DRiLL at VLSP 2025: A Simple Two-Stage Retrieval Framework for Vietnamese Legal Document Retrieval

Xuan-Duong Hieu and Thanh-Dat Do
FPT.AI - FPT Smart Cloud

Abstract

The DRiLL shared task at VLSP 2025 aims to advance Vietnamese Legal NLP by retrieving relevant legal articles from a large corpus in response to given legal questions. The task requires identifying articles that contain or can answer queries, addressing the challenges of semantic understanding and efficiency in legal text processing. To tackle this, our team proposes a simple two-stage retrieval system. The first stage employs hybrid retrieval methods, combining sparse and dense techniques, to efficiently fetch initial candidate articles from the corpus. The second stage uses a cross-encoder ranker based on fine-tuned BERT-like models to refine and rank the candidates for higher precision. Additionally, we incorporate document-based chunking to add titles to articles in the corpus, thereby enhancing contextual accuracy. We also employ a simple fine-tuning strategy for the reranker, utilizing hard-negative mining to further improve performance. Experimental results show that our approach achieves top-2 on the public test, top-3 on the private test, and top-2 on the post-challenge evaluation, demonstrating the effectiveness of this straightforward pipeline in low-resource legal retrieval scenarios despite its simplicity.

1 Introduction

Information Retrieval (IR) has long been a cornerstone of Natural Language Processing (NLP), aiming to fetch relevant documents from large corpora given a user query (Manning et al., 2008). Classical IR methods, such as TF-IDF and BM25, rely on lexical matching and statistical weighting to rank documents (Robertson and Zaragoza, 2009). While effective in many scenarios, these methods often struggle with semantic mismatches, particularly in complex domains where synonyms, paraphrases, and contextual nuances are prevalent. The advent of dense retrieval, leveraging bi-encoder architecture from pre-trained language models such as SBERT (Reimers and Gurevych, 2019), DPR (Karpukhin et al., 2020), and ColBERT (Khattab and Zaharia, 2020), has significantly improved semantic alignment between queries and documents. Moreover, re-ranking methods based on BERT-based cross-encoders (Nogueira and Cho, 2020) or sequence-to-sequence rerankers such as MonoT5 (Nogueira et al., 2020) have further refined retrieval pipelines by modeling fine-grained query–document interactions.

Legal text retrieval represents a specialized and particularly challenging subset of IR. It requires retrieving statutory articles or legal documents that can directly answer or entail legal questions. Compared with general IR, legal retrieval must cope with highly domain-specific terminology, intricate sentence structures, and the need for precise entailment judgments. These challenges are amplified in low-resource languages such as Vietnamese, where annotated data is scarce and pre-trained legal domain models remain limited (Nguyen et al., 2025; Vuong et al., 2025).

The DRiLL shared task at VLSP 2025 (Vuong et al., 2025) directly addresses these challenges by requiring participants to retrieve relevant legal articles from a Vietnamese legal corpus given natural language legal questions. Unlike general-purpose IR, DRiLL emphasizes not only relevance but also legal reasoning, making it a valuable benchmark for evaluating retrieval systems in underexplored low-resource legal settings.

In this work, we present a simple yet effective three-component framework tailored for the DRiLL task. Our solution is as follows:

1. **Document-based chunking with titles:** We introduce a lightweight preprocessing strategy that augments articles with their corresponding document titles, improving retrieval accuracy

by enriching contextual signals.

2. **Two-stage retrieval framework:** We first apply a *hybrid retriever* that combines BM25 (Robertson and Zaragoza, 2009) with dense retrieval using bge-m3[1], leveraging both lexical precision and semantic generalization. The retrieved candidates are then refined with a *cross-encoder re-ranker*, bge-reranker-v2-m3[2], which models fine-grained query–article interactions.
3. **Hard-negative fine-tuning:** We further fine-tune the re-ranker with a simple hard-negative mining technique to better align it with the legal domain.

2 Methodology

2.1 Document-based chunking with titles

We observed that the original corpus lacks explicit titles of laws, chapters, and articles, which causes two main issues: (i) retrieval accuracy is limited because queries often refer to provisions by their titles, and (ii) chunks in the corpus are not sufficiently distinguishable, making it harder for models to discriminate between similar articles.

