Legal Information Retrieval Using Traditional Approaches and Binary Relevance Classification Approach on Translated Text

Venkatesh Murugadas
Faculty of Computer Science
Otto von Guericke University
Magdeburg, Germany
venkatesh.murugadas@st.ovgu.de

Tousifur Rahman Khan
Faculty of Computer Science
Otto von Guericke University
Magdeburg, Germany
tousifur.khan@ovgu.st.de

***Abstract*—The Competition on Legal Information Extraction and Entailment (COLIEE) is an international competition that focuses on improving Legal Information Retrieval and Entailment on Japanese case law and statute law. This project concentrates on Task 3 of COLIEE which is Information Retrieval on Japanese statute laws. Classical approaches such as TF-IDF, BM25 with various indexing methods, and modern approaches of using static word embeddings on Word Mover's Distance and contextual word embeddings from a DistilBERT base model are used to implement a Binary Relevance Classification model.Surprisingly classical approaches outperform the modern approaches with a larger margin. Despite the incorporation of semantic information encoded in modern approaches, they are not able to perform better. Some factors on poor performance along with the future direction for such models are also discussed in this project.**

***Index Terms*—legal information retrieval, TF-IDF, BM25, WMD, DistilBERT, binary relevance classification.**

I. Introduction

The recent advancements in computer sciences and computational power, has increased the ability to analyze large text corpora drastically allowing computational linguists and other professionals to perform text analysis on a large scale. Natural Language Processing(NLP) consists of various applications and one such widely used application is Information Retrieval. Information Retrieval (IR) refers to the extraction of relevant documents for a specific natural language query given by the user. This ability to retrieve relevant information whenever needed by a user efficiently transformed our ability to access information in this modern age.

One of the industries that use information retrieval for due diligence quite extensively is the legal industry where for a given specific query or scenario relevant information from the corpus of statute laws or case laws must be extracted. To solve this problem, law practitioners, in general, employ simple rule-based IR systems to advanced semantic search-based IR models to retrieve the relevant information among corpora of extensive statute laws and case laws. An effective IR system in the hands of a legal professional saves time and money to extract the relevant information in less time increasing productivity and workflow.

A common problem in the field of IR and the legal industry is retrieving the relevant information from corpora given a natural language query. A query in natural language might not contain exact words present in relevant article but semantically similar words, query might be just a few words or sentence which makes it difficult to understand the information need of the user and retrieve the relevant document. This is one of the problems that is tackled by an international competition named "Competition on Legal Information Extraction and Entailment" (COLIEE). COLIEE concentrates on improving the Information retrieval and entailment tasks on Japanese case laws and statute laws which would improve the performance of due diligence in the legal industry.[1].

This project focuses on few experiments to improve the legal information retrieval task on Japanese statute laws and legal bar exam questions which is Task 3 in COLIEE. Task 3 is defined as the retrieval of relevant statute laws/articles (S1, S2,..., SN) given a natural language query Q which is a Japanese legal bar exam question. These legal texts tend to be more precise detailed descriptions and definitions with complex structures. Legal language structures are difficult to comprehend by someone who has no prior knowledge about the legal domain.

Thus, the objective of this project is to tackle the following research questions.

1) How do classical information retrieval methods perform with various feature sets for indexing translated Japanese legal texts for retrieval?
2) How do static legal domain-specific and general-purpose word embeddings perform on legal information retrieval?
3) Does approaching an information retrieval as a Binary Relevance classification problem using contextual word embeddings improve the performance of the legal information retrieval?

These are the research questions based on which the experiments are set up in this project.

[1]https://sites.ualberta.ca/ rabelo/COLIEE2020/

Section 2 discusses various relevant works regarding legal information retrieval methods. Section 3 discusses the background technical information required for the information retrieval models implemented in the experiments. Section 4 provides information about the dataset used in this project. Section 5 describes the methodology of various experiments implemented in this project to tackle the above-mentioned research questions. Section 6 discusses the technical details of the experiment implementations in this project. Section 7 provides the experiment results and the metrics used for evaluating the information retrieval models. Section 8 discusses the interpretations of the experiment results, limitations and future work. Finally, section 9 is about the conclusion.

II. Related Work

Information Retrieval as a task has been around for a long time which allowed researchers to experiment with various techniques. The survey paper by Dong et al. (2008) discusses the traditional approaches used in IR which serves as a basis for some of the classical approach-based experiments. IR models such as TF-IDF and BM25 in combination with Word Embeddings were used by Sugathadasa et al. (2018) for implementation of a legal document retrieval system. The word embeddings employed in these experiments are legal domain-specific embeddings. In addition to the traditional vector space models, query expansion is also one of the commonly used techniques for representing the queries with more information. Maxwell and Schafer (2010) studies the challenges and responses for using query expansion in legal information retrieval.

In recent times due to the advancement in Transformer-based neural network architectures and Contextual Word Embeddings, several methods were employed in the field of IR for semantic search. Transformer based models were used for relevance ranking and information retrieval by Pang et al. (2017), Yang et al. (2019), Dai and Callan (2019), Reimers and Gurevych (2019), MacAvaney et al. (2019). Contextual embeddings of BERT are used along with traditional approaches such as TF-IDF or BM25 for retrieval and re-ranking as implemented by Althammer et al. (2021).

The approach for experimenting with an IR task as a binary relevance classification system was inspired by the experiments implemented by Yang et al. (2019) where BERT was used for ad-hoc document retrieval on microblogs and Shao et al. (2020) implemented binary relevance classification model using BERT based model on Japanese law in COLIEE task 3. These above mentioned works serve as a basis for the experiments implemented in this project.

III. Background

The IR models used in this project can be categorized into classical approach-based and modern approach-based models. TF-IDF and Okapi BM25 are classical approach-based models whereas Word Mover's Distance (WMD) and DistilBERT are modern approach-based models. The theoretical concepts of these above-mentioned models are described in brief in this section.

A. TF-IDF Model

The TF-IDF (Shahmirzadi et al. (2019)) is an acronym for Term Frequency - Inverse Document Frequency. Term frequency (TF) refers to the number of times a particular word/term occurs in a particular document.The Inverse Document Frequency (IDF) calculates a weight for each token which indicates the significance of a term with respect to a document. TF and IDF are combined to find the significance of a word in a document by assigning a weight to each word present in a document. The weight is assigned based on the equation 1.

$$TFIDF_{i,j} = TF_{i,j} * log(\frac{N}{df_i}) \tag{1}$$

where *i* is the term , *j* is the document and *df* refers to the document frequency of a particular term.

For a multi-term query *Q* we used the equation 2 to obtain the score of the relevance between the query *Q* and the document *D*.

$$Score(Q, D) = \sum_{i \in Q} TFIDF_{i,D} \tag{2}$$

The higher the score the more relevant the document is to the query.

B. BM25 Model

The BM25 is an improved version of the TF-IDF model used for ranking a set of documents given a query *Q*. TF-IDF considers documents of various lengths as equal which is one of the disadvantages. So to overcome this issue, Okapi-BM25 (Robertson and Zaragoza (2009)) was introduced. The Okapi-BM25 algorithm controls the term frequency saturation and considers the document length.

Given query Q containing terms $q_1, ...q_n$, the BM25 score of a document D is calculated using the equation 3.

$$score(D, Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{avgdl})} \tag{3}$$

where IDF(q_i) is the Inverse Document Frequency of the term q_i, $f(q_i, D)$ is the term frequency in document *D* , $\mid D \mid$ is the length of the document *D* and $avgdl$ is the average document length of the corpus from which the documents are drawn. There are two hyper parameters named "k_1" and "b" used in the BM25 ranking function. The hyper parameter "k_1" is used to determine the term frequency saturation that limits the impact of a single query term on scoring for a given article. The hyper parameter "b" is used to impact the effect of length of an article compared to the average length of the article collection.

C. Static Word Embeddings

The representations of a word in N dimensions are calculated using distributional hypothesis Harris (1954) assuming that the context of a word can be modeled by knowing the neighboring words present in a text. This was first applied to language modeling by Bengio et al. (2003). Using this technique Mikolov et al. (2013) efficiently developed Word2Vec algorithm for the representation of words in N dimensions. This method represents the words in N-dimensional space which models the meaning of the word concerning the neighboring words using a context window to learn the co-occurrences of the word. Word2Vec has two methods to calculate the word embeddings which are the Continuous Bag of Words (CBOW) and Skip-gram method. The word embeddings are calculated using a shallow neural network model with huge corpora as an input to the network where the embeddings are calculated for each word present in the vocabulary. These word embeddings are pre-trained which can be directly used as an input in any NLP downstream tasks, in this case, Information Retrieval. In Word2Vec, for a word in the vocabulary it consists of only one embedding/vector which is a combined representation of all different senses of the word. For instance, in the sentence "I saw him duck near a duck pond" contains a word "duck" with two different senses. The first "duck" represents an action of dodging or bending whereas the other "duck" is a noun indicating the bird Duck. These two different word senses will not be captured in the Word2Vec algorithm.Thus it is considered as static or non-contextual word embedding.

In this project, two types of static word embeddings are used. A general-purpose word embedding named GloVe (Pennington et al. (2014)) and a domain-specific word embedding named Law2Vec (Chalkidis and Kampas (2019)). GloVe embeddings were created using general-purpose texts such as Wikipedia texts whereas Law2Vec was trained on English legal texts such as European legislation, English-translated Japanese legal texts.

