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]; Ruvan Weerasinghe[2]

[1] Robert Gordon University (RGU), Aberdeen, UK · [2] Informatics Institute of Technology (IIT), Sri Lanka · † Equal contribution

1. Introduction

- SCaLe-QA is a Sri Lankan legal question-answering effort that “learns” from Supreme Court cases so it understands our courts’ language and reasoning patterns. [1]
- it first **finds** the right passages in judgments, then those passages can be used to **draft** answers (the “retrieve, then generate” idea). [4].
- focuses on **passage-level** retrieval (not whole documents) because lawyers usually need the exact paragraph that supports an argument.
- The training data spans **2009–2024** Supreme Court judgments (over 1,500 cases),

[1] A. Louis, G. van Dijck, and G. Spanakis, “Interpretable Long-Form Legal Question Answering with Retrieval-Augmented Large Language Models,” 2023. Available: http://arxiv.org/abs/2309.17050 (arXiv:2309.17050).
[3] S. Jayasinghe, L. Rambukkanage, A. Silva, N. de Silva, S. Perera, and M. Perera, “Learning Sentence Embeddings in the Legal Domain with Low Resource Settings,” in *Proc. 36th Pacific Asia Conf. on Language, Information and Computation (PACLIC)*, 2022, pp. 494–502. Available: 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, and D. Kiela, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” in *Proc. NeurIPS*, 2020.

2. Finetuning Methodology

- **Prepare** judgments: clean up PDFs (OCR if needed), fix formatting, and make text machine-readable. [1]
- **Split** each judgment into manageable **chunks** and then **sentences**; this makes long cases searchable without losing context.
- **Create learning examples**: for each sentence, pick one that's similar (positive) and one that's clearly different (negative) using a classic search method (BM25). [7][8]
- Maintain two kinds of "vectors": one for **grouping similar questions** (intra) and one for **finding passages for a question** (inter).
- Evaluate with **top-K retrieval** metrics that mirror a lawyer's workflow: "Is the right passage in my first few hits?" [15]

[3] S. Jayasinghe *et al.*, "Learning Sentence Embeddings in the Legal Domain with Low Resource Settings," PACLIC 2022. Available:https://aclanthology.org/2022.paclic-1.55