To address these issues, we augmented each article with its hierarchical titles through the following steps:

1. **Source identification.** For each article in the corpus, we used its `law_id` to identify the corresponding original legal document. After examining multiple sources, we determined that the most reliable and comprehensive repository is ThuVienPhapLuat.vn. To automatically locate the official document URL, we employed the Google Search API (SerpAPI[3]) with the `law_id` as the query. We then selected the top-ranked URL returned by SerpAPI as the canonical link for the law text.
2. **Crawling and chunking.** We crawled the full text of each law from the identified URL and segmented it into small chunks that follow the same granularity as the provided DRiLL corpus, ensuring consistency in article boundaries.
3. **Title mapping.** For each article, we reconstructed its hierarchical context by attaching the chain of titles from the crawled document. Specifically, we extracted the nested heading structure in the order:

   Title Level 1
   Title Level 2
   ...
   Article Content

   The article in the corpus is then enriched with a metadata field `titles = ["Title Lv1", "Title Lv2", ...]` that captures its full hierarchical context.

Practical challenges. During crawling, we encountered several limitations of the automated pipeline. Among the 2,156 statutes in the DRiLL corpus, 41 returned incorrect URLs from the Google Search API (SerpAPI) and had to be manually corrected. Moreover, out of nearly 60,000 article chunks, approximately 1,000 could not be mapped precisely at the *section*-level. The main causes were:

- **Content discrepancies:** the online sources had been updated compared to the static dataset released for DRiLL, leading to mismatches.
- **Crawl errors:** in some cases, the crawler failed to capture the full content of a statute, resulting in incomplete chunks.
- **Structural outliers:** a few webpages used layouts that deviated from the dominant pattern, causing the chunk extraction algorithm to fail.

While some issues remained unresolved within the competition timeline, manual inspection confirmed that the corpus is highly consistent overall. We acknowledge these challenges and view systematic error quantification and reconciliation as important directions for future work.

This preprocessing ensures that each article chunk carries both its textual content and the surrounding hierarchical titles, thereby providing richer contextual signals for retrieval and improving discrimination among articles that would otherwise appear similar.

2.2 Two-stage retrieval framework

2.2.1 Stage 1: Hybrid retrieval

The DRiLL shared task at VLSP 2025 is formulated as a **legal article retrieval** problem.

[1] https://huggingface.co/BAAI/bge-m3
[2] https://huggingface.co/BAAI/bge-reranker-v2-m3
[3] https://serpapi.com

- **Input:** a query (typically a natural-language legal question) and a large collection of **articles of law**, where each article corresponds to one provision in Vietnamese legislation.
- **Output:** a ranked list of **relevant articles** that can answer or entail the query.

Formally, given a query q and a corpus $\mathcal{C} = \{a_1, a_2, \ldots, a_N\}$ of articles, the retrieval system must learn a scoring function $f(q, a)$ such that relevant articles receive higher scores.

To maximize recall, we employ a **hybrid retrieval approach** that leverages both sparse and dense signals:

- **Sparse retrieval (BM25).** BM25 (Robertson and Zaragoza, 2009) computes the relevance score based on exact lexical matches between query terms and article terms, adjusted by term frequency and inverse document frequency. For a query q and article a, the BM25 score is:

$$\text{BM25}(q, a) = \sum_{t \in q} IDF(t) \cdot \frac{f(t, a) \cdot (k_1 + 1)}{f(t, a) + k_1 \cdot \left(1 - b + b \cdot \frac{|a|}{\text{avgdl}}\right)} \quad (1)$$

  where $f(t, a)$ is the frequency of term t in article a, $|a|$ is the length of the article, avgdl is the average article length, and k_1, b are hyperparameters.

- **Dense retrieval.** We use the `bge-m3` model. The model encodes both query q and article a into dense vectors $\mathbf{h}_q, \mathbf{h}_a \in R^d$. The relevance score is defined as their cosine similarity:

$$\text{Dense}(q, a) = \cos(\mathbf{h}_q, \mathbf{h}_a) = \frac{\mathbf{h}_q \cdot \mathbf{h}_a}{\|\mathbf{h}_q\| \, \|\mathbf{h}_a\|}. \quad (2)$$

To combine the two signals, we apply a **weighted sum of scores**:

$$\text{Score}(q, a) = \lambda \cdot \text{BM25}(q, a) + (1 - \lambda) \cdot \text{Dense}(q, a), \quad (3)$$

where $\lambda \in [0, 1]$ is a hyperparameter tuned on the validation set. This simple yet effective strategy balances the precision of sparse retrieval with the semantic generalization of dense retrieval.