D. Word Mover's Distance (WMD)

The Earth Mover's Distance (EMD) metric (Rubner et al. (1998)) is used to calculate the distance between two distributions in a given region. The metric WMD is derived from EMD for calculating the distance between two documents or texts which is represented in a vector space based on a word embedding (Huang et al. (2016)). WMD calculates the travel cost between two documents. This cost is usually determined by the Euclidean distance in the embedding space which incorporates semantic similarity between words. Then the cumulative sum of all the word pairs of two documents is taken as document distance. Documents of different lengths can be used to measure the distance between two documents. The only constraint for calculating WMD between two documents is that at least one word must have a word embedding in each document. In this project, static word embeddings such as GloVe and Law2Vec were used as inputs for WMD calculation.

E. DistilBERT

Transformer architecture proposed by Vaswani et al. (2017) consists of encoder and decoder blocks which model the context with the help of the self-attention mechanism. The transformer architecture provides different embeddings for words with different senses as they also consider the position of a word in a sequence. This way the embeddings generated are context dependent. A variant of the Transformer architecture named Bidirectional Encoder Representation from Transformers (BERT) proposed by Devlin et al. (2018) encodes the input text both forward and backward to capture the context of the words using the self-attention mechanism in the transformer blocks only with an encoder which accepts Token embeddings, segment embeddings, and position embeddings as input. BERT creates contextual word embeddings as output for each token and also a word embedding for the whole input text. The BERT model is trained on huge corpora as language models which encode the contextual information. These pre-trained large language models can be then used in further downstream tasks such as Next Sentence Prediction, Text classification, Token-level classification, Machine Translation by training the model on a specific task. New data can be then used to fine-tune the pre-trained language models for a specific task.

BERT architecture consisting of encoder stacks has two models which are BERT-BASE and BERT LARGE. BERT-BASE model has 12 layers of encoder stack with 12 attention heads whereas the BERT-LARGE model has 24 layers of encoder stack with 16 attention heads. These are higher than the original transformer architecture which was proposed by Vaswani et al. (2017). Due to the architecture they also have larger number of parameters. BERT-BASE model has 110 million parameters and BERT-LARGE model has 340 million parameters.

Due to the large number of parameters present in these language models it takes a considerable amount of time for training and inference. To adapt to a scenario with less computational power and time, an architecture that performs at par with the BERT-BASE architecture but with less amount of parameters was proposed by Sanh et al. (2019) called the DistilBERT. It is the "cheap, light, fast and smaller version" of the BERT base model where the knowledge distillation is performed at the pre-training phase to reduce the size by 40% when compared to the BERT base model and retains almost 97% of the language modelling capabilities while being 60% faster than the BERT base model as mentioned by Sanh et al. (2019). DistilBERT base model has 66 million parameters. To enhance the inductive biases the authors had introduced a triple loss that combined language modelling, distillation, and cosine-distance losses.

The above mentioned concepts are discussed with respect to their application in the legal information retrieval in the methodology section.

IV. Dataset

A. Dataset Introduction

The data used in this project is obtained from the Competition on Legal Information Extraction and Entailment (COLIEE). The dataset used for the statue law retrieval task (Task 3) in COLIEE consists of 715 English-translated Japanese Civil Code Articles (S1, S2, S3...) and a "Yes/No" legal bar exam question (Q). The datasets used in this project are obtained from Task 3 of COLIEE 2019 and COLIEE 2020.

The dataset is constructed from a query which is a legal bar exam question and a subset of Japanese civil code articles that are relevant to the query. The dataset was provided as an XML file which consists of $\langle t1 \rangle$ and $\langle t2 \rangle$ tags which contain the relevant English translated Japanese civil code articles and legal bar exam questions respectively. The $\langle t1 \rangle$ would also consist of an id attribute that is unique to each article and query pair. This pair can be considered as the ground truth or a training dataset. The test dataset on the other hand only consists of $\langle t2 \rangle$ the legal bar exam questions which are used by the retrieval models to find the relevant articles and evaluate the retrieval models.

In this project, the datasets from COLIEE 2019 and 2020 were used for training the DistilBERT models. COLIEE 2020 Task 3 test dataset was used for the evaluation of all the models in classical and modern approaches.

B. Dataset Analysis

To understand the properties of the corpus a manual and statistical analysis was carried out on the provided English translated corpus of Japanese statute laws.

The manual analysis was done on the ground truth corpus which contains the query and the relevant statute laws to understand the features that relate to both the queries and statute laws. Among the pairs of queries and statute laws, a random sample of 25 pairs was selected for manual analysis. This leads to an understanding of the legal language which is complex for an outsider without legal domain knowledge. It also made us notice that there were noticeable patterns of bigrams and trigrams in the pairs. There were few pairs in the analysis which had no similar words between the queries and statute laws which made it harder to determine the relevance.

A statistical analysis was conducted to find the average number of relevant articles per query, identifying the stop-words in the corpus, identifying the relevance of similar words between the query and relevant statute laws with and without query expansion, identifying the most common bigrams and trigrams.

From the ground truth corpus, the number of relevant statute laws per query was obtained to determine the number of statute laws to be retrieved for each query. The average number of relevant statute laws per query is 1.92 which indicates most of the queries have 1 or 2 relevant statute laws. 94% of the queries have two or less than two relevant statute laws and 6% have three or more relevant statute laws which can be seen from Figure 1.

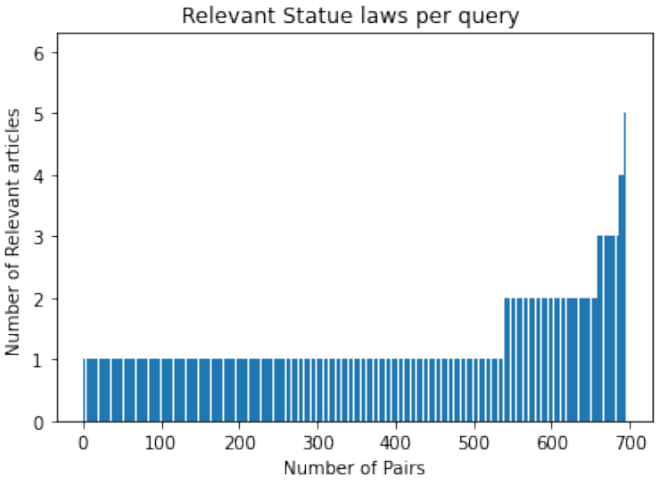

Fig. 1: Relevant Statute laws per query

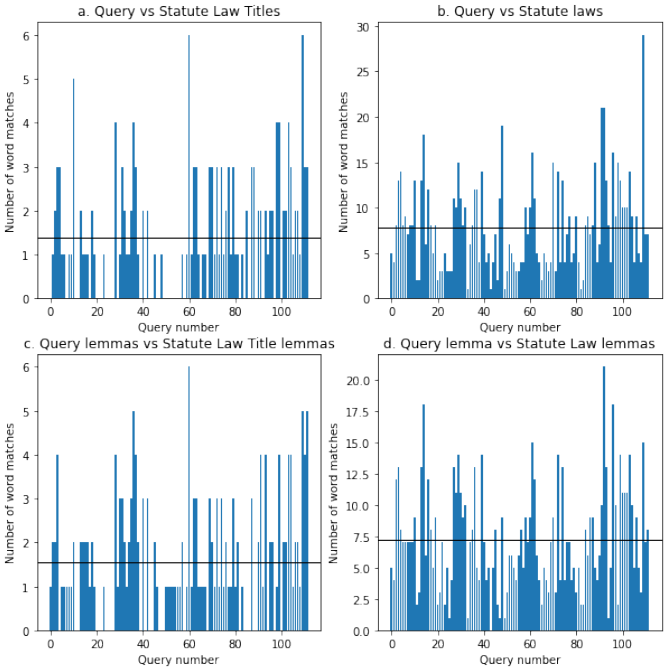

Fig. 2: Similar word matches between Queries and statute laws

Figure 2 indicates the word match between the queries and the statute laws with query expansion using the WordNet Miller et al. (1990) [2]. Graphs a and b represent the word matches of tokens without lemmatization and graphs c and d represent the word matches of tokens with lemmatization. From figure 2 it can be seen that there were 8 matches on average between the queries and statute laws and 2 matches between queries and statute law titles in both the cases with and without lemmatization.This helped us to understand that simple statistical methods with token based and lemma based indexing might capture the relevant articles given a query.

[2]https://www.nltk.org/howto/wordnet.html

An ngram analysis was also performed to find the most commonly occurring bigrams and trigrams in the queries and statute laws. Figure 3 and Figure 4 visually represent the top bigrams and trigrams present in statute laws.

The Exploratory Data Analysis helped to understand the nature of the dataset and finalize the steps necessary for data pre-processing.

V. Methodology

The legal information retrieval on statute laws was set up based on two approaches. They are classified into classical and modern approach. The classical approach uses traditional IR models such as TF-IDF and OkapiBM25. The modern approach employs Word Mover's Distance and DistilBERT model for retreival. The classical and modern approaches are discussed in detail in the following sections.

A. Classical Approach

The classical approach is a blend of the traditional information retrieval process along with various feature sets for indexing the articles and queries. The traditional approaches use term frequency and inverse document frequencies as a weighting mechanism to represent a document in a Bag Of Words (BOW) fashion which contains statistical information of the document. These models with just statistical information might be useful in capturing the patterns occurring in a complex and formal language such as legal texts.The methods involved in setting up the models for TF-IDF and OkapiBM25 are discussed in the following sections.

1) Classical Approach Dataset Preparation: The articles and the queries consist of tokens that do not carry much information for the retrieval task such as the stopwords and punctuations. The stopwords list for the corpus of articles and queries were a combination of words that were obtained from the analysis and the NLTK package Loper and Bird (2002). Legal specfic stopwords were also chosen from LexNLP (Bommarito II et al. (2021)). These stopwords and punctuations were removed from the articles and queries after the tokenization process. These tokens are then converted into their respective lemmas using a lemmatization algorithm from the SpaCy package (Honnibal and Montani (2017)) for lemma-based features. The preprocessed corpus is then indexed for the information retrieval experiments.

