Dense vs. Sparse Baselines for Legal Case Retrieval: A Comparative Study of BERT-base, LegalBERT, and BM25 on COLIEE 2025 Task 1

Rayan Mehta z5478609 UNSW
MATH3856 Assignment 3

***Abstract*—Legal professionals increasingly rely on automated systems to retrieve relevant case law, as manual review is time-consuming and error prone. Retrieval Augmented Generation (RAG) systems enhance large language models (LLMs) by supplying external context retrieved from a corpus, improving factual accuracy and traceability.**

RAG is fundamentally underpinned by Information Retrieval (IR), the process of finding relevant information from a large collection of unstructured data to respond to queries. Sparse IR models such as BM25 which rely primarily on term matching and vocabulary are able to provide a strong baseline for IR tasks as they rely on interpretable statistical approaches. In recent times, Dense IR models, such as BERT, have utilized deep learning in order to allow the semantic meaning of documents to be captured in a latent vector space often outperforming their statistical counterparts in many IR benchmarks. Domain specific BERT variants, such as LegalBERT are further pre-trained on legal texts to capture specialised terminology and reasoning patterns, and have been shown to improve the performance of downstream specialised Natural Language Processing (NLP) tasks.

Legal Case Retrieval (LCR) is an IR task that requires a system to rank cases within a given corpus based on their relevance to a user submitted query. In this paper we aim to set a baseline for LCR with the COLIEE 2025 Task 1, comparing traditional sparse IR methods (BM25) against a naive and reproducible BERT based pipeline, contrasting general purpose BERT-base with the domain specific LegalBERT.

Our results showed that the sparse BM25 baseline remains superior to naive dense retrieval architectures (Mean Reciprocal Rank (MRR) of 0.233 vs. 0.178), though domain-specific pre-training proved effective, with LegalBERT outperforming BERT-base by approximately 22% on MRR. These findings highlight the "information bottleneck" inherent in single-vector representations of documents, which struggle to capture fine-grained legal details, while simultaneously validating that token-level pooling and PCA dimensionality reduction are essential strategies for denoising latent semantic spaces.

Keywords—Legal Case Retrieval, BERT, Transformers, BM25, LegalBERT

I. Introduction

The practice of law, which relies heavily on precedent, requires legal professionals to efficiently navigate vast and ever growing repositories of case law. Legal Case Retrieval (LCR), a specialised form of Information Retrieval (IR), is the task of automatically identifying or ranking prior cases that are legally relevant and provide guidance for a given query [1]. Due to the sheer volume of documentation, manual review is time-consuming and prone to human oversight, creating a critical need for robust automated systems.

This demand has driven the adoption of modern natural language processing (NLP) techniques [2], particularly Retrieval Augmented Generation (RAG) systems. RAG systems function by enhancing large language models (LLMs) with external, authoritative context retrieved from a knowledge corpus [3]. This process is crucial across various knowledge intensive domains, including medical diagnosis [4], technical support [5] and enterprise search [6], as it significantly reduces LLM hallucination, improving factual accuracy and traceability.

The quality of a RAG pipeline is fundamentally reliant on high fidelity Information Retrieval (IR). Historically, IR has employed Sparse IR models [7], such as BM25 [8], which rely on term frequency vectors and explicit lexical matching to identify relevant documents. These methods are interpretable and provide a strong statistical baseline. Conversely, modern pipelines rely on Dense IR to perform semantic search [9] [10] [11]. Dense IR encodes text into fixed size vector embeddings, numerical representations that capture the contextual meaning of documents in a latent vector space. Relevance is then determined by calculating the vector distance such as cosine similarity, between the query embedding and the case embeddings [12].

The canonical model for generating these dense embeddings is the Bidirectional Encoder Representations from Transformers (BERT) model [13], [14]. BERT employs an encoder-only Transformer architecture, pre-trained on a massive general-purpose corpora (e.g Wikipedia), to generate powerful contextual representations. However, the complex, unique vocabulary and structure of legal documents often pose a challenge for these general models. To address this and similar challenges, domain specific BERT variants, such as LegalBERT [15], have been developed and further pre-trained on vast corpuses of legal texts to capture specialised

terminology and reasoning patterns demonstrating improved performance on specialised downstream NLP tasks.

The LCR task presents unique challenges that complicate a straightforward application of these dense IR models [1] [16]. Firstly, extreme document length means that most cases exceed the token limit of standard BERT models. This necessitates robust preprocessing strategies, such as dividing the document into smaller chunks before a single case level embedding can be generated through a pooling strategy. Secondly legal relevance extends beyond simple topical similarity and often requires identifying alignment in key facts and judicial reasoning. The difficulty of LCR has led to a wide variety of sophisticated pipeline approaches in annual benchmarks like the COLIEE (Competition on Legal Information Extraction and Entailment) [17]. State-of-the-Art (SOTA) LCR approaches, particularly those utilizing dense embedding models, primarily diverge in their strategies for document segmentation (chunking) and subsequent representation aggregation (pooling). Furthermore, many advanced methods incorporate dimensionality reduction to ensure computational efficiency during large-scale retrieval.

In this study, we aim to establish a comparative baseline for LCR with the COLIEE 2025 Task 1 dataset, addressing the gap by evaluating the impact of domain-specific pre-training. COLIEE Task 1 is the competition task of finding a set of relevant prior cases from a large legal corpus that support or conflict with a given query case, thereby modeling the fundamental process of locating legal precedent. We implement a simple, reproducible Dense IR pipeline, comparing the retrieval accuracy of the general-purpose BERT-base model against the domain-specific LegalBERT model, using the sparse BM25 model as a strong statistical baseline. Our study uses a basic data cleaning which all models (BM25, BERT-base, and LegalBERT) are evaluated on. For our Dense models, we generate case-level embeddings using mean pooling and explore varied chunking and pooling strategies (including with/without overlapping window and pooling word tokens or just summary CLS tokens). We also evaluate the impact of dimensionality reduction, such as Principle Component Analysis (PCA) [18] on the resultant vector space. Investigating the effect of PCA provides insights into the intrinsic dimensionality and redundancy of the semantic information captured by the two models. Successful reduction suggests the latent space is robust and efficient, which is crucial for scalable, production-grade IR systems. Conversely, a sharp drop in performance due to PCA would indicate that the necessary legal nuance is distributed across all dimensions. Furthermore, analyzing whether PCA affects the general-purpose BERT-base latent space differently than the domain-specific LegalBERT latent space could reveal key differences in how each model organizes complex legal concepts.

II. Related Work

The task of Legal Case Retrieval (LCR) exists at the intersection of Information Retrieval (IR) and domain specific Natural Language Processing (NLP). Our research is informed by a broad body of work in these fields, which can be categorized into the evolution of retrieval models from sparse to dense, the development of domain-specific language models, and the state-of-the-art methods that currently define the field.

