Semantic Legal Searcher:
Neural Information Retrieval-based Semantic Search for Case Law

ACL 2023 Submission

Abstract

This study aims to build a highly performant semantic search model in the field of law by applying neural information retrieval techniques. With classical keyword-based search models, it is difficult for users without domain knowledge of the law to obtain information by searching with appropriate legal terms. In order to solve this problem, we propose a *Semantic Legal Searcher* (SLS), a neural information retrieval-based case law search model. It enables users to search and gain access to legal information even with simple queries rather than professional legal terms. Specifically, the *SLS* process starts with generating good-quality embeddings from a pre-trained language model we created. Next, latent keywords are extracted by a parallel clustering-based topic modeling and then relevance between input queries and legal documents and keywords is estimated by a *multi-interactions* paradigm we developed. Lastly, the *SLS* provides users with semantic similar case laws based on the estimated scores. Experimental results demonstrate that our semantic search model provides relevant precedents for users by understanding legal text and is a powerful tool for information retrieval. The *SLS* can be useful for a lot of real-life applications and allows the general public to easily access legal information.

1 Introduction

The word "Semantic" refers to the meaning associated with language. In the field of search engines, semantic search is meant to improve search accuracy by learning representations of the meaning of the words called embedding. This is a real-valued vector that encodes the meaning of a word such that words closer in the vector space are similar in meaning (Jurafsky et al., 2000). A recent popular approach for generating contextualized embedding is using pre-trained language models (PLMs) like BERT (Devlin et al., 2018). This idea has been extended to sentences-level named sentence embeddings where entire sentences are mapped to vectors. For example, Sentence-BERT (Reimers and Gurevych., 2019) that modified BERT by adding a pooling layer and using Siamese and triplet network structure (Schroff et al., 2015) can produce sentence embeddings. Several embedding techniques with PLMs have quickly dominated the search landscape over recent years.

Classical searches like keyword-based searches have a simple and intuitive process. When a user enters a query to look for, it will return varying results corresponding exactly or well with the query. However, with this traditional method, some users unfamiliar with jargon in a field may find difficulty in accessing the specific database such as legal. To remedy this, we introduce a semantic-based search technique, which is possible for even non-experts in law can more to easily find related precedents by simply entering queries with non-legal terms. This is possible because the semantic search model understands the complex relationships between legal and colloquial terms in embedding space.

In this work, we propose a *Semantic Legal Searcher* (SLS) which is a new conceptual search model based on neural information retrieval. Our main contributions are as follows:

1. We introduce **a Clean Korean Legal Corpus (CKLC)**. This corpus consists of 5.3 pre-processed million sentences of Korean legal text published from 1954 to the present year.
2. We release a language model named **KRLawBERT** that pre-trained Transformer-based models on the CKLC to generate high-quality embeddings and better understand texts in legal domains. We benchmark a series of state-of-the-art pre-training techniques: Masked Language Modeling (Devlin et al., 2018, Liu et al., 2019) and Transformer-based Sequential Denoising Auto-Encoder (Wang et al., 2021).

3. We propose the ***Semantic Legal Searcher* (SLS)** framework by combining semantic document search with clustering-based topic modeling, a method to extract latent keywords within documents. Moreover, the *SLS* includes two new concepts of neural information retrieval. The first technique, *split-merge*, is developed to separate documents into sentences and integrate all encoded sentence-level embeddings. The second technique, *multi-interactions*, is introduced to score semantically similar relevance by matching similarities between queries, documents, and extracted keywords from topic modeling.

Semantic Legal Searcher can find accurate legal information for users' queries, regardless of whether the user is a lawyer or not. In addition, we have verified the practicality of the model in experiments with three specific tasks: Natural language inference (Bowman et al., 2015; Williams et al., 2018), semantic textual similarity (Cer et al., 2017) and legal question-answering tasks. The data, code, and models are available at `https://anonymous.4open.science/r/Semantic-Searcher-F231/`

2 Background

Recently most case law search engines have been designed as keyword-based models and operated on the web. Besides, more than 90% of users of these search engines are lawyers with legal knowledge. However, wouldn't it be possible to create a case law search model easily accessible to the general public with a state-of-art semantic vector technique? To address this question, we first need to understand semantics precisely.

2.1 Semantics

Before the computational linguistics approach, we define the meaning of a word driven by the linguistic study called semantics. The definition of semantics consists of five lexical semantics components and sentence-level semantics: 1) Synonymy; 2) Word similarity; 3) Word relatedness; 4) Semantic frame; 5) Connotation; 6) Sentence semantics.

Synonymy. Two words are synonymous when they are substitutable for one another in any sentence without changing the truth conditions of the sentence, the situations that the sentence would be true. We also say in this case that the two words have the same positional meaning or identical meaning. Synonyms in legal terms include such pairs as *decision / verdict; judgment / ruling; prison / jail; lawyer / solicitor*.