2) Feature set:
The classical approach is experimented by introducing various methods for indexing articles and queries. This is done by choosing a variety of feature sets for experimentation. This section explains the different features which were used in the classical approach for indexing. Example for each feature set is provided in the appendix section for further clarification.

a) Token-based features: The articles and queries are pre-processed by just removing the stop words . These pre-processed articles and queries are then indexed only with the tokens which are used in the similarity or distance calculation functions to retrieve the relevant articles.

b) Lemma-based features: The articles and queries are pre-processed by removing the stopwords and the tokens are converted into their respective lemmas using the lemmatization process. Lemmatization is a procedure where each token is converted to its root form by removing the inflections present in them (Kanerva et al. (2020)). For instance, "running" when lemmatized is converted to its root form "run". This method allows us to decrease the number of words/tokens in the vocabulary and increase the chance of occurrence of a word in the query and article. The converted tokens are indexed which are then used in the similarity or distance function to retrieve the relevant articles.

c) Ngram-based features: Token level ngrams are used in the place of a token or a lemma to index the articles and queries. A token level ngram considers n tokens at a given time for the analysis. This is helpful to represent the phrases available in the corpus. In this project, Bi and Trigram features are considered for encoding the articles and queries. The articles and queries are subjected to a minimal preprocess of removing the stopwords, then the bigrams and trigrams are extracted. The bigram and trigram features have been experimented with separately as two different methods. These ngrams are indexed and then used in the similarity or distance function to retrieve the relevant articles.

d) Query Expansion based features: Query expansion (Maxwell and Schafer (2010)) is a famously used method to expand the terms present in the queries by finding their synonyms from a lexical resource such as WordNet (Miller et al. (1990)) to increase the number of words/terms present in the queries. For each word in the query, its synonym is obtained from the lexical resource which is then appended to the pre-processed query tokens. This increases the chances of matching the query with its relevant article. During the experimentation of this feature, only the query is modified whereas the article remains the same.

e) Combined feature set: The final feature set is a combination of the lemma, Ngram, and query expansion features. The articles are pre-processed by removing the stop words and then represented by combining lemmas, bigrams, and trigrams. A similar approach is taken in the queries by removing the stopwords, converted into lemma and expanding the queries using the query expansion method explained above. Lemma, Bigram and Trigram are obtained from the original query which are then combined with the expanded query and then searched using a similarity or distance function to retrieve the relevant articles.

3) TF-IDF experiments: The Term Frequency - Inverse Document Frequency (TF-IDF) based vectorization method is used to encode the statute laws and the queries to find the similarities between them. The similarities are calculated by a cosine angle between the article vector and a query vector using a cosine similarity function. The similarity function is applied between a query and every article to calculate the similarity where the top N similar articles are retrieved as relevant articles for a given query as shown in Figure 5. The TF-IDF-based approach is used to model the retrieval