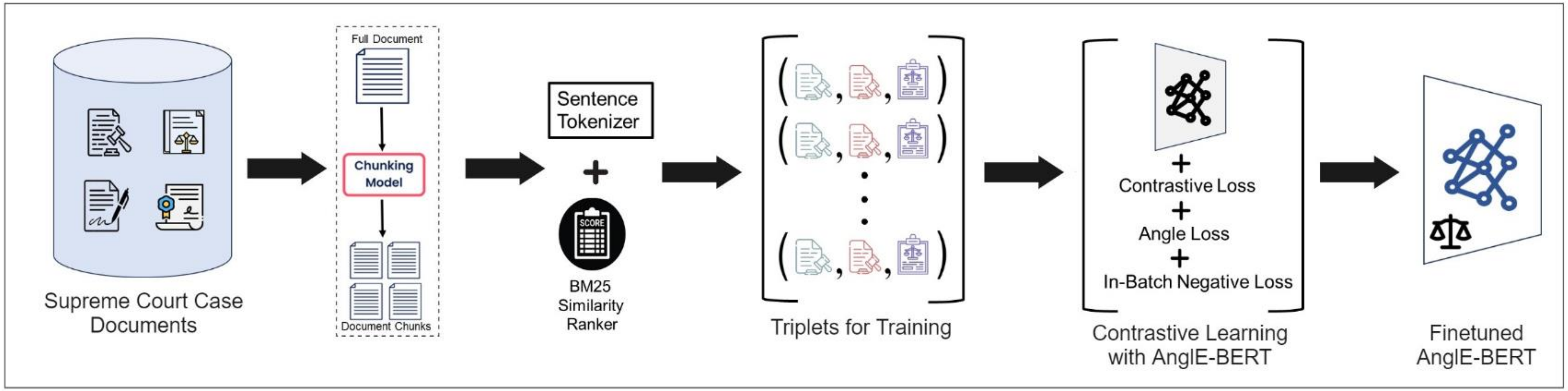


Figure 1: Workflow for Finetuning Process

2.1. Data Source

- **Corpus**: ~1,541 Supreme Court judgments (2009–2024), covering fundamental rights, appeals, writs, constitutional questions, criminal/civil matters, and CHC issues. [1]
- **Acquisition**: scraping public judgments; many files were already text, others needed OCR + manual correction to ensure accuracy.
- **Why it's important**: the model learns *local phrasing* (e.g., "leave to proceed," "arbitrary and capricious," "quash by certiorari") and citations common to Sri Lankan practice.
- **Outcome**: a legally grounded dataset suitable for training and fair evaluation.

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

2.2. Document to Sentence Segmentation

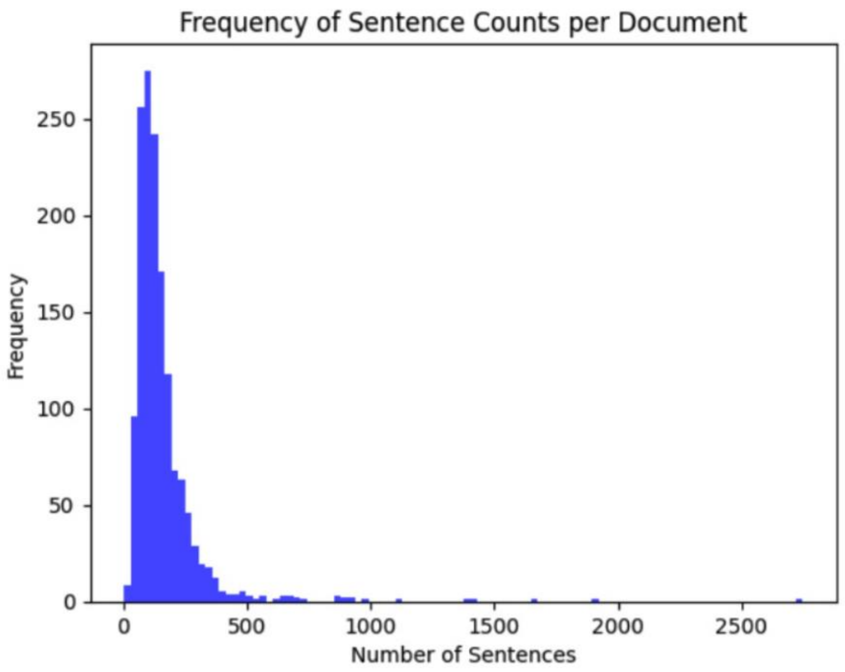
- **Chunk first, then sentence**: long decisions are divided into context-preserving chunks (about a few paragraphs) before sentence splitting, which keeps meaning intact. [4]
- **Why chunks?** They prevent the model from mixing up far-apart parts of a judgment, so retrieved sentences still “live” in coherent neighborhoods.
- **Sentence units**: the final searchable pieces are sentences (or short sentence pairs), which lawyers find easiest to cite.

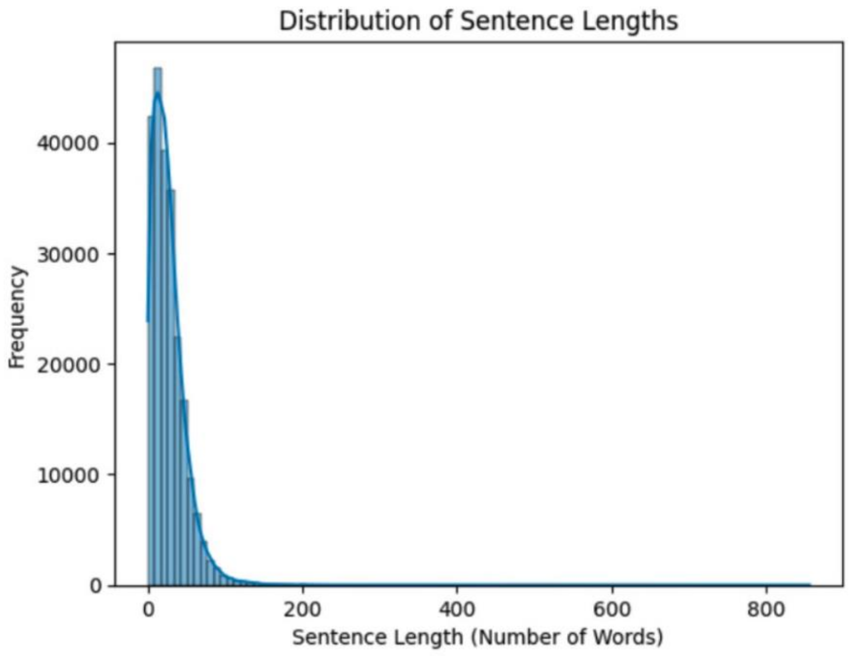
[4] P. Lewis *et al.*, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” NeurIPS 2020.