A. Traditional Approaches to LCR (Sparse IR)

Early and foundational approaches to LCR are built on sparse retrieval models, which are non-learning based statistical models. The most prominent and enduring of these are the TF-IDF (Term Frequency Inverse Document Frequency) [7] and its probabilistic successor BM25 [8]. BM25, in particular, remains a powerful and computationally efficient baseline. It improves upon simple keyword matching by calculating the relevance score based on the Term Frequency (TF) of query terms in a document, the Inverse Document Frequency (IDF) of those terms across the whole corpus and a document length normalisation factor. While highly interpretable and fast, these methods are fundamentally limited by the failure to capture deeper semantic relationships, such as synonyms or legal concepts expressed with different vocabularies [19].

B. Neural Approaches to LCR (Dense IR)

To overcome the limitations of sparse models, researchers began using dense retrieval models. These methods employ neural networks to encode text into a vector space where semantic proximity corresponds to relevance.

The first generation of dense models, such as Word2Vec [20] and GloVe [21], learned high quality non-contextual vector representations for words. While revolutionary, these models were critically flawed in that they assigned a static vector to a word regardless of its meaning in a sentence.

Following this, early neural IR approaches such as CNNs [22], BIDAF [23] and SMASH-RNN [24] used neural architectures to read entire sentences or paragraphs to make more context aware representation.

This paradigm shifted completely with the introduction of the Transformer architecture, specifically BERT (Bidirectional Encoder Representations from Transformers). BERT replaced static embeddings by generating dynamic, contextual embeddings. By processing the entire sentence at once using its self attention mechanism, BERT is able to disambiguate word meaning and capture complex semantic nuance which lead to a massive leap in performance [14].

However, the application of BERT to LCR immediately highlighted the long text problem, in that legal cases exceed the standard 512 token limit input of BERT. This necessitated the development of chunking and pooling strategies (e.g., segmenting the document and averaging the chunk embeddings) to create a single, representative vector for an entire case [1].

C. Towards State Of The Art (SOTA) Approaches

The failure of simple dense retrieval models to fully capture legal nuance is a primary driver of SOTA research. Primarily, single vector BERT approaches that rely on mean pooling fail in LCR because they average out the specific, fine-grained details of key facts and judicial reasoning, resulting in a low-resolution embedding. Fundamentally, these single case dense embeddings, in their compression of information, fail to identify the key circumstances and elements that might make a case relevant even when the general semantic meaning may not be.

To address this challenge, SOTA LCR systems, particularly those submitted to competitions like COLIEE Task 1, still utilise semantic embeddings but have shifted away from single-vector representations toward fine-grained and interaction-based methods.

C-1. Fine-Grained and Late-Interaction Models

A key innovation is the use of late-interaction models, exemplified by ColBERT [25]. Instead of compressing the entire document into one vector, ColBERT independently encodes the query and the document (often segmented into tokens or small chunks) into numerous vectors. These vectors are then condensed using dimension reduction (for faster processing of subsequent steps). Relevance is then calculated by comparing these small vectors dynamically at search time, maximizing the interaction between query tokens and document tokens. Another successful approach is BERT-PLI (BERT-Paragraph Level Interaction) [26], which utilizes BERT to capture semantic relationships at the paragraph-level and then infers the relevance between two cases by aggregating these paragraph-level interactions through a Recurrent Neural Network (RNN) structure.

Sentence level embedders have also emerged as a middle ground between coarse paragraph representations and expensive token level late interactions. Models such as Sentence-BERT [27], fine-tune BERT with siamese or triplet objectives on inference tasks so that producing a single dense vector for each sentence that is specifically optimised for cosine similarity comparisons. In legal retrieval, sentence-level embeddings enable fine-grained matching of key factual propositions or judicial holdings without the quadratic cost of token-level interaction. These approaches retain BERT's contextual power while keeping index size and query latency manageable, bridging the gap between single-vector baselines and full late-interaction models.

C-2. Structural and Semantic Feature Enhancement

Beyond raw text, SOTA models incorporate legal structural knowledge to enhance retrieval. An enhanced feature set including pairwise similarity sentence level embeddings at the proposition level as well as judge names was shown to have strong performance. The COLIEE competition has also seen the application of Graph Neural Networks (GNNs) [28], which model structural relationships between cases or legal elements, capturing complex dependencies a pure semantic embedding cannot. Beyond this, many successful attempts integrated generative Large Language Models (LLMs) in the preprocessing to summarize cases before document encoding [17, pp. 37–46].

C-3. Hybrid Solution

As many of these fine-grained or interaction-heavy dense embedding methods are more computationally expensive, the hybrid approach remains the dominant design pattern in SOTA LCR. This involves combining a fast, sparse IR method (like BM25) in a first stage to filter the massive corpus down to a small, manageable subset. This subset can then be safely and accurately searched or re-ranked using the highly precise, but more resource-intensive, dense methods discussed above [1].

D. Domain-Specific Language Models: LegalBERT

General purpose BERT models have achieved state of the art results in several downstream NLP tasks on generic datasets such as GLUE [29], SQuAD [30] and RACE [31] [15].

Naturally, these general-purpose BERT models have struggled to handle highly specialised terminology and complex reasoning structures found in specialised areas. This has led to the introduction of domain-adaptated BERT variations which are commonly derived from either further training BERT-base or completely training BERT architecture from scratch on a domain specified corpus.

LegalBERT [15] is a domain-adapted BERT specified for downstream legal NLP tasks. The specific version utilized in this study, the LegalBERT-base-uncased model, is pre-trained from scratch on a massive, domain-specific corpus of over 12 GB of legal text (including EU legislation, court cases, and contracts). This crucial decision means that LegalBERT does not simply adjust general-domain weights, but learns a new, specialised vector space from the ground up, utilizing a legal-specific vocabulary (trained via Sentence-Piece on the same corpus). This makes LegalBERT fundamentally different from BERT-base; it is designed to accurately interpret the specific legal meaning of terms and capture the structured logical flow required for legal analysis, thereby creating a more specialised and higher-quality latent vector space. Compared with BERT-base, LegalBERT showed stronger performance in the tasks of text classification and sequence tagging using EURLEX57K, ECHR-CASES and cases from the European Court of Human Rights.

E. The Value of Domain Adaptation

The superior performance of LegalBERT is not an isolated finding but is consistent with a broader trend in NLP literature demonstrating the critical value of domain-specific adaptation.

Zheng et.al (2022) [32] showed that domain-adaptation for architecture, engineering and construction (AEC) industry information retrieval provided improvements on BERT-base.
Lee et.al (2020) [33] pre-trained BERT-base on biomedical articles to get the BIOBERT model which reported performance improvements on biomedical datasets.
Following on from this, Alsentzer et.al (2019) [34] further trained BIOBERT on clinical notes resulting in further improvement.
Other examples of domain adapted BERT models improving in performance include SCIBERT [35] and PharmBERT [36].