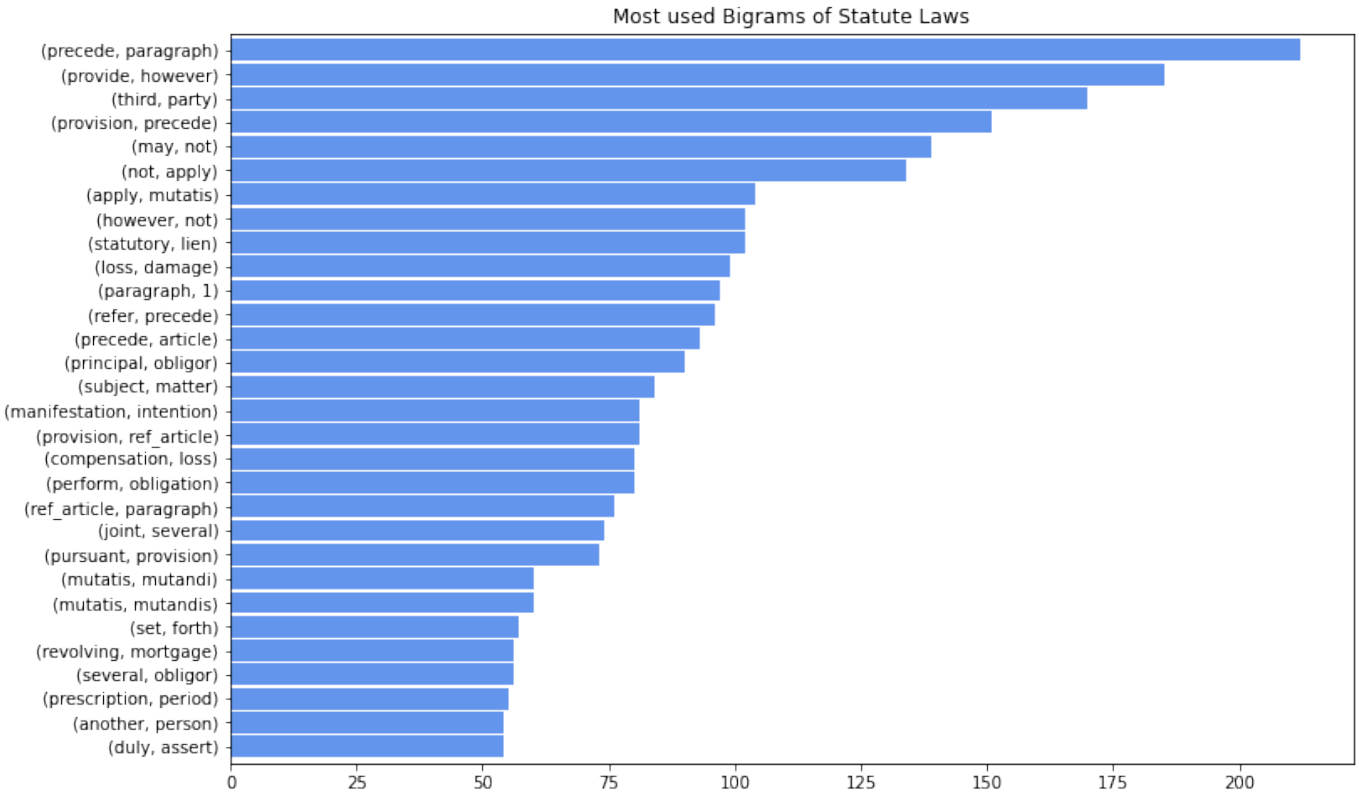

Fig. 3: Most used bigrams in statute laws

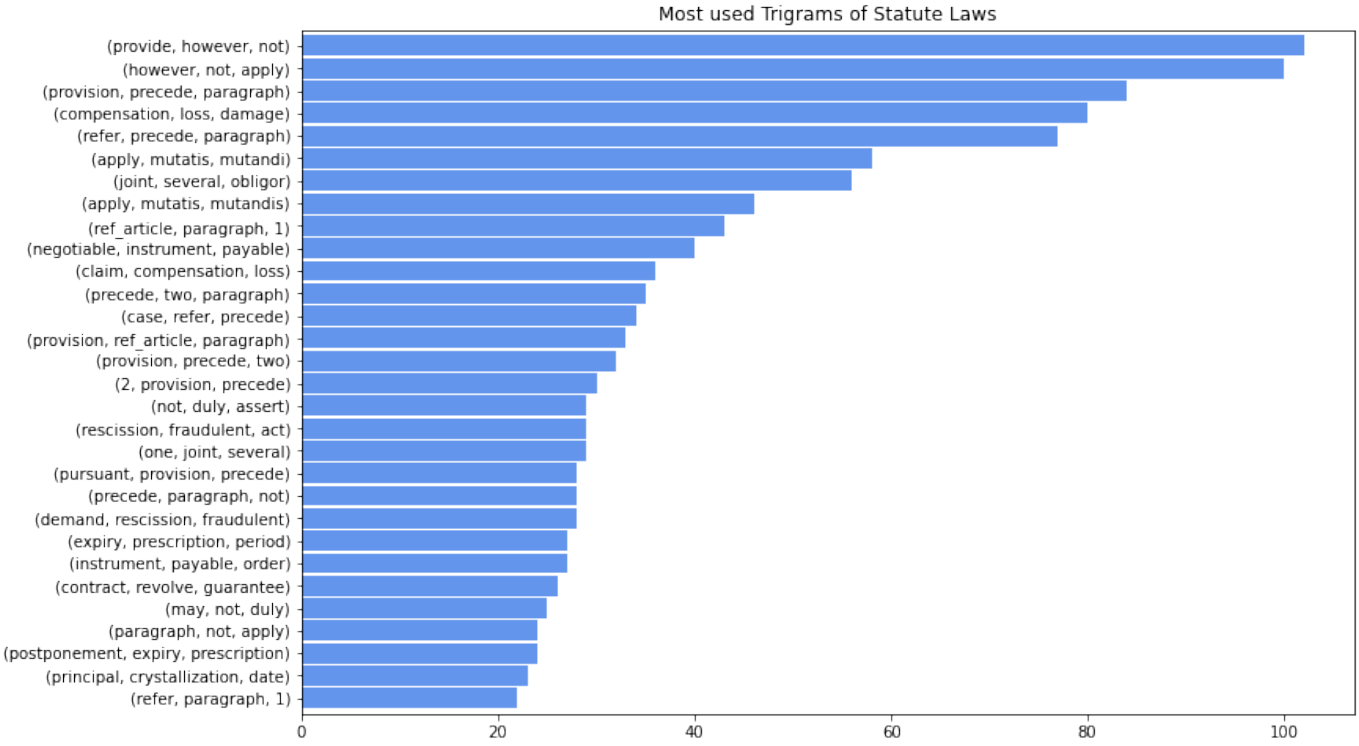

Fig. 4: Most used trigrams in statute laws

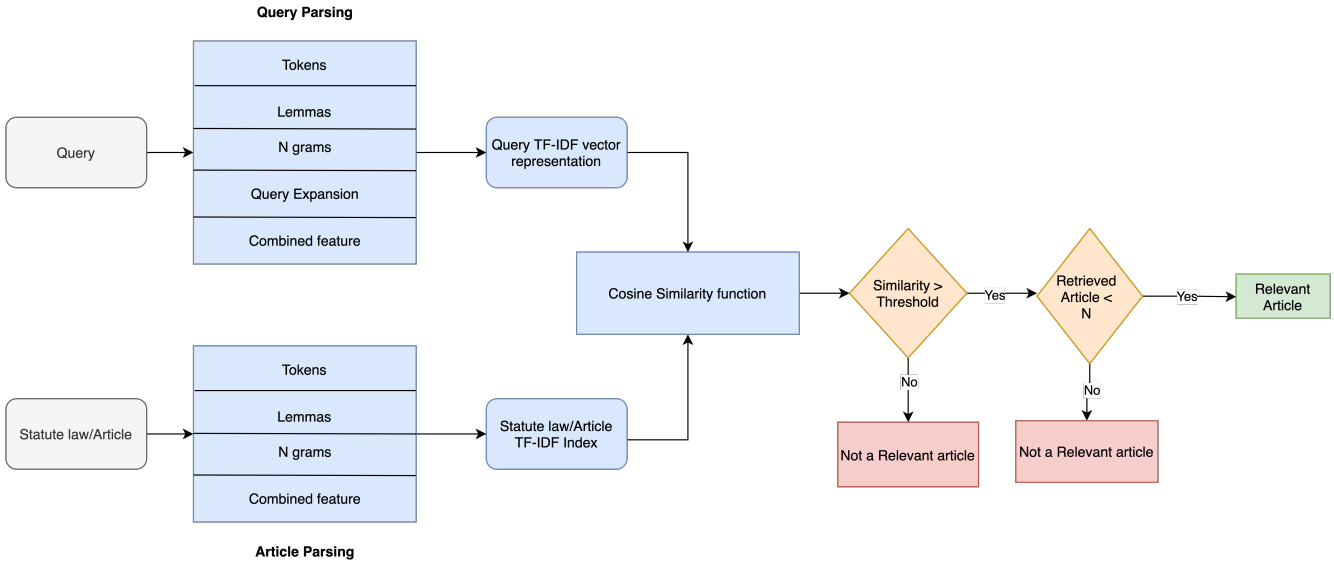

Fig. 5: TF-IDF Experiments setup

process without any semantic information encoded in the indexing process but only with statistical frequency and inverse frequency-based information of the terms used in the articles and queries. This approach is less computationally intensive and is purely statistics-based with no hyperparameters for tuning.

The TF-IDF-based model was experimented by introducing various methods for indexing the articles and queries. The various features used for indexing are described as follows.

- Token-based features
- Lemma-based features
- Query expansion based features
- Ngram-based features (Bigram and Trigram)
- Combined feature set (Lemma, Bigram, Trigram and Query expansion)

The TF-IDF vectors created using various indexing features are then used in the cosine similarity function to calculate the similarity between an article and a query. A higher similarity score indicates higher relevance. Top N similar articles are retrieved for each query which is considered as the relevant articles for a given query.

4) Okapi BM25 experiments: BM25 is a probabilistic retrieval framework used in this legal information retrieval system to rank the articles for a given query based on the query terms. The advantage of ranking the articles based on scores by BM25 over a TF-IDF is the consideration of the average length of the articles along with the Term Frequency and Inverse Document Frequency of a query term. BM25 also ranks the articles based on no semantic information encoded but only on statistical information about the query terms and length of the articles along with two free parameters which can be tuned for better performance. The BM25 function provides a weight or a score that is used to rank each article for a given query. The similarity between an article and a query is calculated based on the rank of an article provided by the score as shown in Figure 6.

The OkapiBM25 approach also experimented with various feature sets for indexing which were used to retrieve the relevant articles. The feature set is similar to the TF-IDF-based experiments and it is listed below.

- Token-based features
- Lemma-based features
- Query expansion based features
- Ngram based features (Bigram and Trigram)
- Combined feature set (Lemma, Bigram, Trigram and Query expansion)

These features are used to index the articles and queries which are used in the ranking function of BM25 to rank the articles for retrieving relevant articles for a given query. The scores are ranked in descending order where the top N-ranked articles are considered as the relevant articles.

The above-mentioned methods are used in the classical approach for legal information retrieval systems.

B. Modern Approach

The modern approach involves two methods which uses static word embedding generated by Word2Vec and contextual word embedding representations generated by a BERT-based distilBERT model.Word2Vec algorithm models the context or semantics based on the words present within a window size in the provided corpus based on the assumption that the surrounding words define the context of a word. DistilBERT based models encode context using self attention mechanism. Thus the The modern approach is considered to encode semantic information of the articles and query for implementation of semantic search. The first method employs

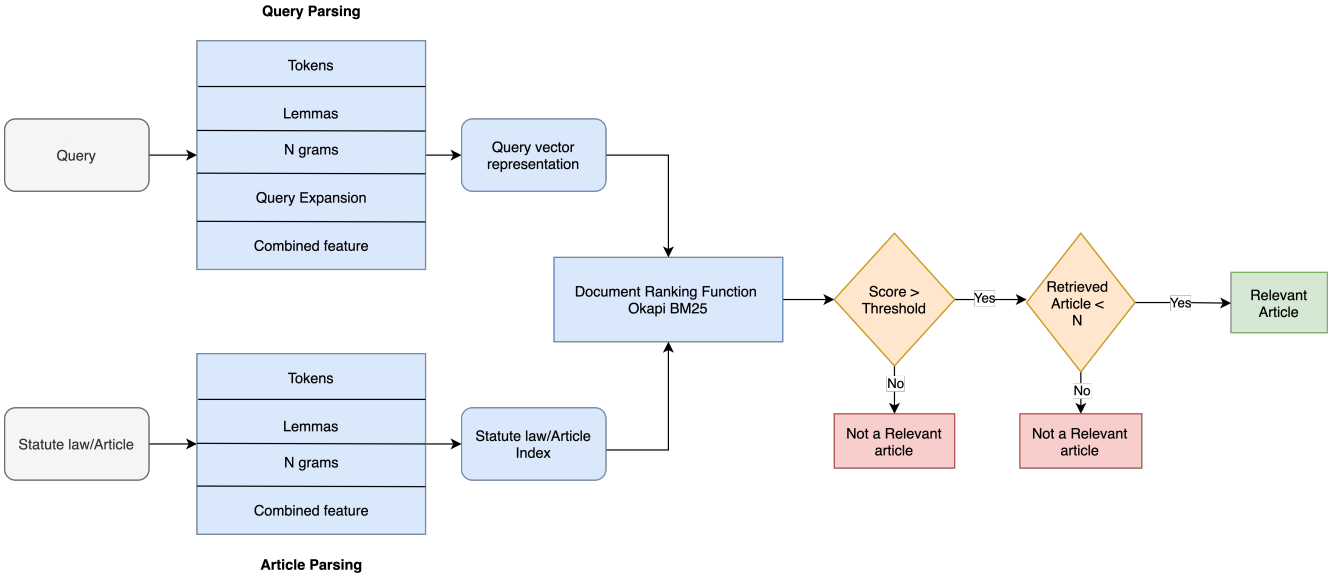

Fig. 6: Okapi BM25 Experiment setup

a distance measure named Word Mover's Distance (WMD) to calculate the distance between articles and queries using static word embeddings. The second method involves the implementation of a distilBERT-based model which approaches the information retrieval as a binary relevance classification problem to retrieve the relevant articles for a given query. The details in setting up the Word Mover's Distance (WMD) and distilBERT methods are discussed in the following sections.

1) Word Mover's Distance (WMD) experiments: The articles and queries are encoded by a static word embedding which is generated using a Word2Vec (Mikolov et al. (2013)) algorithm. These pre-trained word embeddings represent the articles and queries which are then fed into Word Mover's Distance function to calculate the distance between them, then this distance measure is converted into a similarity score. Higher scores indicate higher relevance between an article and a query. The pre-trained word embeddings used in this project for encoding the articles and queries are Law2Vec (Chalkidis and Kampas (2019)) and GloVe (Global Vectors) (Pennington et al. (2014)). In this context, Law2Vec is a legal domain-specific word embedding that is particularly trained on various legal texts in the English language whereas GloVe is a general purpose word embedding that consists mostly of Wikipedia texts.The Out Of Vocabulary (OOV) tokens are not considered during the WMD calculation. If either of the document contains only OOV tokens then the WMD function returns an infinity. The objective of using domain-specific and general purpose word embedding is to understand the impact of the semantic information encoded in legal specific embeddings and general purpose embedding in a legal information retrieval scenario. The setup of WMD for domain-specific and general purpose word embedding is based on the lemma and query expansion-based features which are described below in detail. The word embeddings used in these setup are:

- Domain specific word embedding - Law2Vec
- General purpose word embedding - GloVe

a) Lemma-based features: The article and query tokens are converted into their respective lemmas using the lemmatization process as described in the previous section. This feature reduces the vocabulary and tries to avoid the out of vocabulary issue when a token is not present in the pre-trained embedding vocabulary. The logic behind the conversion of tokens to lemmas for calculating distance between articles and queries is to reduce the distance when most of the inflected words are converted to their root form which might increase the probability of relevance between articles and a query.