2.2. Document to Sentence Segmentation

- *[{ "case_id": "SC_FR_0001_2013", ”*
  - *chunk_id": 1, "text": "This is an application under Article 126 alleging unlawful arrest...", "token_len": 210 },*
  - *"chunk_id": 2, "text": "Counsel for the Petitioner argued that the procedure under Section 32...", "token_len": 190 },*
  - *{ "chunk_id": 3, "text": "The Attorney-General submitted that...", "token_len": 185 }]*

2.2. Distribution of the sentences





2.3. Triplet Creation

- For each **anchor sentence**, BM25 ranks other sentences by similarity. The **top one** becomes the **positive** (it "fits"). [7]
- A **negative** is picked from the least similar candidates (clearly "doesn't fit"), which teaches stronger discrimination. [8]
- This is **weak supervision**: no human labels are needed—lexical/semantic overlap guides the learning at scale.
- Produces **hundreds of thousands** of examples, giving the model broad coverage. [8]

[7] S. Robertson and H. Zaragoza, "BM25 and Beyond," 2009.
[8] L. Wang *et al.*, "Weakly-Supervised Contrastive Pre-training," 2024.

2.3. Triplet Creation Sample

- S1 (anchor): "The detention was authorized under Section 32 of the Code."
- S2: "Police produced the suspect before the Magistrate within 24 hours."
- S3: "Section 32 of the Code permits arrest without warrant in limited cases."
- S4: "The appeal concerns a land title dispute in Galle."
- S5: "Gazette 1465/19 concerns share certificates."

Triplet (sample output)

```
{
  "anchor": "The detention was authorized under Section 32 of the Code.",
  "positive": "Section 32 of the Code permits arrest without warrant in limited cases.",
  "negative": "Gazette 1465/19 concerns share certificates."
}
```

2.4. Embedding Finetuning

- The model (AnglE-BERT) is trained so **similar pairs** (anchor–positive) move **closer**, **dissimilar pairs** (anchor–negative) move **apart**. [9]. There is a compound loss function that is being used .

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

The first term $Lc(SU, SL)$, weighted by $w1$, uses the standard cosine similarities between the anchor and positive (or like) instance, $SL = cos(Xa, XL)$, and the anchor and negative (or unlike) instance, $SU = cos(Xa, XU)$.

The second term, weighted by $w2$, 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 SL and SU respectively.

The third term weighted by $w3$, is similar to the first but uses a refined similarity metric, S', where the embeddings of Xa, XL and XU are split in half

Refs used:
[8] L. Wang *et al.*, “Weakly-Supervised Contrastive Pre-training,” 2024.
[9] X. Li and J. Li, “AnglE-Optimized Text Embeddings,” 2024. Available: http://arxiv.org/abs/2309.12871
[15] N. Reimers and I. Gurevych, “Sentence-BERT,” EMNLP 2019.

2.4. Embedding Finetuning

- Anchor: “Was **Art. 12(1)** infringed by the transfer policy?”
- Positive: “Equal protection under **Art. 12(1)** regarding transfers and promotions.”
- Negative: “**Costs** awarded in Rs… (unrelated section).”

- In this the Term 1: Forces cos(anchor,positive) >> cos(anchor,negative)
- Term 2: ensures the anchor prefers its positive over anyone else’s positive in the batch (all others act as difficult negatives).
- Term 3: ensures this preference is consistent across both halves of the embedding, discouraging brittle geometry.