Word Similarity. Even words that do not have synonyms can be similar to each other. For example, *prisoner* and *criminal* are not synonymous, but similar. While synonyms indicate limited relations between word senses, word similarity indicates extended relationships between all words. Knowing the similarity between two words can help in computing how similar the meanings of two sentences or documents are. This is a core component of word meaning for semantic search.

Word Relatedness. The meaning of two words can be related in ways other than similarity. One such type of connection is named word relatedness. Considering the meaning of the words' *prisoner* and *jail*, the two words are not similar words but are certainly related. They are used together in many contextual sentences. One common kind of relatedness between words is whether they belong to the same semantic field which is a lexical set of words grouped semantically that refers to a specific subject (Jackson et al., 2000; Faber et al., 1999).

Semantic Frame. A semantic frame is a conceptual structure that provides a background of beliefs and experiences necessary to interpret the word's meaning (Fillmore et al., 2001). The idea is that the meaning of a word cannot be understood without access to all the knowledge that relates to that word because each word has semantic roles. A legal case, for example, is connected to words such as *accuse*, *crime*, and *judgment*. Knowing that *accuse* and *crime* have this connection makes it possible for a system to know that a sentence like "*Tom has accused Sam of a violent crime.*" could be understood as "*Sam committed a violent crime.*" and that *Tom* has the role of the *prosecutor* in the frame and *Sam* is the *perpetrator*.

Connotation. Some words have affective meanings that are related to a writer's emotions or evaluations. Connotation is a sentiment aspect of a word's meaning. It can be either positive, negative, or neutral. For example, "*The lawyer was small and thin*" has neutral connotations because it is simply a statement of fact. However, the same sentence rewritten as "*The lawyer was small and slender*" has positive connotations, and "*The lawyer was small and emaciated*" has negative connotations.

Sentence Semantics. Sentence-level semantics deal with the meaning of syntactic units larger than lexical semantics, such as phrases, clauses, sentences, and the semantic relationships between them. When understanding the context and intention in long texts, using only individual words would be limited and requires entire sentence-level semantics.

2.2 Limitation of Keyword Search

Keyword-based search is a conventional information retrieval technique based on the occurrence of words in documents. This method is useful for finding information in the database and getting results within a certain amount of time. However, keyword-based search is not able to provide relevant search results excluding entered queries because it suffers from the fact that it does not know the meaning of the queries as we saw in the previous section (§2.1.). The problems of keyword-based search can be summarized in the following: 1) It does not understand the lexical and sentence-level semantics; 2) It cannot search long and complex queries; 3) It cannot provide flexible results to users who lack domain knowledge in the specialized fields. In these problems, the general public is restricted from accessing specific domain databases, such as legal, through keyword searches.

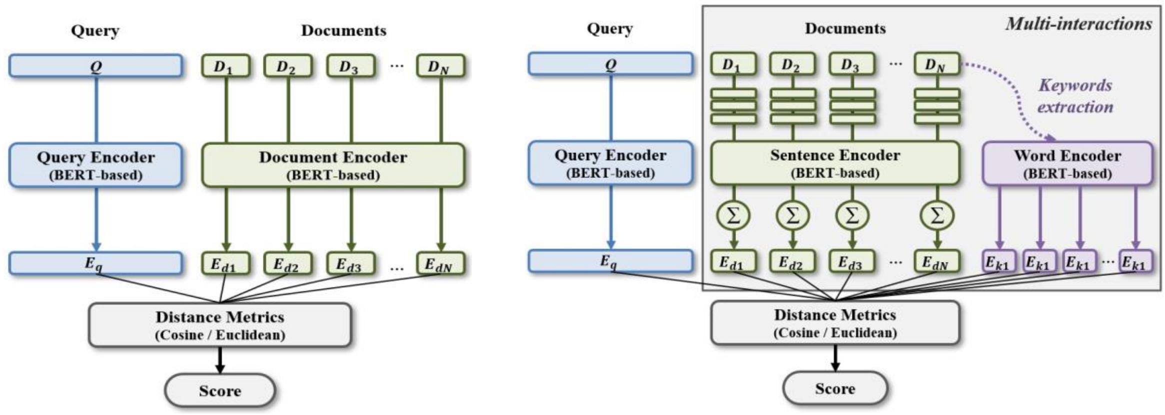
2.3 Semantic Vector

We now turn our attention to semantic-based search. This method keeps the semantic meaning of the text data (§2.1.) by representing each word as a vector. By doing so, we can solve most of the problems from keyword-based searches (§2.2.). The main idea of a semantic vector is that two words that occur in very similar distributions in the vector space have similar semantics. In other words, the semantic vector is meant to represent a word as a point in a multidimensional vector space which is derived from the distributions of word neighbors. These dense vectors for representing words are called word embedding. And the vector representations extended from individual words to entire sentences are sentence embedding. The sentence embedding allows the search model to understand the context, intention, sentiment, and other nuances in the whole text. The semantic-based search uses these embeddings to compare the semantics of an input query and documents rather than performing simple word matching. In semantic-based search areas, embeddings are the key factors in which the search engine improved the understanding of complex queries and recognized the relationship between texts in the database and the input query.