The hybrid retriever returns the top-k candidate **articles** for the second-stage reranker.

2.2.2 Stage 2: Cross-encoder reranking

While the first-stage retriever is effective for recall, the top-k retrieved articles may still contain irrelevant results. To improve precision, we employ a **reranking model** that re-scores candidate articles given the query.

Cross-encoder reranker. We adopt the `bge-reranker-v2-m3`, a cross-encoder that jointly encodes the query–article pair and predicts a relevance score. Formally, for a query q and candidate article a, the reranker computes:

$$\text{Rerank}(q, a) = \text{CE}_\theta(q, a), \quad (4)$$

where CE_θ denotes the cross-encoder with parameters θ.

Filtering with threshold. After reranking with the cross-encoder, we apply a strict filtering mechanism to retain only highly confident candidates. Concretely, we consider the top-10 reranked results and keep those whose relevance score exceeds a threshold of 0.99. If no candidate satisfies this strict criterion, we fall back to selecting the top-2 results by reranker score to avoid returning an empty set. This strategy ensures that the final output contains only highly reliable legal articles while still guaranteeing non-empty retrieval results. Empirically, this post-ranking filter significantly boosts precision with only a minor reduction in recall, leading to a notable improvement in the overall F2 score.

The threshold of 0.99 and the fallback policy of keeping the top-2 results were empirically tuned based on the validation set, which included both the training set and the public test portion of the shared task.

2.3 Hard-negative fine-tuning

Hard Negative Mining. To further improve the reranker, we construct a training dataset with both positive and hard negative samples. The positives are directly taken from the ground-truth-relevant articles. To ensure data quality, we discard any article whose tokenized length exceeds 1024 tokens.

For hard negatives, we leverage the reranking outputs. Specifically, for each query, we take up to the top-30 retrieved articles from the reranker results that are not part of the gold relevant set. Articles that exceed the maximum token length are filtered out. To avoid introducing false negatives, we skip the top-5 hits, since these articles

are often either near-duplicates of positives or semantically very close to the query and may still contain partially relevant information. Including them in the negative set would risk confusing the model. By shifting the selection window, we ensure that negatives are still semantically challenging but not overly close to the positives. From the remaining candidates, we randomly sample 10 articles if more than 15 are available, otherwise we sample 5.

This strategy ensures that the hard negatives are sufficiently difficult while avoiding mislabeled near-positives, thereby providing a strong and reliable training signal for contrastive learning. The detailed procedure is illustrated in Algorithm 1.

Algorithm 1 Hard Negative Mining Strategy

Require: Retrieval results R, corpus C, maximum length $L = 1024$
Ensure: Training set D with positives and hard negatives

for each query $q \in R$ **do**
  $P \leftarrow []$ ▷ positives
  for each $a \in q.relevants$ **do**
    if len(tokenize($C[a]$)) $\leq L$ **then**
      add $(a, C[a])$ to P
    end if
  end for
  if $P = \emptyset$ **then**
    continue
  end if
  $H \leftarrow []$ ▷ hard negatives
  for each $h \in$ top-30($q.rerank\_hits$) **do**
    if $h \notin q.relevants$ and len(tokenize($C[h]$)) $\leq L$ **then**
      add $(h, C[h])$ to H
    end if
  end for
  if $H = \emptyset$ **then**
    continue
  end if
  discard top-5(H)
  if len(H) > 15 **then**
    $H' \leftarrow$ random_sample(H, 10)
  else
    $H' \leftarrow$ random_sample(H, 5)
  end if
  add $\{q, P, H'\}$ to D
end for

Fine-tuning Reranker. After mining positives and hard negatives, we fine-tune the cross-encoder model `bge-reranker-v2-m3` using the Sentence-Transformers[4] framework. Each training instance is constructed by pairing a query with either a relevant article (label $y = 1$) or a mined hard negative article (label $y = 0$). The reranker outputs a relevance score $\hat{y} \in [0, 1]$, which is interpreted as the probability that the article is relevant to the query.

