Legal Norm Retrieval with Variations of the BERT Model Combined with TF-IDF Vectorization

Sabine Wehnert
sabine.wehnert@gei.de
Georg Eckert Institute
Leibniz Institute for International Textbook Research
Germany
Otto von Guericke University
Magdeburg, Germany

Viju Sudhi
Shipra Dureja
Libin Kutty
Saijal Shahania
<firstname>.<lastname>@st.ovgu.de
Otto von Guericke University
Magdeburg, Germany

Ernesto W. De Luca
deluca@gei.de
Georg Eckert Institute
Leibniz Institute for International Textbook Research
Germany
Otto von Guericke University
Magdeburg, Germany

ABSTRACT

In this work, we examine variations of the BERT model on the statute law retrieval task of the COLIEE competition. This includes approaches to leverage BERT's contextual word embeddings, fine-tuning the model, combining it with TF-IDF vectorization, adding external knowledge to the statutes and data augmentation. Our ensemble of Sentence-BERT with two different TF-IDF representations and document enrichment exhibits the best performance on this task regarding the F2 score. This is followed by a fine-tuned LEGAL-BERT with TF-IDF and data augmentation and our third approach with the BERTScore. As a result, we show that there are significant differences between the chosen BERT approaches and discuss several design decisions in the context of statute law retrieval.

CCS CONCEPTS

• **Applied computing** → **Law**; • **Information systems** → **Document representation**; *Language models*; *Similarity measures*; *Relevance assessment*; • **Computing methodologies** → Neural networks.

KEYWORDS

contextual word embeddings, document enrichment, data augmentation, legal information retrieval

ACM Reference Format:
Sabine Wehnert, Viju Sudhi, Shipra Dureja, Libin Kutty, Saijal Shahania, and Ernesto W. De Luca. 2021. Legal Norm Retrieval with Variations of the BERT Model Combined with TF-IDF Vectorization. In *ICAIL '21: International Conference on Artificial Intelligence and Law, June 21–15, 2021, online.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3462757.3466104

1 INTRODUCTION

In this paper, we describe our approach for the retrieval task 3 of the COLIEE competition, based on the English version of the Japanese Civil Code. Legal Statute Retrieval is a challenging task due to the short, abstract nature of law articles and at times very specific scenarios described in a query. Having a requirement of high recall and reasonable precision values at the same time, most state-of-the-art retrieval approaches are not reliable enough for real-life scenarios. Since their first use in the Competition on Legal Information Extraction/Entailment (COLIEE), BERT (Bidirectional Encoder Representations from Transformers) approaches received criticism regarding their explainability compared to other traditional machine learning methods that have been employed. A term frequency - inverse document frequency (TF-IDF) document representation may appear more reliable to a legal practitioner, since ranking based on term statistics and its side effects are interpretable. Nevertheless the ongoing success of BERT-based methods may justify their continued use. A big part in the performance of BERT models is attributed to the rigorous pre-training on huge corpora and the large model capacity, for example with 768 embedding dimensions in the base model. A pre-trained language model has encountered a big diversity of text and has been therefore exposed to enough examples to form a decent representation for the contextual use of words. Hence, substantial knowledge from a distributional semantics point of view appears to be helpful in this task. However, using a standard BERT model on the plain training data is not sufficient. The COLIEE data for task 3 is quite limited in its size due to massive human effort behind its creation. This poses a challenge in the training of deep learning models in addition to the legal jargon, which differs from the language the models may be pre-trained on. We investigate whether data manipulations such as augmentation and the decomposition of relevant articles helps in this issue.

In previous COLIEE editions, machine learning models have also benefited from other types of external knowledge, for example by using ontologies for information extraction. This motivates us to also enrich the training data and to encode the additional content jointly with the original text by using a BERT model. In addition, several BERT approaches have emerged, with variants specializing on our domain, such as LEGAL-BERT [3]. Aside from the model selection, we can also choose between using the BERT model in a supervised setting as a relevance classifier or to extract contextual word embeddings and use them directly for similarity scoring.

We use our three runs in the competition to test a few BERT variations based on those considerations. The contributions of our three runs are:

- We combine Sentence-BERT with modifications on TF-IDF vectorization and document enrichment strategies

ICAIL '21, June 21–15, 2021, online

ACM ISBN 978-1-4503-8526-8/21/06…$15.00
https://doi.org/10.1145/3462757.3466104

- We perform dataset manipulations to train a BERT classifier for retrieval and combine it with TF-IDF vectorization
- We test similarity scores obtained from the BERTScore

The remainder of this work is organized in the following way: In Section 2, we describe approaches using TF-IDF, BERT models and their various use in the past editions of the COLIEE competition. Section 3 contains conceptual descriptions for each of our three runs: Sentence-BERT Embedding with TF-IDF, LEGAL-BERT with TF-IDF, and BERTScore. Section 4 consists of more details on our experimental setup, results and a following discussion. In the final section we conclude our results and indicate future research potential.

2 RELATED WORK

In this section, we describe related research of retrieval methods we used. In particular, we investigate past uses of the respective method within the COLIEE competition. First, we briefly review the TF-IDF (term frequency - inverse document frequency) vectorization and its place within the competition. Second, we collect approaches which are similar to our methods which are using BERT (Bidirectional Encoder Representations from Transformers) and also make a distinction between our methods for the three runs we submitted and the existing work.