In this setup, the articles and query tokens are converted to their respective lemma and then it is encoded with word embeddings. This is then used in the Word Mover's Distance function to calculate the distance between article and query pair which is then converted into a similarity score. For each query, top N similar articles are chosen as relevant articles.

b) Query Expansion: The query expansion is the expansion of the query terms as described in the previous section of the classical approach. With respect to Word Mover's Distance, it increases the probability of decreasing the distance between an article and query by expanding the query with its synonyms.

The expanded queries are encoded with the word embeddings and then fed as an input to the Word Mover's Distance function to calculate the distance between an article and a query which is then converted into a similarity score. For each query, top N similar articles are chosen as relevant articles.

Two sets of experiments were conducted based on lemma based features and query expansion with both Law2Vec and

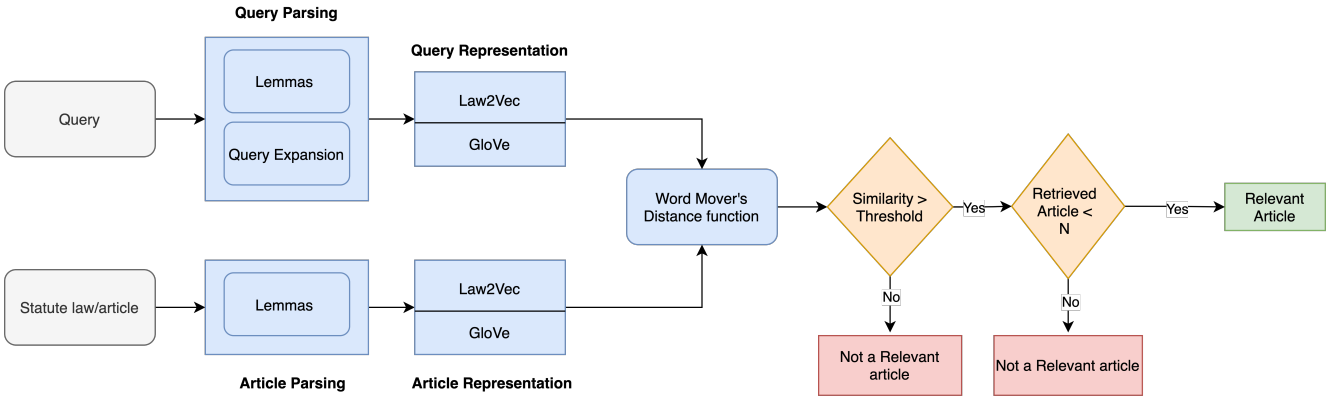

Fig. 7: WMD Experiment setup

GloVe embeddings. In this way, there are 4 experimental setups as shown in the Figure 7.

1) Articles and queries are parsed using lemma based features and represented using Law2Vec word embeddings.
2) Articles are parsed using lemma based features and represented using Law2Vec word embeddings . Queries are parsed using lemma based features along with query expansion and represented using Law2Vec word embeddings.
3) Articles and queries parsed using lemma based features and represented using GloVe word embeddings.
4) Articles are parsed using lemma based features and represented using GloVe word embeddings. Queries are parsed using lemma based features along with query expansion and represented using GloVe word embeddings.

2) DistilBERT experiments: The information retrieval process is converted into a binary relevance classification problem by using a classifier based on BERT which classifies an article as relevant or irrelevant with respect to a query. This approach is inspired by one of the submissions from COLIEE 2020 Shao et al. (2020) which had used BERT-based classification as a retrieval system on articles and queries in the Japanese language. Since transformer models are computationally intensive and time consuming for training, the model considered in this project is DistilBERT which is a smaller version of the BERT base model with lesser parameters when compared to the original model (Sanh et al. (2019)).

DistilBERT model set up for binary relevance classification is shown in Figure 8. The article and query are concatenated into a single string which is given as an input to the DistilBERT model. The DistilBERT model calculates the context embeddings and the first token ”CLS” which is the representation of the whole input string is fed into a feed-forward network and a softmax layer for classifying the text into relevant or irrelevant pairs.

3) DistilBERT classifier Dataset Preparation: The query and article cannot be directly used in a classification model for training and testing of the model. For the data to be used as an input in the DistilBERT model, the article and query pair have to be combined as a single instance and a label indicating the relevance has to be added to the instance. Since transformers are models that capture the contextual information using a self-attention mechanism, the input data consisting of article and query pairs combined is not subjected to pre-processing such as removal of stopwords because of the need to preserve the contextual information in the model during the training phase.

Since it is a text classification problem, the input to the model requires a sequence of text and labels. In this context, the sequence is created by combining the query and an article from the statute law separated by whitespace as shown in Table I. If the query is relevant to the article in the sequence then label ”1” is provided to indicate the relevance else ”0” indicating the irrelevance pair. This way, a single query is combined with all the articles present in the statute law separately to create the sequence of text which is fed as an input to the DistilBERT model as shown in Table I. For instance, there are 715 articles in the Japanese statute law, and a single query which is a ”Yes/No” bar exam question is combined with all 715 articles separated by whitespace. Out of the 715 sequences of texts, there might be N relevant articles for the given query where N $<<$715, for these N sequences the label of ”1” is provided to indicate the relevance whereas for the remaining (715 - N) sequences are provided with a label ”0” indicating the irrelevance of the article to the given query. This way, a retrieval problem is converted into a binary relevance classification problem.

The sequences with the true labels are considered as the training dataset whereas the sequences with no labels are considered as the test dataset for which the relevance has to be inferred from the fine-tuned DistilBERT binary relevance classification model.

The input sequence starts with CLS token which is the representation of whole input sequence, adds PAD token to fill the max sequence length of the input sequence and ends with SEP token. These input sequences are then tokenized accordingly and fed as an input to the DistilBERT binary

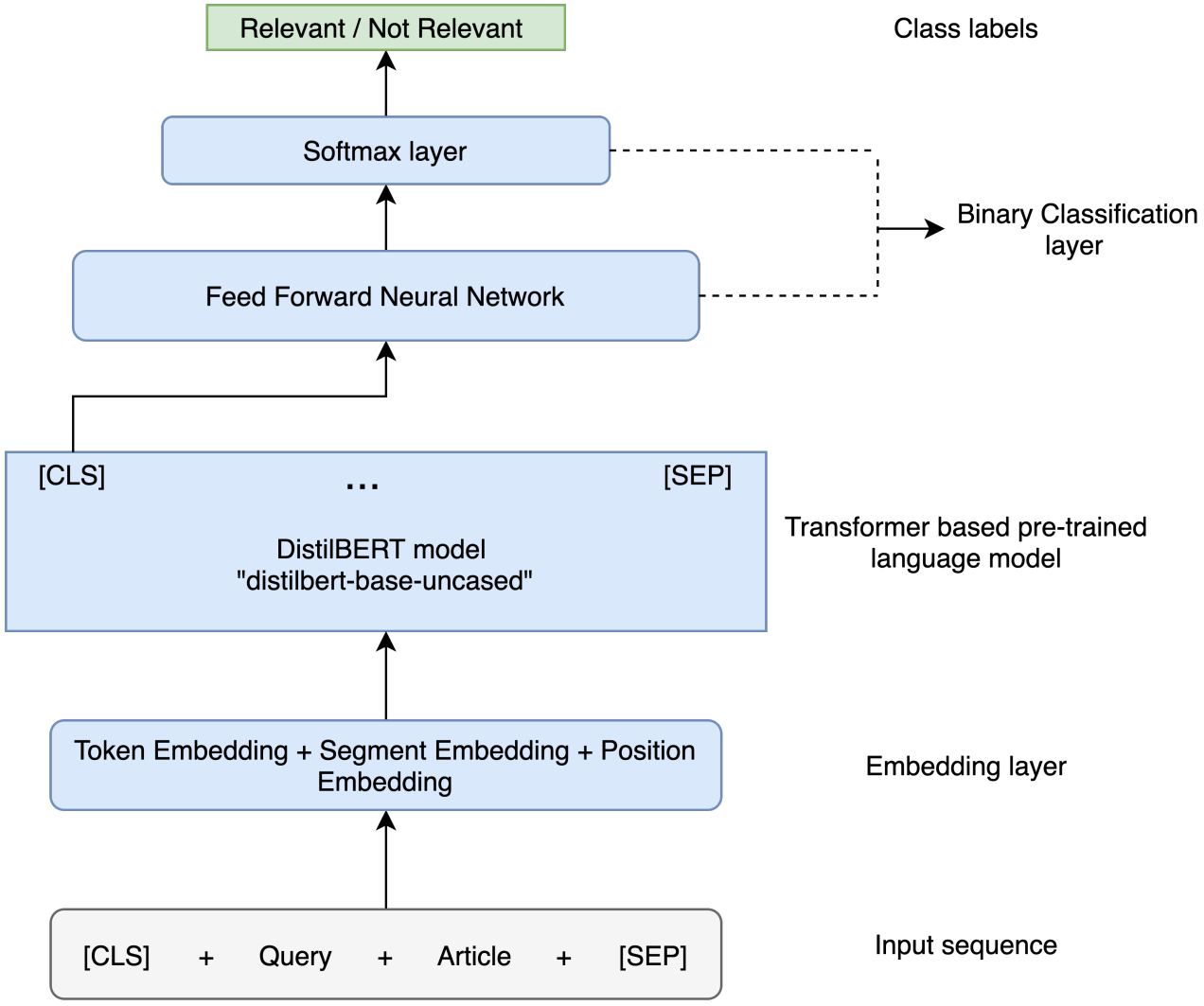

Fig. 8: DistilBERT Experiment setup

relevance classifier model.

