Retrieval for extremely long queries and documents with RPRS: a highly efficient and effective transformer-based re-ranker

Askari, A.; Verberne, S.; Abolghasemi, M.A.; Kraaij, W.; Pasi, G.

Retrieval for Extremely Long Queries and Documents with RPRS: A Highly Efficient and Effective Transformer-based Re-Ranker

ARIAN ASKARI, SUZAN VERBERNE, AMIN ABOLGHASEMI, and WESSEL KRAAIJ,
Leiden University, The Netherlands
GABRIELLA PASI, University of Milano-Bicocca, Italy

Retrieval with extremely long queries and documents is a well-known and challenging task in information retrieval and is commonly known as Query-by-Document (QBD) retrieval. Specifically designed Transformer models that can handle long input sequences have not shown high effectiveness in QBD tasks in previous work. We propose a Re-Ranker based on the novel Proportional Relevance Score (RPRS) to compute the relevance score between a query and the top-k candidate documents. Our extensive evaluation shows RPRS obtains significantly better results than the state-of-the-art models on five different datasets. Furthermore, RPRS is highly efficient, since all documents can be pre-processed, embedded, and indexed before query time that gives our re-ranker the advantage of having a complexity of $O(N)$, where N is the total number of sentences in the query and candidate documents. Furthermore, our method solves the problem of the low-resource training in QBD retrieval tasks as it does not need large amounts of training data and has only three parameters with a limited range that can be optimized with a grid search even if a small amount of labeled data is available. Our detailed analysis shows that RPRS benefits from covering the full length of candidate documents and queries.

CCS Concepts: • **Information systems → Novelty in information retrieval;**

Additional Key Words and Phrases: Query-by-document retrieval · Sentence-BERT based ranking, Neural information retrieval

ACM Reference format:
Arian Askari, Suzan Verberne, Amin Abolghasemi, Wessel Kraaij, and Gabriella Pasi. 2024. Retrieval for Extremely Long Queries and Documents with RPRS: A Highly Efficient and Effective Transformer-based Re-Ranker. *ACM Trans. Inf. Syst.* 42, 5, Article 115 (April 2024), 32 pages.
https://doi.org/10.1145/3631938

1 INTRODUCTION

Query-by-document (QBD) retrieval is a task in which a seed document acts as a query—instead of a few keywords—with the aim of finding similar (relevant) documents from a document collection [31, 79, 80]. Examples of QBD tasks are professional, domain-specific retrieval tasks such

Authors' addresses: A. Askari, S. Verberne, A. Abolghasemi, and W. Kraaij, Leiden University, The Netherlands; e-mails: a.askari@liacs.leidenuniv.nl, s.verberne@liacs.leidenuniv.nl, m.a.abolghasemi@liacs.leidenuniv.nl, w.kraaij@liacs.leidenuniv.nl; G. Pasi, University of Milano-Bicocca, Italy; e-mail: gabriella.pasi@unimib.it.

1046-8188/2024/04-ART115 $15.00
https://doi.org/10.1145/3631938

as legal case retrieval [5, 65, 68, 83], patent prior art retrieval [19, 49, 51], and scientific literature retrieval [15, 42]. In the literature, three other terms are used to refer to this type of task: Query-by-Example retrieval, document similarity ranking, and document-to-document similarity ranking.

Transformer-based ranking models [69], such as BERT-based rankers [46, 78] have yielded improvements in many IR tasks. However, the time and memory complexity of the self-attention mechanism in these architectures is $O(L^2)$ for a sequence of length L [28]. For that reason, architectures based on BERT have an input length limitation of 512 tokens. This causes challenges in QBD tasks where we have long queries and documents. For instance, the average length of queries and documents in the legal case retrieval task of the **Competition on Legal Information Extraction and Entailment (COLIEE)** 2021 [5] is more than 5k words. Variants of Transformers that aim to cover long sequences such as LongFormer [7] and Big Bird [84] have not shown high effectiveness, which could be due to either their scattered attention mechanism [63] or to the limited number of training instances for QBD tasks [5]. To overcome these limitations, in this article we propose an effective and efficient Transformer-based re-ranker for QBD retrieval that covers the whole length of the query and candidate documents without memory limitations. We focus on re-ranking in a two-stage retrieval pipeline following prior studies that address retrieval tasks in a multi-stage retrieval pipeline [2, 13, 46, 47].

In domain-specific tasks, there is often the need for expensive professional searchers as annotators to create QBD test collections, e.g., lawyer annotators for case law retrieval tasks. This makes the annotation process for QBD tasks expensive and, as a result, low-resource training sets are a significant issue in these tasks [5]. One such example is the training set of the COLIEE 2021 dataset for case law retrieval that consists of only 3,297 relevant documents for 900 queries as training instances, which is very few compared to general web search datasets, e.g., MS MARCO [45] has approximately 161 times more training instances (532,761 queries). This limitation is overcome by our proposed method, a **Re-ranker based on Proportional Relevance Score computation (RPRS)**, which does not need large amounts of training data, because it has three parameters with a limited range that can be—but do not necessarily need to be—optimized using a grid search if a small amount of labelled data is available. Compared to BM25, our proposed method has only one more parameter, which we optimize over 10 different possible values. The two other parameters' ranges are equal to those of BM25.

Given a query document and a set of candidate documents, we split the query and document text into sentences and use **Sentence-BERT (SBERT)** bi-encoders [56] to embed the individual sentences. Then, a relevance score is computed by the RPRS relevance model we propose in this article, which is based on the similarity of individual sentences between a query and a document. The intuition behind the proposed method is based on the following assumption: a candidate document from the ranked list of documents in response to a query is more likely to be relevant if it contains a relatively high number of sentences that are similar to query sentences compared to other documents' sentences in the ranked list. Since RPRS has been designed based on this assumption, we conduct an experimental investigation to assess the quality of our proposed method by addressing the following research questions:

RQ1: What is the effectiveness of RPRS compared to the state-of-the-art models for QBD retrieval?

To answer this question we evaluate the effectiveness of RPRS on legal case retrieval (i.e., given a legal case, find the related cases) using the COLIEE 2021 dataset [53]. As RPRS uses a SBERT model to embed query and document sentences, we first run a series of experiments to find the SBERT model with which RPRS achieves the highest effectiveness. Additionally, we investigate two different unsupervised and self-supervised domain adaption approaches on the top-two most effective SBERT models without domain adaption. We conclude that RPRS achieves higher effectiveness

with domain-specific Transformers in the SBERT architecture when adapting them to the domain by using TSDA [73]. We continue the experiments based on this finding. We also compare RPRS to two prior models: (1) the state-of-the-art model on the COLIEE dataset, MTFT-BERT [2], and (2) **Self-Supervised Document Similarity Ranking (SDR)** [21], which also uses bi-encoders for sentence-level representation to compute relevance scores and is the state of the art on two Wikipedia document similarity datasets.

For relevance estimation with RPRS, we split both the query and document content into sentences. However, there are other ways for splitting query and document. Therefore, we investigate if sentences are the most appropriate and informative textual units for RPRS by addressing:

RQ2: How effective is RPRS with shorter or longer text units instead of sentences?

Next, for analyzing the cross-data generalizability of the proposed method we address the following question:

RQ3: What is the effectiveness of RPRS with parameters that were tuned on a different dataset in the same domain?

To address this question, we have evaluated the proposed method on the *Caselaw* dataset [36] with the parameters that were tuned on the COLIEE dataset without doing domain adaption on the collection for the Transformer model. In other words, our goal is to analyze how much the three optimized parameters of RPRS on the COLIEE dataset are transferable to another dataset in the same (i.e., legal) domain. Next, we assess the generalizability of our method by addressing RQ4.

RQ4: To what extent is RPRS effective and generalizable across different domains with different type of documents?

In this regard, we evaluate our method on two different domains using datasets of patents and Wikipedia webpages. For patents, we use CLEF-IP 2011 [51], a patent prior art retrieval dataset, and for Wikipedia we assess the effectiveness of our method on WWG and **Wikipedia wine articles (WWA)** datasets, which are two datasets for similarity ranking between Wikipedia pages in the wine and video game domains. We show that the proposed method achieves higher ranking effectiveness over the state of the art in all five datasets (COLIEE, Caselaw, CLEF-IP 2011, WWG, WWA).[1]

In summary, our contributions are as follows:

(1) We propose an effective and highly efficient re-ranker (RPRS) for QBD tasks that covers the full text of query and candidate documents without any length limitation. The efficiency of our method is justified by using bi-encoders and its effectiveness is evaluated on five datasets.
(2) RPRS is suitable for QBD retrieval datasets with low-resource training data, since it has only three parameters with limited range that can be optimized using a grid search even if a relatively low amount of labelled data is available.
(3) We show how the use of various SBERT models and adapting them on domain-specific datasets affects the effectiveness of RPRS. This results in three SBERT models in the legal, patent, and Wikipedia domains, which we will make publicly available on Huggingface.
(4) Our proposed model outperforms the state-of-the-art models for each of the five benchmarks, including SDR [21], which was also proposed as a sentence-based ranker for QBD tasks.
(5) An ablation study shows that Document Proportion—the proportion of the document that is relevant to the query—is the most important component in RPRS. In addition, we found the crucial role of covering the full length of queries and documents for QBD tasks by studying the effect of feeding RPRS with truncated queries and documents.

The structure of the article is as follows: After a discussion of related work in Section 2, we describe the proposed method in Section 3 and the details of the BM25 and SDR baselines in Section 4.

[1]The implementation is available on https://github.com/arian-askari/rprs

The experiments and implementation details are covered in Section 5. The results are examined and the research questions are addressed in Section 6. In Section 7, we investigate and discuss our proposed method in more-depth. Finally, the conclusion is described in Section 8.

2 RELATED WORK

In the following, we first introduce QBD retrieval tasks. Then, we provide an overview on prior methods for sentence embedding and pre-training. Finally, we give an overview of previous works on sentence-level retrieval methods.

2.1 QBD Tasks

Legal case retrieval. In countries with *common law* systems, finding supporting precedents to a new case, is vital for a lawyer to fulfill their responsibilities to the court. However, with the large amount of digital legal records—the number of filings in the U.S. district courts for total cases and criminal defendants was and is 544,460 in 2020[2]—it takes a significant amount of time for legal professionals to scan for specific cases and retrieve the relevant sections manually. Studies have shown that attorneys spend approximately 15 hours in a week seeking case law [30]. This workload necessitates the need for information retrieval (IR) systems specifically designed for the legal domain. The goal of these systems is to assist lawyers in their duties by exploiting AI and traditional information retrieval methods. One particular legal IR task is case law retrieval.

The majority of the work on case law retrieval takes place in COLIEE [53, 55]. Askari and Verberne [5] combine lexical and neural ranking models for legal case retrieval. They optimize BM25 and obtain state-of-the-art results with lexical models. However, recently, on top of the optimized BM25, Abolghasemi et al. [2] present multi-task learning as a re-ranker for QBD retrieval and set the new state of the art. The limitation of the method in Reference [2] is that the input is limited by the BERT architecture (512 tokens) [17]. Therefore, it cannot cover the full text of long documents in QBD datasets like COLIEE.

Patent prior art retrieval. Patents serve as stand-ins for various domains including technological, economic, and even social activities. The **Intellectual Property (IP)** system encourages the disclosure of innovative technology and ideas by granting inventors exclusive monopoly rights over the economic value of their inventions. Therefore, according to Piroi et al. [51], patents have a considerable impact on the market value of companies. With the number of patent applications filed each year continuing to rise, the need for effective and efficient solutions for handling such huge amounts of information grows more essential. There are various patent analysis tasks. In this work, we focus on patent prior art retrieval, the aim of which is to find patent documents in the target collection that may invalidate a specific patent application [49, 51].

Patent retrieval is a challenging task: The documents are lengthy, and the language is formal, legal, and technical with long sentences [70]. Shalaby et al. [64] provide a detailed review of research on patent retrieval that shows that the most successful methods on the CLEF-IP [51] benchmarks are traditional lexical-based methods [50, 71], sometimes extended with syntactic information [18]. There are more recent studies that focus on patent retrieval in passage-level assessment [4, 25]. However, recent studies do not address document-level patent retrieval due to the limitation of Transformer-based methods on taking into account the full length of the documents. Mahdabi and Crestani [40] propose an effective approach for patent prior art retrieval, which is based on collecting a citation network using a specific API that is not provided by the organizers of the CLEF-IP dataset. Taking into account the fact that this setup is different from the original setup, we do not consider this approach as our baseline for a fair comparison.

[2] https://www.uscourts.gov/statistics-reports/judicial-business-2020

Document similarity for Wikipedia-based datasets. Estimating the similarity within Wikipedia pages is useful for many applications such as clustering, categorization, finding relevant web pages, and so on. Ginzburg et al. [21] propose two new datasets on Wikipedia annotated by experts. The documents come from the wine and video game domains; we refer to the collections as (1) **Wikipedia video games (WVG)** and (2) WWA. For both datasets, the task is finding relevant Wikipedia pages given a seed page.