2.1 Retrieval with TF-IDF

TF-IDF vectorization gives an idea of how relevant a particular term is within a document and within the document collection. TF-IDF vectors represent a document by assigning a higher weight to terms which appear relatively frequent in few documents - compared to their usual occurrence in the rest of the corpus - by discounting the term frequency with the inverse document frequency. As Beel et al. [1] comment, the TF-IDF vectorization scheme is the most widely used approach for content-based filtering for recommender systems and related text mining domains. In the COLIEE competition multiple teams in the previous years used TF-IDF vectors with or without other representation methods to retrieve the relevant articles given a query [8, 10]. In legal information retrieval, TF-IDF only is still a valuable baseline model because its results are easy to interpret for domain experts. However, in previous editions of the competition, a mere TF-IDF approach could not reliably achieve winning scores. When used in conjugation with any other embedding techniques, competitive results were attainable. One such approach has been employed by Rabelo et al. [9] to address the case law entailment task. They employ two different cosine similarity approaches and a confidence score from BERT [4] to improve the extraction/entailment results. We adopt a similar approach in our second run by combining TF-IDF similarity scores with the softmax scores obtained from fine-tuned BERT models. However, we also differ in the way of choosing documents to calculate similarity and in thresholding for the retrieval task.

2.2 Retrieval with BERT

Nowadays, many pre-trained deep learning-based language models are available, coming from neural network architectures for Natural Language Processing (NLP) with significant improvements for various downstream tasks, such as single sentence classification, question answering tasks, sentence tagging tasks and paraphrase identification. BERT introduced by Devlin et al. [4] is currently a common choice for such downstream tasks, replacing various traditional NLP pipelines. Following this, there has been an exhaustive study about the applications of BERT and experiments to investigate different fine-tuning methods for these pre-trained models by Sun et al. [12]. They present various fine-tuning strategies for BERT on a text classification task, providing a general solution for achieving state-of-the-art results on a variety of text classification datasets. We follow a few of these best practices for better performance with our selected models, such as:

(1) the use of the right combination of different hyperparameters that directly affect the learning,
(2) the importance of the selection of the correct value of warm-up steps,
(3) how concentrating on the decay rate can help to converge towards the minima and when the learning rate decay should start, and
(4) the right combination of batch-size with the number of epochs and warm-up steps.

2.2.1 Fine-tuning BERT. When Devlin et al. proposed the BERT model, they described its use on downstream tasks in two phases: a pre-training and a fine-tuning phase [4]. Therefore, its intended use for any further task is to first fine-tune it in order to achieve the desired performance. Nowadays, BERT is not always fine-tuned, sometimes the pre-trained model and its embeddings perform well enough, if the domain is not substantially changed compared to what the model was pre-trained on. However, for the legal domain it can be worthwhile to adapt an existing BERT model to the different use of vocabulary in that context. This can be also observed in the past COLIEE competitions. For the task on statute law retrieval, Nguyen et al. [8] use an ensemble of BERT models. The publicly available *bert-base-uncased*[1] model is pre-trained on the English language Wikipedia and BooksCorpus [14] and then fine-tuned by Nguyen et al. on the COLIEE training data. The model is combined with another special *bert-base-uncased* model that is further trained with the masked language model (MLM) on the entire COLIEE data (BERT-CC) and fine-tuned on training data to obtain a measure of relevance. This ensemble of BERT achieved the best F2 score for the validation data. As their BERT-CC focuses on legal domain knowledge, we reviewed further special BERT models. We found RoBERTa (Robustly Optimized BERT pre-training Approach) [7] with its variants, and LEGAL-BERT [3] as promising models for task 3. RoBERTa [7] is optimized with some alterations to essential hyperparameters in BERT and trained with relatively bigger batches over a large training data size. It also excludes BERT's next-sentence prediction task, allowing it to improve on MLM over BERT. This leads to a better performance on various baseline NLP downstream tasks [7]. Similarly, LEGAL-BERT [3] is an adaption of BERT in the legal domain where pre-training is carried out on a collection of several fields of English legal text, such as contracts, court cases, and legislation. This special BERT model has been performing better than the original version of BERT on legal domain-specific tasks [3].

[1] https://huggingface.co/bert-base-uncased

2.2.2 *Contextual Embeddings from BERT.* Aside from further training the whole BERT language model and using it on a classification task, we can also use contextual word embeddings from a pretrained BERT model to determine semantic similarity of the query and the article(s). Contextual word embeddings are computed at runtime. In particular, we obtain different vectors for the same word, when it is used in another context or position in a sentence. In that way, we can also distinguish homonyms when they are accompanied by enough words in the appropriate context. Since the contextual embedding type is quite recent, there is no final consensus in the research community of how to compute the distance between two contextual word embedding sequences. The most common methods are: using the [CLS] token which is often seen as a representation of a whole sentence, using the individual word embeddings or averaging all individual word embeddings in a sentence and then computing a similarity score using the Word Mover's Distance [6] or cosine similarity. In the experiments by Reimers et al. [11], using the mean of the individual contextual word embeddings outperformed the approach with the [CLS] token. A recent approach related to this is the BERTScore [13]. After computing the pairwise cosine similarity of all token-wise contextual embeddings from two sentences, BERTScore selects token pairs between the two sentences which have the highest cosine similarity. Those similarities are summed up and discounted by the words in the sentence to obtain precision, recall and the according F1-score. Optionally, the BERTScore can also incorporate IDF weighting. We employ the BERTScore in our third run to test whether the mere embeddings of BERT can also capture enough context in the training data, compared to using document enrichment or fine-tuning on a BERT model for relevance classification.

