from [10]. This approach differentiates itself from existing contrastive learning methods discussed in [11, 12]. Specifically, the loss function combines three key objectives:

$$L = w_1\ L_c(S_U, S_L) + w_2 \left(-\sum_b \sum_m \log \left(\frac{\exp\left(\frac{S_L}{\tau}\right)}{\sum_j \exp\left(\frac{S_U}{\tau}\right)} \right) \right) + w_3\ L_c(S'_U, S'_L) \tag{2}$$

- The first term $L_c(S_U, S_L)$, weighted by w_1, uses the standard cosine similarities between the anchor and positive (or like) instance, $S_L = cos(X_a, X_L)$, and the anchor and negative (or unlike) instance, $S_U = cos(X_a, X_U)$. Here the general contrastive loss function is defined as: $L_c(S_U, S_L) = \log\left(1 + \sum \exp\left(\frac{S_U - S_L}{\tau}\right)\right)$. This encourages the model to ensure that positive pairs have higher similarity than negative pairs.
- The second term, weighted by w_2, applies in-batch negative sampling, comparing the anchor-positive pairs within a batch and treating the remaining pairs as negatives. Again using cosine similarities to arrive at S_L and S_U respectively.
- The third term weighted by w_3, is similar to the first but uses a refined similarity metric, S', where the embeddings of X_a, X_L and X_U are split in half. The similarities S'_L and S'_U are calculated by averaging the cosine similarity over the two halves of the embeddings.

Here τ: is a temperature scaling parameter that controls the sensitivity of the model to differences in similarity scores. Lower values of τ increase the sharpness, while higher values soften the distribution. Parameters m and b represent the batch size and number of batches, respectively.

The loss function was applied to fine-tune AnglE-BERT, a model specifically designed for angle-optimised contrastive learning, as introduced by [9]. AnglE-BERT was initially trained with the angle optimisation mechanism that adjusts the angles between embeddings in the latent space, which is particularly effective for distinguishing between similar and dissimilar pairs of sentences. Each fine-tuning run for AnglE-BERT was carried out using the triplets formed from the BM25-ranked sentences, executed on an NVIDIA RTX A100. The training phases were conducted with a batch size of 32, over

10 epochs, spanning around 14 GPU hours, during which the model was exposed to just over 230,000 triplets. The extensive training was aimed at refining the model's ability to learn the nuances of the legal texts seen in the triplets.

2.5. Model Training Dualities

As introduced in AnglE-BERT [9], two distinct flavours of embeddings were fine-tuned using contrastive learning, each optimised for different retrieval purposes. In this work, these embeddings will be categorised as **Intra-Embeddings** and **Inter-Embeddings**, which are designed to serve specific retrieval and matching tasks.

- **Intra-Embeddings (f(Q))**: These embeddings are optimised for attribute matching within the same type of content, such as comparing questions with questions. This type of embedding is particularly useful for semantic textual similarity tasks, where the focus is on finding sentences with closely related meanings, even if they are phrased differently.
- **Inter-Embeddings (g(Q))**: These embeddings are designed for broader information retrieval scenarios, where the goal is to match content across different types of attributes, such as comparing a query with relevant passages, entities, or supporting texts. This allows for more flexible retrieval tasks, where the query might need to be matched with various types of contextual information.

The fine-tuning process outline before for embeddings was conducted separately on both the intra and inter embeddings, ensuring that each representation was optimised for its respective task. This dual fine-tuning approach allows the model to perform well across both precise attribute matching and broader retrieval tasks. Conceptually, this approach is akin to a form of query rewriting[13], where each type of embedding acts as a different representation of the input query, tailored to optimise retrieval for specific purposes.

Table 1 provides an example illustrating the difference between intra-embedding and inter-embedding for a sentence used in the training process.

Table 1
Comparison of an example sentence with and without the *Cue* text (in blue) to create inter and intra embeddings.

Embedding	Sentence
intra	*f*("The primary issue arises due to the inclusion of 'share certificate' in Gazette No. 1465/19.")
inter	*g*("Represent this sentence for searching relevant passages:" + "The primary issue arises due to the inclusion of 'share certificate' in Gazette No. 1465/19.")

These dual embeddings can form the foundation for flexible and robust retrieval systems that can handle both precise and contextually broad queries within the legal domain.

3. Evaluation

Prior to evaluating the performance of the embedding models, there are two key stages: *Casebase Creation* and *Test Set Creation*. These stages are crucial for building the necessary datasets for evaluating the retrieval models in the subsequent retrieval evaluation.