2.4 Related Work to Semantic Search

Semantic-based search have been on the rapid rise and dominated the search landscape by leveraging neural information retrieval (IR). Since the introduction of BERT (Devlin et al., 2018), which can generate fixed-sized contextual embeddings, several neural IR approaches have been tried to apply it to semantic search. A common approach is to feed the query and document pair through BERT and use distance metrics on top of BERT's [CLS] token embedding to generate a relevance score. In subsequent work, Sentence-BERT (Reimers and Gurevych., 2019) generates sentence-level embeddings, and it's possible to estimate the semantic relevance of a pair of documents given a query. ColBERT (Khattab and Zaharia., 2020) introduces the *late interaction* paradigm, where query and document are encoded at fine granularity into token-level multi-embeddings, and relevance is estimated using a *MaxSim operator* between these two sets of vectors. Several other methods leverage multi-vector representations, including PreTTR (MacAvaney et al., 2020) and MORES (Gao et al., 2020). Recently, COIL (Gao et al., 2021) generates token-level document embeddings similar to ColBERT and performs *token interactions* by matching between query and document terms.

The architecture of *Semantic Legal Searcher* (SLS) is a new neural IR approach optimized for legal datasets as shown in Figure 1 (b). Unlike common methods Figure 1 (a), we extend our search model by introducing two information retrieval techniques. First, a *split-merge* technique is introduced to contain as much document information as possible in embeddings. In other words, we perform additional embedding modelization that splits each document into sentences and merges encoded sentence-level embeddings to minimize the loss of information in converting the whole document text into embedding. Secondly, a *multi-interactions* technique is introduced to improve the quality of semantic similarity measures. *SLS* is a search framework that combines semantic search and topic modeling to find relevant documents and simultaneously can extract keywords from each document. Therefore, it is possible to generate keyword embedding in *SLS*. The *multi-interactions* paradigm is that input queries, documents, and keywords are encoded into vectors and then relevance is measured not only by two sets of vectors from queries and documents but also by keyword embeddings.



(a) Common information retrieval approach (b) *SLS* approach

Figure 1: Contrasts existing approach with the proposed *Semantic Legal Searcher*

3 Semantic Legal Searcher

The process of the *SLS* is divided into four steps as shown in Figure 2. In the first step, each document in the legal database is encoded into embeddings and then fulfilled embedding modelization called *split-merge*. In the next step, these embeddings are parallelly clustered quickly, and then keywords are extracted by our topic modeling technique. In the third step, named *multi-interactions*, both the relevance of the query vector to the legal document embeddings and to the keyword embeddings are estimated by distance metrics. Lastly, the model provides user search results based on their relevance score.

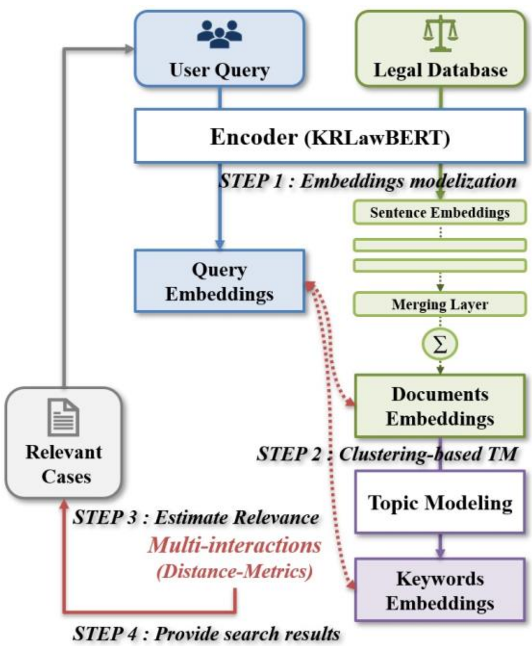


Figure 2: Semantic Legal Search Procedure

3.1 Clean Korean Legal Corpus

We created a Clean Korean Legal Corpus (CKLC), a new dataset of Korean legal texts. It is a pre-processed corpus consisting of 150 thousand cases of judicial decisions from the Supreme Court of Korea and statutes published from 1954 to the current year. The total number of sentences in CKLC is 5.3 million.

The dataset consists of five distinct sections for each law case: 1) case name; 2) case number; 3) judgment issue, 4) judgment summary; 5) full-text; 6) label. In detail, the judgment issue section contains the gist of the important legal issues of the cases and the judgment summary includes the main points of the full judgment text. The full-text section contains the official ruling of the court, the reasoning consisting of logical reasons and grounds for the conclusion, and related statutes. Lastly, the label section is labeled as to whether each case was dismissed or admitted.