In this setup of training dataset creation, there is a drastic imbalance in the relevance and irrelevance classes. Relevance classes are small in number compared to the vast number of irrelevant class instances. The imbalanced dataset creates a bias in the learning algorithm by classifying instances mostly with the class consisting of a higher number of instances. Thus the model will be subjected to a high bias and low variance. To tackle these problems caused due to an imbalanced dataset, few techniques are employed in an experimental setup which is explained in detail in the following section.

4) DistilBERT experimental setup: The pre-trained DistilBERT Sequence classification model is fine-tuned using the English translated Japanese statute law for binary relevance classification. The techniques used for solving the imbalanced dataset problems are listed below.

- Oversampling relevant class instances.
- Downsampling irrelevant class instances with and without class weights.
- Downsampling irrelevant class instances and oversampling relevant class instances with and without class weights.

Oversampling and downsampling are famous techniques used for dealing with imbalanced datasets (Rout et al. (2018)). Downsampling and oversampling methods are coupled with class weights to handle imbalanced datasets. Oversampling is a procedure of increasing the number of samples present in a certain class ”N” number of times just by duplication or through data augmentation methods. Downsampling on the other hand is a procedure of decreasing the number of samples present in a certain class to counteract the problem of the imbalanced dataset by eliminating certain instances or creating

ID	Sequence	Label
H21-1-1	Acceptance made by a minor that received an offer of gifts without burden without getting consent from his/her statutory agent may not be rescinded. (Juridical Acts by Minors) Article_5 (1) A minor must obtain the consent of the minor's legal representative to perform a juridical act; provided, however, that this does not apply to a juridical act for merely acquiring a right or being released from an obligation. (2) A juridical act in contravention of the provisions of the preceding paragraph is voidable. (3) Notwithstanding the provisions of paragraph (1), a minor may freely dispose of property that the legal representative has permitted the minor to dispose of for a specified purpose, to an extent that falls within the scope of that purpose. The same applies if the minor disposes of property that the legal representative has permitted the minor to dispose of without specifying a purpose.	1
H21-1-1	Acceptance made by a minor that received an offer of gifts without burden without getting consent from his/her statutory agent may not be rescinded. (Fundamental Principles) Article_1 (1) Private rights must be congruent with the public welfare. (2) The exercise of rights and performance of duties must be done in good faith. (3) Abuse of rights is not permitted.	0

TABLE I: Binary Relevance classification example from training dataset

a smaller sample of that particular class. These methods can be complemented by the introduction of class weights. The idea of using a class weight is to weigh the loss computed for different classes differently based on whether they belong to the majority or minority classes. The class weight penalizes the minority class's misclassification more rather than the misclassification of the majority class tending the model to not overfit the majority class (Rout et al. (2018)). The experimental setups for DistilBERT binary relevance classification models are explained below in detail.

a) Oversampling relevant class instances: As a technique to deal with the imbalanced dataset, the class with fewer instances is over-sampled. In this case, the relevant class has significantly fewer instances and they are oversampled to create an almost balanced dataset. The oversampling could be done by just duplicating the relevant class instances N number of times in this scenario. In this setup, it is implemented in 2 experimental scenarios where the relevant class is oversampled 100 times in the first scenario and 500 times in the second scenario.

b) Downsample irrelevant class instances: The irrelevant class instances are downsampled by eliminating instances from the majority class and keeping only M instances that are chosen randomly or based on a particular heuristic approach. With this setup, there are 4 scenarios for experimentation.

- In the first set of scenarios, downsampling the irrelevant class instances randomly by choosing M instances for each relevant instance, drastically reduces the irrelevant class instances. It is then implemented with and without class weights.
- In the second set of scenarios, for each relevant class instance, the top 10 similar irrelevant class instances are chosen by a similarity function such as TF-IDF and cosine similarity method. These top M similar irrelevant class instances are kept and others are eliminated which drastically reduces the number of irrelevant class instances. This setup is experimented with and without class weights.

c) Downsampling irrelevant class and oversampling relevant class instances: In this setup, the irrelevant class instances are downsampled by eliminating instances from the majority class and keeping only M number of instances either randomly chosen or based on a particular heuristic approach. Along with the downsampling, the relevant classes are also oversampled N number of times to balance the irrelevant classes. Here the oversampling is just a duplication of the instances N number of times. With this setup, there can be 4 scenarios for experimentation.

- Downsampling the irrelevant class instances randomly by choosing M instances for each relevant class instance and oversampling for each relevant class instance by N times increases the relevant class instances. In this way, the number of irrelevant class instances is drastically reduced and the relevant class instances are increased. This scenario is implemented with and without class weights.
- Downsampling the irrelevant class instances based on top M similar irrelevant class instances for each relevant class instance using a similarity function such as TF-IDF method and oversampling for each relevant class instance by N times increases the relevant class instances. In this way, the irrelevant class instances are statistically related to the relevant class instances providing the learning algorithm to consider the similarity between the relevant and irrelevant class instances while modelling the decision boundaries between the binary relevance classes. This scenario is experimented with and without class weights.

Overall, there are 10 experimentation setups for the DistilBERT based binary relevance classification model that uses the contextual word embeddings created from the self-attention mechanism by transformer architecture. The experiment names are listed below.

1) 100 times oversampled dataset (without class weights)
2) 500 times oversampled dataset (without class weights)
3) Top 10 random irrelevant class downsample and oversampling of relevant class dataset (without class weights)

4) Top 10 similar irrelevant class downsample and oversampling of relevant class dataset (without class weights)
5) Only Top 10 random Irrelevant class downsampled dataset (without class weights)
6) Only top 10 similar Irrelevant class downsampled dataset (without class weights)
7) Top 10 random irrelevant class downsample and oversampling of relevant class dataset (with class weights)
8) Top 10 similar irrelevant class downsample and oversampling of relevant class dataset (with class weights)
9) Only Top 10 random Irrelevant class downsampled dataset (with class weights)
10) Only top 10 similar Irrelevant class downsampled dataset (with class weights)

These experiments summarise the approaches carried out by a binary relevance classification model for a legal information retrieval system.

VI. Experiment Implementation

This section describes the various libraries, architectures, and hyperparameter settings used in the experimental implementations for legal information retrieval in COLIEE Task 3 described in the Methodology section. The experiments are implemented using the Python programming language.

A. Classical Approach

The TF-IDF-based classical approach is implemented using Gensim library[3] with functions such as TfidfModel and similarities. The TfidfModel[4] function is used to generate the TF-IDF vectors whereas the similarities function implements the sparse matrix cosine similarity between an article and a query. The parameters involved for tuning the TF-IDF model are the number of articles to be retrieved for a query which is denoted as "q" and threshold value for similarity scores which is denoted as "th". These values were optimized to get a better result for Tf-IDF retrieval.

The tokenization, lemmatization features are implemented using SpaCy[5] python natural language processing library. The Ngram features are implemented by using NLTK[6] function called ngrams[7] which is used to create the bigrams and trigrams. The Query expansion was implemented by using a WordNet[8] lexical resource python wrapper provided by NLTK library. This was used to obtain the synonyms for the query terms.

The OkapiBM25 model was implemented by rank_bm25[9] python library which consists of basic OkapiBM25 implementation as well as the modified versions of the BM25 algorithm (Robertson and Zaragoza (2009)). The article scores for each query are calculated using the function BM25Okapi. These scores are used to rank the articles for a given query, where top N articles are chosen as relevant articles. The number of articles to be retrieved which is denoted as "q" and the difference between the top score and score for the retrieved article is be a threshold denoted as "th" is chosen during the retrieval . Therefore "q" and "th" are optimized for better performance of the model. The hyperparameters K1 and b are not optimized and chosen a constant value of 1.5 and 0.75 respectively.

B. Modern Approach

The Word Mover's Distance model was implemented by a word_mover_distance python library[10] which largely reused gensim library implementation of wmdistance function which calculates the distance between an article and a query. The static pre-trained word embeddings for articles and queries were obtained from Law2vec[11] domain-specific word embedding vectors and GloVe[12] general-purpose word embedding vectors. The number of articles "q" to be retrieved by the model and the threshold "th" for the similarity above which the relevant articles are chosen is provided during the retrieval as a hyper-parameter which impacts the retrieval performance of the model.

Lemmatization of the articles and queries are implemented using SpaCy[13] python natural language processing library. The Query expansion was implemented by using a WordNet [14] lexical resource python wrapper provided by NLTK library. This is used to obtain the synonyms for the query terms similar to the classical approach query expansion method.

Information retrieval as a binary relevance classification system is implemented using a BERT-based DistilBERT Sanh et al. (2019) model. Due to the unavailability of computing resources, a distilled version of the original BERT base model is chosen with fewer parameters that are pretrained on general-purpose English text. The transformer-based DistilBERT model was implemented using Huggingface python library[15] which is a huge repository for transformer-based models. The function used for implementing the DistilBERT based binary relevance classifier was TFDistilBertForSequenceClassification[16] which uses the pretrained DistilBERT model named "distilbert-base-uncased".

The architecture of the classifier model involves DistilBERT transformer blocks and a feed-forward network and a softmax layer implemented on top of it with two nodes to determine the probability of the relevant and not relevant classes. An adam optimizer is used with a learning rate of 5e-5 and a cross-entropy loss function is implemented. The pretrained DistilBERT is fine-tuned on the Japanese civil code and legal

[3]https://radimrehurek.com/gensim/
[4]https://radimrehurek.com/gensim/models/tfidfmodel.html
[5]https://spacy.io/
[6]https://www.nltk.org/
[7]https://www.nltk.org/api/nltk.html?highlight=ngram
[8]https://www.nltk.org/howto/wordnet.html
[9]https://pypi.org/project/rank-bm25/
[10]https://pypi.org/project/word-mover-distance/
[11]https://archive.org/details/Law2Vec
[12]https://nlp.stanford.edu/projects/glove/
[13]https://spacy.io/
[14]https://www.nltk.org/howto/wordnet.html
[15]https://huggingface.co/
[16]https://huggingface.co/transformers/model_doc/distilbert.html

bar exam question dataset with an objective of binary relevance classification. The model is fine-tuned with a batch size of 8 and is run for 2 epochs. The DistilBERT binary classifier model is fine-tuned on Graphics Processing Unit (GPU) for faster fine-tuning.