2.5. Model Training Dualities

- Two distinct embeddings were fine-tuned, each optimized for different retrieval purposes,
  - Intra-Embeddings (f(Q)): These embeddings are optimized for attribute matching within the same type of content, such as comparing questions with questions
  - Inter-Embeddings (g(Q)): These embeddings are designed for broader information retrieval

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 optimize retrieval for specific purposes.

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

2.5. Model Training Dualities

- *Intra call (question -> question)*
  Input. -"Is production before a Magistrate mandatory within 24 hours of arrest?”
  - Output - use to find **similar questions** (“Must a suspect be produced promptly?”).
- *Inter call (question -> passage)*
  - "Represent this sentence for searching relevant passages: Is production before a Magistrate mandatory within 24 hours of arrest?"
  - Output - use to search **case passages** stating Article 13(2) / Sec. 32 requirements.

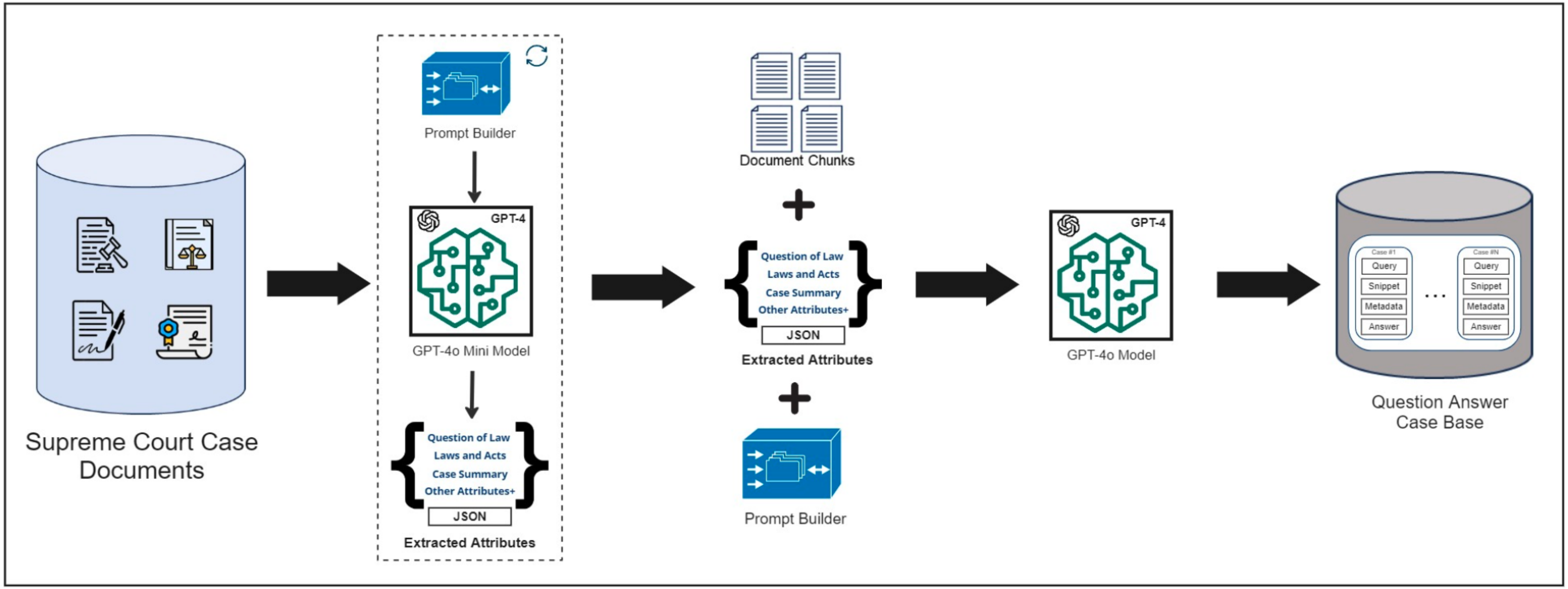
3. Evaluation

- Prior to evaluating the performance of the embedding models, there are two key stages:
  1. Casebase Creation
  2. Test Set Creation

3.1. Casebase Creation

- The documents were segmented into manageable chunks of 384 tokens, and the key attributes were extracted.
- Each chunk gets labels: court, parties, questions of law, summary, laws/acts cited, judgment/relief.
- Stored as JSON, so tools can filter quickly (e.g., “only FR cases with Art. 12(1) and transfers”).
- Why this helps: retrieval can be focused (filter first, then rank), reducing irrelevant hits.