3.1. Casebase Creation

The first stage in the evaluation process involved constructing the casebase, using the scraped Supreme Court documents. As illustrated in Figure 3, this process involved multiple steps of attribute extraction from legal documents. The documents were segmented into manageable chunks of 384 tokens, and the following key attributes were extracted using the *GPT-4o mini*[7] model:

[7] https://platform.openai.com/docs/models/gpt-4o-mini

- **Court Details**: Including court name, case number, case year, and case type.
- **Parties Involved**: Identifying plaintiffs, defendants, and their respective roles.
- **Questions of Law**: Legal questions presented in the case, particularly those considered by the Supreme Court.
- **Case Summary**: A brief summary of the case, including judgment details, legal issues, and key findings.
- **Laws and Acts Referenced**: Listing specific laws or legal acts cited during the judgment.
- **Judgment Details**: Including the decision outcome, key findings, and legal conclusions reached by the court.

Once extracted, these attributes were compiled into structured JSON records, providing a metadata-rich view of each case. This enabled efficient question-answer generation by providing the necessary context of the full legal case in a condensed structure, for each document chunk, rather than requiring the system to process the entire case document.

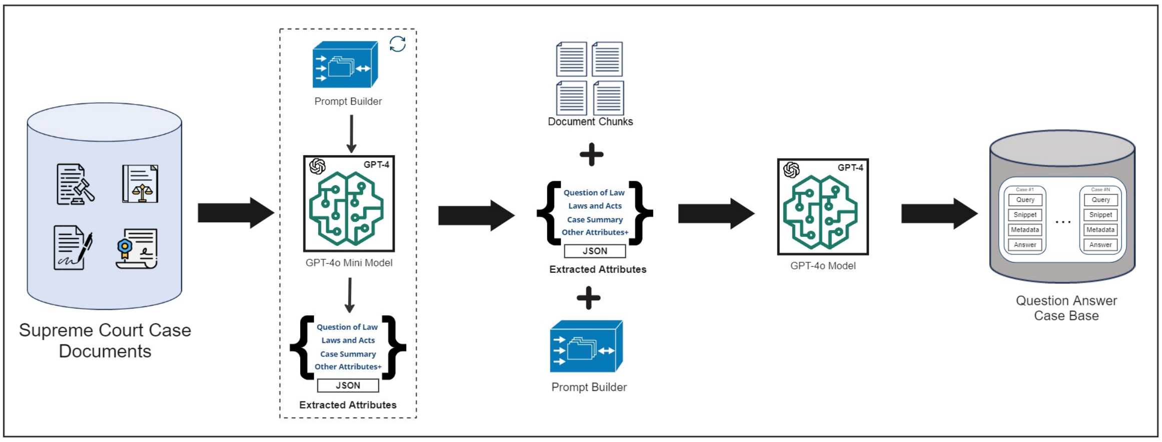


Figure 3: Casebase Creation Workflow

3.2. Test Set Creation

The second stage involved creating a robust test set from the casebase. This process, depicted in Figure 4, starts by filtering the entire casebase to group cases that share overlapping legal references, such as common laws or acts. Metadata filtering was applied in such a way that for each case in the casebase, other cases with overlapping laws were identified. These similar cases were ranked based on how closely related they were, but ensuring that they originated from different legal documents, thus enhancing diversity in the source material.

The cases selected through this filtering process were further refined by applying a cosine similarity ranking mechanism, using the Ada-002 embedding model [14] to identify closely related case pairs using the query of each case. These pairs were then used to generate a new hybrid question and answer through a prompt designed for the GPT-4o [8] LLM. The generated question was complex and required the context of both related case snippets to be answered correctly.

A human-in-the-loop system was employed to review the generated question-answer pairs. The evaluators (authors of this paper) assessed the quality of each pair, filtering out those that lacked relevance or quality. This rigorous review ensured the creation of a high-quality test set for further retrieval evaluation.

This test set contained 1000 high-quality question-answer pairs to evaluate the embedding retrieval.