3.2 KRLawBERT

We can use existing PLMs such as BERT in the *SLS* framework. However, this way is less competitive in the field of legal information retrieval. Therefore, we release a KRLawBERT pre-trained on CKLC (§3.1.) by benchmarking two popular techniques: Masked Language Modeling (MLM) and Transformer-based Sequential Denoising Auto-Encoder (TSDAE).

MLM. BERT (Devlin et al., 2018) is a Bi-directional Transformer for pre-training over a lot of text data to learn a word-level language representation. Its performance improvement could be attributed to an outstanding innovation named masked language modeling which allows bi-directional training in Transformer-based architecture. MLM is a fill-in-the-blank task, where a model uses the context words surrounding a mask token to try to predict what the masked word should be. BERT is pre-trained by a static masking modeling that executes a random selection of input tokens to train a deep bidirectional representation. Roberta (Liu et al., 2019) is an enhanced language model by retraining BERT with its inventive strategies. Roberta introduces a dynamic masking technique so that the masked token changes during the MLM training epochs.

TSDAE. Transformer-based sequential denoising auto-encoder (Wang et al., 2021) is recently another self-supervised learning technique. TSDAE is a task of reconstructing damaged sentences. Provided with input sequences damaged from deleting or swapping words, the model tries to generate the most likely substitution sentences. Specifically, TSDAE introduces noise to input sentences by removing about 55 – 60% of the tokens. These damaged sentences are encoded by the Transformer encoder into sentence vectors and then the decoder network attempts to predict the original input sentences from the damaged encoding vectors. This may seem similar to MLM, but they arguably differ in that while the decoder in MLM has access to full-length word embeddings for every single token, the TSDAE decoder only has access to the sentence vector produced by the encoder. Notice that each Transformer encoder in MLM and TSDAE produces token-level and sentence-level embeddings, respectively.

MLM and TSDAE are great ways to train a language model in self-supervised training without labels. In addition, both methods make the language model better understand the particular use of language (Korean) in a more specific domain (legal). Such a model can then be fine-tuned to accomplish several supervised NLP tasks.

Fine-tuning. To adapt the KRLawBERT to produce semantic legal embeddings, it needs a more supervised fine-tuning approach. We fine-tune KRLawBERT on the following three datasets: 1) Natural language inference (NLI) pairs; 2) semantic textual similarity (STS); 3) parallel legal data. Both NLI and STS datasets contain labeled sentence pairs. The parallel legal datasets consist of 1.2 million pairs of semantically similar legal sentences based on CKLC (§3.1.). The KRLawBERT learns how to distinguish between similar and dissimilar sentence pairs using the optimization functions like softmax loss or cosine similarity loss. Figure 3 shows the whole procedure of how to train KRLawBERT. Notice that since MLM-based KRLawBERT generates word-level embeddings, we need to add a pooling layer, however TSDAE-based KRLawBERT that can generate sentence-level embedding is fine-tuned directly on NLI, STS, and parallel legal datasets.

Any other embedding learning techniques can be used at this stage if the language model leads to generating semantically similar embeddings. Hence, the quality of searching in *SLS* will increase as improved legal language models are developed and legal datasets for fine-tuning are collected.

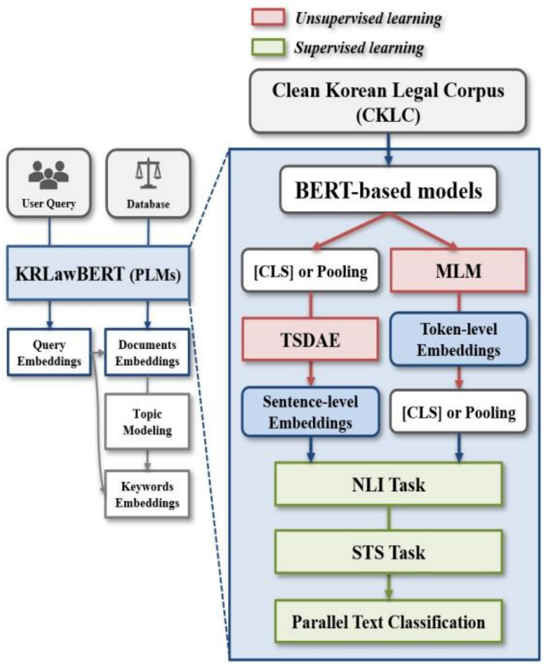


Figure 3: Language model Training Procedure

3.3 Embeddings modelization

Encoder. The next step in building the *SLS* framework is to encode the text into a dense vector. Transformers-based language models such as KRLawBERT (§3.2.) can produce a fixed-size embedding for each word in text data ($E_{num\ of\ text \times 512 \times 768}(\mathbb{R})$). The most common way to get sentence embedding is simply averaging these word vectors or using [CLS] special token that appears at the start of a sentence ($E_{num\ of\ text \times 768}(\mathbb{R})$). However, it turns out that the embedding generated by these methods is not rich in information.