2.2 Sentence Embedding

Sentence embedding is a well-studied topic and it is suitable for measuring sentence similarity, clustering, information retrieval via semantic search, and so on. BERT-based sentence embedding methods employ Transformer models to effectively and efficiently embed sentences. The challenge is that BERT's architecture makes it inappropriate for semantic similarity search in its original configuration, since it requires both sentences to be concatenated into the network and applies multiple attention layers between all tokens of both sentences, which has a significant computational overhead. For instance, to find the most similar pair in a collection of 10,000 sentences BERT needs roughly 50 million inference calculations, which takes around 65 hours [56].

Humeau et al. [26] introduce poly-encoders to tackle the runtime overhead of the BERT cross-encoder and present a method to calculate a score employing attention between the context vectors and pre-computed candidate embeddings. However, Poly-encoders have the disadvantage that their score function is not symmetric and that their computing cost is excessively high and requires $O(n^2)$ score calculations. Reimers and Gurevych [56] therefore proposed SBERT as a variant of the pre-trained BERT model that uses siamese and triplet network architectures to produce semantically meaningful embeddings that can be compared using cosine similarity. SBERT decreases the time it takes to find the most similar pair from 65 hours to around 5 seconds while keeping BERT's performance [56]. In this work, we utilize SBERT as our embedding model, making the efficiency of RPRS much higher than cross-encoder based re-rankers. We exploit a selection of SBERT models that are tuned on different datasets.

It is noteworthy to mention that while our proposed methods leverage bi-encoders, specifically Sentence-BERT Reimers and Gurevych [56], to re-rank the first-stage retriever's ranked list, our approach fundamentally diverges from dense passage retrievers that employ bi-encoders as first stage retrievers in terms of both methodology and application [27]. Dense retrievers emphasize on the alignment of query and relevant document representations by optimizing a targeted loss function, such as the mean square error between the query and relevant document vectors while we do not have such training in our proposed. Moreover, optimizing dense retrievers on the query-by-document task with extremely lengthy queries or documents is not computationally possible due to BERT model's maximum word limit of 512 words, which prevents them from representing each query or document by a single pass to a bi-encoder and, as a result, prevents them from being able to optimizing the representation of query and document, as they do not have one union representation. There could be future work on studying how applying dense passage retrieval on query by document task is out of the scope of this study.

2.3 Unsupervised Sentence Embeddings Training

Previous works on unsupervised sentence embeddings learning have achieved promising results on semantic textual similarity tasks by combining pre-trained Transformers with various training objectives. Carlsson et al. [10] propose Contrastive Tension, which views identical and different sentences as positive and negative examples, respectively, and trains two independent encoders. BERT-flow [32] debiases the embedding distribution toward Gaussian to train the model.

SimCSE [20] uses contrastive learning [12, 23] to classify identical sentences with different dropout masks as positive instances.

Wang et al. [73] propose an unsupervised state-of-the-art method called TSDAE that is based on Transformers and denoising auto-encoders that encode damaged sentences into fixed-sized vectors and require the decoder to reconstruct the original sentences from these sentence embeddings [73]. Wang et al. [74] propose **Generative Pseudo Labeling (GPL)**, which combines a query generator with pseudo labeling from a cross-encoder. However, the GPL methodology is not suitable for QBD tasks as it relies on cross-encoders that are limited both in length and efficiency. We evaluate the effect of two state-of-the-art pre-training methods in Section 6.

2.4 Sentence-level Retrieval

Addressing relevance of candidate documents by leveraging sentence-level evidence has been studied for long documents retrieval in the past years [3, 24, 33, 81, 82]. However, there is no work on sentence-based relevance score computation using Transformer-based models (i.e, SBERT [56]) on QBD retrieval tasks similar to our approach. Yilmaz et al. [81] apply inference at the sentence level for each document and aggregate the sentence-level inference by learning a weight for each top-scoring sentence in each candidate document. Zhang et al. [85] observe that the "best" sentence or paragraph in a document gives a decent proxy for document relevance, which was the inspiration for Yilmaz et al. [81] as well. We also use this intuition in our approach to QBD Retrieval, where both the query and the documents are long texts. Mysore et al. [41] propose a scientific document similarity model based on sentence-level similarity that leverages co-citation sentences as a source of document similarity.

Recently, Ginzburg et al. [21] proposed an unsupervised ranker called SDR that computes the final relevance score by computing two sentence- and paragraph-level matrices. They evaluate SDR's effectiveness on two new datasets annotated by human experts. We replicate SDR [21] as the most recent and comparable methodology to RPRS, because (1) although SDR's mechanism is dissimilar to RPRS fundamentally, it also is a sentence-level relevance scoring model designed for QBD tasks that covers the full length of both queries and candidate document texts using sentence embeddings, and (2) similarly to RPRS, SDR has a complexity of $O(N)$, where N is count of sentences—due to the utilization of bi-encoder sentence embeddings. Therefore, besides comparing RPRS with the state-of-the-art model on each dataset, we compare its effectiveness to SDR. SDR's architecture makes it suitable for both full-ranking and re-ranking settings.

3 PROPOSED METHOD: RPRS

As mentioned, we assume that a candidate document d is likely to be relevant to a query document q if a large proportion of d is similar to q and a large proportion of q is similar to d. A specific challenge in QBD tasks is that a document can be very long (more than 10k words), and it may contain several topics. For a long query document q, this can cause an irrelevant candidate document d positioned on top, because only one topic of d is very similar to one topic of q. Similarly, a long irrelevant candidate document could be ranked on top because one of its topics is very similar to only one topic of query document. We address this problem in our method by integrating the length of the query and document into Equations (4) and (5) that we elaborate on in the following. In the following, we define our concepts and proposed methodology.

3.1 Definitions

In Table 1, we show the legend of the symbols we employ to formally introduce our methodology. In our sentence-based relevance model, we first retrieve the set of documents TK_q for q with an initial ranker. Given q, we compute the cosine similarity of each $\overrightarrow{q_s}$ with each $\overrightarrow{d_s}$, i.e., we compute the

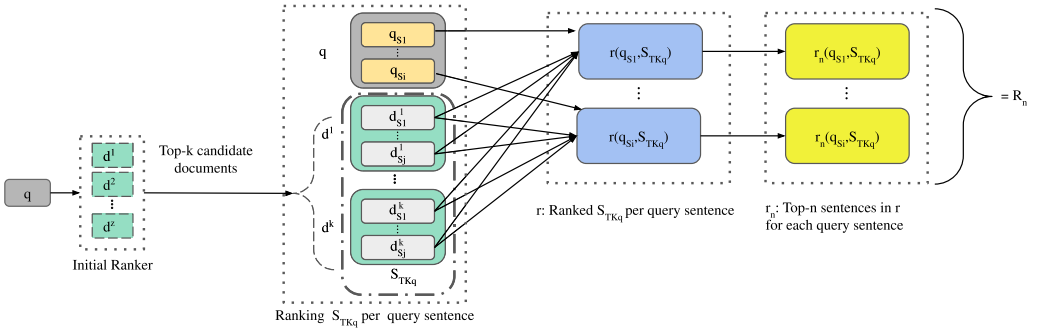

Fig. 1. The workflow of computing R_n given a query document q and set of top-k candidate documents; i and j refer to the last sentence of each query and document.

Table 1. Meaning of Symbols

Symbol	Meaning
q and d	query and candidate document
q_s and d_s	a sentence of q and d respectively
$\overrightarrow{q_s}$ and $\overrightarrow{d_s}$	vector representation (sentence embedding) of q_s and d_s
S_d and S_q	the sets of all sentences of d and q.
TK_q	set of top-k candidate documents retrieved for q
S_{TK_q}	set of all sentences from TK_q
$r(q_s, S_{TK_q})$	a ranked list of S_{TK_q} for q_s
$r_n(q_s, S_{TK_q})$	top-n sentences from $r(q_s, S_{TK_q})$
$R_n(S_q, S_{TK_q})$	contains all $r_n(q_s, S_{TK_q})$ for all sentences of S_q

The bold face is used to represent highest effectiveness in terms of a metric across the models.

similarity of each query sentence with each sentence in S_{TK_q}. The result $r(q_s, S_{TK_q})$ is a ranked list of sentences from S_{TK_q} for q_s. The set $r_n(q_s, S_{TK_q})$ contains the top-n sentences from $r(q_s, S_{TK_q})$, i.e., the document sentences most similar to q_s. From now on, we call a d_s "*most similar*" to q_s if d_s is a member of $r_n(q_s, S_{TK_q})$, meaning that it is among the top-n sentences from the ranked list for sentences from the top-k candidate documents. This indicates that the document sentence d_s is placed among the top-n highest most similar document sentences to the query sentence q_s.

The set $R_n(S_q, S_{TK_q})$ contains all $r_n(q_s, S_{TK_q})$ for all query sentences. Figure 1 shows the workflow of computing R_n step by step. Based on our assumption, d and q are likely to be relevant if a large proportion of S_d occurs in a large proportion of each r_ns from R_n.

3.2 Re-ranker based on Proportional Relevance Score

We formally define the proportional relevance score for *RPRS* as

$$RPRS(q, d, S_{TK_q}, n) = QP(S_q, S_d, S_{TK_q}, n) \times DP(S_d, S_q, S_{TK_q}, n), \quad (1)$$

where we use the **Proportional Relevance Score (PRS)**, QP is Query Proportion, and DP is Document Proportion. Given q, we compute PRS for d based on the R_n and S_{TK_q}. The parameter n controls the number of top-n similar sentences per query sentence in r_n and R_n. In the following, we define two functions, $q_{R_n}(S_q, S_d, S_{TK_q})$ and $d_{R_n}(S_d, S_q, S_{TK_q})$, that we use in the later equations for computing QP and DP,

$$q_{R_n}(S_q, S_d, S_{TK_q}) = \sum_{q_s}^{S_q} min(1, |S_d \cap r_n(q_s, S_{TK_q})|), \tag{2}$$

where $|x|$ denotes the cardinality of set x and the min function returns 1 if at least one of the sentences of d (S_d's sentences) is in $r_n(q_s, S_{TK_q})$, zero otherwise. Here we do not take into account the repetition of S_d's sentences in each r_n for a q_s because of the min function. However, we incorporate that in a controllable way in the extended variation ($RPRSw/freq$) in Section 3.3 by defining parameter $k1$. The function q_{R_n} counts for how many sentences of q, at least one sentence of the candidate document sentences (S_d) occurs at least one time in the set of q_s's r_n. Next, we define d_{R_n}, which is the main component for computing DP as

$$d_{R_n}(S_d, S_q, S_{TK_q}) = \sum_{d_s}^{S_d} min\left(1, \sum_{r_n}^{R_n} |\{d_s\} \cap r_n|\right), \tag{3}$$

where $\{d_s\}$ denotes a singleton, i.e., a set with only one sentence of S_d. For a query document q and a candidate document d, the function d_{R_n} iterates over all sentences of S_d and counts how many sentences of d occur at least one time in R_n. Given the parameter n, it is possible that more than one sentence of d occurs in r_n for a query sentence. We argue that Equations (2) and (3) are complementary to each other as each equation assess the relevance from either the query or candidate document perspective. We show the impact of each equation by the ablation study in Section 7.1. We now define QP and DP based on Equations (2) and (3),

$$QP(S_q, S_d, S_{TK_q}, n) = \frac{q_{R_n}(S_q, S_d, S_{TK_q})}{\text{count of } q\text{'s sentences}}, \tag{4}$$

$$DP(S_d, S_q, S_{TK_q}, n) = \frac{d_{R_n}(S_d, S_q, S_{TK_q})}{\text{count of } d\text{'s sentences}}. \tag{5}$$

It is noteworthy that the denominator of QP (count of q's sentences) has the same value for all candidate documents, and thus it could be ignored for ranking; however, we keep it as it makes relevance scores comparable for score analysis.

Intuitively, the relevance score of d should increase by having more of its sentences in R_n for query q. The advantage of this design is that the candidate document d receives the highest relevance score if all r_ns' sentences in R_n are from d. However, d receives the lowest relevance score if none of its sentences are in r_ns' sentences in R_n. However, there is a disadvantage in this design, as it does not consider the repeated appearances of candidate document sentences in r_n's. Therefore, we propose another variation on RPRS called RPRS w/freq that takes into account this in a controllable way explained in the next section.