For us, it is particularly interesting that there are recent approaches for fine-tuning a language model specifically to obtain meaningful sentence embeddings [2, 11]. In the previous COLIEE edition, the *cyber* team achieved the best performance among all teams using the universal sentence encoder, TF-IDF and a support vector machine for the case law retrieval task [10]. Hence, we assume that TF-IDF combined with sentence embeddings could also work well on the statute law retrieval task. A new advancement on sentence embeddings has been made by Reimers et al. [11] who introduce Sentence-BERT. It outperforms the existing embedding methods and is found useful for multiple downstream tasks. It is based on a Siamese network architecture which ties the weights of two BERT models (one for each input sentence) that are updated during fine-tuning. As a default, the mean is used to pool the obtained contextual word embeddings from each BERT model. Then, the two resulting sentence embeddings are concatenated with their element-wise difference, so that the final softmax layer can predict a class. We have made use of this state-of-the-art embedding approach to create a richer and more meaningful numeric representation of each article and query pair in our first run. In a previous COLIEE edition, Kim et al. [5] have employed a Siamese Deep Convolutional Neural Network for the entailment task, which results in better performance compared to regular Convolutional Neural Networks. They attribute their success to the Siamese architecture which requires less parameters due to the weight sharing mechanism and a lower risk of overfitting. For this reason, we assume that a sentence embedding based on a similar architecture may be a good fit for the COLIEE data and also perform well on the retrieval task.

Table 1: Methods for each run for task 3

Run Name	Method
OvGU_run1	Sentence-BERT + TF-IDF + data enrichment
OvGU_run2	LEGAL-BERT + TF-IDF + data augmentation
OvGU_run3	BERTScore

Overall, TF-IDF vectorization and BERT-based approaches have already been tried in course of the past COLIEE editions. Nevertheless, there are many options to employ both methods, while fine-tuning can affect the outcome substantially.

3 STATUTE RETRIEVAL TASK

This section describes in detail the three different methods we proposed and implemented for task 3 in COLIEE 2021, as mentioned in Table 1. While the first method exploits Sentence-BERT coupled with TF-IDF vectors and data enrichment, the second method uses LEGAL-BERT with TF-IDF vectors and data augmentation. The third method applies the BERTScore to solve the problem at hand.

3.1 Sentence-BERT Embedding with TF-IDF

The first run involves a combination of 2-stage TF-IDF vectorization with Sentence-BERT embeddings. This run was the best out of all the runs submitted for task 3 in COLIEE 2021. An overview of the approach is depicted in Figure 1 and described in the following.

We start by enriching the training data with multiple adjustments as described in Table 2. This enrichment helps us to create vectors for each article in the Civil Code which are more unique than those the training data itself could deliver. A concrete example of the enrichment process for Article 177 can be found in Table 3. We enrich each article in the training data as follows:

- **Metadata:** We add structural information using the section titles in the Civil Code. In that way, hierarchical relations between articles within the same Part, Chapter, Section and even Subsection are modeled.
- **Crawled data:** We crawl Japanese open source commentary on the Civil Code articles and thereby potentially enrich the original article text with general remarks, corner cases, previous versions, related articles and a reasoning for the relation.
- **Relevant queries from training data:** We parse the training data labels of task 4 (entailment) to enrich our training data of task 3 with queries that have a positive entailment relationship. With a positive entailment relationship we can be sure that the queries added correspond to the meaning of the article and can help in determining relevance, too.

After data enrichment, we encode the enriched texts with the TF-IDF vectorizers and the Tokenizer[2] for our Sentence-BERT and progress to the final relevance score calculation with the following steps:

[2] https://huggingface.co/distilroberta-base

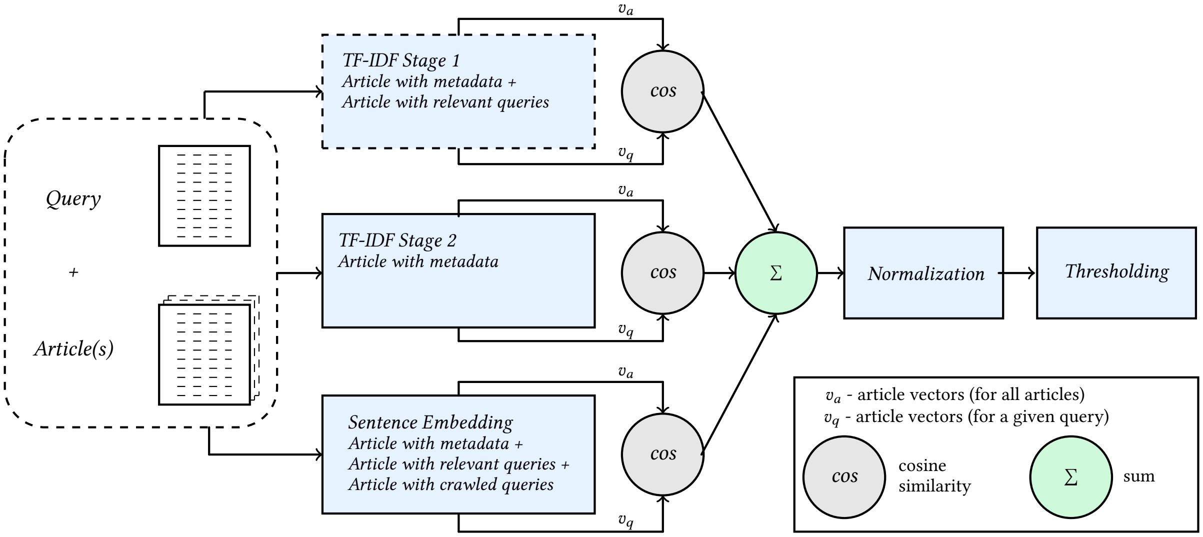

Figure 1: Overview of the approach using Sentence-BERT embeddings with TF-IDF.

Table 2: Data enrichment for the statute retrieval task

	Description
Articles with metadata	training data + details regarding *Part*, *Chapter*, *Section* and *Subsection*.
Articles with crawled data	training data + translated crawled data from the website https://ja.wikibooks.org/
Articles with relevant queries	training data + queries from training data if the entailment label is *Y* for the respective article.

(1) TF-IDF vectors are computed for queries and articles together as a two-stage process. In the first stage, we rely on sub-linear term frequency scaling and L2 normalization while computing the vectors. Articles are enriched by a combination of *Articles with relevant queries* and *Articles with metadata*.