[8] https://platform.openai.com/docs/models/gpt-4o

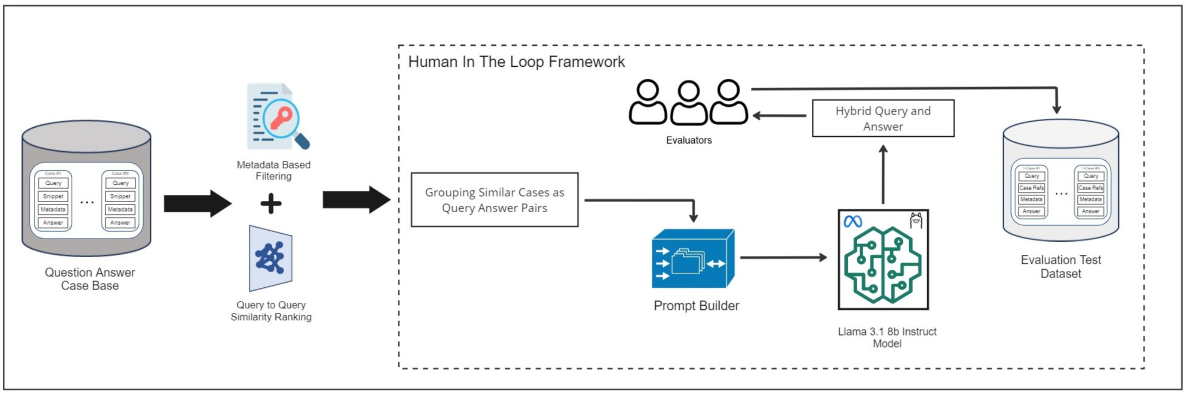


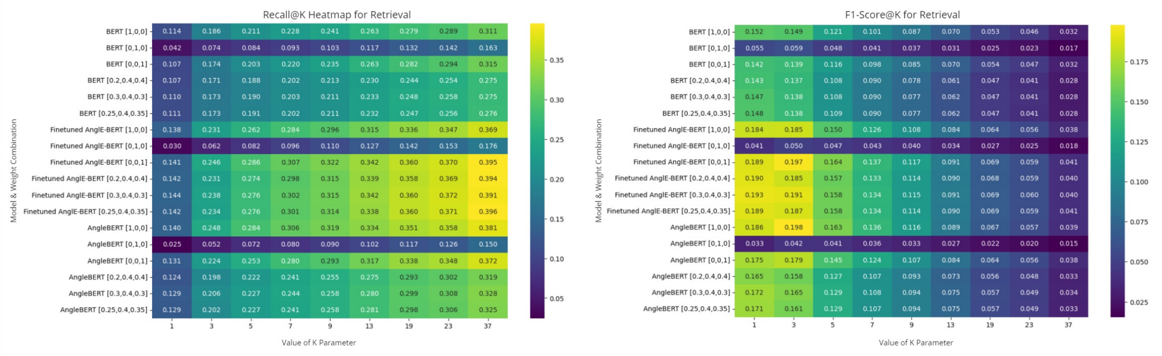
Figure 4: Test Set Creation Workflow

3.3. Retrieval Analysis

The embedding model performance was assessed using multiple *Retrieval@K* evaluations, which helps in understanding how well the model ranks and retrieves relevant information based on the hybrid test cases. The retrieval evaluation included analysing both the **F1-score@K** and **Recall@K**, which provide insights into the balance between precision and recall during the retrieval process.

To conduct this evaluation, we used k-Nearest Neighbors (k-NN) based retrieval, exploring a range of *k* values between 1 and 37. These prime values allowed us to investigate the optimal retrieval size for the legal documents used in the study.

We evaluated our fine-tuned AnglE-BERT model for both intra and inter-embeddings, comparing it against the standard BERT[15] and AnglE-BERT models[9]. Figure 5 shows the heat maps comparing Recall@K and F1-score@K for these models and their respective weight configurations.