3.3 Taking into Account Frequency (RPRS w/freq)

In the previous section, by using the minimum function in Equations (2) and (3), we only accounted for the occurrence of *at least one* of the candidate documents' sentences in r_n and R_n. In other words, we did not consider the frequency of the occurrences in our definition for functions q_{R_n}

and d_{R_n}. To empower our model, we define Fq_{R_n} and Fd_{R_n} as the modified versions of q_{R_n} and d_{R_n} that take into account the frequency,

$$Fq_{R_n}(S_q, S_d, S_{TK_q}) = \sum_{q_s}^{S_q} \frac{|S_d \cap r_n(q_s, S_{TK_q})|}{|S_d \cap r_n(q_s, S_{TK_q})| + k1\left((1-b) + \frac{b \cdot dl}{avgdl}\right)}, \tag{6}$$

$$Fd_{R_n}(S_d, S_q, S_{TK_q}) = \sum_{d_s}^{S_d} \frac{\sum_{r_n}^{R_n} |\{d_s\} \cap r_n|}{\sum_{r_n}^{R_n} |\{d_s\} \cap r_n| + k1\left((1-b) + \frac{b \cdot dl}{avgdl}\right)}. \tag{7}$$

Here, inspired by the BM25 Okapi schema, we take the frequency into account in a controllable way. To this aim, we introduce a modified version of RPRS, which we denote by RPRS w/freq, which relies on two parameters: $k1$, the frequency saturation parameter,[3] for controlling the effect of frequency of occurrence of a candidate document's sentence in R_n, and b, the document length normalization parameter, for controlling the effect of the candidate document length. Similarly to the BM25 formula, *dl* refers to document length and *avgdl* refers to average length of documents. The advantage of the frequency saturation parameter ($k1$) of the proposed method that works similarly to BM25's "term saturation" mechanism, is that the occurrence of multiple sentences of a candidate document in only one r_n of R_n has a lower impact on the relevance score than the occurrence of multiple sentences of a candidate document each of which occurs only once in several different r_ns of R_n. This characteristic is in line with BM25's concept of frequency saturation, which prevents a document from obtaining a high score solely based on the repetition of a single word that matches just one word from the query. We parameterize the degree of normalizing the relevance according to the document length (i.e., number of sentences of a document) with the parameter b, which works similarly to BM25's length normalization parameter [34, 58]. If $b = 0$, then the relevance score is not normalized by the document length at all, because the right side of the denominator in Equations (6) and (7) will be $k1((1-b) + \frac{b \cdot dl}{avgdl}) = ((1-0) + \frac{0 \cdot dl}{avgdl}) = k1((1) + 0) = k1$. Therefore, only $k1$ will be kept in the denominator and the whole denominator will be $|S_d \cap r_n(q_s, S_{TK_q})| + k1$ and $\sum_{r_n}^{R_n} |\{d_s\} \cap r_n| + k1$ for Equations (6) and (7), respectively. As b increases from 0 toward 1, the impact of document length—compared to the average length of documents (dl/adl)—in normalizing the relevance score will be higher. Consequently, $b = 1$ means the relevance score will be fully normalized based on the document length, because the right side of the denominator in Equations (6) and (7) will be $k1((1-b) + \frac{b \cdot dl}{avgdl}) = ((1-1) + \frac{1 \cdot dl}{avgdl}) = k1((0) + \frac{1 \cdot dl}{avgdl}) = k1 \cdot \frac{dl}{avgdl}$. Therefore, the document length will be divided by the average length of documents and will be multiplied to $k1$. As a result, the denominator will be fully normalized based on the ratio of document length to the average length of all documents in the denominator, and the whole denominator will be $|S_d \cap r_n(q_s, S_{TK_q})| + k1 \cdot \frac{dl}{avgdl}$ and $\sum_{r_n}^{R_n} |\{d_s\} \cap r_n| + k1 \cdot \frac{dl}{avgdl}$ for Equations (6) and (7), respectively.

In summary, the final proposed method has three parameters: $k1$ and b, as described above, and n that controls the number of top-n sentences in r_n per query sentence. All parameters can be tuned on the training set. Furthermore, the parameters can be used with default values or with values that are obtained by tuning the method on another dataset. With *RPRS* and *RPRS w/freq*, we only need to compute the cosine similarity between embeddings of query sentences and the document

[3]To gain a deeper insight into k1 and frequency saturation, we recommend referring to the original BM25 paper Robertson and Walker [58]. For a more accessible explanation of frequency saturation and its effects, you can explore the details provided in Reference [60]. This resource offers a step-by-step breakdown that should make it easier to understand how this parameter works.

sentences. Our proposed method is efficient, because all documents can be pre-processed, embedded, and indexed before query time. At query time, producing the embedding for query sentences using SentenceBERT is highly efficient [56]. This gives either *RPRS* or *RPRS w/freq* the advantage of having a complexity of $O(N)$, where N is the total number of sentences in the query and candidate documents compared to re-rankers based on Cross-encoders with $O(N^2)$. Furthermore, calculating cosine similarity is a simple operation, thus resulting in a fast inference time.

4 BASELINE RETRIEVAL MODELS

In this section, we introduce BM25 and SDR that are the two main baselines in our experiments.

4.1 BM25

Lexical retrievers estimate the relevance of a document to a query based on word overlap [57]. Many lexical methods, including vector space models, Okapi BM25, and query likelihood, have been developed in previous decades. We use BM25 because of its popularity as first-stage ranker in current systems and its strong effectiveness on QBD tasks [59]. Based on the statistics of the words that overlap between the query and the document, BM25 calculates a score for the pair:

$$s_{lex}(q,d) = BM25(q,d) = \sum_{t \in q \cap d} rsj_t . \frac{tf_{t,d}}{tf_{t,d} + k_1 \left\{(1-b) + b\frac{|d|}{l}\right\}}, \tag{8}$$

where t is a term, $tf_{t,d}$ is the frequency of t in document d, rsj_t is the Robertson-Spärck Jones weight [58] of t, and l is the average document length; k_1 and b are parameters.

4.2 Birch

Yilmaz et al. [81] present a simple yet effective solution for applying BERT to long document retrieval: The inference is applied to each sentence in a candidate document, and sentence-level evidence is aggregated for ranking documents as follows:

$$Score_d = a \cdot S_{doc} + (1-a) \cdot \sum_{i=1}^{n} w_i \cdot S_i, \tag{9}$$

where S_{doc} represents the original document score and S_i denotes the ith top-scoring sentence according to BERT. The parameters a and w_i's can be learned or used with default values. To replicate the "3S: BERT" models, referred to as 3S-Birch hereafter, we utilize the official implementation from the Birch paper, which uses the three top-scoring sentences.[4] We report the 3S-Birch approach, as it yielded the highest effectiveness, even though we experimented with the top-1 and top-2 scoring sentences. Until now, the effectiveness of Birch has only been analyzed and proven for short queries and long documents. In contrast, our investigation focuses on evaluating its performance in situations where both queries and documents are extremely long. We use the same BERT model that we use for our proposed method per each dataset.

4.3 SDR [21]

We replicate SDR [21] as the most recent and comparable methodology to RPRS, because (1) although SDR's mechanism is dissimilar to RPRS fundamentally, it also is a sentence-level relevance scoring model designed for QBD tasks that cover the full length of both queries and candidate document texts using sentence embeddings, and (2) similarly to RPRS, SDR has a complexity of $O(N)$ where N is count of sentences—due to the utilization of bi-encoder sentence embeddings

[4] https://github.com/castorini/birch

instead of cross-encoders—for computing relevance scores in contrast to other techniques with $O(N^2)$ [6, 17]. Therefore, besides comparing RPRS with the state-of-the-art model on each dataset, we compare its effectiveness with SDR. SDR's architecture makes it suitable for both full-ranking and re-ranking settings; as a result, we evaluate it in both configurations and only report the result of the setup (ranker or re-ranker) in which it performs best. In the following, we introduce SDR's pre-training and inference methodology.

4.3.1 Self-supervised Pre-training. Given a collection of documents D, SDR samples sentence pairs from the same paragraph of a given document (intra-samples) and sentence pairs from different paragraphs taken from different documents (inter samples) with equal probability (i.e., 0.5 for each type of sampling). It tokenizes sentences and aggregates them into batches, and it randomly masks them in a similar way to the RoBERTa pre-training paradigm. The authors use the Roberta model for their implementation [35]. The pretraining objective of SDR comprises a dual-term loss: (1) a standard MLM loss adopted from Reference [17] that allows the model to specialize in the domain of the given collection of documents [22], and (2) a contrastive loss [23] aims to minimize the distance between the representations of sentence-pairs from the same paragraph (intra-samples), while maximizing the distance between the representations of sentence-pairs from different paragraphs (inter-samples).

4.3.2 Inference. SDR computes the relevance score between a query q and a candidate document d by computing a two-stage hierarchical similarity score: (1) SDR creates a sentence similarity matrix M for all possible pairs of query paragraphs and candidate document paragraphs. Each cell in M represents the cosine similarity between a sentence from paragraph i of the query and a sentence from paragraph j of the candidate document. Specifically, the rows of M correspond to sentences from the query paragraph, and the columns correspond to sentences from the candidate document paragraph. Therefore, a cell $M^{i,j}_{a,b}$ represents the cosine similarity between sentence a from paragraph i of the query and sentence b from paragraph j of the candidate document. (2) Next, a paragraph similarity matrix P is created for the candidate document d based on all pairs of paragraphs (query paragraph vs. document paragraph). Each cell in P represents the similarity between a paragraph from the query and a paragraph from the candidate document and is computed using the maximum cosine similarity between sentences in the two paragraphs one from the query and one from the candidate document. The motivation for the creation of matrix P is that similar paragraph pairs should incorporate similar sentences that are more likely to correlate under the cosine metric.

Finally, P is normalized globally ($NRM(P)$), based on other candidate documents' paragraph similarity matrices. Based on the $NRM(P)$ generated for a candidate document, the total similarity score S is obtained by computing an average of the highest cosine similarity scores in each row of P. The motivation for S is that the most correlated paragraph pair only contributes to the total similarity score. In contrast to RPRS, SDR does not take into account the length of candidate document, because the denominator in its formula for computing matrix P is count of sentences in a query's paragraph and for total score S is the count of paragraphs in the query. However, this issue has been handled in our re-ranker by integrating the length of the query and document into QP and DP (see Section 3.2). Moreover, if one paragraph of a candidate document be the most similar paragraph to all paragraph of query, then SDR does not penalize this repetition while we take that into account with $k1$ parameter in *RPRS w/freq* that controls frequency saturation (see Section 3.3). In addition, the cosine score is used directly into the approach, while we only consider that score for ranking sentences in R_n (see Section 3.2 and Figure 1).

5 EXPERIMENTS

In this section, we first describe the datasets, the pre-trained Transformers that we use, and our implementation details for the retrieval models. Finally, we provide information about parameter tuning and pre-processing of the dataset.

5.1 Datasets

We evaluate our models on three QBD retrieval tasks: legal case retrieval, patent prior art retrieval, and document similarity ranking for Wikipedia pages.

COLIEE'21. We first use the COLIEE'21 dataset to evaluate the effectiveness of the proposed method (PRS) on legal case retrieval. There are 650 query documents in the train set and 250 in the test set [53], with 4,415 documents as candidate documents in both sets. The candidate documents' average length is 5,226 (tokenized by SparkNLP [29], see below), with outliers reaching 80,322 words.

Caselaw. To investigate the generalizability of our model, *RPRS w/freq*, we tuned it on the COLIEE'21 dataset and evaluate the tuned model on a different dataset: the Caselaw dataset [36]. Therefore, we use the Caselaw dataset as a test set to assess how well the tuned *RPRS w/freq* model can generalize to new and unseen data. The Caselaw dataset contains 100 query documents, 2,645 relevance assessments, and 63,916 candidate documents. In Caselaw, the candidate documents' average length is 2,344, with outliers up to 124,092 words.

CLEF-IP 2011. For patent prior art retrieval [51], we experiment on the CLEF-IP 2011 dataset that contains 300 and 3,973 query documents in the train set and test set, respectively, both of which have document-level relevance assessments. The count of candidate documents per query is about 3 million. In this work, we select the English subset of CLEF-IP 2011 that contains about 900,000 candidate documents per query and 100 and 1,324 query documents in the train set and test set, respectively. We concatenate title, abstract, description, and claims as the whole patent document. The candidate documents' average length in CLEF-IP is 10,001 words, with outliers up to 407,308 words.

Wikipedia. For the Wikipedia datasets, we use the WVG and the WWA datasets [21] that contain 21,935 and 1,635 candidate documents and 90 and 92 query documents, respectively. The WWG collection consists of articles reviewing video games from all genres and consoles and the WWA collection consists of a mixture of articles discussing different types of wine categories, brands, wineries, grape varieties, and more. The documents' average length is 1, 061 and 966, with outliers up to 23,048 and 13,081 words for the WVG and WWA datasets, respectively.

5.2 Baselines

BM25 has previously been shown to be a strong baseline for QBD retrieval [59], and it holds the state of the art among all lexical models on COLIEE [5]. We implement BM25 as the initial ranker using Elasticsearch on all five datasets. In addition to BM25 on the complete query document text, we employ BM25 on the top-10 percent of query document terms that are extracted and scored using **Kullback–Leibler divergence for Informativeness (KLI)** [72] following prior work [5, 36]. As KLI has been shown to be an effective approach for making shorter queries for BM25 [36], we employ KLI on documents of all three datasets and refer to that in the result tables as "BM25 + KLI."

Moreover, we replicate the SDR [21] ranker as SDR_{inf} where "inf" refers to inference. SDR_{inf} is a comparable sentence-level baseline to *RPRS w/freq*. It should be noted that wherever SDR is mentioned in the tables, it is referring to the best-performing SDR variant, either as a re-ranker

after the best first-stage ranker, or as a full ranker. We experiment with Roberta besides the other SBERT's models as SDR's authors use the Roberta model for their implementation [35].