Sentence-BERT (Reimers and Gurevych., 2019) which is a modified version of the BERT by adding a pooling layer allows us to build powerful sentence embeddings. Sentence-BERT can produce semantically meaningful embeddings ($E_{num\ of\ text \times 768}(\mathbb{R})$) of long-text sequences beyond the word-level through additional supervised fine-tuning tasks (§3.2.).

Split-Merge. Encoding the entire document with the encoder cannot contain all text into embeddings and lead to important information being lost. To avoid information loss, we need additional embedding modelization techniques which convey much information to embeddings. Inspired by a dynamic switching gate (Yang et al., 2019), we propose the *split-merge* to control the amount of information flowing from the PLMs as well as combine separated embeddings. This technique consists of split and merge parts. Following steps can summarize the function of *split-merge*:

1. **Split**: from input documents $D = \{d_1 \dots, d_n\}$, split each document into sentences $d_i = \{s_1 \dots, s_m\}$, BERT-based encoder computes a set of feature vectors $H_i = \{h_1 \dots, h_m\}$ where h is the hidden state of the encoder.
2. **Merge**: an embedding gate g looks at the input signals from sequential sentence-level embeddings H_i and outputs range from 0 (utterly important information) to 1 (utterly trivial information):

$$g = \sigma(Wh_j + Uh_{j+1} + b) \qquad (1)$$

where σ is a logistic sigmoid function.

Then, we reconstruct document-level embeddings $E_d = \{e_1 \dots, e_n\}$ by integrating all separated sentence-level embeddings H_i:

$$e_i = \sum_{j=1}^{m-1} g_i \odot h_j + (1 - g_i) \odot h_{j+1} \qquad (2)$$

where $\odot$ is an element-wise multiplication.

3.4 Clustering based Topic Modelling

Topic modeling is an unsupervised method to extract latent keywords and uncover latent themes within documents. Clustering-based topic modeling is an advanced technique using various clustering frameworks with embeddings for topic modeling. Adding topic modeling in the semantic search process has distinct advantages in interpretability and search quality. Firstly, representations of the search results are interpretable since literal topics in the latent vector space are discovered from each cluster and extracted keyword. Secondly, the PLMs can generate not only document embeddings but also keyword vector representation. Thus, *SLS* can increase search accuracy through the *multi-interactions* paradigm (§3.5.) which measures the relevance of not a single set of vectors from queries and documents but multi-sets of vectors by adding keywords embeddings. We create a parallel clustering-based topic modeling technique focused on speed, as shown in Figure 4.

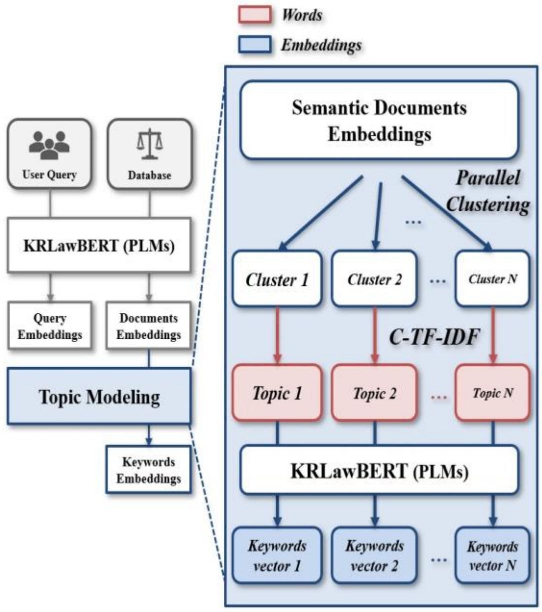


Figure 4: Topic Modeling with Parallel Clustering

Parallel Clustering. A parallel clustering algorithm is the main component of our topic modeling architecture. Primarily this algorithm attempts to parallelly assign all objects to their closest fixed *K* centroids and merge the clustered groups based on their nearest centroids. Here, the distance measures used can be Euclidean or cosine. The function of parallel clustering can sum up as follows:

1. **Initialize**: a random *K* is selected of *N* data points as the centroids.
2. **Assign & Filtering**: each data point should be parallelly associated with the closest centroids, and some data lower than the threshold *t* be filtered out.
3. **Merge**: each group centroid should be merged if the distance is higher than *t*, and the merged groups re-compute centroids of newly created groups.
4. **Stack**: the clustered *N* data are stacked in order of cluster size.

Steps 2 and 3 can be repeated multiple times until the cluster assignments stop changing.

As a result of the parallel clustering, legal documents are grouped into semantically similar embeddings and rearranged by cluster size.

Keyword extraction. In the next place, each cluster is regarded as a topic and then we select representative words from each cluster through the class-based TF-IDF formula introduced in BERTopic (Grootendorst., 2022). The class-based TF-IDF is a variation of TF-IDF (Joachims., 1996) and the formula is:

$$W_{t,c} = tf_{t,c} \times log\ (1 + \frac{A}{df_t}) \quad (3)$$

where each cluster is converted into a single document and *tf* is the frequency of words *t* in class *c* that refers to the cluster and *idf* is the one added to the average number of words per class *A* divided by the frequency of words *t* across all classes. Like with the TF-IDF formula, we can extract the local keywords by simply multiplying adjusted TF with IDF to get the importance score per word in each cluster. This formula allows us to interpret statistical distributions of important words for each cluster.