The vectors $\vec{v}$ are computed by the following equations 1 - 4,

$$tf_{t,d} = (1 + log(tf_{t,d})) \tag{1}$$

$$idf_t = log(\frac{N}{1 + df_t}) \tag{2}$$

$$w_{t,d} = tf_{t,d} * idf_t \tag{3}$$

$$\vec{v} = \frac{\vec{w}}{\sqrt{\sum_i w_{i,d}^2}} \tag{4}$$

where,

- $tf_{t,d}$ is the term frequency - frequency of term t in document d. Here, documents are the individual articles of the Civil Code.
- N is the total number of documents in the collection.
- df_t is the document frequency - frequency of term t in the collection.
- $w_{t,d}$ is the weight which is the product of term frequency and inverse document frequency.

The vectors after a single stage of TF-IDF vectorization yielded significant precision-recall trade-offs reflected in the relatively lower F2 scores. This prompted us to provide a different, but unique representation of the articles, which ended up in a second stage of TF-IDF where query and article vectors have been created considering only the *Articles with metadata* enrichment. The combination of both stages acts as a counter-balance in the trade-off.

(2) Sentence-BERT embeddings for each article are created with the enrichment described in Table 2. We rely on the implementation[3] by Reimers et al. [11] and use the pre-trained *paraphrase-distilroberta-base-v1* model to create the article and query embeddings. We select the aforementioned paraphrase model because it was trained on millions of paraphrase examples and is reportedly performing well on natural language inference tasks[4].

(3) Finally, for each query-article pair we compute the cosine similarity to determine the relevance of each article for the respective query. For each pair, we obtain three different similarity scores from the first stage TF-IDF, second stage TF-IDF and Sentence-BERT embeddings. The sum of these scores is then normalized and we empirically determine a threshold to filter out the best relevant articles for each test query.

[3] https://github.com/UKPLab/sentence-transformers

[4] https://www.sbert.net/docs/pre-trained_models.html#paraphrase-identification

Table 3: Example data enrichment for Article 177 of the Civil Code

training data
Article 177: Acquisitions of, losses of and changes in real rights on immovables .. and other laws regarding registration.
Metadata
Part: II Real Rights *Chapter:* I General Provisions *Section:* 3 Extinctive Prescription *Subsection:* Requirements of Perfection of Changes in Real Rights on Immovables
Crawled data
Comprehensive succession - The range of changes in property rights that require registration has been determined .. Legal evidence theory - What kind of person is referred to as "a person who has a legitimate interest .. .. (161 unique words in total, shortened to conserve space)
Relevant queries from training data
– *H19-11-3:* In a case where A bought a registered building owned by B .. his/her acquisition of ownership of that building. – *H21-24-E:* If a mortgage creation contract has the agreement of the mortgagee .. there is no registration of its creation. – *R01-6-A:* In cases A sold Land X belonging to A and B sold it to C, C may be asserted .. for sales without security.

For developing an explanatory dialogue in a real setting, the additional text we gained in the enrichment steps can be marked in a different font style. Then, we can highlight important keywords based on the scores of each TF-IDF stage. Since we did not apply any weighting during the cosine similarity computation of the Sentence-BERT embeddings, the similarity between the word vectors of query and article can be visualized using a heatmap.

3.2 LEGAL-BERT with TF-IDF

For our run 2, we treat task 3 as a sentence-pair classification task to predict the relevance of 1 if the given article is related to the query and 0 otherwise. Considering its good performance on previous retrieval tasks, we choose to work with a BERT model. A variety of BERT models that are pre-trained on different datasets can be used for addressing domain-specific tasks with fine-tuning.

3.2.1 BERT configuration. Following this convention with fine-tuning, we initially used bert-base-uncased which has 12 hidden layers with 768 hidden units in each layer 12 attention heads. A classification head is added on top of the base model consisting of a single layer of fully connected linear neurons. We use the softmax function to get a probability distribution for the two labels and use cross-entropy loss with Adam optimizer to fine-tune the model. We split the training dataset into two parts for fine-tuning (∼ 85 % training) the model and use the rest of the dataset for validation (all queries starting with id "R01-*").

3.2.2 Data Pre-processing. An overview of the pre-processing for the LEGAL-BERT with TF-IDF approach is illustrated in Figure 2. We pre-process both the training and validation splits in the following manner:

(1) **Data Decomposition:** This is performed to extract each relevant article for a given query to form separate instances. For every query, there is one or more than one article associated and relevant to it. We take individual articles to create a new instance in the training dataset so that the query can be divided into multiple instances against all of its relevant articles. An example is shown in the Table 4 for the query with the Pair ID "H27-22-4":
Query Q: *"In the contract for deposit for value, if the performance of the obligation to return deposited Thing has become impossible due to reasons not attributable to the depositary, he/she may not claim remuneration from the depositor, with respect to the period after the impossibility of performance of the agreed duration."*
After achieving better results with data decomposition than with the original dataset, we further extract referenced articles from each relevant article of the query using regular expressions and append that as well to form multiple instances of query-article pairs for each query. The same example is extended further for Approach 2 in Table 5.
However, this extensive decomposition of referenced articles did not optimize our recall further. We assume this is plausible as these articles are supporting articles to the relevant article content but are not directly relevant to the query. We compare the results with and without data decomposition and summarize them in Table 6. We decided to go with Approach 1 where we have a better recall score.