F. Study Scope

While the advanced approaches discussed in this section represent the cutting edge of LCR, the goal of this study is not to engineer a novel SOTA pipeline. Instead, this research performs a diagnostic and comparative analysis to understand the fundamental capabilities of domain-adapted models. By comparing the baseline performance of BM25, BERT-base, and LegalBERT, and by using PCA to probe the intrinsic dimensionality of their respective latent spaces, this study aims to reveal how pre-training affects the organization of legal semantics for scalable retrieval.

III. Methodology

A. Problem Statement and Data

For COLIEE task 1 we are given a corpus of Federal Court of Canada case laws. We say that a case is 'noticed' to a query if the case is referenced by the query case. As our test and training labels we are given a list of cases within the corpus each with their own set of 'noticed' cases. This is presented in a JSON file of the following format

```
{
  "000001.txt": ["000005.txt", "012101.txt"],
  "003423.txt": ["398421.txt", "012101.txt",
  "173651.txt"],
  "012831.txt": ["000001.txt"],
  ...
}
```

The goal of the COLIEE task is, for each query case (which is also part of the corpus) to return a subset of the corpus that is noticed by that query case. The performance is then judged off of F1 score, precision and recall. To simplify our …, in this paper we alter this aim slightly, instead our pipelines are built to rank each case in the original corpus, based on how likely we think they are to be noticed. The performance is then judged on metrics such as Mean Reciprocal Rank, Precision@K, Recall@K, F1-score@K.

The training set contains 7350 documents in the corpus as well as 1678 query cases. The test dataset contains 2159 documents with 400 query cases. Since none of our pipelines required training, we just used the train dataset.

B. Preprocessing

Raw Canadian case law files contain substantial noise, including <FRAGMENT_SUPPRESSED> placeholders, editor footnotes, page headers, court metadata, and "[End of document]" markers. We applied aggressive but safe cleaning: all editor lines, page numbers, fragment markers, and standard boilerplate headers were removed using regular expressions. Leading judge name lines (e.g., "Manson, J.") and "[Translation]:" prefixes were stripped, and empty or near-empty lines (containing only punctuation) were discarded. The remaining text was converted to lowercase and collapsed to single whitespace. This preprocessing preserves substantive legal content while eliminating token budget waste and ensuring consistent, embeddable strings across the corpus. The same procedure was applied for the BM25 pipeline and the BERT based pipelines.

C. Sparse Information Retrieval Basline: BM25

To provide a robust, interpretable baseline for comparison against our dense retrieval pipeline we implemented a standard Sparse Information Retrieval (IR) pipeline utilizing the BM25 algorithm. Sparse IR models like BM25 rely on lexical matching and term frequency rather than semantic meaning. BM25 is a powerful and computationally efficient algorithm that serves as a probabilistic extension upon the earlier Term Frequency-Inverse Document Frequency (TF-IDF) [7]. It extends upon the basic TF-IDF framework by incorporating principles from the Probabilistic Retrieval Model (PRM). This probabilistic approach calculates the relevance score between a query and a document, framing it as the probability that the document is relevant given the query terms. The score is based on three main factors: (a) the Term Frequency of query terms in a document, (b) the Inverse Document Frequency of those terms across the whole corpus and (c) a document length normalisation factor (to prevent excessively long documents from being favoured).

In our pipeline, for a given query, the algorithm calculates a relevance score between this query and every document in the corpus. The documents ordered by relevance will form the model's prediction for likeliness to be noticed in the COLIEE task 1.

D. BERT Background

Our dense retrieval pipeline is founded on the BERT model, which is used to encode both the query case and corpus documents into fixed size vector embeddings. This section details the BERT fundamentals and specific strategies we employ to adapt it for document level legal case retrieval.

BERT employs an encoder only Transformer [37] architecture. It is a stack of Transformer encoder layers, each utilizing a self-attention mechanism to process the input sequence bidirectionality, allowing it to gather context from both the left and right sides of a sentence. This capability is what enables BERT to generate dynamic, contextual embeddings, which can

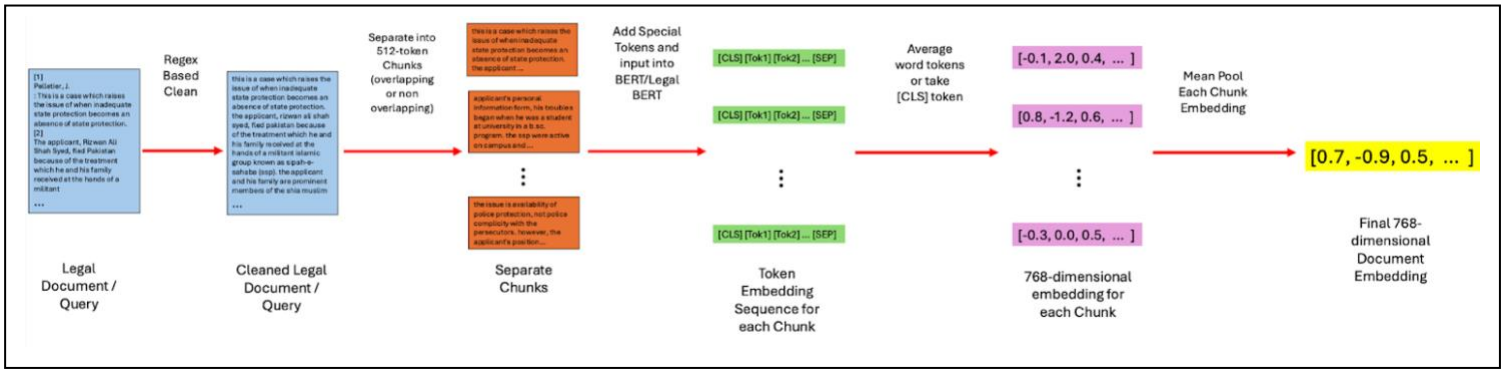

Figure 1: Pipeline for Embedding a Single Legal Document / Query

disambiguate word meaning and capture complex semantic nuance, overcoming the limitations of static models like Word2Vec and GloVe [14] [38].

The text input relies on WordPiece tokenization [39], which dissects the vocabulary to convert text into a list of IDs, which are then mapped to an initial vector using a learned embedding matrix.

BERT is pre-trained on a massive general purpose corpus using two unsupervised tasks. The first is Masked Language Modelling (MLM) where the model is trained to predict words that have been randomly masked in the input sequence. Secondly, BERT is trained on Next Sentence Prediction (NSP), where the model must predict whether two sentences are contiguous in the original text. As BERT is not specifically designed for these tasks but instead to generate token level representations of text, for any downstream task (including training), a prediction head must be concatenated to the model. Training on these tasks enables the outputs of the BERT model (without the prediction head) to meaningfully encode semantic meaning in text.

E. The CLS Token and Task Adaptation

The classification token, [CLS], is always prepended as the first token of every input sequence. Since the prediction head of the NSP task is based on the CLS token alone, after training it is able to serve as the sentence level summary representation of the entire input. Consequently, we can interpret the final state of each word token to capture what that word means in the context of the sentence.