3.5 Measure the Relevance of Embeddings

Multi-Interactions. As distance metrics, normalized dot product and Euclidean are good measurements to quantify the similarity between two or more vectors. *SLS* computes the *multi-interactions* that both the relevance of the input query *Q* to the legal document *D* and to the keyword *K* are estimated by distance metrics. Let E_q, E_d, E_k (where *N* is the fixed length of the token sequence;) be the final vector sequences derived from *Q*, *D*, *K*. The *multi-interactions* scoring mechanism is given as follows:

$$Score_{q,d,k} = \sum_{i=1}^{N} E_{q_i} \cdot \{\omega E_{k_i} + (1-\omega) E_{d_i}\} \quad (4)$$

where $\cdot$ is a normalized dot product and ω is a scalar weight assigned. In addition, we benchmark two calculation approaches to extract top *k* relevant documents: 1) All distance metric; 2) Restricted distance metric.

All Distance Metric. The most naive way to retrieve relevant legal documents would be to measure the similarity between the input query (E_q) and all target vectors (E_d, E_k) and then find the top *k* document embeddings with a high similarity score. This method has high accuracy but is too slow to be applied to a large dataset.

Restricted Distance Metric. Another approach is dividing all target embeddings (E_d, E_k) into partitions. This method computes the distance between the centroid of each partition and the input query vector (E_q) and then restricts the search area to the partition containing the centroid nearest to the input query. Since this approach is based on a few regions of the vector space, it reduces the search scope of *SLS* and speeds it up by effectively calculating the similarity scores.

SLS can be performed slowly with high accuracy or quickly with low accuracy depending on the number of partitions *p*. *SLS* allows the users to choose one of the two computational strategies above and flexibly sets the parameter *p* by finding the best balance between the accuracy and search speed.

4 Experimental Setup

All codes related to the *SLS*, are run on a machine with 2 cores Intel(R) Xeon(R) CPU @ 2.30 GHz and Tesla T4 GPU.

4.1 Models

Several pre-trained Transformer models for language tasks have been proposed, inspired by the BERT architecture, and redesigned to handle multilingual inputs. In this paper, to produce semantic legal embeddings, we designed a language model named KRLawBERT (§3.2.) based on unsupervised learning (MLM, TSDAE) and supervised fine-tuning (NLI, STS, parallel legal data). Moreover, we follow a baseline model as KoBERT (SKTBrain et al., 2020), which is pre-trained on a large-scale Korean text corpus.

4.2 Evaluation

We conducted three different NLP downstream tasks for evaluating performance of KRLawBERT in *SLS* framework: 1) Korean Natural Language Inference; 2) Korean Semantic Textual Similarity; 3) Legal Question Answering.

NLI & STS. KorNLI and KorSTS are NLI and STS datasets in Korean (Ham et al., 2020). In the KorNLI task, the BERT-based models receive a pair of Korean sentences and classifies their relationship into one out of three categories: entailment, contradiction, and neutral. The KorSTS is a task that assesses the gradations of semantic similarity between two Korean sentences. The similarity score ranges from 0 (completely dissimilar) to 5 (completely equivalent). This task is commonly used to evaluate either how well the language model grasps the semantic closeness of two sentences or how well it generates the semantic representation of the sentence.

Legal Question Answering. We report three metrics for legal question-answering: namely Precision-k, Recall-k, and Hit-k. These metrics as part of human validations can evaluate whether the top *k* search results really include law cases and are satisfied with ordinary people.

Precision-k is concerned about how many search results are relevant among the provided results:

$$\boldsymbol{P} = \frac{\#\ of\ model's\ search\ results\ that\ are\ relevant}{\#\ of\ Law\ cases\ recommended\ by\ the\ model}$$

Recall-k focuses on measuring how many search results are provided among all values:

$$\boldsymbol{R} = \frac{\#\ of\ \ model's\ search\ results\ that\ are\ relevant}{\#\ of\ All\ the\ possible\ relevant\ Law\ cases}$$

Hit-k is meant for a percentage of users who are satisfied with the search results among the total users:

$$\boldsymbol{Hit} = \frac{\#\ of\ Hit\ Users}{\#\ of\ Users}$$

For the statistical comparison experiment, five questions that consist of two or three words and questions of five natural languages were randomly chosen from an online legal question table. Subsequently, ranging from 1 to 10 question queries, the above three metrics scores were calculated at each step for each model.