For COLIEE 2021, we compare the proposed method to the lexical state-of-the-art model, which is $BM25_{optimised}$ with optimized parameters ($b = 1, k = 2.8$) [5]. Additionally, we do comparisons with two re-rankers: the BERT re-ranker [46], and the multi-task fine-tuned BERT re-ranker (MTFT-BERT), which is the neural state of the art on COLIEE [2]. For Caselaw [36], we re-use the lexical and neural models that we fine-tuned on COLIEE to analyze the generalizability of them and the proposed method. The state-of-the-art method for Caselaw is K [36].[5] We found that K's run file is the most effective initial ranker, and thus we used that as our initial ranker on Caselaw [36].

For patent retrieval, there is no Transformer-based method baseline, which could be due to the fact that average length of documents in the patent dataset is around 10,001 and thus cannot be handled by Transformer models straightforwardly. Therefore, we compare our result with the best two methods for the English language in the CLEF IP 2011 competition based on Figure 3 from Piroi et al. [51]: Ch.2 and Hy.5. We found that Ch.2 is the most effective method on CLEF-IP 2011, and thus we used that as our initial ranker.

For the Wikipedia datasets, we compare our result with SDR_{inf}, which proposed these datasets recently and is the state-of-the-art model for them.

5.3 Pre-trained Sentence BERT Models

Reimers and Gurevych [56] have published a set of SBERT embedding models that they trained and evaluated extensively with respect to their effectiveness for semantic textual similarity (Sentence Embeddings) and Semantic Search tasks on 14 and 5 different datasets respectively.[6] We exploit the following top-4 ranked embedding models[7] that are different in terms of training data or architecture:

- **all-mpnet-base-v2**,[8] **all-distilroberta-v1**,[9] and **all-MiniLM-L12-v2**[10] are SBERT models trained based on *mpnet-base* [66], *distilbert-base-cased* [62], and *MiniLM-L12-H384-uncased* [75] models respectively on more than 1 billion sentence pairs as general purpose models for sentence similarity;
- **multi-qa-mpnet-base-dot-v1**[11] is a *mpnet-base* model [66] trained on 215M question-answer pairs from various sources and domains, including StackExchange, Yahoo Answers, Google, and Bing search queries and many more as a model for question answering and IR tasks.

Besides the above-mentioned SBERT models, we utilize the following domain-specific and general well-known variants of BERT and Roberta models in the architecture of SBERT to investigate their effectiveness, considering that they have the efficiency of the SBERT's architecture:

- BERT base uncased, BERT large uncased, Roberta base uncased, and Roberta large uncased are pre-trained on a large corpus of English raw data [17, 35];
- Legal BERT base uncased [11] is a light-weight model of BERT-BASE (33% the size of BERT-BASE) pre-trained from scratch on 12 GB of diverse English legal text of several types (e.g., legislation, court cases, contracts);

[5]We do not report result with manually created boolean queries for fair comparison.
[6]https://www.sbert.net/docs/pretrained_models.html
[7]Please note that these models are top-ranked at the time of submission and the SBERT list can be updated later.
[8]https://huggingface.co/sentence-transformers/all-mpnet-base-v2
[9]https://huggingface.co/sentence-transformers/all-distilroberta-v1
[10]https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2
[11]https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-dot-v1

— Patent BERT [67] is trained by Google on 100M+ patents (not just U.S. patents) including abstract, claims, description based on BERT large uncased architecture.

We only report the results of the trained SBERT models as they obtained higher than the general well-known variants of BERT models in the architecture of SBERT in all datasets except for the Wikipedia datasets (WWG and WWA). For these, we report the Roberta large model in the architecture of SBERT, which achieved the best result.

5.4 Implementation Details

For SBERT, we first experiment with all the models that are introduced in Section 5.3 and then adapting the two best models for each dataset on the domain using the methods proposed by Wang et al. [73] (TSDAE) and Ginzburg et al. [21] (SDR) for domain adaptation without labeled data.[12] The SDR pre-training method is a dual-term loss objective that is composed of a standard MLM loss adopted from Reference [17]—which allows the model to specialize in the domain of the given collection [22]—and a contrastive loss [23]. For SDR pre-training, we tokenize sentences, aggregate them into batches, and randomly mask them in a similar way to the RoBERTa pre-training paradigm. For TSDAE pre-training, as suggested in Reference [73], we use 10,000 sentences for each domain, which is only 2% of the COLIEE'21, 1% of the Wikipedia (WWG and WWA) datasets, and less than 0.1% of the CLEF-IP 2011 dataset. PyTorch [48], HuggingFace [77], and Sentence BERT [56] are used to implement all of our models.

We use $BM25_{optimised}$ [5] as our initial ranker for COLIEE, K for Caselaw [36], and *Ch.2* for CLEF-IP 2011 [51]. To determine the optimal re-ranking depth, we increase the depth of the initial rank result on the validation set from 15 to 100 in steps of 5. We found 50 to be optimal for COLIEE, 100 for WWG and WWA, and 20 for the CLEF-IP 2011 dataset. As we use Caselaw as test set only, we employ COLIEE's re-ranking depth for Caselaw.

5.5 Parameter Tuning

For tuning parameters, we experiment with values from 1 to 10 for parameter n: $n = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$. For the b and $k1$ parameters of *RPRS w/freq*, we used the same range as when tuning them for BM25 ($b = [0.0, 0.1, 0.2, \ldots, 1]$, $k1 = [0.0, 0.2, 0.4, \ldots, 3.0]$). We found ($b = 1$, $k1 = 2.8$, $n = 4$) for COLIEE'21, ($b = 0.9$, $k1 = 3.0$, $n = 4$) for WWG and WWA, and ($b = 0.8$, $k1 = 2.4$, $n = 5$) for CLEF-IP.

5.6 Pre-processing

We investigated sentence segmentation using three libraries, since legal case retrieval and patent retrieval are challenging domain-specific tasks with long legal sentences [61, 70]. We empirically found that SparkNLP [29] is segmenting the legal text into sentences with a higher quality than NLTK [8] and Stanza [52]. We split long sentences into sequences of 25 words for COLIEE and 30 words for CLEF-IP (average length of sentences in both collections). We then detected French sentences in the COLIEE data using the method in Reference [16] and translated them to English using the Google Translate API.

6 RESULTS

We employ a paired t-test between the proposed method, i.e., *RPRS w/freq*, and the state-of-the-art model for each dataset. We implement four mutual baselines (BM25, BM25+KLI, SDR_{inf}, Birch) across all the datasets. Additionally, we present state-of-the-art approaches for each individual

[12]The resulting models on the patent, legal and Wikipedia domain will be publicly available.

dataset. By doing so, we guarantee the comprehensive coverage of state-of-the-art methodologies for each dataset. It is worth highlighting that, to the best of our knowledge, no previous work have undertaken the challenge of addressing query-by-document tasks across all these datasets collectively. Existing works have been limited to individual datasets, each concentrating on a single dataset.

We answer the following research questions, assessing the effectiveness of our proposed methods, RPRS and RPRS w/freq, from different perspectives:

- **RQ1:** What is the effectiveness of RPRS compared to the state-of-the-art models for QBD retrieval?
- **RQ2:** How effective is RPRS with shorter or longer text units instead of sentences?
- **RQ3:** What is the effectiveness of RPRS with parameters that were tuned on a different dataset in the same domain?
- **RQ4:** To what extent is RPRS effective and generalizable across different domains with different type of documents?

In the following, we first address the choice of SBERT model for the proposed method, *RPRS w/freq*. Next, we analyze the effectiveness of the proposed method on the five domain-specific QBD datasets and discuss the results per domain. In summary, we found that *RPRS w/freq* outperforms the state-of-the-art models significantly for all official metrics in all five datasets. Please note that the initial ranker remains consistent across various re-rankers for all datasets, thereby establishing a fair comparison. Furthermore, each of the BERT-based re-rankers utilizes the identical BERT model employed by RPRS. This choice ensures that RPRS does not gain an unfair advantage from using a potentially superior BERT model compared to the baseline re-rankers.

6.1 Choice of SBERT Model for *RPRS w/freq*

To address **RQ1**, we first run a set of experiments with different SBERT models to find with which SBERT model the *RPRS w/freq* achieve higher effectiveness. Table 2 shows that our re-ranker achieves a higher effectiveness than the strong initial ranker (line a) [5] using any sentence embedding models (lines b–j). This shows the effectiveness of our re-ranker is more dependent on the proposed method rather than sentence embedding models, while our re-ranker effectiveness could be improved with a sentence embedding model that captures the legal context more accurate. Morover, the table shows that out of the four SBERT models (lines b–e), *all-MininLM-L12-v2*, achieves highest effectiveness. Interestingly, this is the SBERT model with the lowest number of parameters (33M, compared to 82M, 110M, and 110M for the other three). This could indicate that between SBERT models that have shown their effectiveness in other (general) domains, the chance of generalizing to a new domain-specific dataset for a SBERT model that has lowest parameters might be higher than others. This finding is in line with the fact that, in the context of traditional bias-variance tradeoffs, neural models with fewer parameters are more likely to generalize to the other domains [86, 87]. The higher effectiveness for *multi-qa-mpnet-base-dot-v1* (line c) compared to *all-mpnet-base-v2* (line b) while they both have same number of parameters and same base architecture could be due to the fact that *multi-qa-mpnet-base-dot-v1* was trained on (question, pair) instances, and as a result, it is suitable for semantic search task and aligns with our re-ranking task while all-mpnet-base-v2 is trained on various tasks.

The second highest effectiveness is achieved by *Legal SBERT*, which is the domain-specific Legal BERT model [11] that is loaded into the SBERT architecture [56] and has 36M parameters (more than MiniLM and less than our three other SBERT models). The effectiveness of our proposed method with the usage of Legal SBERT model could be due the fact it is pre-trained on the legal

Table 2. Results of the Proposed Method, *RPRS w/freq*, on COLIEE'21 Using Different Transformer Models Embedding

	Model Name	Precision	Recall	F1
Initial Ranker				
a	$BM25_{optimized}$ + KLI	0.1700	0.2536	0.2035
Top SBERT models				
b	all-mpnet-base-v2	0.1760	0.2478	0.2058
c	multi-qa-mpnet-base-dot-v1	0.1840	0.2625	0.2163
d	all-distilroberta-v1	0.1900	0.2632	0.2206
e	all-MiniLM-L12-v2	0.1920	0.2745	0.2259
Domain-Specific				
f	Legal SBERT	0.1880	0.2720	0.2223
Domain Adaptation				
g	all-MiniLM-L12-v2-SDR	0.1817	0.2594	0.2137
h	all-MiniLM-L12-v2-TSDAE	0.1840	0.2584	0.2149
i	Legal SBERT-SDR	0.1890	0.2762	0.2244
j	Legal SBERT-TSDAE	**0.1960**	**0.2891**	**0.2336**

Top SBERT models are top-4 trained and extensively evaluated models by SBERT and are publicly available. Legal SBERT is the Legal BERT [11] that is loaded into the SBERT [56] architecture. Domain adaption has been done using Self-Supervised, SDR, and Unsupervised, TSDAE, methods on all-MiniLM-L12-v2 and Legal SBERT models as the models show top-two highest effectiveness without Domain Adaption.
The bold face is used to represent highest effectiveness in terms of a metric across the models.

documents. However, with domain adaptation on COLIEE corpus using TSDAE method [73], i.e., Legal SBERT-TSDAE, *RPRS w/freq* achieves the highest effectiveness (row *j*). We found the same pattern for SDR and RPRS without frequency (Section 3.2), i.e., they achieve highest effectiveness with Legal-BERT-TSDAE model embeddings, but we only report it for *RPRS w/freq*, which is our complete proposed method.

Furthermore, an observation by comparing rows (g,i) and (h,j) of Table 2, is that the domain adaption with TSDA results in higher effectiveness of our re-ranker compared to SDR domain adaption method that is a contrastive-learning domain adaption method. We suppose the higher effectiveness of our re-ranker with domain adaption by TSDAE compared to SDR could be due to the fact that TSDAE removes words from the input randomly and as the result forces the network to produce robust embeddings. In contrast, the input sentences for a contrastive-learning approach such as SDR are not modified, resulting in less stable embeddings. This finding was confirmed previously by Wang et al. [73].

Additionally, by comparing rows *e* with *g* and *h*, we can see that domain adaption on all-MiniLM-L12-v2 results in drops in the effectiveness of our re-ranker. This observation is in line with the finding in prior work [73] that shows that domain adaption with TSDAE after supervised training results in a drop in the effectiveness. It is noteworthy to mention that all-MiniLM-L12-v2 is trained on 1 billion samples in a supervised manner. A better strategy in this case, would be first doing domain adaption as a pre-training step with TSDAE and then doing supervised training with

Table 3. Results for COLIEE'21

	Model	Precision	Recall	F1
Probabilistic lexical matching baselines				
a	BM25	0.0770	0.1959	0.1113
b	BM25 + KLI	0.0983	0.1980	0.1313
c	BM25optimized + KLI	0.1700	0.2536	0.2035
d	TLIR	0.1533	0.2556	0.1917
Cross-encoders				
e	BERT	0.1340	0.2263	0.1683
f	Legal BERT	0.1440	0.2463	0.1817
g	MTFT-BERT [2] (Previous state of the art)	0.1744	**0.2999**	0.2205
Sentence-based baseline				
h	SDR_{inf}	0.1470	0.2063	0.1716
i	Birch	0.1721	0.2577	0.2064
Proposed methods				
j	RPRS without frequency	0.1890	0.2799	0.2256
k	RPRS w/freq	**0.1960**†*	0.2891*	**0.2336**†*

† and * indicate a statistically significant improvement over state of the art (MTFT-BERT, line g) and $BM25_{optimized}$ (line c), respectively, according to a paired-test ($p < 0.005$) with Bonferroni correction for multiple testing. The winning team in the COLIEE 2021 competition for legal case retrieval is TLIR [39]. The initial ranker for all re-rankers (BERT, MTFT [5], SDR, and RPRS) is $BM25_{optimized}$+KLI. Legal SBERT-TSDAE is used to embed sentences for SDR_{inf}, RPRS, and *RPRS w/freq*.
The bold face is used to represent highest effectiveness in terms of a metric across the models.