(2) **Data Augmentation:** We use the non-relevant articles to reduce data imbalance. For this, we enriched this decomposed dataset using the top 50 non-relevant articles for each query instance. These non-relevant articles are based on the highest cosine similarity between TF-IDF vectors of the relevant article to all the articles excluding the other relevant ones for the respective query. This approach is similar to the implementation by Nguyen et al. [8], where they considered query-article similarity. However, we assume that article-article similarity is better suited than query-article similarity since we find that articles are more related to each other in terms of cosine similarity than they are to the queries. Based on the cosine similarity, we select only the top 50 non-relevant articles as training examples, since we did not intend to reintroduce the data imbalance that we attempted to overcome with augmentation.

Table 4: Approach 1 - Data decomposition of multiple articles for each query into multiple instances

Queries	**Articles**
Before Pre-processing	
Query Q	**Article 665** The provisions of Articles 646 through 648, Article 649, and Article 650, paragraphs ...
	Article 648 (1) In the absence of any ... (2) ... the provisions of Article 624 ... (3) ... course of performance.
	Article 536 (1) If the performance ... (2) ... obligee for the benefit.
After Pre-processing	
Query Q	**Article 665** The provisions of Articles 646 through 648, Article 649, and Article 650, paragraphs ...
Query Q	**Article 648** (1) In the absence of any ... (2) ... the provisions of Article 624 ... (3) ... course of performance.
Query Q	**Article 536** (1) If the performance ... (2) ... obligee for the benefit.

Table 5: Approach 2 - Data decomposition of multiple articles and their referenced articles for each query into multiple instances

Queries	**Articles**
Before Pre-processing	
Query Q	**Article 665** The provisions of **Articles 646 through 648**, **Article 649**, and **Article 650**, paragraphs ...
	Article 648 (1) In the absence of any ... (2) ... the provisions of **Article 624** ... (3) ... course of performance.
	Article 536 (1) If the performance ... (2) ... obligee for the benefit.
After Pre-processing	
Query Q	**Article 665** The provisions of Articles 646 through 648, Article 649, and Article 650, paragraphs ...
Query Q	**Article 646** (1) A mandatary must deliver to the mandator monies and other things received during ...
Query Q	**Article 647** If the mandatary has personally consumed monies that were to be delivered to the mandator ...
Query Q	**Article 648** (1) In the absence of any special agreements, the mandatary may not claim remuneration ...
Query Q	**Article 649** If costs will be incurred in administering the mandated business, the mandator must ...
Query Q	**Article 650** (1) If the mandatary has expended costs found to be necessary for the administration ...
Query Q	**Article 624** (1) An employee may not demand remuneration until the work the employee promised ...
Query Q	**Article 536** (1) If the performance ... (2) ... obligee for the benefit.

Table 6: Results on the validation set for different data pre-processing approaches of run 2

Model	**Prec**	**Recall**
bert-base-uncased without decomposition	0.1392	0.3973
bert-base-uncased with Approach 1	0.2529	**0.4421**
bert-base-uncased with Approach 2	0.1179	0.4300

(3) **Augmenting the Original Dataset:** To ensure that the original data could still influence the model, we also append the original data. In other words, for each query without any data decomposition, all relevant articles are processed in an instance as they are given in the dataset. This increases the number of relevant articles for each query at the cost of generating some duplicates, since for queries which have only one relevant article, those are already obtained at the step of data decomposition. Overall, the three pre-processing steps increase the number of training instances by the factor 10.

3.2.3 Fine-Tuning. On comparing the results with the legal domain-specific pre-trained BERT, *bert-base-uncased* was outperformed by *legal-bert-base-uncased* and *legal-RoBERTa* on similar hyperparameters. We finally choose *legal-bert-base-uncased* as it indicated the most satisfactory results to further test with different experimental setups in Section 4.1. To extract relevant articles for a given query during testing, we combine each query with all the articles. For LEGAL-BERT, we applied the softmax function to the logits predicted from our model. For each query-article pair, we obtain two softmax probability values, indicating the non-relevance and relevance of the article to the query. We only consider the softmax probabilities of the relevance column. To avoid the underflow of softmax probabilities of top relevant articles, we max-normalize these scores. At the same time, we also calculate the query-article cosine similarity of all the articles for each query. The similarity scores are also max-normalized for the same reasons as stated above. We ultimately compute an average of these two normalized scores. To select the top-n relevant articles we use a threshold value selected based on the precision-recall trade-off for the validation set. The time for training the LEGAL-BERT model increases from 2 minutes on the original dataset to 2 hours on the fully enriched dataset[5]. The larger amount of text in the enriched data does not have a significant impact during the test phase. At runtime, we directly process the new query and all pre-stored enriched articles

[5] We used an NVIDIA Quadro RTX 8000 to accelerate training.

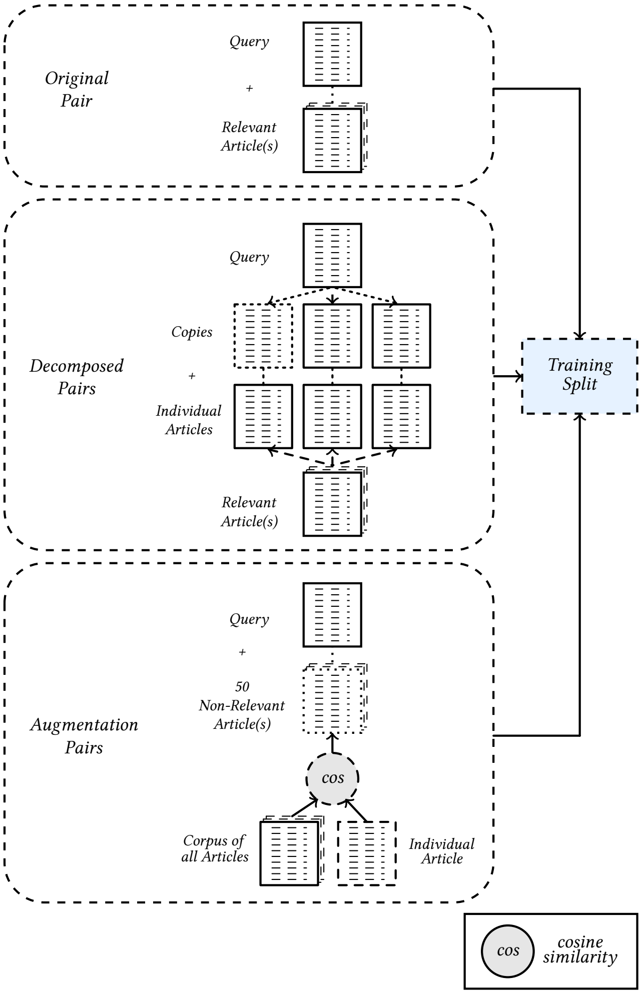

Figure 2: Pre-processing for LEGAL-BERT with TF-IDF.