4.3 Results

In Table 1 upper side, we show the performance of the language models on the *SLS* process. All of the language models we created showed better performance than baseline. The TSDAE-based KRLawBERT achieved the highest score in NLI and STS tasks. That indicates the TSDAE-based model encodes semantically meaningful information better than others. In particular, evaluation results show that our model performs fairly well in legal question-answering tasks. Compared to the baseline, the metric scores of KRLawBERT are dramatically up by 30 – 40% points. In Table 1 lower side, we also find that both the *split-merge* and the *multi-interactions* mechanisms help improve semantic search accuracy by 14 – 20%. It demonstrates that they are suitable approach in neural information retrieval (IR) without KRLawBERT. Therefore, we expect *SLS* to show potential for expansion with powerful neural IR tools and could consider a performance comparison to recent neural IR methods as future work.

Models	P-10	R-10	Hit-10	NLI	STS
Baseline Retrievers					
KoBERT	0.50	0.48	0.60	0.69	0.78
KRLawBERT Retrievers (Ours)					
BERT-MLM	0.60	0.55	0.65	0.79	0.85
RoBERTa-MLM	0.65	0.65	0.75	0.79	0.85
TSDAE	**0.70**	**0.65**	**0.78**	**0.79**	**0.86**
With Information Retrieval Tech (Ours)					
Single-inter (q, d)	0.70	0.65	0.78	-	-
Multi-inter (q, d, k)	**0.75**	**0.68**	**0.80**	-	-
split-merge	**0.80**	**0.75**	**0.80**	-	-
Multi-inter + split- merge	**0.85**	**0.80**	**0.85**	-	-

Table 1: Information Retrieval Evaluation

5 Conclusion

In this paper, we propose the *Semantic Legal Searcher* (SLS), a highly effective semantic case law search model. By leveraging the KRLawBERT (§3.2.) that a language model pre-trained on a large-scale Korean legal corpus and the *split-merge* embedding modelization technique (§3.3.), we can generate high-quality semantic embeddings. In addition, our *SLS* architecture improves the information retrieval performance through parallel clustering-based topic modeling (§3.4.) and the *multi-interactions* (§3.5.).

The *SLS* framework is not limited to the Korean language and the fields of Law. Since this framework is a vector-based architecture with various embedding techniques consisting of semantic search and topic modeling, it can be extended to multi-lingual datasets and other domain sectors. Furthermore, by separating the process of embedding modelization, parallel clustering-based topic modeling, and semantic search, flexibility can be given in the model allowing for ease of usability.

Our experiment (§4.2., §4.3.) demonstrates that the *SLS* has good enough performance across legal questions-answering. We conclude that our semantic search model can effectively retrieve the relevant case law and provide users with meaningful results in real-life applications.

Number	Random Queries
Q1	"Drunk driving fines" "음주운전 벌금"
Q2	"Landlord-Tenant Dispute" "임대인-세입자 분쟁"
Q3	"Sexual Assault" "성폭력"
Q4	"Criminal Livelihood" "생계형 범죄"
Q5	"juvenile delinquency" "소년 범죄"
Q6	"My car collided with a vehicle in the next lane while trying to avoid another vehicle changing from lane 1 to 2." "1차선에서 2차선으로 바꾸는 차량을 피하려다가 옆 차선 차와 충돌하였습니다."
Q7	"The tenant does not pay rent to me, the landlord, for 3 months." "세입자가 3개월째 집주인인 저에게 월세를 주지 않습니다."
Q8	"I have been mentally harmed by an illegally installed camera in the bathroom." "화장실에 카메라를 설치하여 정신적 피해를 보았습니다."
Q9	"I couldn't make a meal for three days in a row, so I got starved and stole bread from the bakery." "3일째 끼니를 해결하지 못해 배고픈 나머지 빵집에서 빵을 훔쳤습니다."
Q10	"Juveniles who had known that they weren't entitled to criminal punishment deliberately committed violent crimes." "형사처분을 받지 않는다는 걸 알고 촉법소년들이 고의로 폭력 범죄를 저질렀습니다."

Table 2: Random Input Queries Examples

6 Limitations

We need to discuss the limitations of *Semantic Legal Searcher* in three areas: 1) Language models; 2) Clustering issue; 3) Objectivity in evaluation.

Language Models. we create pre-trained language models to utilize in *SLS* architecture. KRLawBERT takes both unsupervised and supervised learning strategies to offer powerful legal-based embeddings for semantic search (§3.2., §3.3.). As a result, although KRLawBERT improves linguistics task performance in the legal field, do not benefit from linguistics information that leads to more general representations to help adapt to new tasks and domains. In addition, this model is not a multi-lingual model. Since KRLawBERT pre-trained in Korean languages with a large scaled legal corpus, it cannot make a difference between Korean and other languages. However, *SLS* is the architecture composed of vector-based models (§3., §5.). Therefore, language models pre-trained on various domains and languages can be flexibly applied in *SLS*. We conducted the experiments of *SLS* on the arXiv papers English dataset (Cornell University., 2022), [1] and the results of experiments show the *SLS*'s successful search performance even in the English environment. The downside of KRLawBERT paradoxically demonstrates the elasticity of *SLS*.