The training objective is the binary cross-entropy (BCE) loss:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\Big(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\Big), \tag{5}$$

where $y_i \in \{0, 1\}$ is the ground-truth label for the i-th query–article pair, and $\hat{y}_i$ is the predicted probability.

This fine-tuning process enables the reranker to better separate relevant articles from challenging hard negatives, thereby improving overall retrieval accuracy on the DRiLL task.

3 Experiments

3.1 Implementation Details

Stage 1 (Hybrid Retrieval). We use `Elasticsearch` for both sparse and dense retrieval. For the sparse component, we adopt BM25 with k=0.5 and b=0.5. For dense retrieval, we encode each *article of law* with `bge-m3` at a maximum input length of 1024 tokens, and index the vectors in `Elasticsearch`. Scores are combined by a weighted sum with λ=0.6:

$$\text{Score}(q, a) = \lambda \cdot \text{BM25}(q, a) + (1-\lambda) \cdot \text{Dense}(q, a).$$

The hybrid retriever returns the top-100 candidate **articles**.

Stage 2 (Reranking and Filtering). We rerank the 100 candidates using `bge-reranker-v2-m3` and keep the top-10 by reranker score. The choice of retaining the top 10 candidates was also empirically tuned in the validation set (training + public test). Finally, we apply the filtering algorithm described above.

Fine-tuning the Reranker. The reranker is fine-tuned with mined hard negatives (Section 1) using the `SentenceTransformers` framework for 2 epochs, batch size 16, learning rate 2×10^{-5}, and binary cross-entropy loss.

[4] https://sbert.net/

Additional Notes on Hyperparameter Choices. The parameters of BM25 ($k = 0.5, b = 0.5$), the weighted sum coefficient $\lambda = 0.6$, and the cut-off of the top 100 candidates were chosen mainly based on previous practices and evidence from related retrieval studies. Due to time and computational constraints, we did not conduct extensive hyperparameter tuning. Instead, we adopted widely established settings from the IR literature, with the aim of striking a practical balance between effectiveness and efficiency.

For the hard negative mining procedure, we also followed fixed settings, we discarded the top-5 hits to avoid trivial cases, considered up to the top-30 retrieved non-relevant articles as potential negatives, and then randomly sampled 10 candidates (or 5 if fewer were available) for training. These hyperparameters were similarly chosen as a pragmatic compromise between difficulty of negatives and computational feasibility.

3.2 Evaluation Metrics

System performance is evaluated using **Precision**, **Recall**, and the **macro-**F_2 score.

$$\text{Precision} = \frac{1}{|Q|} \sum_{q \in Q} \frac{|\text{Retrieved}(q) \cap \text{Relevant}(q)|}{|\text{Retrieved}(q)|}$$

$$\text{Recall} = \frac{1}{|Q|} \sum_{q \in Q} \frac{|\text{Retrieved}(q) \cap \text{Relevant}(q)|}{|\text{Relevant}(q)|}$$

$$F_2 = \frac{5 \times \text{Precision} \times \text{Recall}}{4 \times \text{Precision} + \text{Recall}}$$

where Q is the set of all queries, Retrieved(q) denotes the set of articles retrieved for query q, and Relevant(q) denotes the relevant articles for the ground truth.

3.3 Systems Compared

We evaluate the following configurations:

- **Default**: hybrid retrieval $\rightarrow$ reranking; return top-3 articles (no score filtering).
- **+ Title**: same as Default, but each article is enriched with its *article title* prepended to the text during encoding.
- **+ Title + Filter**: add the post-reranking thresholding (>0.99) instead of returning a fixed top-3.
- **+ Title + Filter + FT**: further fine-tune the reranker with mined hard negatives.
- **+ Title + Filter + FT***: submitted checkpoint during the private phase (an intermediate checkpoint).