F. Generating Case Level Embeddings

Legal cases often exceed the standard 512 token input limit of BERT, presenting a significant challenge for document encoding. The naive method, implemented for our baseline for producing a single representative vector for an entire legal case, is done by segmenting the document and aggregating the resulting chunk embeddings.

To manage document length, each case is divided into smaller equal size chunks (with 512 tokens). We compare two chunking strategies. First we try a sliding window where chunks are created with overlap (216 token overlap), which helps preserve context and continuity across chunk boundaries. Secondly we try splitting the document based on contiguous and non overlapping chunks.

In a given chunk, the BERT model will give us encodings for each token, thus it remains how we should combine these to get an encoding for the whole chunk. We try taking the [CLS] token as a representation of the entire chunk as well as averaging out all of the word tokens (excluding [CLS] and other special tokens).

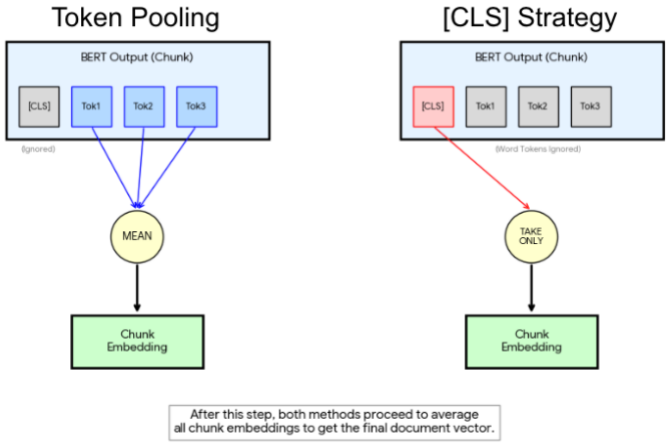

Figure 2: difference between token pooling or taking [CLS] tokens for a chunk level embedding

For both of these strategies we applied mean pooling of the chunk level encodings to then get a document level encoding.

G. Full BERT based Retrieval and Ranking Pipeline

Using the pipeline outlined in figure 1, with a specific choice of BERT/LegalBERT, overlapping/non-overlapping sliding window and word-token/CLS tokens we produce a 768-dimensional embedding for each document in the corpus.

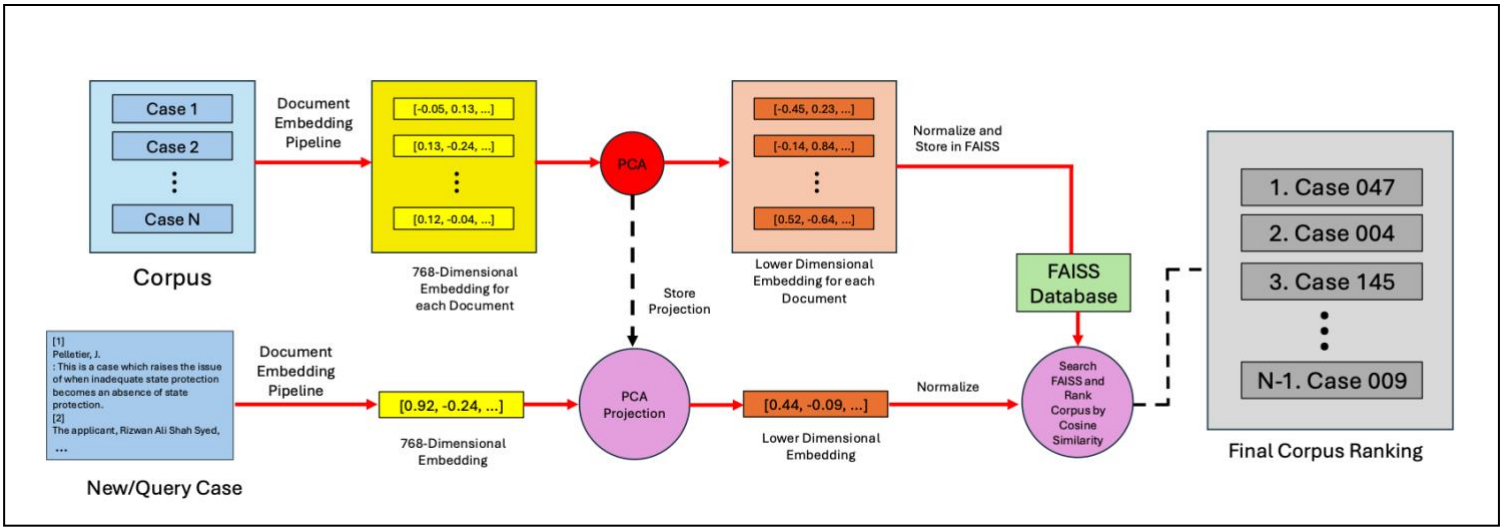

Figure 3: Full BERT based pipeline

We then experimented with using Principal Component Analysis (PCA) to project the embeddings of the corpus onto a lower dimensional space. We varied using no PCA, to projecting onto a 348 and 128 dimensional subspaces, storing the projection used each time. Following this each vector was normalised to unit length.

Then for each query case in the test set we would gather a document embedding using the same pipeline and if necessary, project the final vector down using the same projection, and then scale it down to unit length. Then to gather the final ranking of likelihood of being 'noticed', we sorted all cases in the corpus (excluding the query case) based on cosine similarity to the final unit vector of the query. This pipeline is described in figure 3.

H. Experiment Setting and Metrics

For a given query case in the test set, the BM25 pipeline and the BERT based pipelines all produce a ranking of the corpus which is then compared to the true list of 'noticed' cases.

Formally we have a query case q with a corresponding set of true 'noticed' cases $N_q = \{q_1, q_2, \dots, q_L\}$ and a permutation of the corpus (excluding the query case) ranked from most to least likely to be 'noticed' $[x_1, x_2, \dots, x_{N-1}$. We compute the mean reciprocal rank

$$MRR = \frac{1}{|N_q|} \sum_{i \in N_q} \frac{1}{rank(i)}.$$

For a given K we can get the K most likely cases to be noticed $[x_1, \dots, x_K]$. Then comparing the K most likely cases to the true amount of noticed cases we can get a count of the True Positives (TP), False Positives (FP) and False Negatives (FN) which allows us to compute the following metrics

$$Recall@K = \frac{TP}{TP + FN},$$

$$Precision@K = \frac{TP}{TP + FP},$$

$$F1@K = \frac{2 \cdot Recall@K \cdot Precision@K}{Recall@K + Precision@K}.$$

For each proposed pipeline we computed the mean of the MRR, R@10, P@10 and F1@10 metrics across the entire test set.