Clustering Issue. Parallel Clustering performance is critical to topic modeling and generating keyword embeddings for semantic search (§3.4.). Unfortunately, parallel clustering is not a perfect algorithm and has two drawbacks. One of the weak points of parallel clustering is that results will differ based because of random centroid *K* initialization. This means that users can run parallel clustering on the same document dataset multiple times and get different clustered results. This issue causes inconsistency problems in topic modeling on small datasets. Second, picking the optimal value of parameters such as centroids *K*, threshold *t*, and max iteration is a challenging model selection problem. Parallel clustering might involve some manual labor for adjusting those significant parameters. Nevertheless, parallel clustering shows

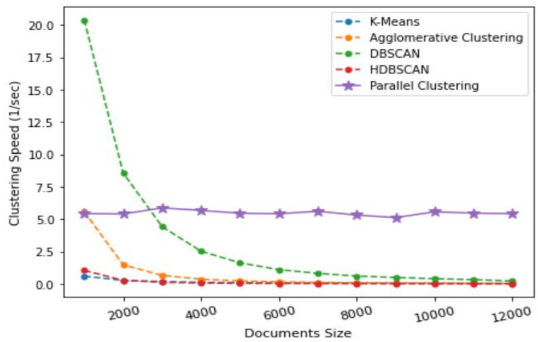


Figure 5: Clustering Speed Comparison Chart

[1] https://anonymous.4open.science/r/Semantic-Searcher-F231/(Colab)2_SLS_on_Eng.ipynb

strengths in large-scale text classification and leads to fast information retrieval. Figure 5 shows the clustering performance comparison on the MovieLens text dataset (Harper and Konstan., 2016). Experimental results demonstrate that our parallel clustering is faster and more coherent in document clustering than other famous clustering methods.

Objectivity in Evaluation. The legal question-answering metrics for information retrieval evaluation (§4.2.) are substitutes for what is fundamentally a subjective evaluation. One user might judge the relevance of a case law search results differently from another user. Accordingly, even if this measure can be used to get an indication of a search model's performance, they are just that, an indication. To solve this limitation, we attempt to create a lawyer-validated legal question table and score the model's answers by attorneys. This table contains frequently asked legal case queries online. Table 2 shows some of the question queries.

References

Jurafsky, Daniel; H. James, Martin. 2000. Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. In *Upper Saddle River, N.J.: Prentice Hall. ISBN 978-0-13-095069-7.*

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. In *CoRR, abs/1810.04805.*

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese best networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics.*

Florian Schroff, Dmitry Kalenichenko, James Philbin. 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. *arXiv preprint arXiv:1503.03832.*

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692.*

Kexin Wang, Kils Reimers, Iryna Gurevych. 2021. TSDAE: Using Transformer-based Sequential Denoising Auto-Encoder for Unsupervised Sentence Embedding Learning. *arXiv preprint arXiv:2104.06979v4.*

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo LopezGazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2017).

Howard Jackson, Etienne Zé Amvela, *Words, Meaning, and Vocabulary*, Continuum, 2000, p14. ISBN 0-8264-6096-8

Pamela B. Faber, Ricardo Mairal Usón, *Constructing a Lexicon of English Verbs*, Walter de Gruyter, 1999, p67. ISBN 3-11-016416-7

Fillmore, Charles J., and Collin F. Baker. "Frame semantics for text understanding." Proceedings of WordNet and Other Lexical Resources Workshop, NAACL. 2001.

Omar Khattab, Matei Zaharia. 2020. ColBERTv2: Effective and Efficient Retrieval via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020,* pages 39–48. ACM

Sean MacAvaney, Franco Maria Nardimi, Raffaele Perego, Nicola Tonellotto, Nazi Goharian, and Ophir Frieder. 2020. Efficient Document Re-Ranking for Transformers by Precomputing Term Representations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 49–58.*

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2020. Modularized Transfomer-based Ranking Framework. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4180–4190.*

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. COIL:Revisit Exact Lexical Match in Information

Retrieval with Contextualized Inverted List. *arXiv preprint arXiv:2104.07186.*

Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Yong Yu, Weinan Zhang, Lei Li. 2019. Towards Making the Most of BERT in Neural Machine Translation. *arXiv preprint arXiv:1908.05672*.

Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794.*

Thorsten Joachims. 1996. A probabilistic analysis of the Rocchio algorithm with TF-IDF for text categorization. Technical report, Carnegie-Mellon univ Pittsburgh pa dept of computer science.

SKTBrain. 2020. Korean BERT pre-trained cased (KoBERT). *https://github.com/SKTBrain/KoBERT*.

Jiyeon Ham, Yo Joong Choe, Kyubyong Park, IIji Choi, Hyungjoon Soh. 2020. KorNLI and KorSTS: New Benchmark Datasets for Korean Natural Language Understanding. *arXiv preprint arXiv:2004.03289.*

F Harper and Joseph Konstan. 2016. The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems.TiiS*.