The class weights for penalizing the minority class were implemented by a class_weight function from sklearn python library[17].

Based on the above-mentioned architecture and hyperparameters all the experiments for the DistilBERT binary relevance classification system are conducted.

VII. Evaluation

The legal information retrieval system is evaluated using metrics such as precision, recall, and F2 measure as used in COLIEE and are calculated as mentioned in equation 4, 5 and 6.

$$Precision = average\ of \frac{(the\ number\ of\ correctly\ retrieved\ articles\ for\ each\ query\ )}{(the\ number\ of\ retrieved\ articles\ for\ each\ query\ )} \tag{4}$$

$$Recall = average\ of \frac{(the\ number\ of\ correctly\ retrieved\ articles\ for\ each\ query\ )}{(the\ number\ of\ relevant\ articles\ for\ each\ query\ )} \tag{5}$$

$$F2\ score = \frac{(5\ *\ Precision\ *\ Recall)}{(4\ +\ Precision\ +Recall\ )} \tag{6}$$

Precision is used to measure the reliability of the IR systems whereas Recall is used to measure the probability of relevant documents retrieved by the IR systems. F2 score on the other hand is a weighted harmonic mean of precision and recall which provides a holistic measure for an IR system. In COLIEE, the F2 score as mentioned in equation 6 is considered as a final score for evaluation of the models.

The evaluation results of the IR models are provided in Table II. The results of the experiment turned out to be unexpected as the classical approach models outperformed the modern approaches. Various indexing methods used in the classical approach of TF-IDF and BM25 combined together perform far better in comparison with respect to the modern approaches of static word embeddings and contextual embeddings based on the Binary relevance classification model. Thus it is evident from the experiment results that the TF-IDF model with lemmas and combination of lemmas, query expansion, ngrams had similar F2 scores indicating that the addition of features did not yield improvement in the performance of the model. Whereas, the BM-25 model with lemmas alone was able to outperform the models with other features.

The modern approach performs poorly as the F2 score remains near 0 percent throughout all the experiments. The Word Mover's Distance with the general purpose and domain-specific word embeddings seem to have no impact on the retrieval process. The Binary relevance classification top 10 similar irrelevant class downsampled model with class weights achieve the highest recall score but nearly 0 precision as the learning algorithm has captured the bias towards the relevant classes and tagged most of the test instances as relevant class producing a huge number of false positives. The modern approach though encoded with semantic information struggles to perform at least on par with the classical models due to the distance measure and architecture designs proposed in this project.

Based on these results TF-IDF with lemmas, BM25 with lemmas, and bigrams perform the best when compared to the other experimental models proposed.

VIII. Discussion and Future works

The classical approach with various indexing methods proved to yield significant results. TF-IDF and BM25 models just with Lemma as an indexing method were able to provide a decent F2 score which indicates that reducing the words to their root form with Lemmatization does have some impact on the IR process. Bigram feature indexing in the BM25 model provides equivalent performance to Lemma indexing. Overall all the indexing methods performed more or less equally which reveals that statistical frameworks yield similar results with various indexing methods.

From the experimental results, it is evident that the semantic information encoded into the models either with static word embeddings or contextual word embeddings did not have a significant impact on the information retrieval process given the retrieval models used along with the input features provided in this project.

The Word Mover's distance proved to be inefficient with both general-purpose and domain-specific word embeddings which might have been due to the distance measure employed in finding the similarity between an article and a query in a normalized bag of words fashion. The travel cost between the words must be high enough because of the embedding space which leads to higher distance and lower similarity in the retrieval process. Another reason would be the Out Of Vocabulary (OOV) problem where there is no word embedding for a word/token present in an article and query pair. Out of 9333 unique article and query tokens (including expanded query tokens) there were 3803 OOV tokens in Law2Vec embeddings and 3008 OOV tokens in GloVe embeddings. This would cause loss of information during the search for a relevant article only with a limited amount of words/tokens present. Since only static word embeddings were used in the previous approach, the next approach inspired by Shao et al. (2020) employed contextual word embeddings in a distilBERT base binary relevance classification model for retrieving relevant articles. The original approach by Shao et al. (2020) topped COLIEE 2020 Task 3 with an F2 measure of 65.9% whereas a similar approach was implemented on English-translated Japanese statute laws and legal bar exam questions which performed poorly with near 0% F2 scores. This could have happened for so many factors. Since this is a deep learning-based approach explainability of the output

[17]https://scikit-learn.org/stable/index.html

might be difficult. Due to this many reasons were hypothesized with the pieces of evidence present at hand. One of the major reasons is the difference in the language used for retrieving the relevant articles.Shao et al. (2020) used the Japanese language for the retrieval process whereas this project used English-translated text. This difference could contribute to the difference in the language understanding capabilities of the model. Also, the original approach used the Japanese BERT base model[18] which is larger and has more Japanese language understanding capabilities whereas we had used the lighter version of BERT named the DistilBERT model for the English language. Though distilBERT performs comparatively well to its BERT counterpart the language understanding capabilities of an English-translated Japanese text is debatable. Apart from these two major differences, the hyperparameter tuning used in their implementations is unknown which also contributes to the output of the Task. It should also be noted that the input sequence consisting of an article and a query were combined with whitespace and note using a [SEP] token. This might have also contributed to the lower performance of the distilBERT model.

In the DistilBERT based experiments, most of the work had been done in dealing with imbalanced datasets such as oversampling, downsampling, and introducing class weights. There seems to be very less effect except for the number of retrieved articles. Other than that there seems no significant difference in the retrieval performance of the models.

Since explainability of the results produced by DistilBERT based experiments was out of scope for this project, we tried to probe the final softmax classifier layer of the Binary relevance classification model to understand how confident the model is when predicting a class label either relevant or irrelevant. During this, it was evident that the model was not able to clearly distinguish between the two classes in most of the classes due to the low margin in its prediction between the two classes. For simplicity, let us just visualize two models classifier layer in which one of the two models had classified all the instances as relevant (”Only top 10 similar Irrelevant class downsampled dataset with class weights”) and the other model has classified all the instances as irrelevant (”Only Top 10 random Irrelevant class downsampled dataset (with class weights)”).

From figure 10 it is evident that the model is confident in its prediction of all the instances as an irrelevant class and there is a large margin between both the classes. Whereas from figure 9 we can see that the model is not very confident about its prediction where the margin between both the classes are very less. This indicates that the signals picked up in the models are not useful to create the decision boundary between two classes but it is very difficult to understand the reason why the model behaves this way.

Using static or contextual word embeddings for a semantic search opens up a lot of possibilities for experimentation. Contextual word embeddings can be used along with classical

[18]https://huggingface.co/cl-tohoku/bert-base-japanese

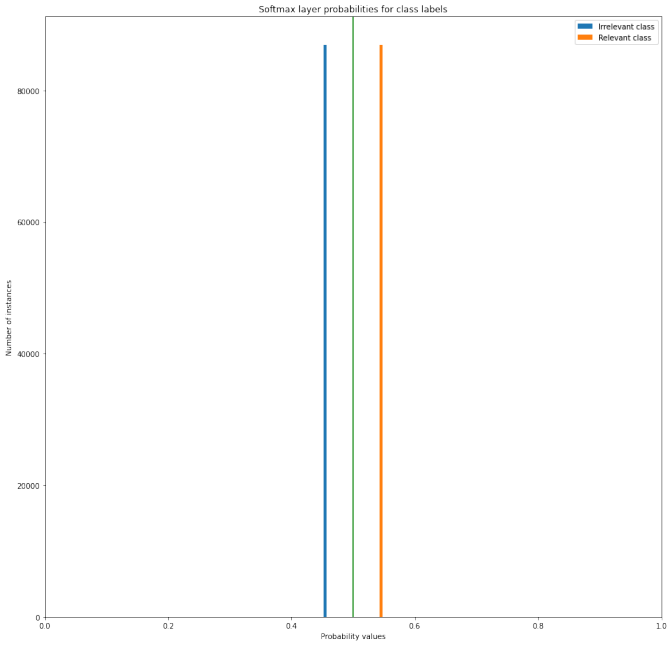

Fig. 9: Only Top 10 similar Irrelevant class downsampled dataset (with class weights) model which classified all instances as Relevant class.

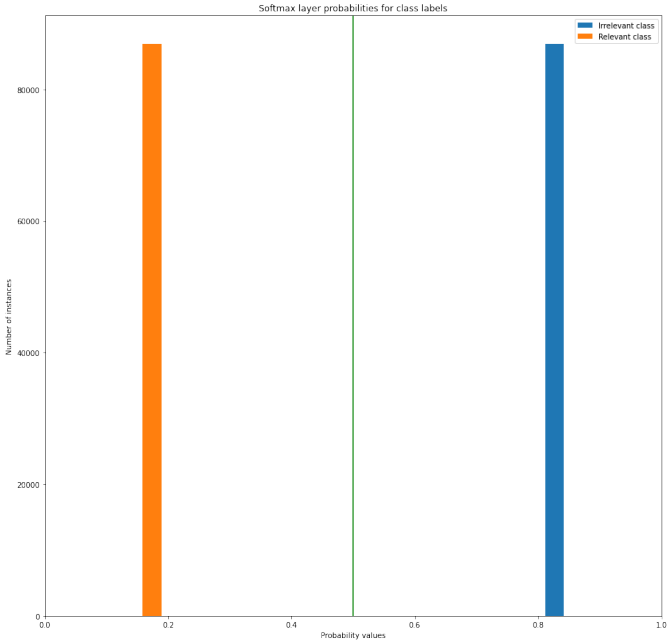

Fig. 10: Only Top 10 random Irrelevant class downsampled dataset (with class weights) model which classified all instances as Irrelevant class.

approaches such as TF-IDF or BM25. Initial retrieval can be done by classical approaches and contextual word embeddings can be used to rerank the articles according to the similarity based on the semantic information such as implementations by Choudhary et al. (2020) and Qadrud-Din et al. (2020). Also as future work, it would benefit the community of information retrieval to understand why pretrained language models with semantic information such as DistilBERT perform poorly on binary relevance classification as happened in the above-presented experiments.