I. Software Suite

The entire retrieval pipeline was developed in Python 3.10, with computational tasks accelerated by NVIDIA GPU hardware and the PyTorch framework. The implementation relied on distinct libraries for the dense and sparse retrieval components. Most of the code was run on Kaggle TPU's for faster performance.

The Dense Retrieval Pipeline (BERT-base and LegalBERT) was primarily built using the Hugging Face Transformers library for loading and managing the pre-trained models, specifically "bert-base-uncased" and "nlpaueb/legal-bert-base-uncased". All tensor operations and model execution were handled by PyTorch. The efficiency of the vector search operations, including the final calculation of Cosine Similarity across the corpus, was achieved using the highly optimized Faiss library. Finally, data manipulation and metrics calculation utilized NumPy and Pandas.

The Sparse Retrieval Pipeline using the BM25 algorithm was implemented using BM250kapi, a dedicated toolkit that provides efficient indexing and querying functionality for the lexical retrieval component.

IV. Results

In this section, we present the comparative performance of the Sparse BM25 baseline against the Dense BERT-base and LegalBERT pipelines. Performance is primarily evaluated on Mean Reciprocal Rank (MRR), as this metric best reflects the system's ability to place relevant 'noticed' cases higher in the retrieval list. We also report the Precision@K, Recall@K and F1@K (for K=10) to provide a more holistic view of the retrieval quality.

A. Sparse vs. Dense Retrieval Performance

The comparison between the sparse and dense retrieval approaches demonstrates that the Sparse IR baseline (BM25) achieved the highest performance across all metrics. As shown in Table 1, BM25 achieved the best Mean Reciprocal Rank (MRR) with 0.233.

In comparison the best performing dense model configuration (LegalBERT with token-pooling, overlapping sliding windows and PCA reduction to 348 dimensions) achieved a mean MRR of 0.178. This represents a performance gap of approximately 24%. Similarly BM25 achieved the highest mean values of Recall@10, Precision@10 and F1@10 with values of 0.070, 0.220, 0.096 respectively compared to 0.052, 0.170 and 0.072 for the best dense models.

Model	MRR	Precision@10	Recall@10	F1@10
BM25	0.233	0.070	0.220	0.096
BERT-base	0.153	0.043	0.139	0.059
LegalBERT	0.178	0.052	0.169	0.072

Table 1: The performance of the sparse (BM25) pipeline vs the best configuration BERT-base (token-pooling, overlapping sliding window, PCA to 348 dimensions) and LegalBERT (token pooling, overlapping sliding window, PCA to 348 dimensions) pipelines.

B. Impact of Domain Specific Pre-training

We observed that when controlling for preprocessing variables (Token-pooling, overlapping/non overlapping sliding windows and domain reduction), LegalBERT consistently outperformed BERT-base as shown in figure 4. Across all configurations, the LegalBERT pipeline outperformed the BERT-base pipeline by an average of 22% in MMR. On the optimal configuration, the LegalBERT pipeline outperformed BERT-base by 0.025 in MMR corresponding to a 16% boost.

C. Intrinsic Dimensionality and PCA Analysis

We analyzed the effect of dimensionality reduction on retrieval performance using Principal Component Analysis (PCA). As demonstrated in table 2, both the 348 and 128 dimensional projections resulted in higher MMR scores than the full 768-dimensional embedding. This suggests that reducing dimensionality increases MRR but only up to a certain point.

Dimension	Mean MRR		
	BERT-base	LegalBERT	All
128	0.134	0.162	0.148
348	0.140	0.169	0.154
768 (no PCA)	0.129	0.158	0.143

Table 2: Mean MMR values averaged over all configurations of the dense pipelines

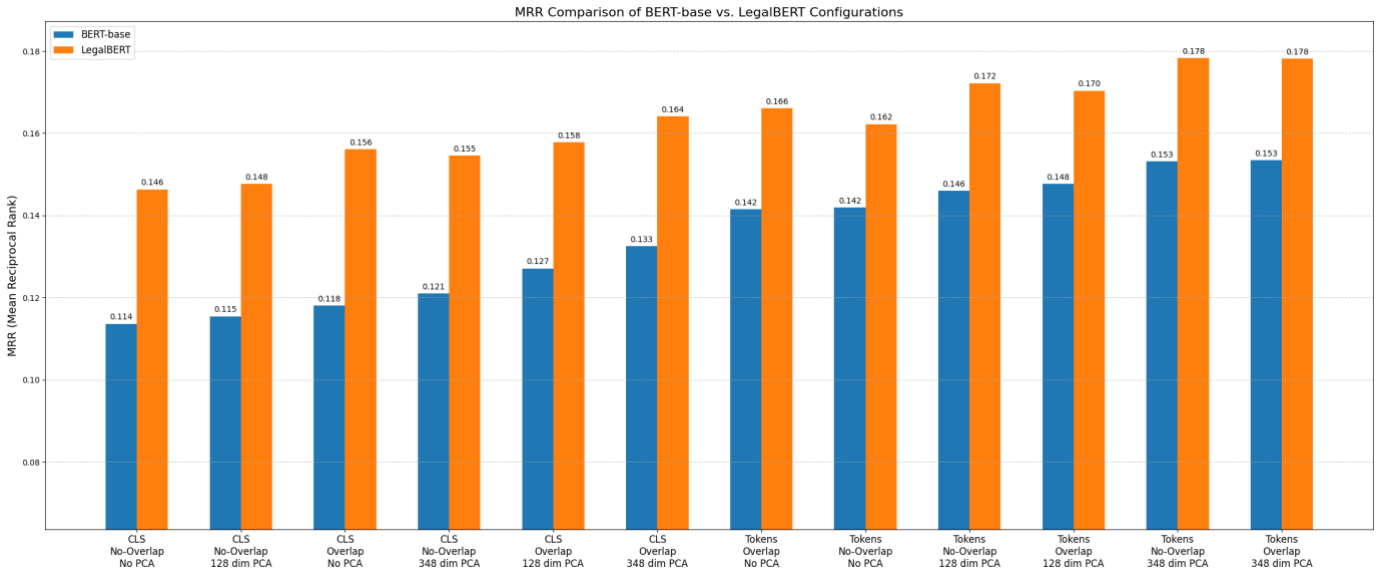

Figure 4: Relative performance on mean MMR over all dense pipeline configurations between BERT-base and LegalBERT

D. Ablation: Pooling and Segmentation Strategies

We also analyzed the impact of documentation segmentation (chunking) and aggregation (pooling) strategies to determine the most effective method for encoding the long legal documents.

D-1 Token vs [CLS] Pooling

Our results indicate that mean-pooling all word tokens is consistently superior to using the [CLS] chunk summary tokens. Across all model configurations, token pooling yielded higher MMR scores. For the general-purpose BERT-base, token pooling outperformed [CLS] pooling by an average of ~20-25%. For the domain-adapted LegalBERT, the gap was smaller but still distinct at approximately ~6-10%. This suggests that for long legal texts, a single classification token is insufficient to capture the dense semantic information required for relevance. Averaging the embeddings of all tokens preserves a richer representation of the legal context.