Recall@K Heatmap for Retrieval

Model & Weight Combination	1	3	5	7	9	13	19	23	37
BERT [1,0,0]	0.114	0.186	0.211	0.228	0.241	0.263	0.279	0.289	0.311
BERT [0,1,0]	0.042	0.074	0.084	0.093	0.103	0.117	0.132	0.142	0.163
BERT [0,0,1]	0.107	0.174	0.203	0.220	0.235	0.263	0.282	0.294	0.315
BERT [0.2,0.4,0.4]	0.107	0.171	0.188	0.202	0.213	0.230	0.244	0.254	0.275
BERT [0.3,0.4,0.3]	0.110	0.173	0.190	0.203	0.211	0.233	0.248	0.258	0.275
BERT [0.25,0.4,0.35]	0.111	0.173	0.191	0.202	0.211	0.232	0.247	0.256	0.276
Finetuned AnglE-BERT [1,0,0]	0.138	0.231	0.262	0.284	0.296	0.315	0.336	0.347	0.369
Finetuned AnglE-BERT [0,1,0]	0.030	0.062	0.082	0.096	0.110	0.127	0.142	0.153	0.176
Finetuned AnglE-BERT [0,0,1]	0.141	0.246	0.286	0.307	0.322	0.342	0.360	0.370	0.395
Finetuned AnglE-BERT [0.2,0.4,0.4]	0.142	0.231	0.274	0.298	0.315	0.339	0.358	0.369	0.394
Finetuned AnglE-BERT [0.3,0.4,0.3]	0.144	0.238	0.276	0.302	0.315	0.342	0.360	0.372	0.391
Finetuned AnglE-BERT [0.25,0.4,0.35]	0.142	0.234	0.276	0.301	0.314	0.338	0.360	0.371	0.396
AngleBERT [1,0,0]	0.140	0.248	0.284	0.306	0.319	0.334	0.351	0.358	0.381
AngleBERT [0,1,0]	0.025	0.052	0.072	0.080	0.090	0.102	0.117	0.126	0.150
AngleBERT [0,0,1]	0.131	0.224	0.253	0.280	0.293	0.317	0.338	0.348	0.372
AngleBERT [0.2,0.4,0.4]	0.124	0.198	0.222	0.241	0.255	0.275	0.293	0.302	0.319
AngleBERT [0.3,0.4,0.3]	0.129	0.206	0.227	0.244	0.258	0.280	0.299	0.308	0.328
AngleBERT [0.25,0.4,0.35]	0.129	0.202	0.227	0.241	0.258	0.281	0.298	0.306	0.325

Value of K Parameter

F1-Score@K for Retrieval

Model & Weight Combination	1	3	5	7	9	13	19	23	37
BERT [1,0,0]	0.152	0.149	0.121	0.101	0.087	0.070	0.053	0.046	0.032
BERT [0,1,0]	0.055	0.059	0.048	0.041	0.037	0.031	0.025	0.023	0.017
BERT [0,0,1]	0.142	0.139	0.116	0.098	0.085	0.070	0.054	0.047	0.032
BERT [0.2,0.4,0.4]	0.143	0.137	0.108	0.090	0.078	0.061	0.047	0.041	0.028
BERT [0.3,0.4,0.3]	0.147	0.138	0.108	0.090	0.077	0.062	0.047	0.041	0.028
BERT [0.25,0.4,0.35]	0.148	0.138	0.109	0.090	0.077	0.062	0.047	0.041	0.028
Finetuned AnglE-BERT [1,0,0]	0.184	0.185	0.150	0.126	0.108	0.084	0.064	0.056	0.038
Finetuned AnglE-BERT [0,1,0]	0.041	0.050	0.047	0.043	0.040	0.034	0.027	0.025	0.018
Finetuned AnglE-BERT [0,0,1]	0.189	0.197	0.164	0.137	0.117	0.091	0.069	0.059	0.041
Finetuned AnglE-BERT [0.2,0.4,0.4]	0.190	0.185	0.157	0.133	0.114	0.090	0.068	0.059	0.040
Finetuned AnglE-BERT [0.3,0.4,0.3]	0.193	0.191	0.158	0.134	0.115	0.091	0.069	0.060	0.040
Finetuned AnglE-BERT [0.25,0.4,0.35]	0.189	0.187	0.158	0.134	0.114	0.090	0.069	0.059	0.041
AngleBERT [1,0,0]	0.186	0.198	0.163	0.136	0.116	0.089	0.067	0.057	0.039
AngleBERT [0,1,0]	0.033	0.042	0.041	0.036	0.033	0.027	0.022	0.020	0.015
AngleBERT [0,0,1]	0.175	0.179	0.145	0.124	0.107	0.084	0.064	0.056	0.038
AngleBERT [0.2,0.4,0.4]	0.165	0.158	0.127	0.107	0.093	0.073	0.056	0.048	0.033
AngleBERT [0.3,0.4,0.3]	0.172	0.165	0.129	0.108	0.094	0.075	0.057	0.049	0.034
AngleBERT [0.25,0.4,0.35]	0.171	0.161	0.129	0.107	0.094	0.075	0.057	0.049	0.033

Value of K Parameter

Figure 5: Recall and F1 Score Analysis of Retrieval@K

The results indicate that fine-tuning on AnglE-BERT improves both the Recall@K and F1-score@K across different retrieval levels. Specifically, the fine-tuned AnglE-BERT model with different weight configurations has shown a robust retrieval performance whereas AnglE-BERT has performed well in only query-to-query matching. These findings suggest that the fine-tuned model's ability to retrieve relevant legal cases was robust and well-adapted to the unique characteristics of the legal domain.