3.4 Results and Analysis

Setting	P	R	F_2
Default	0.2594	0.6471	0.4982
+ Title	0.3030	0.7564	0.5822
+ Title + Filter	0.4598	0.7335	0.6555
+ Title + Filter + FT	**0.5723**	0.7492	**0.7055**
+ Title + Filter + FT*	0.5097	**0.7652**	0.6955

Table 1: Private test results. P=Precision, R=Recall. *FT* denotes fine-tuning with hard negatives. * indicates the submitted intermediate checkpoint.

Effect of Title Enrichment. Adding the *article title* yields consistent gains over Default: ΔP=$+0.0436$, ΔR=$+0.1093$, ΔF_2=$+0.0840$. The improvement is especially pronounced for recall, indicating that explicit article-level context helps the retriever/reranker align queries with the correct legal provisions.

Effect of Score Filtering. Introducing the > 0.99 threshold substantially increases precision ($+0.1568$) with a modest recall drop (-0.0229), improving F_2 by $+0.0733$ over *+Title*. This confirms that a strict post-filter effectively removes uncertain candidates while preserving most true positives.

Effect of Fine-tuning. Fine-tuning the reranker with mined hard negatives further boosts precision ($+0.1125$) and slightly raises recall ($+0.0157$) over *+Title+Filter*, delivering the best F_2 ($+0.0500$). Overall, relative to *Default*, the final system improves precision by $+0.3129$ (about $+121\%$ relative), recall by $+0.1021$ ($\sim$$+16\%$ relative), and F_2 by $+0.2073$ ($\sim$$+41.6\%$ relative).

Takeaways. (1) Title enrichment is a simple, high-impact change that primarily raises recall. (2) Strict score filtering (> 0.99) is crucial for precision with minimal recall loss. (3) Fine-tuning with hard negatives improves both precision and F_2, suggesting better discrimination among near-miss legal articles under the DRiLL evaluation.

4 Conclusion

In this work, we presented a simple yet effective two-stage retrieval framework for the VLSP

2025 DRiLL shared task on Vietnamese legal document retrieval. Our approach combined three key components: (i) document-based chunking with titles to enrich article representations, (ii) a hybrid retrieval strategy integrating BM25 with dense semantic embeddings for high recall, and (iii) a cross-encoder reranker further improved with hard-negative fine-tuning and strict post-filtering for precision.

Despite its simplicity, the system achieved competitive results, ranking top-2 on the public leaderboard, top-3 on the private test, and top-2 in the post-challenge evaluation. These outcomes highlight that careful engineering of established IR techniques, together with lightweight domain-aware augmentations, can deliver strong performance even in low-resource legal NLP settings.

For future work, we plan to explore domain-adaptive pretraining for Vietnamese legal language models, more advanced retrieval architectures such as late-interaction models, and multi-task learning setups that jointly optimize retrieval with downstream legal reasoning tasks. We also view LLM integration as a promising direction. While our current system emphasizes lightweight and efficient retrieval, future extensions may incorporate LLM-based rerankers or retrieval-augmented generation. In particular, we aim to experiment with Vietnamese-legal-adapted LLMs as rerankers or post-retrieval reasoning modules, which could provide richer interpretability and stronger semantic alignment in complex legal queries.

Acknowledgments

We would like to thank **FPT Smart Cloud** for sponsoring computational resources and providing GPU infrastructure that enabled us to conduct the experiments in this work.

References

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 39–48.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.

Tan-Minh Nguyen, Hoang-Trung Nguyen, Trong-Khoi Dao, Xuan-Hieu Phan, Ha-Thanh Nguyen, and Thi-Hai-Yen Vuong. 2025. Vlqa: The first comprehensive, large, and high-quality vietnamese dataset for legal question answering. *Preprint*, arXiv:2507.19995.

Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage re-ranking with bert. *Preprint*, arXiv:1901.04085.

Rodrigo Nogueira, Jimmy Yang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. In *Findings of EMNLP 2020*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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)*, pages 3982–3992.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Thi-Hai-Yen Vuong, Tan-Minh Nguyen, Hoang-Trung Nguyen, Trong-Khoi Dao, and Le Hoang-Quynh Nguyen, Ha-Thanh. 2025. Overview of the vlsp 2025 challenge on drill: Deep retrieval in the expansive legal landscape. In *Proceedings of the 11th International Workshop on Vietnamese Language and Speech Processing*.