D-2 Impact of Overlapping/Non-Overlapping Windows

We observed a distinct interaction between the pooling strategy and the window stride (overlap). The performance of token pooling was remarkably robust to window sizing. Switching from non-overlapping windows to overlapping windows resulted in negligible performance changes (e.g., the MRR delta was often less than 0.002). In contrast, [CLS] pooling showed a consistent, albeit marginal, improvement when using overlapping windows. Reducing the stride to 256 resulted in an average MRR increase of approximately 0.01 across configurations. While small in absolute terms, this improvement was present in every test case, suggesting that [CLS] tokens are more sensitive to boundary effects than mean-pooled representations.

Hence, we conclude that mean-pooling word tokens appears to be the more effective and robust strategy for this task, achieving higher performance without requiring the computational overhead of overlapping windows.

V. Discussion

A. Persistence of the Lexical Gap

A primary finding of this report is the resilience of the sparse retrieval baseline (BM25) against the dense retrieval architectures. Despite the theoretical ability of Transformer models to capture semantic nuance [9], BM25 achieved a Mean Reciprocal Rank (MRR) of 0.233, outperforming the best dense configuration (LegalBERT with PCA) which achieved 0.178. This performance gap of approximately 24% highlights specific constraints within the legal domain and the architectural limitations of "naive" dense retrieval.

Legal Case Retrieval (LCR) differs significantly from general open domain question answering. Relevance in case law often hinges on precise lexical markers, specific statute numbers or rigid legal phrases (e.g “onus of proof") rather than broad semantic similarity, which the dense pipelines were designed to capture. BM25, which relies on exact term matching and frequency, naturally excels at capturing these high signal keywords. When a rare but critical identifier is tokenized and embedded, its specific identity can be “smeared” or diluted amongst the mean pooling on the high dimensional vector space, leading to the retrieval of documents that are topically similar but legally irrelevant.

The underperformance of our dense models can be attributed to what Khattab and Zaharia [25] describe as the limitation of "representation-focused" retrieval. Our pipeline relied on mean pooling, which forces the model to aggregate thousands of tokens into a single fixed-size vector.

Primarily, single vector BERT approaches that rely on mean pooling fail in LCR because they average out the specific, fine-grained details of key facts and judicial reasoning, resulting in a low-resolution embedding. This aggregation creates a critical loss of signal. Legal cases are not referenced because they are about the same general topic, but more because one specific part of a case is relevant to another specific part of that case. For example, a precedent might be cited solely for its definition of "gross negligence," even if the rest of the case facts are factually distinct. By averaging the embedding for "gross negligence" with hundreds of other tokens representing general legal boilerplate, the specific signal is washed out.

Fundamentally, these single case dense embeddings, in their compression of information, fail to identify the key circumstances and elements that might make a case relevant even when the general semantic meaning may not be. While the model successfully clustered cases by broad legal domain (e.g., "Immigration Law"), it lacked the resolution to distinguish the specific legal mechanisms required for prediction.

This dominance of sparse retrieval is consistent with the broader landscape of the COLIEE 2025 competition [17]. An analysis of top-performing teams in the proceedings reveals that pure dense retrieval is rarely used in isolation. For instance, the SIL team employed a cascading framework that relied on lexical features alongside semantic ones to reduce the search space, while UQLegalAI utilized a graph-based approach to capture connectivity that pure embeddings miss. The fact that our dense pipeline could not beat BM25 validates the industry standard of using Hybrid Systems, where BM25 is used for high-recall initial retrieval (the "first stage"), and dense models are reserved for re-ranking a smaller subset of candidates where semantic understanding is required to distinguish between lexically similar cases.

B. Efficacy of Domain Adaptation (LegalBERT vs. BERT-base)

While sparse models dominated the overall leaderboard, a direct comparison between the dense architectures reveals the critical value of domain-adapted BERT models. Across all comparable configurations, LegalBERT consistently

outperformed BERT-base (see figure 4) achieving a relative improvement of ~15-25% in MMR.

This performance disparity validates the hypothesis that legal text constitutes a distinct “sublanguage” [40] [41] which general-purpose models fail to fully capture [15]. BERT-base is trained on a generic corpora and consequently, its latent space is organized around general semantic relationships where words like "consideration" or "party" retain their common usage definitions.

In contrast, LegalBERT was pretrained on specialised legal text which allows it to learn a specialised vector space where polysemous terms are disambiguated correctly (e.g., "consideration" as a contractual exchange rather than "thoughtfulness"). By aligning the pre-training distribution with the downstream task distribution, LegalBERT reduces the "domain shift" that typically degrades the performance of transfer learning in specialised fields like AEC or Law [32].

A nuanced finding from our ablation study provides deeper insight into how this adaptation improves performance. As noted in Section IV-D, while token-pooling (averaging all word vectors) outperformed [CLS]-pooling (using the single summary token) for both models, the performance gap was significantly smaller for LegalBERT (~6-10%) than for BERT-base (~20-25%). This discrepancy suggests a fundamental difference in the quality of the classification token itself.
During training, the BERT-base [CLS] is trained to aggregate information from general encyclopedic text. It struggles to generate a representative summary vector for a complex legal document because it was never taught to prioritize legal concepts over general narrative elements. Conversely the LegalBERT [CLS] token has effectively learned to function as a "legal summarizer." It is optimized to aggregate complex statutory logic and judicial reasoning into a single vector representation.

C. Dimensionality of Latent Space Representations

Perhaps the most counter-intuitive finding of this study was the impact of dimensionality reduction on retrieval performance. Standard assumptions in deep learning suggest that compressing high-dimensional vectors (768d) into lower-dimensional spaces (128d or 348d) should result in information loss and, consequently, lower retrieval accuracy.

However, our results demonstrate that dimensionality reduction actually acted as a performance enhancer. For the LegalBERT pipeline, reducing the vector size from 768 to 348 dimensions improved the MRR from 0.143 to 0.154 on average, and even the highly compressed 128-dimensional vectors outperformed the full 768-dimensional embeddings (0.148 vs 0.143 on average).

We posit that this improvement is due to the PCA "denoising" effect. The raw 768-dimensional output of BERT models is known to be anisotropic—meaning the embeddings occupy a narrow cone in the vector space rather than being uniformly distributed. When using a "naive" mean-pooling strategy, as we did in this study, the resulting case vectors likely aggregate not only the semantic signal (the legal reasoning) but also the syntactic noise and common-frequency words that dominate the principal components of the vector space.

This finding independently validates the architectural choices made in state-of-the-art late-interaction models like ColBERT. Khattab and Zaharia (2020) [25] explicitly design ColBERT to project BERT's standard 768-dimensional output down to much smaller dimensions (e.g., 128) to make their "MaxSim" interaction step computationally feasible. They reported that even aggressive compression (down to 24 dimensions) resulted in negligible performance loss compared to full-size embeddings.