with the already trained language model, so that the prediction is not causing any noticeable delay in the system's response time.

3.3 BERTScore

In the third run of the retrieval task, we use the BERTScore [13]. In Zhang et al.'s implementation[6], BERTScore outputs precision, recall, and F1 measure. We use the F1 score as the main value for further analysis. To decide how many articles should be retrieved per query, the following steps are used to determine the BERTScore threshold (K):

(1) We calculate the BERTScore for each article given a query, and then rank the result set in descending order.
(2) For each query in the training data we select the top n documents, with a BERTScore value for each n. From this, we select the BERTScore value K as a threshold where the F2 score is maximized, since the task performance is evaluated on the F2 score.

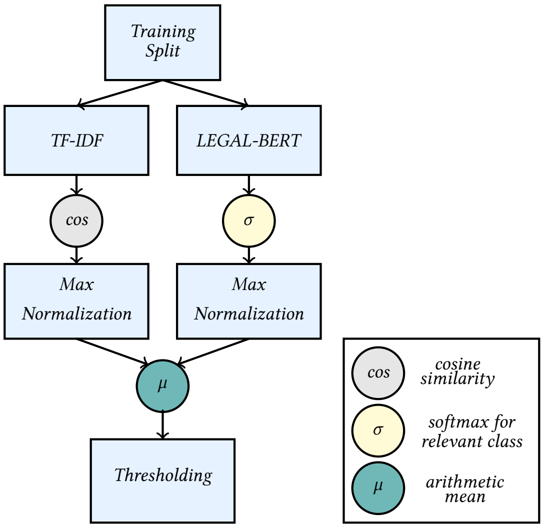

Figure 3: Overview of the approach using LEGAL-BERT with TF-IDF.

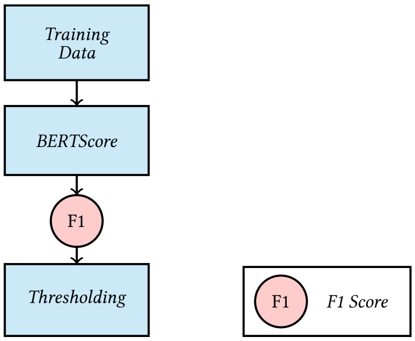

Figure 4: Overview of the approach using BERTScore.

(3) For the test data, we take the average K of all the BERTScore thresholds from the training data.

4 EVALUATION

In this section, we evaluate the previously presented methods. First, we describe details of the experimental setup. Second, we proceed to show the respective results on the competition task. Third, we discuss our findings. We evaluate our runs for the statute retrieval task on our validation split using variations in hyperparameters for training different models. The evaluation contributes a quantitative and qualitative analysis of how the runs perform with different hyperparameter settings - if applicable - and if they are comparable to each other. We discuss a few of the experiments below.

4.1 Experimental Setup

4.1.1 Sentence-BERT Embedding with TF-IDF. To enrich the articles further, we make use of the crawled content[7]. We extract all the

[6] https://github.com/Tiiiger/bert_score

[7] https://ja.wikibooks.org/wiki/民法第<*id*>条 , where <*id*> stands for the Article ID of the different articles in the Civil Code.

Table 7: Two stages of TF-IDF counter-balancing the precision-recall trade-off with Sentence-BERT.

	F2	Prec	Recall
Validation data			
with 1st stage TF-IDF	54.67	50.16	61.98
with 2nd stage TF-IDF	53.74	49.54	60.27
with both stages	**56.52**	**52.60**	**63.39**
COLIEE 2021 test data			
with 1st stage TF-IDF	72.98	66.77	78.40
with 2nd stage TF-IDF	73.02	66.28	**79.63**
with both stages	**73.02**	**67.49**	77.78

paragraph tags (<*p*>) and the list tags (<*ol*>, <*ul*>, <*dl*>, <*li*>) to get relevant information about the articles. This is motivated by the team *TRC3* in the previous year of COLIEE [10], where they used the content in Japanese itself. However, we translate the fetched content to English using the *google-trans-new* package [8]. To vectorize these enriched articles and queries we used the TfidfVectorizer from *scikit-learn.*

To address the problem of the precision-recall trade-off, we use two-stages of TF-IDF, which is motivated from previous experiments we conducted on queries starting with the id "R01-*", as shown in Table 7. It is evident on the validation data that the two-stage TF-IDF can counter-balance the classical trade-off between precision and recall, considering the improved F2 scores. For the COLIEE 2021 test data the second stage has a positive effect on the F2 score as well, though it is not as significant as we found it for our validation split.

The threshold value to filter out the top n relevant articles was found empirically. After normalizing the sum of the scores from the two stages of TF-IDF and Sentence-BERT embeddings, we considered the top 4 articles. This was purely based on our validation data, where none of the queries had relevant articles exceeding a count of four. This is true with COLIEE 2021 test data as well, where none of the articles have more than 4 relevant articles. To find a threshold for the scores of these articles, an index-based threshold was found to be better than a single value for the whole set. Accordingly, we take the article in the 1st index (with a score of 1.0) and then set a threshold of 0.91 or higher for the articles if found in the 2nd index and a threshold of 0.85 or above if found in the subsequent indices.