3.4. Embedding Distribution

The embedding distribution, as illustrated in Figure 6, was obtained by calculating the cosine similarity between the query (i.e., the question for each case) and its corresponding snippet or context.

For the standard BERT and AnglE-BERT models, the similarity distribution is skewed to the left. This left-skewed distribution indicates that these models classify more query-snippet pairs as having

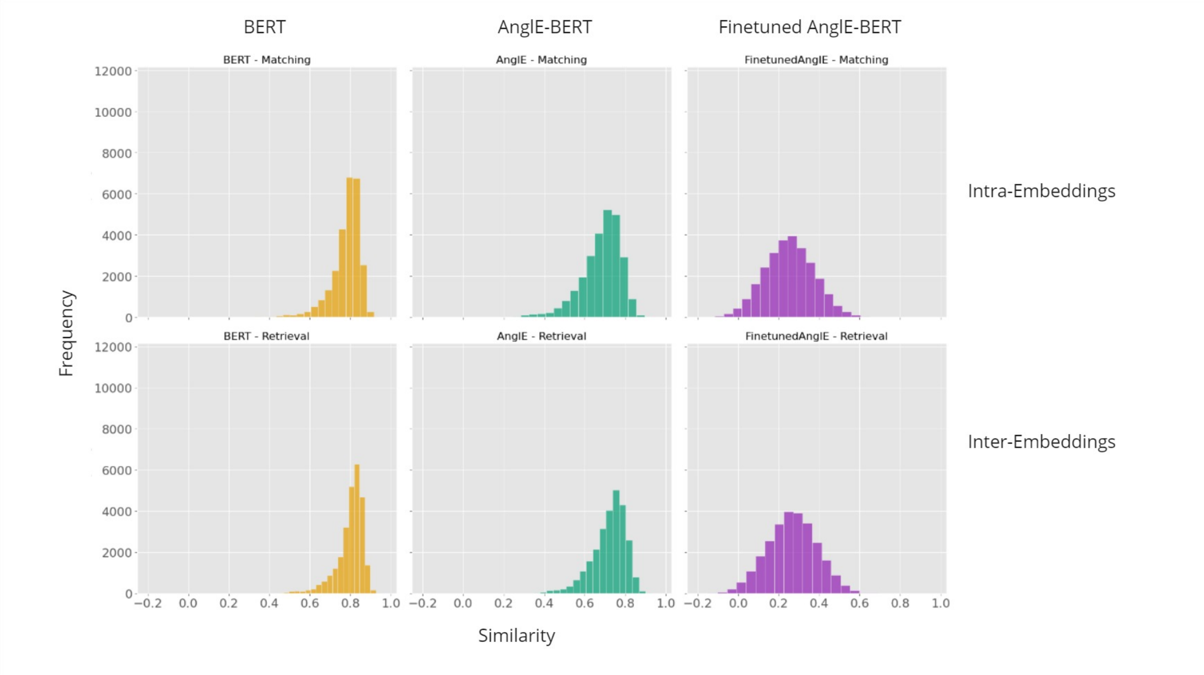


Figure 6: Analysis of Cosine Distributions between Query and Snippet

a relatively high similarity score. This behaviour suggests that BERT and AnglE-BERT may not be capturing the nuanced relationships between legal queries and snippets effectively, potentially leading to a higher number of false positives in retrieval tasks. In contrast, the fine-tuned AnglE-BERT model exhibits a more normal-like distribution. This shift suggests that the fine-tuning process has improved the model's ability to differentiate between relevant and irrelevant cases, balancing the similarity scores across query-snippet pairs. The more centred distribution may be an indicator that the fine-tuned model is better adapted to the legal domain, likely learning the domain-specific vocabulary and complex semantic relationships within legal texts. As a result, it performs more robustly in distinguishing between cases with subtle variations in meaning, leading to improved retrieval performance.

4. Conclusion

In this work, we developed SCaLe-QA, a foundational system tailored to the specific requirements of Sri Lankan Legal Question Answering (LQA) tasks by using domain-specific embeddings derived from Supreme Court cases. Our work primarily focused on enhancing the retrieval accuracy of a RAG system using CBR by fine-tuning embeddings, using BM25 ranking for triplet generation and contrastive learning methods. An interesting finding was the creation of dual representations for the query depending on the attributes being compared for retrieval. Finetuning in this manner resulted in superior F1 scores. Future work will involve integrating Case-Based Reasoning (CBR) to build more comprehensive question-answering models, as well as expanding the scope of SCaLe-QA to attribute-focused embedding models.