Our results confirm that this is not merely a computational shortcut, but a sound strategy. The core semantic information required for legal relevance is preserved, and potentially even clarified, in lower-dimensional spaces. This suggests that future legal IR systems can aggressively compress vectors to reduce index size without fearing a loss of retrieval accuracy.

D. Architectural Nuance

A detailed ablation of our pooling strategies reveals a significant interaction between the model architecture and the aggregation method. Consistent with our initial hypothesis, token-pooling (mean-pooling of all word vectors) outperformed [CLS]-pooling across all configurations.

For the general-purpose BERT-base, the gap was substantial: token-pooling outperformed [CLS]-pooling by approximately 20–25% in MRR. This confirms that the [CLS] token, which is optimized during pre-training for a generic "Next Sentence Prediction" task on Wikipedia text, fails to capture the dense, specific semantic information required for legal relevance when summarizing chunks with 512 tokens. It acts as a "lossy" summary that discards too much critical detail.

In contrast, for LegalBERT, this performance gap shrank significantly to 6–10%. This suggests that while token-pooling remains mechanically superior for long documents, LegalBERT's [CLS] token is more suitable for this task. Because it was pre-trained on legal corpora (including case law and legislation), its internal attention mechanisms have learned to aggregate information in a way that preserves more legally relevant signals than the general-purpose model. However, the fact that token-pooling still wins indicates that even a domain-adapted summary token cannot fully replace the granular signal provided by averaging the entire document's token embeddings.

E. Limitations and Future Work

While this study establishes a strong baseline for domain adaptation in legal retrieval, several limitations in our "naive"

diagnostic approach point toward fruitful avenues for future research.

E-1 Representation Bottleneck

Our reliance on mean-pooling (and even PCA-reduced embeddings) inherently suffers from an information bottleneck. Compressing a large legal case into a single vector inevitably washes out fine-grained details, such as specific citational relationships or dissenting opinions, which are often the decisive factors in relevance. Future work should move beyond "representation-focused" architectures to "interaction-focused" models like ColBERT. By retaining token-level embeddings and computing relevance via a "late interaction" (MaxSim) step, such models can match specific legal tests between cases without the lossy compression of pooling. Furthermore, validating whether our PCA efficiency gains hold true for ColBERT's multi-vector architecture would be a critical next step.

E-2 Sentence Level BERTs: Optimising [CLS] token

Our results showed that the [CLS] token, while improved by domain adaptation in LegalBERT, still lagged behind token pooling. This is expected, as the native BERT [CLS] token is not optimized for cosine similarity, typically yielding poor sentence embeddings when used out-of-the-box [27]. In response to this, Siamese-network training patterns like in Sentence-BERT have been developed to help optimise the [CLS] token to represent semantic meaning through training on a cosine-similarity loss. Future work should be dedicated to re-implementing siamese network training using domain adapted BERT bases such as what was done by Chauhan [42].

E-3 The Hybrid Pipeline: Combining Exact Match with Semantic Density

Given that BM25 outperformed all dense models in this study, and dense models excelled at capturing domain-specific semantics, a Hybrid Pipeline represents the most logical evolution. Pure dense retrieval models often fail to capture specific legal identifiers (e.g., statute numbers), whereas sparse models miss synonymous legal concepts. A production-grade system should utilize BM25 as a fast, high-recall "first stage" retriever to filter the massive corpus down to a manageable subset, followed by a Legal Sentence BERT or ColBERT "re-ranker" to apply semantic reasoning to the candidates. This approach aligns with the methodologies of top-performing teams in COLIEE 2025 [17], which employed multi-stage retrieval integrating lexical filtering with semantic re-ranking.

E-4 Full Survey of COLIEE Pipelines

An interesting avenue of research would be to review a collection of SOTA COLIEE task 1 pipelines that rely on BERT for some base. Re-implementing these advanced pipelines with a LegalBERT base instead of BERT-base would further demonstrate the effectiveness of domain-adapted BERT models in specialised tasks.

VI. Conclusion

In this study, we established a comparative baseline for Legal Case Retrieval (LCR) on the COLIEE 2025 Task 1 dataset, evaluating the performance of sparse (BM25) and naive dense retrieval architectures (BERT-base and LegalBERT). Our results clearly demonstrate the superiority of the sparse BM25 baseline (MRR of 0.233) over the best dense configuration (MRR of 0.178), validating that LCR's reliance on precise lexical markers creates a persistent lexical gap that single-vector dense models struggle to bridge. Crucially, the comparison between dense models confirmed the critical value of domain adaptation, with LegalBERT consistently outperforming BERT-base by approximately 15–25% in MRR, and its classification token proving to be a significantly more effective legal summarizer. Finally, we showed that Principal Component Analysis (PCA) acts as a performance enhancer, suggesting that lower-dimensional embeddings are both more efficient and "denoised." These findings collectively validate the industry-standard approach: a Hybrid Pipeline, using BM25 for high-recall initial retrieval, followed by a LegalBERT-based re-ranker, represents the most logical and effective evolution for production-grade LCR systems.

References

[1] Y. Feng, C. Li, and V. Ng, "Legal Case Retrieval: A Survey of the State of the Art," in Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand: Association for Computational Linguistics, 2024, pp. 6472–6485. doi: 10.18653/v1/2024.acl-long.350.

[2] K. R. Chowdhary, "Natural Language Processing," in Fundamentals of Artificial Intelligence, New Delhi: Springer India, 2020, pp. 603–649. doi: 10.1007/978-81-322-3972-7_19.

[3] M. Arslan, H. Ghanem, S. Munawar, and C. Cruz, "A Survey on RAG with LLMs," Procedia Comput. Sci., vol. 246, pp. 3781–3790, 2024, doi: 10.1016/j.procs.2024.09.178.

[4] P. Sun, Y. Chen, X. Li, and X. Chu, "The Multi-Round Diagnostic RAG Framework for Emulating Clinical Reasoning," Aug. 05, 2025, arXiv: arXiv:2504.07724. doi: 10.48550/arXiv.2504.07724.

[5] H.-C. Lee, K. Hung, G. M.-T. Man, R. Ho, and M. Leung, "Development of an RAG-Based LLM Chatbot for Enhancing Technical Support Service," in TENCON 2024 - 2024 IEEE Region 10 Conference (TENCON), Singapore, Singapore: IEEE, Dec. 2024, pp. 1080–1083. doi: 10.1109/TENCON61640.2024.10902801.

[6] J. Singh, "Combining Machine Learning and RAG Models for Enhanced Data Retrieval: Applications in Search Engines, Enterprise Data Systems, and Recommendations," J. Comput. Intell. Robot., vol. 3, no. 1, pp. 163–204, Mar. 2023.