4.1.2 LEGAL-BERT with TF-IDF. To decide among the three alternative models that we selected as candidates for our run 2 as discussed in Section 3.2, we validate them on various hyperparameter settings and observe that the default hyperparameters of the Adam Optimizer with a selective change in the learning rate ranging from 1e^{-03} to 1e^{-06}, 1e^{-05} achieve the best results among all three models (see Table 8) when trained on 3 epochs for batch-size 16. Considering the highest recall score, we select *legal-bert-base-uncased* to be our final choice for run 2.

Table 8: Results on validation set for run 2 candidates

Model	Prec	Recall
bert-base-uncased	0.2529	0.4421
legal-bert-base-uncased	0.3447	**0.5357**
legal-RoBERTa	0.2205	0.4866

Table 9: Task 3 Results for COLIEE 2021

Position	Run	F2	Prec	Recall	R_30
1	OvGU_run1	**0.7302**	0.6749	0.7778	0.8515
9	OvGU_run2	0.6717	0.4857	**0.8025**	0.9010
18	OvGU_run3	0.3016	0.1570	0.7006	0.7030

Further, we experiment with warm-up steps, introduce a decay rate and did some hyperparameter tuning to optimize our results. We notice that with 3500 warm-up steps and a decay rate of $0.1^{(1+epoch)}$, we achieve the best performance. We then perform further training on the validation set with an increased batch size of 24. We then create an ensemble of *legal-bert-base-uncased* and TF-IDF vectors, both with max-normalized similarity scores for the article-query pairs, assigning equal weights to both the scores. Finally, we fetch the articles that are above the threshold value of 0.5.

4.1.3 BERTScore. For the BERTScore, we use the model type *bert-base-uncased*, 9 layers and no re-weighting with IDF. This setup was determined based on the performance on our validation data which we also used before (queries starting with the id "R01-*"). The text is processed with the regular Tokenizer of BERT and we pass query and article(s) without further modification to the scorer of the original BERTScore implementation. Our thresholding strategy for this run results in a threshold value of 0.63331205.

4.2 Results

Our first run, OvGU_run1 obtained the first position for its F2 score in the overall task evaluation for COLIEE 2021. OvGU_run2 also has the best recall sharing the position with the run *JNLP.CrossLMultiLThreshold*, closely followed by OvGU_run1. While considering Recall at 30, our runs have the third best (for OvGU_run2) and the fifth best (for OvGU_run1) scores. The results for our runs are summarized in Table 9. Values in bold are the best scores for the corresponding metric.

4.3 Discussion

We assume that our first run provides reliable results because of the combination of contextual Sentence-BERT embeddings with the TF-IDF vectors. This is supported by the test query *R02-1-A: "The family court may decide to commence an assistance also in respect of a person whose capacity to appreciate their own situation is extremely inadequate due to a mental disorder."*,
as shown in Table 10. For this query, only Sentence-BERT embeddings could retrieve the most relevant Article 15 which was not retrieved in either stage of TF-IDF vectorization. The Article 15 has

[8] https://github.com/lushan88a/google_trans_new

Table 10: Comparison of results for query *R02-1-A*

Method	Retrieved articles
With TF-IDF stages	**Article 11**
With Sentence-BERT	**Article 15**, Article 7, **Article 11**
With combination	**Article 11**, **Article 15**
Relevant articles	**Article 15**, **Article 11**

Table 11: Comparison of results for query *R02-24-U*

Method	Top retrieved articles
With Run 1	**Article 563**, Article 566, Article 567
With Run 2	**Article 563**, Article 565, Article 567, **Article 562**
Relevant articles	**Article 562**, **Article 563**

the following content:
"(Decisions for Commencement of Assistance)
Article 15 (1) ***The family court may decide to commence an assistance in respect of a person whose capacity to appreciate their own situation is inadequate due to a mental disorder****, at the request of the person in question, that person's spouse, that person's relative within the fourth degree of kinship, the guardian, the guardian's supervisor, the curator, the curator's supervisor, or a public prosecutor; provided, however, that this does not apply to a person with respect to whom there are grounds as prescribed in Article 7 or the main clause of Article 11. (2) The issuance of a decision for commencement of assistance at the request of a person other than the person in question requires the consent of the person in question. (3) A decision for commencement of assistance must be made concurrent with a decision as referred to in Article 17, paragraph (1) or a decision as referred to in Article 876-9, paragraph (1)."*
It turns out that for this query-article pair, we have a significant term overlap, which may be diluted by the whole article length. In that way, sentence-based approaches in general may work well for this query. Only our run *OvGU_run1* and *TR_HB* have a 100% F2 score for this query.

On comparing our different runs, we find interesting similarities in the articles retrieved by each of them. This might possibly be because of the common TF-IDF coupling in the first two runs. We did not expect that embeddings from a pre-trained model (in run 1) could give more or less comparable results with those from a model further trained on the COLIEE dataset (in run 2).

Another insight from the results is how thresholding plays a significant role in the retrieval task. For example, the test query *R02-24-U:*
"A donor shall assume a duty to retain the subject matter exercising care identical to that he/she exercises for his/her own property until the completion of such delivery.",
retrieved only one relevant article with run 1 but both relevant articles with run 2. This is described in Table 11. Drawing conclusions from this query - out of the many similar queries, we are not surprised to see the fine-tuned model of run 2 retrieve 74 candidate articles and run 1 retrieving only 70 candidate articles from a total of 101. This results in run 2 with the overall best recall of 0.8025.