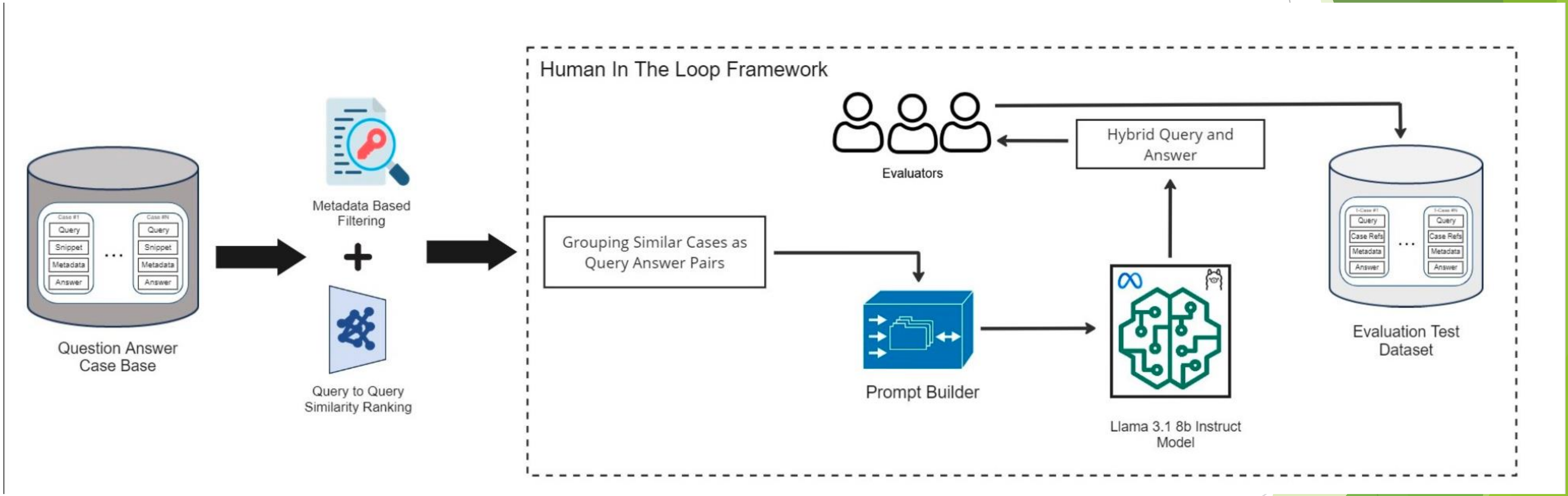
3.1. Casebase Creation



3.2. Test Set Creation

- Start from the casebase. Use metadata (laws/acts/articles, case type) to group cases with overlapping legal references while ensuring different documents for diversity.
- Similarity ranking. For each case's query, compute cosine similarity with others (Ada-002 embeddings) to rank closest cross-document pairs..
- Hybrid Q&A generation. For top-ranked pairs, prompt GPT-4o to write one complex question that requires both snippets to answer, plus a concise answer.
- Human-in-the-loop curation. Authors review and filter: keep items that truly need both snippets, drop low-quality/ambiguous ones.
- Final benchmark. Curated set of 1,000 high-quality Q&A pairs for Retrieval@K evaluation (Recall@K, F1@K).

3.2. Test Set Creation



3.3. Retrieval Analysis

- **Metrics.** Report **Recall@K** (did the correct snippet appear in top-K?) and **F1-score@K** (how clean/useful is that top-K list).
- **Method. k-Nearest Neighbors (k-NN)** over the vector index; vary **K from 1 to 37** (primes) to probe small vs. larger result sets.
- **Models compared.**
- **Fine-tuned AnglE-BERT** (both **intra** and **inter** flavours; multiple loss-weight configs),
- **Vanilla AnglE-BERT** [9], and **BERT** baseline [15].
- **Visualization.** Heatmaps of **Recall@K** and **F1@K** across K and model variants (Figure 5) show performance bands at a glance.