[7] P. Bafna, D. Pramod, and A. Vaidya, "Document clustering: TF-IDF approach," in 2016 International Conference on Electrical, Electronics, and Optimization

Techniques (ICEEOT), Chennai, India: IEEE, Mar. 2016, pp. 61–66. doi: 10.1109/ICEEOT.2016.7754750.
[8] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," Found. Trends® Inf. Retr., vol. 3, no. 4, pp. 333–389, 2009, doi: 10.1561/1500000019.
[9] J. Wang et al., "Utilizing BERT for Information Retrieval: Survey, Applications, Resources, and Challenges," ACM Comput. Surv., vol. 56, no. 7, pp. 1–33, July 2024, doi: 10.1145/3648471.
[10] H. S. Walsh and S. R. Andrade, "Semantic Search With Sentence-BERT for Design Information Retrieval," in Volume 2: 42nd Computers and Information in Engineering Conference (CIE), St. Louis, Missouri, USA: American Society of Mechanical Engineers, Aug. 2022, p. V002T02A066. doi: 10.1115/DETC2022-89557.
[11] G. Izacard et al., "Unsupervised Dense Information Retrieval with Contrastive Learning," Aug. 29, 2022, arXiv: arXiv:2112.09118. doi: 10.48550/arXiv.2112.09118.
[12] K. Juvekar and A. Purwar, "COS-Mix: Cosine Similarity and Distance Fusion for Improved Information Retrieval," 2024, arXiv. doi: 10.48550/ARXIV.2406.00638.
[13] M. V. Koroteev, "BERT: A Review of Applications in Natural Language Processing and Understanding," 2021, arXiv. doi: 10.48550/ARXIV.2103.11943.
[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2018, arXiv. doi: 10.48550/ARXIV.1810.04805.
[15] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The Muppets straight out of Law School," 2020, arXiv. doi: 10.48550/ARXIV.2010.02559.
[16] Y. Ma et al., "Incorporating Structural Information into Legal Case Retrieval," ACM Trans. Inf. Syst., vol. 42, no. 2, pp. 1–28, Mar. 2024, doi: 10.1145/3609796.
[17] Randy Goebel et al., "Proceedings of the Workshop on the Twelfth International Competition on Legal Information Extraction and Entailment (COLIEE 2025)," in Proceedings of the Workshop on the Twelfth International Competition on Legal Information Extraction and Entailment (COLIEE 2025), COLIEE 2025 Organizers, June 2025. [Online]. Available: https://coliee.org/documents/Proceedings/2025-Proceedings.pdf
[18] H. Abdi and L. J. Williams, "Principal component analysis," WIREs Comput. Stat., vol. 2, no. 4, pp. 433–459, July 2010, doi: 10.1002/wics.101.
[19] J. Rayo, R. de la Rosa, and M. Garrido, "A Hybrid Approach to Information Retrieval and Answer Generation for Regulatory Texts," 2025, doi: 10.48550/ARXIV.2502.16767.
[20] K. W. Church, "Word2Vec," Nat. Lang. Eng., vol. 23, no. 1, pp. 155–162, Jan. 2017, doi: 10.1017/S1351324916000334.
[21] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.
[22] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 12, pp. 6999–7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.
[23] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional Attention Flow for Machine Comprehension," 2016, arXiv. doi: 10.48550/ARXIV.1611.01603.
[24] J.-Y. Jiang, M. Zhang, C. Li, M. Bendersky, N. Golbandi, and M. Najork, "Semantic Text Matching for Long-Form Documents," in The World Wide Web Conference, San Francisco CA USA: ACM, May 2019, pp. 795–806. doi: 10.1145/3308558.3313707.
[25] O. Khattab and M. Zaharia, "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, Virtual Event China: ACM, July 2020, pp. 39–48. doi: 10.1145/3397271.3401075.
[26] Y. Shao et al., "BERT-PLI: Modeling paragraph-level interactions for legal case retrieval.," in IJCAI, 2020, pp. 3501–3507.
[27] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," 2019, arXiv. doi: 10.48550/ARXIV.1908.10084.
[28] Y. Tang, R. Qiu, Y. Liu, X. Li, and Z. Huang, "CaseGNN: Graph Neural Networks for Legal Case Retrieval with Text-Attributed Graphs," in Advances in Information Retrieval, vol. 14609, N. Goharian, N. Tonellotto, Y. He, A. Lipani, G. McDonald, C. Macdonald, and I. Ounis, Eds., in Lecture Notes in Computer Science, vol. 14609. , Cham: Springer Nature Switzerland, 2024, pp. 80–95. doi: 10.1007/978-3-031-56060-6_6.
[29] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding," 2018, arXiv. doi: 10.48550/ARXIV.1804.07461.
[30] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," 2016, arXiv. doi: 10.48550/ARXIV.1606.05250.
[31] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "RACE: Large-scale ReAding Comprehension Dataset From Examinations," 2017, arXiv. doi: 10.48550/ARXIV.1704.04683.
[32] Z. Zheng, X.-Z. Lu, K.-Y. Chen, Y.-C. Zhou, and J.-R. Lin, "Pretrained Domain-Specific Language Model for General Information Retrieval Tasks in the AEC Domain," 2022, doi: 10.48550/ARXIV.2203.04729.
[33] J. Lee et al., "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," Bioinformatics, vol. 36, no. 4, pp. 1234–1240, Feb. 2020, doi: 10.1093/bioinformatics/btz682.
[34] E. Alsentzer et al., "Publicly Available Clinical BERT Embeddings," in Proceedings of the 2nd Clinical Natural Language Processing Workshop, Minneapolis, Minnesota, USA: Association for Computational Linguistics, 2019, pp. 72–78. doi: 10.18653/v1/W19-1909.

[35] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A Pretrained Language Model for Scientific Text," 2019, doi: 10.48550/ARXIV.1903.10676.
[36] T. ValizadehAslani et al., "PharmBERT: a domain-specific BERT model for drug labels," Brief. Bioinform., vol. 24, no. 4, p. bbad226, July 2023, doi: 10.1093/bib/bbad226.
[37] A. Vaswani et al., "Attention Is All You Need," 2017, arXiv. doi: 10.48550/ARXIV.1706.03762.
[38] A. Rogers, O. Kovaleva, and A. Rumshisky, "A Primer in BERTology: What We Know About How BERT Works," Trans. Assoc. Comput. Linguist., vol. 8, pp. 842–866, Dec. 2020, doi: 10.1162/tacl_a_00349.
[39] Y. Wu et al., "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," Oct. 08, 2016, arXiv: arXiv:1609.08144. doi: 10.48550/arXiv.1609.08144.
[40] P. M. Tiersma, Legal language, Paperback ed. Chicago, Ill.: Univ. of Chicago Press, 2000.
[41] R. Haigh, Legal English, Fifth edition. London New York: Routledge, Taylor & Francis Group, 2018.
[42] Jayendra Chauhan, "Legal-SBERT: Creating a Sentence Tranformer for the Legal Domain and Generating Data," Custom project, 2022.