With BERTScore, the interesting query to analyse is the test query *R02-17-I:*
"In the case that D manifests the intention to release another obligor (C) from the obligation to D, even if neither D nor B manifests a particular intention, D may not claim the payment of 600,000 to another obligor (A)."
We are able to retrieve 3 out of 4 articles (Article 439, 440 and 441) using this technique which was the highest number when compared to other teams for this query. However, this result can be attributed to the threshold we selected, with high recall and lower precision. The ranking of the articles by the BERTScore is only average even for this query, considering the Mean Average Precision (MAP). The MAP score is only 0.0509 for run 3, while run 1 gets 0.0299 and run 2 achieves 0.1250. The best MAP score for this query with a value of 0.2309 was obtained by the team *JNLP* with their run called *JNLP.CrossLBertJP*. For assessing the final ranking performance of run 3, we can compare its MAP score to other teams. Also here we observe that the BERTScore with a MAP score of 0.5557 is the fourth-lowest performing run in the competition, whereas our run 1 achieves 0.7496 and run 2 has the highest overall MAP score among our runs of 0.7571. The best MAP score of 0.7947 was achieved by the team *JNLP* with their run *JNLP.CrossLMultiLThreshold*. This leads us to the conclusion that the standard BERTScore without IDF-reweighting or any further combined methods may not be sufficient to solve this task, at least with the the query type distribution of this year's test dataset. We also observe how thresholding influences our F2 score in run 1, so that our method scores higher than a run by the *JNLP* team which has a better ranking performance.

From the results and discussion above, our main takeaways from this COLIEE edition for task 3 are:

(1) Contextual embeddings can significantly enhance retrieval performance when coupled with TF-IDF vectors.
(2) Adding external knowledge to the articles in the form of structural information, entailed queries or definitions can help to make them more unique.
(3) Data augmentation techniques are useful to train a BERT classifier for a retrieval task.
(4) An intelligent or rather more effective thresholding mechanism should be devised to further improve precision and maintain a decent F2 score.

5 CONCLUSION AND FUTURE WORK

In this work, we study variations of the language model BERT for task 3 of the COLIEE competition on statute law retrieval. We find a benefit in combining the BERT model with TF-IDF vectorization and in working on a sentence level with contextual embeddings. Furthermore, it is helpful to test different pre-trained models and fine-tuning, as well as adding external knowledge and data augmentation techniques. Our winning approach is an ensemble of Sentence-BERT and two different TF-IDF representations with different extents of document enrichment. In the second run, we fine-tune a BERT classifier for retrieval based on an augmented dataset. The third run is similarity scoring using the BERTScore with thresholding. Future enhancements can consist of an improved thresholding mechanism and of encoding other types of external knowledge, for example named entities.

REFERENCES

[1] Jöran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. 2016. Research-paper recommender systems: a literature survey. *Int. J. Digit. Libr.* 17, 4 (2016), 305–338. https://doi.org/10.1007/s00799-015-0156-0

[2] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, Eduardo Blanco and Wei Lu (Eds.). Association for Computational Linguistics, 169–174. https://doi.org/10.18653/v1/d18-2029

[3] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. *CoRR* abs/2010.02559 (2020). arXiv:2010.02559 https://arxiv.org/abs/2010.02559

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[5] Mi-Young Kim, Yao Lu, and Randy Goebel. 2017. Textual Entailment in Legal Bar Exam Question Answering Using Deep Siamese Networks. In *New Frontiers in Artificial Intelligence - JSAI-isAI Workshops, JURISIN, SKL, AI-Biz, LENLS, AAA, SCIDOCA, kNeXI, Tsukuba, Tokyo, Japan, November 13-15, 2017, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 10838)*, Sachiyo Arai, Kazuhiro Kojima, Koji Mineshima, Daisuke Bekki, Ken Satoh, and Yuiko Ohta (Eds.). Springer, 35–48. https://doi.org/10.1007/978-3-319-93794-6_3

[6] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From Word Embeddings To Document Distances. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015 (JMLR Workshop and Conference Proceedings, Vol. 37)*, Francis R. Bach and David M. Blei (Eds.). JMLR.org, 957–966. http://proceedings.mlr.press/v37/kusnerb15.html

[7] 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. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

[8] Ha-Thanh Nguyen, Hai-Yen Thi Vuong, Phuong Minh Nguyen, Tran Binh Dang, Quan Minh Bui, Vu Trong Sinh, Chau Minh Nguyen, Vu D. Tran, Ken Satoh, and Minh Le Nguyen. 2020. JNLP Team: Deep Learning for Legal Processing in COLIEE 2020. *CoRR* abs/2011.08071 (2020). arXiv:2011.08071 https://arxiv.org/abs/2011.08071

[9] Juliano Rabelo, Mi-Young Kim, and Randy Goebel. 2019. Combining Similarity and Transformer Methods for Case Law Entailment. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law, ICAIL 2019, Montreal, QC, Canada, June 17-21, 2019*. ACM, 290–296. https://doi.org/10.1145/3322640.3326741

[10] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2020. COLIEE 2020: Methods for Legal Document Retrieval and Entailment. https://sites.ualberta.ca/~rabelo/COLIEE2021/COLIEE2020_summary.pdf. Accessed: 2021-05-09.

[11] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 3980–3990. https://doi.org/10.18653/v1/D19-1410

[12] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to Fine-Tune BERT for Text Classification?. In *Chinese Computational Linguistics - 18th China National Conference, CCL 2019, Kunming, China, October 18-20, 2019, Proceedings (Lecture Notes in Computer Science, Vol. 11856)*, Maosong Sun, Xuanjing Huang, Heng Ji, Zhiyuan Liu, and Yang Liu (Eds.). Springer, 194–206. https://doi.org/10.1007/978-3-030-32381-3_16

[13] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=SkeHuCVFDr

[14] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 19–27. https://doi.org/10.1109/ICCV.2015.11