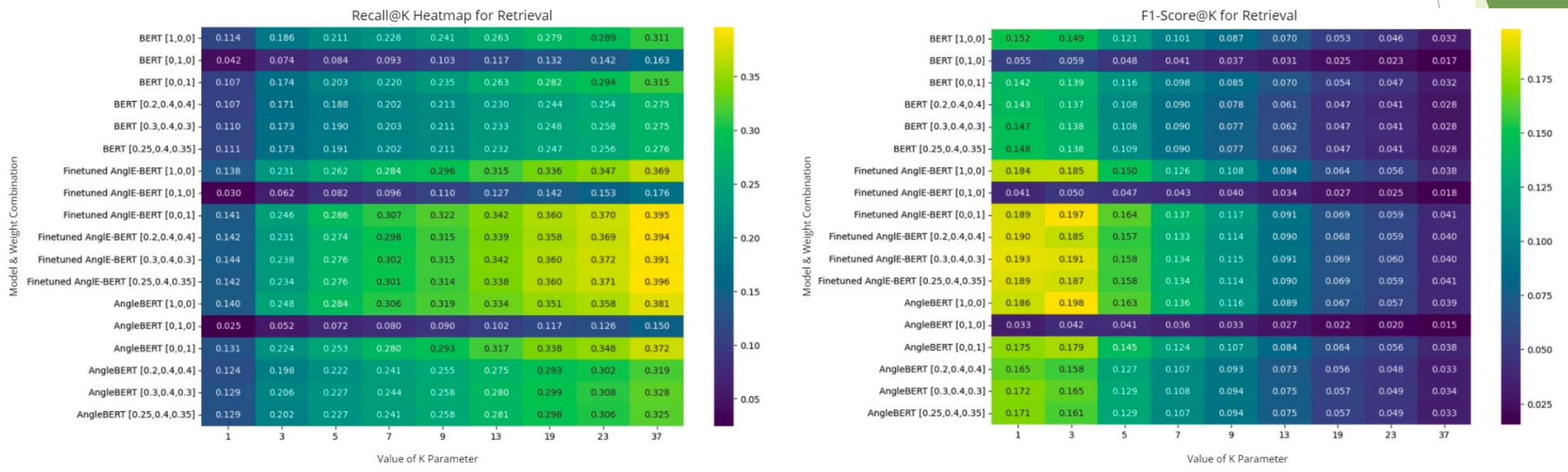
[9] X. Li and J. Li, "AnglE-Optimized Text Embeddings," 2024.
[15] N. Reimers and I. Gurevych, "Sentence-BERT," 2019.

3.3. Retrieval Analysis Contd

- **Metrics.** Report **Recall@K** (did the correct snippet appear in top-K?) and **F1-score@K** (how clean/useful is that top-K list).
- **Method. k-Nearest Neighbors (k-NN)** over the vector index; vary **K from 1 to 37** (primes) to probe small vs. larger result sets.
- **Models compared.**
- **Fine-tuned AnglE-BERT** (both **intra** and **inter** flavours; multiple loss-weight configs),
- **Vanilla AnglE-BERT** [9], and **BERT** baseline [15].
- **Key finding #1. Fine-tuned AnglE-BERT** consistently **improves Recall@K and F1@K** across K, not just at a single setting.
- **Key finding #2. Vanilla AnglE-BERT** is decent for **query↔query** similarity (intra) but **lags** on **query↔passage** retrieval (inter).

[9] X. Li and J. Li, "AnglE-Optimized Text Embeddings," 2024.
[15] N. Reimers and I. Gurevych, "Sentence-BERT," 2019.