1 billion samples as was suggested by the authors of TSDAE [73].[13] However, re-training MiniLM with 1 billion examples was not possible for us due to computational limitations.

Therefore, based on our empirical findings, for the next experiments on each domain, we pick the domain-specific Transformer model and then adapt it into the the target domain, e.g., legal, using TSDAE. We argue that doing this domain adaption is realistic for a real-world application, because it is trained on the document collection that are provided for training and does not need or use any supervision from labelled data.

6.2 Effectiveness on the COLIEE Dataset (RQ1)

Taking into account the finding from previous section, we exploit Legal SBERT-TSDAE in Table 3 to embed sentences for *RPRS*, *RPRS w/freq*, and SDR. Row i of Table 3 show that even without taking into account frequency, *RPRS* achieves higher effectiveness than the initial ranker ($BM25_{optimized}$ + KLI, line c), Birch (line i), and the state-of-the-art re-ranker (MTFT-BERT, line g) on precision and the official metric ($F1$). This is while *RPRS* has only one parameter, n. Row k of the table shows that $RPRSw/freq$ achieve highest effectiveness and significantly better results over the state of the art (MTFT-BERT [2], line g) on precision and the official metric ($F1$), which

[13] Also reported on https://github.com/UKPLab/sentence-transformers/issues/1372#issuecomment-1028525300 (visited February 27, 2023).

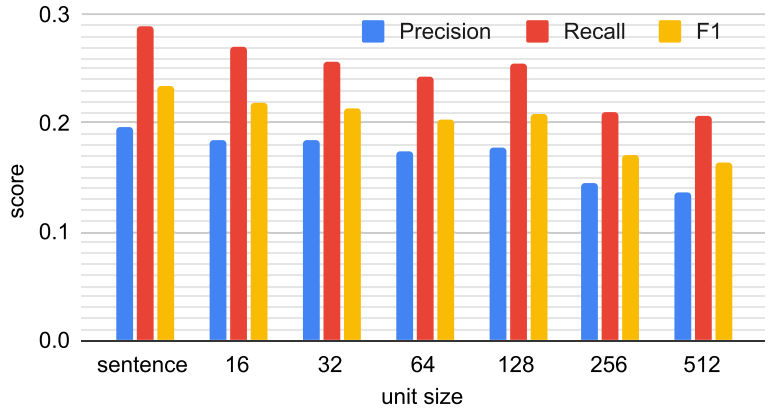

Fig. 2. Effectiveness of RPRS w/freq using fixed-length units instead of sentences on COLIEE 2021 (segment size in tokens).

indicates the importance and role of adding the document length normalization parameter b, and frequency saturation parameter $k1$ in increasing the effectiveness of the proposed method. We emphasize that the official metric for COLIEE is $F1$ and the COLIEE organizers only report $F1$ in their official reports for case law retrieval task [53]. We assume achieving lower recall by our re-ranker, *RPRS w/freq* (line k), compared to MTFT-BERT [2] (line g) could be due to the inverse relationship between recall and precision, which is a common observation based on experimental evidence [9]. Cleverdon [14] provides an in-depth explanation of why precision and recall have often an inverse relationshi. SDR_{inf} (line h) achieves lower performance than $RPRSw/freq$ and *RPRS* results (lines j and k).

6.3 Effect of using other Units than Sentences (RQ2)

We investigate the effect of using embeddings of different textual units than sentences in *RPRS w/freq* to analyze if "sentence" is the most appropriate unit for our re-ranker. To do so, we split the documents in segments of length l with $l \in \{16, 32, 64, 128, 256, 512\}$. Figure 2 shows that sentences are better units than sequences with pre-defined fixed-length, also compared to sequences with fixed-length 16 and 32 that are similar in length to the average and median sentence length (29.4 and 24.8 tokens). This confirms the relevance of sentences as units for retrieval, one of the premises of RPRS.

6.4 Effectiveness without Parameter Optimization on Target Dataset (RQ3)

To address this question, we evaluate the proposed method on the *Caselaw* dataset [36] with the parameters that were tuned on the COLIEE dataset without doing domain adaption on the collection for the Transformer model. In other words, our goal is to analyze how much the three optimized parameters of RPRS on the COLIEE dataset are transferable to another dataset in the same (i.e., legal) domain. Table 4 shows that the effectiveness of our re-ranker is higher than all models including the state-of-the-art model (*K*, line c) for all evaluation metrics with (line i) or without (line j) taking into account frequency. This supports the cross-data generalizability of our re-ranker, i.e., optimizied parameters of our proposed method work effectively for another dataset in the same domain, as we re-use the tuned $RPRSw/freq$ on COLIEE for the Caselaw dataset without optimizing its parameters on the Caselaw dataset. SDR_{inf} (line g) and Birch (line h) baselines achieve lower performance than $RPRSw/freq$ and *RPRS* results (lines i and j).

Table 4. Results for CaseLaw

Model	P@1	R@1	MAP@5	NDCG@5	MRR
Probabilistic lexical matching baselines					
a BM25	0.500	0.025	0.227	0.358	0.618
b BM25 + KLI	0.660	0.115	0.232	0.398	0.726
c K [36] (Previous State-of-the-art)	0.730	0.119	0.307	0.473	0.803
Cross-encoders					
d BERT	0.320	0.064	0.118	0.201	0.438
e Legal BERT	0.330	0.065	0.128	0.217	0.449
f MTFT-BERT	0.360	0.064	0.160	0.252	0.481
Sentence-based baseline					
g SDR_{inf}	0.500	0.070	0.171	0.320	0.585
h Birch	0.680	0.117	0.270	0.425	0.774
Proposed methods					
i RPRS without frequency	0.730	0.128	0.314	0.489	0.807
j RPRS w/freq	**0.780**†	**0.138**†	**0.321**†	**0.496**†	**0.837**†

† indicate a statistically significant improvement over the state of the art (K [36], line c) according to a paired-test ($p < 0.05$) with Bonferroni correction for multiple testing. We pick the optimized values of the proposed method parameters from COLIEE dataset, and use them to analyze the proposed method cross-data generalizability on Caselaw dataset. Legal SBERT-TSDAE is used to embed sentences for SDR_{inf}, RPRS, and *RPRS w/freq*.
The bold face is used to represent highest effectiveness in terms of a metric across the models.

6.5 Generalizability of Porposed Method (RQ4)

Patent domain (CLEF-IP 2011 dataset). Table 5 shows that our re-ranker outperform the state-of-the-art model on the CLEF-IP 2011 for all evaluation metrics with or without taking into account frequency (lines g and h). Row g of Table 5 show that even without taking into account frequency *RPRS* achieves higher effectiveness than the initial ranker (Ch. 2, line d), the previous state of the art on CLEF-IP'11. Patent SBERT-TSDAE is used to embed sentences for SDR_{inf}, RPRS, and *RPRS w/freq*, which is Patent BERT [67] model that is loaded into into the SBERT [56] architecture and adapted into the domain using TSDAE method [73]. SDR_{inf} (line e) and Birch (line f) show competitive results compared to BM25 (lines a and b); however, it obtained much lower results than $RPRSw/freq$.

Wikipeda domain (Wine and Video games datasets). Table 6 shows that SDR_{inf} (line c and *d*) and Birch (line e) obtain competitive results to $BM25 + KLI$ (line b) on SDR's own datasets. However, as we achieved higher recall with BM25+KLI, we use it as the initial ranker for our re-ranker. Results show the generalizability and effectiveness of $RPRSw/freq$ and *RPRS* on a very different domain (Wikipedia) for two datasets as they could outperform the state-of-the-art model on the WWG and WWA datasets for all evaluation metrics with or without taking into account frequency (lines f and g). It is noteworthy to mention that we report all of the official metrics of WWG and WWA datasets. We refer to *Hit Ratio @k* in the original paper [21] as recall in this table as their definition for the *Hit Ratio* is equal to recall in Information

Table 5. Results for Patent Retrieval

	Model	P@5	P@10	R@5	R@10	MAP@5	MAP@10	MRR
Probabilistic lexical matching baselines								
a	BM25	0.084	0.066	0.010	0.147	0.053	0.065	0.187
b	BM25 + KLI	0.095	0.071	0.011	0.167	0.074	0.087	0.221
Top-two teams in English CLEF-IP'11								
c	Hy.5	0.057	0.041	0.071	0.103	0.045	0.051	0.162
d	Ch.2 [51] (Previous state of the art)	0.122	0.089	0.150	0.216	0.102	0.118	0.296
Sentence-based baseline								
e	SDR_{inf} + Ch.2	0.105	0.077	0.120	0.175	0.078	0.090	0.251
f	Birch + Ch.2	0.125	0.087	0.148	0.208	0.104	0.114	0.288
Proposed methods								
g	RPRS + Ch.2	0.131	0.092	0.160	0.221	0.113	0.129	0.319
h	RPRS w/freq + Ch.2	**0.132**†	**0.093**†	**0.167**†	**0.229**†	**0.116**†	**0.132**†	**0.332**†

† indicates the statistically significant improvement over best team on the CLEF-IP11 data (Ch.2 [51]) according to a paired-test ($p < 0.001$) with Bonferroni correction for multiple testing respectively. Patent SBERT-TSDAE is used to embed sentences for SDR_{inf}, RPRS, and *RPRS w/freq*.
The bold face is used to represent highest effectiveness in terms of a metric across the models.

Retrieval. We compute **Mean Percentile Rank (MPR)** using the original implementation by SDR's authors [21].[14]

7 FURTHER ANALYSIS AND DISCUSSION

In this section, we further analyze our results, starting with an ablation study on the components of proposed methods, followed by the effect of document length, document coverage, the parameter space, and a more detailed comparison to the SDR baseline.

7.1 Ablation Study on RPRS w/freq Components

The contribution of each module in RPRS w/freq is assessed by an ablation study in Table 7. The role of Query Proportion (QP), Document Proportion (DP), and the parameters (b and $k1$) has been evaluated in rows a (No QP), b (No DP), and c (no b and k1). For the sake of clarity, we re-write the modified equation according to each row. For row a (No QP), we remove the Equation (4) from *RPRS w/freq* formula:

$$RPRS\,w/freq = DP.$$

Similarly for row b (No DP):

$$RPRS\,w/freq = QP.$$

To do the ablation study of row c (no b and k1), we modify Equation (6), Fq_{R_n}, and Equation (7), Fd_{R_n}, as follows:

$$Fq_{R_n}(S_q, S_d, S_{TK_q}) = \frac{\sum_{q_s}^{S_q} |S_d \cap r_n(q_s, S_{TK_q})|}{\sum_{q_s}^{S_q} |S_d \cap r_n(q_s, S_{TK_q})|}$$

$$Fd_{R_n}(S_d, S_q, S_{TK_q}) = \frac{\sum_{d_s}^{S_d} \sum_{r_n}^{R_n} |\{d_s\} \cap r_n|}{\sum_{d_s}^{S_d} \sum_{r_n}^{R_n} |\{d_s\} \cap r_n|}.$$

[14] https://github.com/microsoft/SDR/blob/main/models/reco/wiki_recos_eval/eval_metrics.py

Table 6. Results for the Video Games (Left) and Wines (Right) Datasets from Ginzburg et al. [21]

	Model	Embedding	Video games (WWG)				Wines (WWA)			
			recall@10	recall@100	MPR	MRR	recall@10	recall@100	MPR	MRR
Probabilistic lexical matching baselines										
a	BM25	—	0.2200	0.4887	0.8606	0.5681	0.1704	0.5041	0.8164	0.4562
b	BM25 + KLI	—	0.2425	**0.5499**	0.8748	0.6150	0.1732	**0.6332**	0.8191	0.4759
Previous state of the art										
c	SDR_{inf}	R SDR	0.2360	0.5400	0.9740	0.6400	0.1700	0.5900	0.8930	0.5090
d	SDR_{inf}	R TSDAE	0.2384	0.5425	0.9761	0.6433	0.1712	0.5980	0.8940	0.5100
Sentence-BERT baseline										
e	Birch	R TSDAE	0.2322	0.5271	0.8683	0.6115	0.1720	0.5599	0.8088	0.4680
Proposed Methods										
f	RPRS	R TSDAE	0.2580†*	**0.5499**	0.9768	0.6501†*	0.1892†*	**0.6332**	0.8955	0.5210†*
g	RPRS w/freq	R TSDAE	**0.2663**†*	**0.5499**	**0.9774**	**0.6676**†*	**0.2015**†*	**0.6332**	**0.8980**	**0.5360**†*

R refers to Roberta large in this table. † and * indicate a statistically significant improvement over previous state of the art (SDR, line b) and BM25+KLI (line d) respectively, according to a paired test ($p < 0.005$) with Bonferroni correction for multiple testing.
The bold face is used to represent highest effectiveness in terms of a metric across the models.