References

[1] A. Louis, G. van Dijck, G. Spanakis, Interpretable Long-Form Legal Question Answering with Retrieval-Augmented Large Language Models, 2023. URL: http://arxiv.org/abs/2309.17050, arXiv:2309.17050 [cs].

[2] A. Abdallah, B. Piryani, A. Jatowt, Exploring the state of the art in legal QA systems, Jour-

nal of Big Data 10 (2023) 127. URL: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-023-00802-8. doi:10.1186/s40537-023-00802-8.

[3] S. Jayasinghe, L. Rambukkanage, A. Silva, N. de Silva, S. Perera, M. Perera, Learning Sentence Embeddings In The Legal Domain with Low Resource Settings, in: S. Dita, A. Trillanes, R. I. Lucas (Eds.), Proceedings of the 36th Pacific Asia Conference on Language, Information and Computation, Association for Computational Linguistics, Manila, Philippines, 2022, pp. 494–502. URL: https://aclanthology.org/2022.paclic-1.55.

[4] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Curran Associates Inc., Red Hook, NY, USA, 2020. Event-place: Vancouver, BC, Canada.

[5] N. Wiratunga, R. Abeyratne, L. Jayawardena, K. Martin, S. Massie, I. Nkisi-Orji, R. Weerasinghe, A. Liret, B. Fleisch, CBR-RAG: Case-Based Reasoning for Retrieval Augmented Generation in LLMs for Legal Question Answering, in: J. A. Recio-Garcia, M. G. Orozco-del Castillo, D. Bridge (Eds.), Case-Based Reasoning Research and Development, volume 14775, Springer Nature Switzerland, Cham, 2024, pp. 445–460. URL: https://link.springer.com/10.1007/978-3-031-63646-2_29. doi:10.1007/978-3-031-63646-2_29, series Title: Lecture Notes in Computer Science.

[6] M.-Y. Kim, Y. Xu, R. Goebel, Applying a Convolutional Neural Network to Legal Question Answering, in: M. Otake, S. Kurahashi, Y. Ota, K. Satoh, D. Bekki (Eds.), New Frontiers in Artificial Intelligence, volume 10091, Springer International Publishing, Cham, 2017, pp. 282–294. URL: http://link.springer.com/10.1007/978-3-319-50953-2_20. doi:10.1007/978-3-319-50953-2_20, series Title: Lecture Notes in Computer Science.

[7] S. Robertson, H. Zaragoza, The Probabilistic Relevance Framework: BM25 and Beyond, Foundations and Trends® in Information Retrieval 3 (2009) 333–389. URL: http://www.nowpublishers.com/article/Details/INR-019. doi:10.1561/1500000019.

[8] L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, F. Wei, Text Embeddings by Weakly-Supervised Contrastive Pre-training, 2024. URL: http://arxiv.org/abs/2212.03533, arXiv:2212.03533 [cs].

[9] X. Li, J. Li, AnglE-optimized Text Embeddings, 2024. URL: http://arxiv.org/abs/2309.12871, arXiv:2309.12871 [cs].

[10] J. Su, Cosent (1): A more effective sentence vector scheme than sentence bert, 2022. URL: https://kexue.fm/archives/8847.

[11] T. Gao, X. Yao, D. Chen, SimCSE: Simple Contrastive Learning of Sentence Embeddings, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 6894–6910. URL: https://aclanthology.org/2021.emnlp-main.552. doi:10.18653/v1/2021.emnlp-main.552.

[12] L. Xu, H. Xie, Z. Li, F. L. Wang, W. Wang, Q. Li, Contrastive Learning Models for Sentence Representations, ACM Trans. Intell. Syst. Technol. 14 (2023). URL: https://doi.org/10.1145/3593590. doi:10.1145/3593590, place: New York, NY, USA Publisher: Association for Computing Machinery.

[13] X. Ma, Y. Gong, P. He, H. Zhao, N. Duan, Query Rewriting for Retrieval-Augmented Large Language Models, 2023. URL: http://arxiv.org/abs/2305.14283, arXiv:2305.14283 [cs].

[14] OpenAI, New and Improved Embedding Model, 2023. URL: https://openai.com/blog/new-and-improved-embedding-model, publisher: OpenAI.

[15] N. Reimers, I. Gurevych, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: http://arxiv.org/abs/1908.10084.