3.3. Retrieval Analysis Cont.



3.4 Embedding Distribution

- For each test item, compute **cosine(query, snippet)** and plot the distribution of scores across all pairs.
- **Baseline shape. BERT** and **vanilla AnglE-BERT** show a **left-skewed** histogram (many pairs getting **higher** similarity than they deserve).
- **Why that's bad.** Left-skew -> the model says "these look alike" **too often** -> more **false positives** in top-K (irrelevant snippets pushed up).
- **After fine-tuning. Fine-tuned AnglE-BERT** shifts toward a **more centered / normal-like** distribution—fewer "everything is similar" judgments.
- **Interpretation.** The tuned model better **separates relevant vs. irrelevant** query–snippet pairs, reflecting **legal nuance** (doctrine applied vs. merely mentioned).

4. Conclusion

- Domain-tuned embeddings for Sri Lankan LQA. Built from Supreme Court cases to “speak” local legal language and structure.
- Creation of dual representations with fine tuning gained higher F1 scores.
- 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, and G. Spanakis, "Interpretable Long-Form Legal Question Answering with Retrieval-Augmented Large Language Models," 2023. Available: http://arxiv.org/abs/2309.17050 (arXiv:2309.17050).
- [2] A. Abdallah, B. Piryani, and A. Jatowt, "Exploring the state of the art in legal QA systems," *Journal of Big Data*, vol. 10, no. 127, 2023. doi:10.1186/s40537-023-00802-8. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-023-00802-8
- [3] S. Jayasinghe, L. Rambukkanage, A. Silva, N. de Silva, S. Perera, and M. Perera, "Learning Sentence Embeddings in the Legal Domain with Low Resource Settings," in *Proc. 36th Pacific Asia Conf. on Language, Information and Computation (PACLIC)*, 2022, pp. 494–502. Available: 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, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Proc. NeurIPS*, 2020.
- [5] N. Wiratunga, R. Abeyratne, L. Jayawardena, K. Martin, S. Massie, I. Nkisi-Orji, A. Liret, B. Fleisch, and R. Weerasinghe, "CBR-RAG: Case-Based Reasoning for Retrieval Augmented Generation in LLMs for Legal Question Answering," in *Case-Based Reasoning Research and Development*, LNCS 14775, Springer, 2024, pp. 445–460. doi:10.1007/978-3-031-63646-2_29. Available: https://link.springer.com/10.1007/978-3-031-63646-2_29
- [6] M.-Y. Kim, Y. Xu, and R. Goebel, "Applying a Convolutional Neural Network to Legal Question Answering," in *New Frontiers in Artificial Intelligence*, LNCS 10091, Springer, 2017, pp. 282–294. doi:10.1007/978-3-319-50953-2_20. Available: http://link.springer.com/10.1007/978-3-319-50953-2_20
- [7] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," *Foundations and Trends in Information Retrieval*, vol. 3, pp. 333–389, 2009. doi:10.1561/1500000019. Available: http://www.nowpublishers.com/article/Details/INR-019
- [8] L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, and F. Wei, "Text Embeddings by Weakly-Supervised Contrastive Pre-training," 2024. Available: http://arxiv.org/abs/2212.03533 (arXiv:2212.03533).
- [9] X. Li and J. Li, "AnglE-Optimized Text Embeddings," 2024. Available: http://arxiv.org/abs/2309.12871
- [10] J. Su, "CoSENT (1): A more effective sentence vector scheme than Sentence-BERT," 2022. Available: https://kexue.fm/archives/8847
- [11] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple Contrastive Learning of Sentence Embeddings," in *Proc. EMNLP*, 2021, pp. 6894–6910. doi:10.18653/v1/2021.emnlp-main.552. Available: https://aclanthology.org/2021.emnlp-main.552
- [12] L. Xu, H. Xie, Z. Li, F. L. Wang, W. Wang, and Q. Li, "Contrastive Learning Models for Sentence Representations," *ACM Trans. Intell. Syst. Technol.*, vol. 14, 2023. doi:10.1145/3593590. Available: https://doi.org/10.1145/3593590
- [13] X. Ma, Y. Gong, P. He, H. Zhao, and N. Duan, "Query Rewriting for Retrieval-Augmented Large Language Models," 2023. Available: http://arxiv.org/abs/2305.14283
- [14] OpenAI, "New and Improved Embedding Model," 2023. Available: https://openai.com/blog/new-and-improved-embedding-model
- [15] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proc. EMNLP*, 2019. Available: http://arxiv.org/abs/1908.10084

Thank you