Table 7. Ablation Study Results on the COLIEE 2021 Legal Case Retrieval Task

		P	R	F1
	Full method (RPRS w/freq)	**0.1960**	**0.2891**	**0.2336**
a	No QP (Equation (4))	0.1670	0.2421	0.1976
b	No DP (Equation (5))	0.1410	0.2180	0.1712
c	No b and k1 params	0.1860	0.2691	0.2199
	RPRS without freq (Equation (1))	0.1890	0.2799	0.2256
d	No min function in PRS	0.1798	0.2644	0.2140

The bold face is used to represent highest effectiveness in terms of a metric across the models.

The first and second highest drop in quality of ranking is caused by removing *DP* and *QP*, which shows the crucial role of both *QP* and *DP* in our re-ranker. This aligns with the assumption for designing our re-ranker: A candidate document and query are likely relevant if a large proportion of their textual content is similar to each other. We further analyze the effect of the min function in *RPRS* for the equations of q_{R_n} (Equation (3)) and d_{R_n} (Equation (2)) in row d. To do so, we modify Equations (2), q_{R_n}, and (3), q_{R_n}, as follows:

$$q_{R_n}(S_q, S_d, S_{TK_q}) = \sum_{q_s}^{S_q} |S_d \cap r_n(q_s, S_{TK_q})|$$

$$d_{R_n}(S_d, S_q, S_{TK_q}) = \sum_{d_s}^{S_d} \sum_{r_n}^{R_n} |\{d_s\} \cap r_n|.$$

In general, Table 7 shows that the full method with all components including QP, DP, and b and $k1$ parameters obtains highest effectiveness and supports the necessity of each component. Morover, it ranks the importance and impact of each component in the effectiveness of the method. The

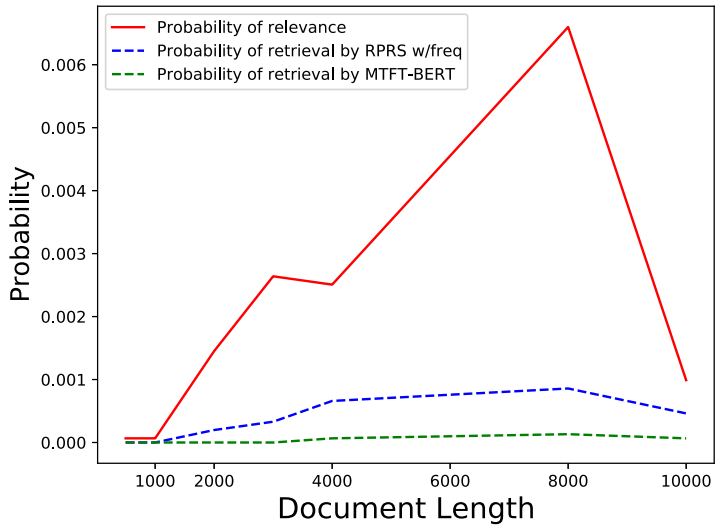

Fig. 3. The probability of relevance and probability of retrieval on COLIEE 2021. In comparison to the probability of relevance, longer documents have a disproportionately smaller probability of being retrieved by MTFT-BERT while *RPRS w/freq* is not biased against retrieving long documents.

importance of the *min* function in *RPRS* supports the intuition that if the repetition of sentences of a candidate document in R_n is not controlled, the quality of the RPRS drops. This supports the *min* function for *RPRS* and motivate proposing *RPRS w/freq* to take into account the repetitions in an intelligent way by adding $k1$ and b.

7.2 Effect of Document Length in Comparison to MTFT-BERT

We plot Figure 3 to analyze the effect of document length on the effectiveness of the proposed method, and compare it with the MTFT-BERT model on COLIEE21. This comparison is justified due to the fact that while in MTFT-BERT the document length is bounded by the maximum input length of BERT, it is the current state-of-the-art model on COLIEE'21. Therefore, we select the most effective available model on this dataset for comparison in our analysis. In Figure 3, probability of relevance $P(relevant|length(doc))$ indicates the chance of having a relevant document with a specific length among all documents, and probability of retrieval $P(ret|length(doc))$ shows the chance of retrieving a relevant document with a specific length among all retrieved documents, within top-k ranks. To compute the probability of retrieval, we set $k = 5$, because the average number of relevant documents per query in the collection is five. We analyze the *probability of retrieval* and *probability of relevance* from relatively shorter documents—with 1,000 words—toward longer documents with 10,000 words. An ideal ranker does not lose effectiveness, probability of retrieval in Figure 3, by the increment in the length of documents. As shown in Figure 3, the *RPRS w/freq* model not only does not loose effectiveness with increasing the document length, but also retrieves the longer relevant documents easier and gets closer to the probability of relevance $P(relevant|length(doc))$ for them. This could be due to the fact that longer documents provide more information, and the *RPRS w/freq* model can take use that information effectively for ranking. This is while MTFT-BERT does not improve by increasing document length and receive similar effectiveness even with the more information that exist in a longer document. This indicates that one of the reasons for gaining a higher effectiveness of F1 score, which is the official metric of COLIEE dataset, by the proposed method compared to the other models could be because of the

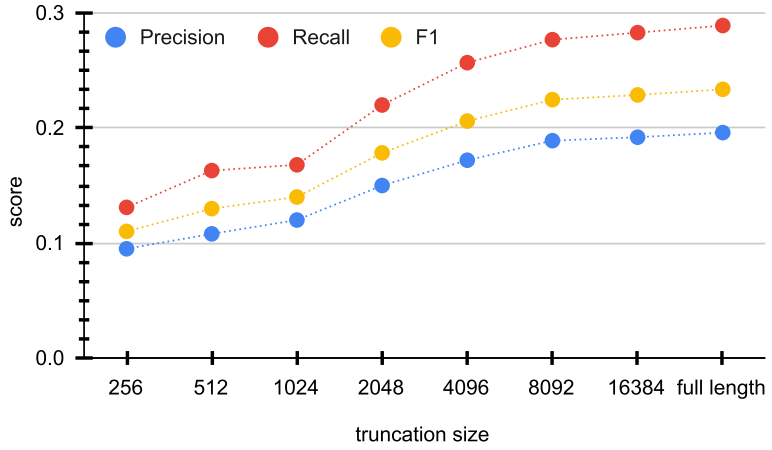

Fig. 4. Effectiveness of RPRS w/freq over varying truncation length for queries and documents on COLIEE 2021 where truncation size is the maximum length input.

fact that *RPRS w/freq* could capture the full information in the lengthy documents and retrieves them more effective than other models such as MTFT-BERT.

7.3 Effect of Covering the Full Length of Queries and Documents

We analyze if taking into account the full length of queries and documents is an advantage for the proposed method in Figure 4. We argue that an ideal semantic re-ranker should achieve higher effectiveness by receiving the full length of queries and documents rather than truncated text as the input. Therefore, we analyze the power of our re-ranker when the length of queries and documents increases: we experiment with different maximum lengths for the input. Given $l \in \{256, 512, 1024, 2048, 4096, 8092, 16384\}$ as the maximum input length, we select the first l tokens of the query and documents as their representation. Moreover, our analysis gives an in-depth insight about the robustness of the proposed method and assess whether it collapse when presented with long queries and documents or not. As Figure 4 shows, our re-ranker takes advantage of seeing the whole content of query and documents in terms of effectiveness as it has the highest effectiveness when we feed it with the full length and the lowest effectiveness with only 256 tokens as the input. The figure also shows the largest leap is between 1,024 and 4,096 tokens, which is approaching the average document length in COLIEE (5226). This confirms that $RPRSw/freq$ takes advantage of the full document length in estimating relevance.

7.4 Parameter Sensitivity for $RPRS\ w/freq$

In addition to the tuned parameters that were found and explained in Section 5.5, we analyze the sensitivity of the proposed parameters in other ranges in the following. Figure 5 shows the effects of changes in parameters n, b, and $k1$ on the overall performance of our method on the COLIEE 2021 dataset. In each diagram, the value of each point represents the F1 score according to different values of the parameters. The following observations can be made from Figure 5:

— A comparison between different values of n—independent from b and $k1$—indicates that the proposed method has the lowest effectiveness by reducing the value of n to 1. The effectiveness of the model increases as n is set to 5, and as we set $n = 10$, we see that model effectiveness declines while still being superior to n set to 1. We argue that this demonstrates the existence of a tradeoff in the value of n, which determines the number of the most similar

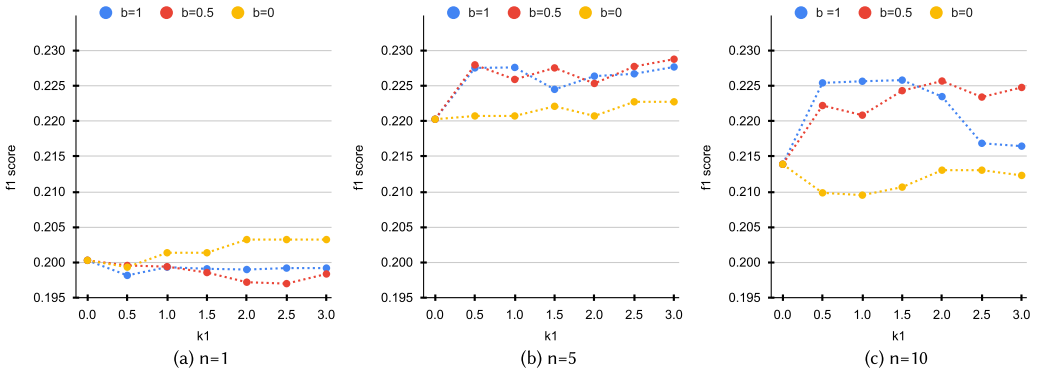

Fig. 5. Sensitivity analysis of our proposed method, *RPRSw/freq*, to changes in parameters n, b, and k1 on the COLIEE 2021 dataset.

sentences per query sentence in set r_n. If n is set to 1, then the selection of *most similar* sentences per query sentence would be very strict, and the effectiveness would decrease. This could be because there may be other sentences from documents that are similar enough to qualify as a *most similar* sentence to a query sentence—in addition to the top-1 most similar sentence—but with $n = 1$, they will not be taken into account in the computation of relevance score. However, if n is set to 10, then some sentences that are not similar enough to qualify as a *most similar* sentence to a query sentence may appear in r_n, and as the result the model obtain less effectiveness compared to $n = 5$, because it would be less strict in creation of r_ns for query sentences.

— If the $k1$ parameter is set to 0 for any value of the b parameter, then the $f1$ score depends only on the value of the n parameter. This is due to the fact that $k1$ multiplies to the b parameter, and if $k1$ is set to zero, then the relevance score is independent of changes in the b parameter's value. Taking this into consideration, we could reduce the search space for parameter optimization by eliminating the necessity of grid search for b values when $k1$ is set to zero.

— By increasing the value of the n parameter the proposed method becomes more sensitive to the document length normalization parameter b, and frequency saturation parameter $k1$. There are larger disparities in the $f1$ scores when the $k1$ or b parameter are changed for $n = 10$ than for $n = 5$, and for $n = 5$ than for $n = 1$. This indicates where the method is less strict in qualifying a sentence as a *most similar* sentence to a query sentence, the impact of term saturation parameter ($k1$) and document length normalization (b) are higher.

— The proposed method performs better in terms of effectiveness without doing length normalization ($b = 0$) when n is set to 1, and, as the result, the selection of the most similar sentences per query sentence is very strict, and it performs better with length normalization ($b = 0.5$ and $b = 1.0$) when n is less strict ($n = 5$ and $n = 10$). This could probably shows that the negative effects on effectiveness by less strictly selecting the most similar sentences per query sentence—by increasing the value of n parameter—could might be controlled with b parameter.

In general, the parameter sensitivity analysis reveals that while optimizing RPRS parameters increases effectiveness, it can still result in increased effectiveness even without optimization by initializing each of the three parameters with the value from middle of their range, e.g., $n = 5$, $k1 = 1.5$, and $b = 0.5$ that obtains 22.75 F1 score, which is still better than current state of the art on the dataset (MTFT-BERT [2]) while not being the ideal result by our *RPRS w/freq* re-ranker.

Table 8. Pearson Product-Moment Correlation Coefficients between Document Length and Relevance Score

Model	Pearson Product-Moment Correlation coefficient				
	COLIEE 2021	Caselaw	CLEF-IP	Video Games	Wines datasets
SDR	0.5768	0.5901	0.612	0.542	0.5366
RPRS w/freq	−0.0565	−0.0401	0.0105	−0.0322	0.0109

The bold face is used to represent highest effectiveness in terms of a metric across the models.

7.5 Further Analysis of RPRS Compared to SDR

We compared RPRS to SDR, because it has some similarities: It also uses sentence-level relevance scoring based on sentence embeddings and it uses a bi-encoder architecture, like RPRS. There are fundamentally significance differences; however, SDR creates a sentence-level similarity matrix M for each pair of query and document paragraphs and based on M matrices, creates a paragraph similarity matrix P for a candidate document. Each cell of P contains the similarity between a pair of paragraphs from query and a candidate document. The matrix P is then normalized to ($NRM(P)$) and the total score S is computed based on $NRM(P)$. The denominator in the formula for computing the matrix P is the number of sentences in a query paragraph and the denominator for the total score S is the number of paragraphs in the query. An important consequence is that the length of the candidate documents is not taken into account, while RPRS explicitly does so. Moreover, if one paragraph of a candidate document is the most similar paragraph to all paragraphs of the query, then SDR does not penalize this repetition, while we take that into account with the $k1$ parameter in *RPRS w/freq*, which controls frequency saturation. In addition, it is noteworthy to mention that SDR uses the cosine similarity directly in their approach while we only consider that score for ranking sentences in R_n.

Losada et al. [37] indicate that an ideal retrieval model should not be tuned to favour longer documents and the relevance scores it produces should not be correlated with document length. They argue that the previous empirical evidence supporting the *scope hypothesis*[15] is over-exaggerated and inaccurate due to the incompleteness of modern collections. This is even more important in QBD retrieval as the length of relevant documents could vary from short to very long. Therefore, the retrieval model should not be biased to the document length. Consequently, to study the sensitivity of our proposed method to document length, and compare it with SDR, we analyze the correlation between candidate document length and the relevance score produced by *RPRS w/freq* and SDR in Table 8. We find that the Pearson Product-Moment Correlation coefficients for *RPRS w/freq* on all five datasets are close to zero while for *SDR* these correlations are much higher, above 0.5, for all datasets. This indicates the strong correlation of SDR relevance score with the document length and however the robustness of *RPRS w/freq* with respect to document length. In other words, *RPRS w/freq* is effective while it is not biased to the length of the document.

In addition to the correlation analysis, we further study the statistics over word counts of documents in all datasets to find out if there is a specific statistical characteristic in the Video games and Wines datasets that makes SDR stronger on those datasets compared to the three datasets in the legal and patent domain. Figure 6 shows the document length distribution over the datasets that indicates higher variance among Patent and Legal datasets compared to Wikipedia datasets (WVG and WWA). The boxes bound the 25th to 75th percentiles, top whisker cover data within

[15]The scope hypothesis in Information Retrieval states that a relationship exists between document length and relevance [58].

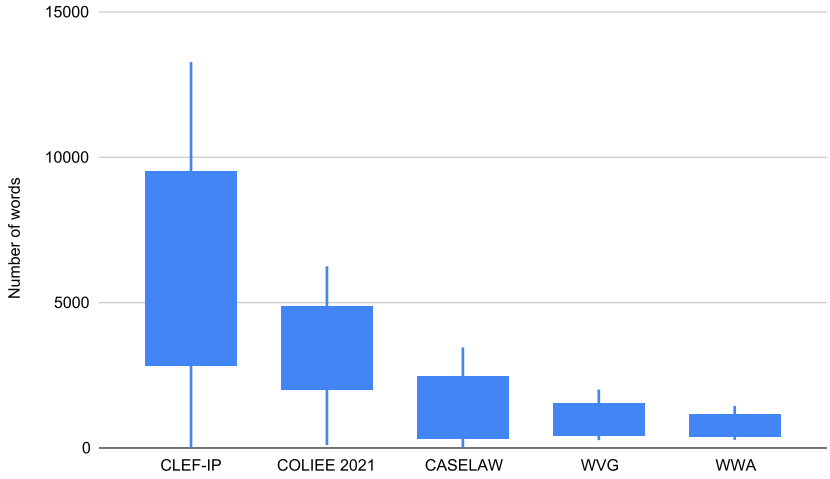

Fig. 6. Distribution of word counts per each document for all of the five datasets: Patent retrieval (CLEF-IP), Case law retrieval (COLIEE 2021 and CASELAW), and Wikipedia (WWG and WWA).

Table 9. Results for COLIEE'2020

	Model	Precision	Recall	F1
	Probabilistic lexical matching baselines			
a	BM25 + KLI	0.6700	0.6117	0.6395
	Baselines			
b	JNLP team [44] (Previous state of the art)	0.8025	0.7227	0.7605
c	Paraformer [43]	0.7346	0.7407	0.7376
	Proposed methods			
d	RPRS without frequency	0.7840	0.7550	0.7692
e	RPRS w/freq	0.7980	**0.7690**†*	**0.7832**†*

† and * indicate a statistically significant improvement over the previous state of the art (JNLP team [44], row b) and BM25 + KLI (line a), respectively, according to a paired test ($p < 0.005$) with Bonferroni correction for multiple testing. The winning team in the COLIEE 2020 competition for legal case retrieval is JNLP [44]. The initial ranker is $BM25_{optimized}$+KLI. Legal SBERT-TSDAE is used to embed sentences for RPRS and *RPRS w/freq*.
The bold face is used to represent highest effectiveness in terms of a metric across the models.

1.5× the inter-quartile range, and outliers are removed. Moreover, the standard deviation of documents length for the Video Games and Wines datasets are 675 and 721, respectively. However, for COLIEE 2021, Caselaw, and CLEF-IP, the standard deviation of documents length are 6, 933, 4, 937, and 11, 402. Thus, the documents in SDR's Wikipedia datasets have less variance compared to the three legal and patent datasets and are shorten than COLIEE and Patent datasets. As a result, we conclude that the lower effectiveness of SDR on the legal and patent datasets is likely caused by SDR not being robust against the length of documents and suffers from the high standard deviation and longer documents in the legal and patent datasets in comparison to its own datasets. Our experimental results in Section 6.5 indicate that our method *RPRS w/freq* also outperforms SDR

Table 10. Results for COLIEE'2022

	Model	Precision	Recall	F1
	Probabilistic lexical matching baselines			
a	BM25 + KLI	0.3000	0.2850	0.2923
	Baselines			
b	UA team [54] (Previous state of the art)	0.4111	0.3389	0.3715
c	Siat team [76]	0.3005	0.4782	0.3691
	Proposed methods			
d	RPRS without frequency	0.4244	0.3607	0.3900
e	RPRS w/freq	**0.4361**†*	0.3904	**0.4120**†*

† and * indicate a statistically significant improvement over the previous state of the art (UA team [54], row b) and BM25 + KLI (line a), respectively, according to a paired test ($p < 0.005$) with Bonferroni correction for multiple testing. The winning team in the COLIEE 2020 competition for legal case retrieval is UA [54]. The initial ranker is $BM25_{optimized}$+KLI. Legal SBERT-TSDAE is used to embed sentences for RPRS and *RPRS w/freq*.
The bold face is used to represent highest effectiveness in terms of a metric across the models.

on their datasets, so the robustness of *RPRS w/freq* is not only beneficial for extremely lengthy documents but also in other domains.

7.6 Effectiveness on Different Versions of the COLIEE Dataset

To analyze the effectiveness of the proposed methods in more depth, we investigate the effectiveness of the proposed methods compared to the previous state-of-the-art methods on different versions of the COLIEE datasets. We analyze this on COLIEE 2020 and COLIEE 2022. Tables 9 and 10 show that our re-ranker outperforms the state-of-the-art model on both versions of COLIEE for all evaluation metrics with or without taking into account frequency (lines d and e). Row d of Tables 5 show that even without taking into account frequency, *RPRS* achieves higher effectiveness than the previous state-of-the-art method (b). Paraformer Nguyen et al. [43] reports F2 in their paper while F1 is the official metric for COLIEE dataset. Therefore, we report F1 instead of F2 in Table 9.[16]

8 CONCLUSION AND FUTURE WORK

In this article, we proposed methods for effectively exploiting sentence-level representations produced by the highly efficient bi-encoder architecture for QBD re-ranking. We proposed a novel model using SBERT representations, with a frequency-based extension inspired by BM25's "term saturation" mechanism and the incorporation of document length normalization into the relevance score computation. Our experiments on five datasets show that our model *RPRS w/freq* takes advantage of the long queries and documents that are common in QBD retrieval. While our *RPRS w/freq* model is unsupervised with only three tunable parameters, it is more effective than state-of-the-art supervised neural and lexical models. In addition, it is highly efficient for retrieval tasks with long documents and long queries, because the operation to compute relevance score by *RPRS w/freq* are based on (1) a bi-encoder, SBERT, which is about 46,800 times faster than the common cross-encoder BERT architecture; (2) the pre-processing, embedding, and indexing of document

[16]It is important to note that Nguyen et al. [43] focus on COLIEE 2020 and report all the baselines on COLIEE 2020 while mistakenly mention COLIEE 2021 as the used dataset in their experiments.

sentences could be done before the query time; (3) the only calculation based on the embeddings are cosine similarity and sorting that are simple and efficient operations.

We show the effectiveness of *RPRS w/freq* on five datasets with low-resource training data, which indicates its suitability for QBD retrieval tasks in which the training data are very limited compared to general web search due to the high cost of dataset creation for these tasks. Therefore, we attain high efficiency and effectiveness while being optimized on low-recourse training data with *RPRS w/freq*.

While we outperform the state-of-the-art models on each dataset with our proposed method, the effectiveness results show the difficult nature of the tasks. One reason for that is the low effectiveness of the first stage retrieval model limits the re-ranker performance. As one direction of improvement for future work, we aim to focus on first-stage retrieval by designing a modification to our proposed method, which is suitable for first-stage retrieval tasks. The parameter sensitivity analysis reveals that while optimizing RPRS parameters increases effectiveness, it can still result in increased effectiveness even without optimization by initializing each of the three parameters with a value from the middle of their range.

For further future improvement on our model, we might be inspired by two other variants of BM25 to make the proposed method parameter-free with dynamically computed parameters. For the frequency saturation parameter, $k1$, a variant of BM25 called "BM25 adaptive" [38], which dynamically computes the $k1$ parameter could be taken into account as inspiration. Additionally, for the document length normalization parameter, b, Lipani et al. [34] propose a length normalization method that removes the need for a b parameter in BM25, which could inspire us to dynamically compute the b parameter of *RPRS w/freq* in future. Considering the intuition of both approaches, working on dynamically initializing the n parameter could make our method completely parameter free, which is an interesting direction for future work. We argue because a sensitivity analysis reveals a pattern that indicates robust and greater effectiveness relative to the baseline for the proposed method with different parameter values, finding a nearly optimal parameter value dynamically might likely be possible.

We believe the proposed method is suitable for other tasks that deal with long documents. In addition, we suggest analyzing the effectiveness of our proposed method on other IR tasks in which we do not have extremely long documents, but the query and candidate document do consist of multiple sentences [1, 15]. We hope that our work will spark more research interest in retrieval for extremely long queries and open up possibilities for highly efficient long-document retrieval.

REFERENCES

[1] Amin Abolghasemi, Arian Askari, and Suzan Verberne. 2022. On the interpolation of contextualized term-based ranking with bm25 for query-by-example retrieval. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR'22)*. Association for Computing Machinery, New York, NY, 161–170. https://doi.org/10.1145/3539813.3545133

[2] Amin Abolghasemi, Suzan Verberne, and Leif Azzopardi. 2022. Improving BERT-based query-by-document retrieval with multi-task optimization. In *Advances in Information Retrieval: Proceedings of the 44th European Conference on IR Research (ECIR'22)*.

[3] Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. Association for Computational Linguistics, 3490–3496. https://doi.org/10.18653/v1/D19-1352

[4] Sophia Althammer, Sebastian Hofstätter, and Allan Hanbury. 2021. Cross-domain retrieval in the legal and patent domains: A reproducibility study. In *European Conference on Information Retrieval*. Springer, 3–17.

[5] A. Askari and S. Verberne. 2021. Combining lexical and neural retrieval with longformer-based summarization for effective case law retrieva. In *Proceedings of the 2nd International Conference on Design of Experimental Search & Information REtrieval Systems*. CEUR, 162–170.

[6] Oren Barkan, Noam Razin, Itzik Malkiel, Ori Katz, Avi Caciularu, and Noam Koenigstein. 2020. Scalable attentive sentence pair modeling via distilled sentence embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3235–3242.

[7] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-document transformer. *CoRR abs/2004.05150*, (2020). Retrieved from https://arxiv.org/abs/2004.05150

[8] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc.

[9] Michael Buckland and Fredric Gey. 1994. The relationship between recall and precision. *J. Am. Soc. Inf. Sci.* 45, 1 (1994), 12–19.

[10] Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. 2020. Semantic re-tuning with contrastive tension. In *International Conference on Learning Representations*.

[11] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. arXiv:2010.02559. Retrieved from https://arxiv.org/abs/2010.02559

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*. PMLR, 1597–1607.

[13] Xiaoyang Chen, Kai Hui, Ben He, Xianpei Han, Le Sun, and Zheng Ye. 2022. Incorporating ranking context for end-to-end BERT re-ranking. In *European Conference on Information Retrieval*. Springer, 111–127.

[14] Cyril W. Cleverdon. 1972. On the inverse relationship of recall and precision. *J. Document.* 28, 3 (1972), 195–201.

[15] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. 2020. Specter: Document-level representation learning using citation-informed transformers. arXiv:2004.07180. Retrieved from https://arxiv.org/abs/2004.07180

[16] M. Danilak. 2014. langdetect: Language detection library ported from Google's language detection. Retrieved January 19, 2014 from https://pypi.python.org/pypi/langdetect/

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805. Retrieved from https://arxiv.org/abs/1810.04805

[18] Eva D'hondt, Suzan Verberne, Wouter Alink, and Roberto Cornacchia. 2011. Combining document representations for prior-art retrieval. In *CLEF (Notebook Papers/Labs/Workshop)*.

[19] Atsushi Fujii, Makoto Iwayama, and Noriko Kando. 2007. Overview of the patent retrieval task at the NTCIR-6 workshop. In *Proceedings of the NTCIR Workshop*.

[20] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. arXiv:2104.08821. Retrieved from http://arxiv.org/abs/2104.08821

[21] Dvir Ginzburg, Itzik Malkiel, Oren Barkan, Avi Caciularu, and Noam Koenigstein. 2021. Self-supervised document similarity ranking via contextualized language models and hierarchical inference. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, 3088–3098. https://doi.org/10.18653/v1/2021.findings-acl.272

[22] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. arXiv:2004.10964. Retrieved from https://arxiv.org/abs/2004.10964

[23] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 2. IEEE, 1735–1742.

[24] Sebastian Hofstätter, Bhaskar Mitra, Hamed Zamani, Nick Craswell, and Allan Hanbury. 2021. Intra-document cascading: Learning to select passages for neural document ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1349–1358.

[25] Sebastian Hofstätter, Navid Rekabsaz, Mihai Lupu, Carsten Eickhoff, and Allan Hanbury. 2019. Enriching word embeddings for patent retrieval with global context. In *European Conference on Information Retrieval*. Springer, 810–818.

[26] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. arXiv:1905.01969. Retrieved from https://arxiv.org/abs/19505.01969

[27] 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 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*. Association for Computational Linguistics, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

[28] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. arXiv:2001.04451. Retrieved from https://arxiv.org/abs/2001.04451

[29] Veysel Kocaman and David Talby. 2021. Spark NLP: Natural language understanding at scale. *Softw. Impacts* (2021), 100058. https://doi.org/10.1016/j.simpa.2021.100058

[30] Steven A. Lastres. 2015. Rebooting legal research in a digital age. Insights Paper (white paper funded by LexisNexis) (July 2013), https://www.lxisnexis.com

[31] Nhat X. T. Le, Moloud Shahbazi, Abdulaziz Almaslukh, and Vagelis Hristidis. 2021. Query by documents on top of a search interface. *Inf. Syst.* 101 (2021), 101793.

[32] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. arXiv:2011.05864. Retrieved from https://arxiv.org/abs/2011.05864

[33] Minghan Li, Diana Nicoleta Popa, Johan Chagnon, Yagmur Gizem Cinar, and Eric Gaussier. 2023. The power of selecting key blocks with local pre-ranking for long document information retrieval. *ACM Trans. Inf. Syst.* 41, 3 (2023), 1–35.

[34] Aldo Lipani, Mihai Lupu, Allan Hanbury, and Akiko Aizawa. 2015. Verboseness fission for bm25 document length normalization. In *Proceedings of the International Conference on the Theory of Information Retrieval*. 385–388.

[35] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv:1907.11692. Retrieved from https://arxiv.org/abs/1907.11692

[36] Daniel Locke, Guido Zuccon, and Harrisen Scells. 2017. Automatic query generation from legal texts for case law retrieval. In *Asia Information Retrieval Symposium*. Springer, 181–193.

[37] David E. Losada, Leif Azzopardi, and Mark Baillie. 2008. Revisiting the relationship between document length and relevance. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*. 419–428.

[38] Yuanhua Lv and ChengXiang Zhai. 2011. Adaptive term frequency normalization for BM25. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. 1985–1988.

[39] Yixiao Ma, Yunqiu Shao, Bulou Liu, Yiqun Liu, Min Zhang, and Shaoping Ma. 2021. Retrieving legal cases from a large-scale candidate corpus. In *Proceedings of the 8th International Competition on Legal Information Extraction/Entailment (COLIEE'21)*.

[40] Parvaz Mahdabi and Fabio Crestani. 2014. Query-driven mining of citation networks for patent citation retrieval and recommendation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. 1659–1668.

[41] Sheshera Mysore, Arman Cohan, and Tom Hope. 2021. Multi-vector models with textual guidance for fine-grained scientific document similarity. arXiv:2111.08366. Retrieved from https://arxiv.org/abs/2111.08366

[42] Sheshera Mysore, Tim O'Gorman, Andrew McCallum, and Hamed Zamani. 2021. CSFCube–A test collection of computer science research articles for faceted query by example. arXiv:2103.12906. Retrieved from https://arxiv.org/abs/2103.12906

[43] Ha-Thanh Nguyen, Manh-Kien Phi, Xuan-Bach Ngo, Vu Tran, Le-Minh Nguyen, and Minh-Phuong Tu. 2022. Attentive deep neural networks for legal document retrieval. *Artificial Intelligence and Law (December 2022)*. DOI: https://doi.org/10.1007/s10506-022-09341-8

[44] Ha-Thanh Nguyen, Hai-Yen Thi Vuong, Phuong Minh Nguyen, Binh Tran Dang, Quan Minh Bui, Sinh Trong Vu, Chau Minh Nguyen, Vu Tran, Ken Satoh, and Minh Le Nguyen. 2020. JNLP team: Deep learning for legal processing in COLIEE 2020. arXiv:2011.08071. Retrieved from https://arxiv.org/abs/2011.08071

[45] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *Choice* 2640 (2016), 660.

[46] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. arXiv:1901.04085. Retrieved from https://arxiv.org/abs/1901.04085

[47] Harshith Padigela, Hamed Zamani, and W. Bruce Croft. 2019. Investigating the successes and failures of BERT for passage re-ranking. arXiv:1905.01758. Retrieved from https://arxiv.org/abs/1905.01758

[48] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* 32 (2019), 8026–8037.

[49] Florina Piroi and Allan Hanbury. 2019. Multilingual patent text retrieval evaluation: CLEF–IP. In *Information Retrieval Evaluation in a Changing World*. Springer, 365–387.

[50] Florina Piroi, Mihai Lupu, and Allan Hanbury. 2013. Overview of clef-ip 2013 lab. In *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, 232–249.

[51] Florina Piroi, Mihai Lupu, Allan Hanbury, and Veronika Zenz. 2011. CLEF-IP 2011: Retrieval in the intellectual property domain. In *CLEF (Notebook Papers/Labs/Workshop)*. Citeseer.

[52] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

[53] Juliano Rabelo, Randy Goebel, Mi-Young Kim, Yoshinobu Kano, Masaharu Yoshioka, and Ken Satoh. 2022. Overview and discussion of the competition on legal information extraction/entailment (COLIEE) 2021. *Rev. Socionetw. Strateg.* 16, 1 (2022), 111–133.

[54] Juliano Rabelo, Mi-Young Kim, and Randy Goebel. 2022. Semantic-based classification of relevant case law. In *JSAI International Symposium on Artificial Intelligence*. Springer, 84–95.
[55] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2020. COLIEE 2020: Methods for Legal Document Retrieval and Entailment. Retrieved from https://sites.ualberta.ca/~rabelo/COLIEE2021/COLIEE_2020_summary.pdf
[56] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv:1908.10084. Retrieved from https://arxiv.org/abs/1908.10084
[57] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retriev.* 3, 4 (2009), 333–389.
[58] Stephen E. Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94*. Springer, 232–241.
[59] Guilherme Moraes Rosa, Ruan Chaves Rodrigues, Roberto Lotufo, and Rodrigo Nogueira. 2021. Yes, BM25 is a strong baseline for legal case retrieval. arXiv:2105.05686. Retrieved from https://arxiv.org/abs/2105.05686
[60] Seitz Rudi. 2020. Understanding TF-IDF and BM-25. Retrieved from https://kmwllc.com/index.php/2020/03/20/understanding-tf-idf-and-bm-25/
[61] George Sanchez. 2019. Sentence boundary detection in legal text. In *Proceedings of the Natural Legal Language Processing Workshop*. Association for Computational Linguistics, 31–38. https://doi.org/10.18653/v1/W19-2204
[62] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv:1910.01108. Retrieved from https://arxiv.org/abs/1910.01108
[63] Ivan Sekulić, Amir Soleimani, Mohammad Aliannejadi, and Fabio Crestani. 2020. Longformer for MS MARCO document re-ranking task. arXiv:2009.09392. Retrieved from https://arxiv.org/abs/2009.09392
[64] Walid Shalaby and Wlodek Zadrozny. 2019. Patent retrieval: A literature review. *Knowl. Inf. Syst.* 61, 2 (2019), 631–660.
[65] Yunqiu Shao, Jiaxin Mao, Yiqun Liu, Weizhi Ma, Ken Satoh, Min Zhang, and Shaoping Ma. 2020. BERT-PLI: Modeling paragraph-level interactions for legal case retrieval. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI'20)*. 3501–3507.
[66] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Adv. Neural Inf. Process. Syst.* 33 (2020), 16857–16867.
[67] Rob Srebrovic and Jay Yonamine. 2020. *Leveraging the BERT Algorithm for Patents with TensorFlow and Big Query*. Technical Report. Google.
[68] Vu Tran, Minh Le Nguyen, Satoshi Tojo, and Ken Satoh. 2020. Encoded summarization: Summarizing documents into continuous vector space for legal case retrieval. *Artif. Intell. Law* 28, 4 (2020), 441–467.
[69] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.
[70] Suzan Verberne, E. K. L. D'hondt, N. H. J. Oostdijk, and Cornelis H. A. Koster. 2010. Quantifying the challenges in parsing patent claims. In *Proceedings of the 1st International Workshop on Advances in Patent Information Retrieval (AsPIRe'10)*.
[71] Suzan Verberne and Eva D'hondt. 2009. Prior art retrieval using the claims section as a bag of words. In *Workshop of the Cross-Language Evaluation Forum for European Languages*. Springer, 497–501.
[72] Suzan Verberne, Maya Sappelli, Djoerd Hiemstra, and Wessel Kraaij. 2016. Evaluation and analysis of term scoring methods for term extraction. *Inf. Retriev. J.* 19, 5 (2016), 510–545.
[73] Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. TSDAE: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning. arXiv:2104.06979. Retrieved from https://arxiv.org/abs/2104.06979
[74] Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2021. GPL: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. arXiv:2112.07577. Retrieved from https://arxiv.org/abs/2112.07577
[75] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Adv. Neural Inf. Process. Syst.* 33 (2020), 5776–5788.
[76] J. Wen, Z. Zhong, Y. Bai, X. Zhao, and M. Yang. 2022. Siat@ coliee-2022: Legal case retrieval with longformer-based contrastive learning. In *Proceedings of the 16th International Workshop on Juris-informatics (JURISIN 22)*.
[77] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. arXiv:1910.03771. Retrieved from https://arxiv.org/abs/1910.03771
[78] Ming Yan, Chenliang Li, Chen Wu, Bin Bi, Wei Wang, Jiangnan Xia, and Luo Si. 2019. IDST at TREC 2019 deep learning track: Deep cascade ranking with generation-based document expansion and pre-trained language modeling. In *Proceedings of the Text Retrieval Conference (TREC'19)*.
[79] Eugene Yang, David D. Lewis, Ophir Frieder, David A. Grossman, and Roman Yurchak. 2018. Retrieval and richness when querying by document. In *Design of Experimental Search & Information REtrieval Systems (DESIRES)*, 68–75.

[80] Yin Yang, Nilesh Bansal, Wisam Dakka, Panagiotis Ipeirotis, Nick Koudas, and Dimitris Papadias. 2009. Query by document. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*. 34–43.

[81] Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Applying BERT to document retrieval with birch. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19): System Demonstrations*. 19–24.

[82] Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. 3490–3496.

[83] Masaharu Juliano Yoshioka. 2021. COLIEE 2020: Methods for legal document retrieval and entailment. In *New Frontiers in Artificial Intelligence: JSAI-isAI'20 Workshops, JURISIN, and LENLS'20 Workshops, Revised Selected Papers*, Vol. 12758. Springer Nature, 196.

[84] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS'20)*.

[85] Haotian Zhang, Mustafa Abualsaud, Nimesh Ghelani, Angshuman Ghosh, Mark D. Smucker, Gordon V. Cormack, and Maura R. Grossman. 2017. UWaterlooMDS at the TREC 2017 common core track. In *Proceedings of the Text Retrieval Conference (TREC'17)*.

[86] Ruiqi Zhong, Dhruba Ghosh, Dan Klein, and Jacob Steinhardt. 2021. Are larger pretrained language models uniformly better? comparing performance at the instance level. arXiv:2105.06020. Retrieved from https://arxiv.org/abs/2105.06020

[87] Helong Zhou, Liangchen Song, Jiajie Chen, Ye Zhou, Guoli Wang, Junsong Yuan, and Qian Zhang. 2021. Rethinking soft labels for knowledge distillation: A bias-variance tradeoff perspective. arXiv:2102.00650. Retrieved from https://arxiv.org/abs/2102.00650

Received 28 February 2023; revised 31 August 2023; accepted 16 October 2023