IX. Conclusion

Legal Information Retrieval using classical and modern approaches have been experimented in this project and the results indicate that classical approaches outperform the modern approaches that were implemented in this project. This is very specific to the architecture design and distance measure used in modern approaches. Despite better results achieved by Shao et al. (2020) using a similar Binary relevance classification method, we had discussed the potential factors which might affect the performance of the DistilBERT based experiments implemented in this project. But just statistical frameworks with various indexing methods alone or combined can produce decent retrieval outputs in the legal domain. The future path to these approaches would be understanding the performance and behavior of models implemented in modern approaches which will shed light on the bottlenecks present in large pretrained language models and word embeddings being used in IR settings.

References

Sophia Althammer, Mark Buckley, Sebastian Hofstätter, and Allan Hanbury. Linguistically informed masking for representation learning in the patent domain. *arXiv preprint arXiv:2106.05768*, 2021.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155, 2003.

Michael J Bommarito II, Daniel Martin Katz, and Eric M Detterman. Lexnlp: Natural language processing and information extraction for legal and regulatory texts. In *Research Handbook on Big Data Law*. Edward Elgar Publishing, 2021.

Ilias Chalkidis and Dimitrios Kampas. Deep learning in law: early adaptation and legal word embeddings trained on large corpora. *Artificial Intelligence and Law*, 27(2):171–198, 2019.

Sneha Choudhary, Haritha Guttikonda, Dibyendu Roy Chowdhury, and Gerard P Learmonth. Document retrieval using deep learning. In *2020 Systems and Information Engineering Design Symposium (SIEDS)*, pages 1–6. IEEE, 2020.

Zhuyun Dai and Jamie Callan. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 985–988, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Hai Dong, Farookh Khadeer Hussain, and Elizabeth Chang. A survey in traditional information retrieval models. In *2008 2nd IEEE International Conference on Digital Ecosystems and Technologies*, pages 397–402. IEEE, 2008.

Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

Gao Huang, Chuan Guo, Matt J Kusner, Yu Sun, Fei Sha, and Kilian Q Weinberger. Supervised word mover's distance. *Advances in neural information processing systems*, 29, 2016.

Jenna Kanerva, Filip Ginter, and Tapio Salakoski. Universal lemmatizer: A sequence-to-sequence model for lemmatizing universal dependencies treebanks. *Natural Language Engineering*, pages 1–30, 2020.

Edward Loper and Steven Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.

Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. Cedr: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1101–1104, 2019.

Tamsin Maxwell and Burkhard Schafer. Natural language processing and query expansion in legal information retrieval: challenges and a response. *International Review of Law, Computers & Technology*, 24(1):63–72, 2010.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, 1990.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. Deeprank: A new deep architecture for relevance ranking in information retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 257–266, 2017.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Javed Qadrud-Din, Ashraf Bah Rabiou, Ryan Walker, Ravi Soni, Martin Gajek, Gabriel Pack, and Akhil Rangaraj. Transformer based language models for similar text retrieval and ranking. *arXiv preprint arXiv:2005.04588*, 2020.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence

embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Stephen Robertson and Hugo Zaragoza. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc, 2009.

Neelam Rout, Debahuti Mishra, and Manas Kumar Mallick. Handling imbalanced data: a survey. In *International proceedings on advances in soft computing, intelligent systems and applications*, pages 431–443. Springer, 2018.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 59–66. IEEE, 1998.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Omid Shahmirzadi, Adam Lugowski, and Kenneth Younge. Text similarity in vector space models: a comparative study. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 659–666. IEEE, 2019.

Hsuan-Lei Shao, Yi-Chia Chen, and Sieh-Chuen Huang. Bert-based ensemble model for statute law retrieval and legal information entailment. In *JSAI International Symposium on Artificial Intelligence*, pages 226–239. Springer, 2020.

Keet Sugathadasa, Buddhi Ayesha, Nisansa de Silva, Amal Shehan Perera, Vindula Jayawardana, Dimuthu Lakmal, and Madhavi Perera. Legal document retrieval using document vector embeddings and deep learning. In *Science and information conference*, pages 160–175. Springer, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Wei Yang, Haotian Zhang, and Jimmy Lin. Simple applications of bert for ad hoc document retrieval. *arXiv preprint arXiv:1903.10972*, 2019.

X. APPENDIX

A. Feature set examples

The feature set is explained with an example using a query from test data and article from statute law.

- Query : A contract of sales concluded by a minor may not be rescinded if it relates to daily life, even in cases the consent of the parental authority is not obtained.
- Relevant article : (Juridical Acts by Minors) Article_5 (1) A minor must obtain the consent of the minor's legal representative to perform a juridical act; provided, however, that this does not apply to a juridical act for merely acquiring a right or being released from an obligation. (2) A juridical act in contravention of the provisions of the preceding paragraph is voidable. (3) Notwithstanding the provisions of paragraph (1), a minor may freely dispose of property that the legal representative has permitted the minor to dispose of for a specified purpose, to an extent that falls within the scope of that purpose. The same applies if the minor disposes of property that the legal representative has permitted the minor to dispose of without specifying a purpose.

a) Token based feature:

- Query : contract, sales, concluded, minor, may, not, rescinded, relates, daily, life, even, cases, consent, parental, authority, not, obtained
- Relevant Article : juridical, acts, minors, article_5, 1, minor, must, obtain, consent, minor, ”s”, legal, representative, perform, juridical, act, provided, however, not, apply, juridical, act, merely, acquiring, right, released, obligation, 2, juridical, act, contravention, provisions, preceding, paragraph, voidable, 3, notwithstanding, provisions, paragraph, 1, minor, may, freely, dispose, property, legal, representative, permitted, minor, dispose, specified, purpose, extent, falls, within, scope, purpose, applies, minor, disposes, property, legal, representative, permitted, minor, dispose, without, specifying, purpose

b) Lemma based feature:

- Query : contract, sale, conclude, minor, may, not, rescind, relate, daily, life, even, case, consent, parental, authority, not, obtain
- Relevant item : juridical, act, minor, article_5, 1, minor, must, obtain, consent, minor, ”s”, legal, representative, perform, juridical, act, provide, however, not, apply, juridical, act, merely, acquire, right, release, obligation, 2, juridical, act, contravention, provision, precede, paragraph, voidable, 3, notwithstanding, provision, paragraph, 1, minor, may, freely, dispose, property, legal, representative, permit, minor, dispose, specified, purpose, extent, fall, within, scope, purpose, apply, minor, dispose, property, legal, representative, permit, minor, dispose, without, specify, purpose

c) Ngram based features:

- Query bigrams: juridical act, act minor, minor article 5, article 5 1, 1 minor, minor must, must obtain, obtain consent, consent minor, ”minor s”, ”s legal”, legal representative, representative perform, perform juridical, juridical act, act provide, provide however, however not, not apply, apply juridical, juridical act, act merely, merely acquire, acquire right, right release, release obligation, obligation 2, 2 juridical, juridical act, act contravention, contravention provision, provision precede, precede paragraph, paragraph voidable, voidable 3, 3 notwithstanding, notwithstanding provision, provision paragraph, paragraph 1, 1 minor, minor may, may freely, freely dispose, dispose property, property legal, legal representative, representative permit, permit minor, minor dispose, dispose specified, specified purpose, purpose extent, extent fall, fall within, within scope, scope purpose, purpose apply, apply minor, minor dispose, dispose property, property legal, legal representative, representative permit, permit

minor, minor dispose, dispose without, without specify, specify purpose

- Relevant article bigrams : contract sale, sale conclude, conclude minor, minor may, may not, not rescind, rescind relate, relate daily, daily life, life even, even case, case consent, consent parental, parental authority, authority not, not obtain
- Query trigrams : contract sale conclude, sale conclude minor, conclude minor may, minor may not, may not rescind, not rescind relate, rescind relate daily, relate daily life, daily life even, life even case, even case consent, case consent parental, consent parental authority, parental authority not, authority not obtain
- Relevant article trigrams : juridical act minor, act minor article 5, minor article 5 1, article 5 1 minor, 1 minor must, minor must obtain, must obtain consent, obtain consent minor, "consent minor s", "minor s legal", "s legal representative", legal representative perform, representative perform juridical, perform juridical act, juridical act provide, act provide however, provide however not, however not apply, not apply juridical, apply juridical act, juridical act merely, act merely acquire, merely acquire right, acquire right release, right release obligation, release obligation 2, obligation 2 juridical, 2 juridical act, juridical act contravention, act contravention provision, contravention provision precede, provision precede paragraph, precede paragraph voidable, paragraph voidable 3, voidable 3 notwithstanding, 3 notwithstanding provision, notwithstanding provision paragraph, provision paragraph 1, paragraph 1 minor, 1 minor may, minor may freely, may freely dispose, freely dispose property, dispose property legal, property legal representative, legal representative permit, representative permit minor, permit minor dispose, minor dispose specified, dispose specified purpose, specified purpose extent, purpose extent fall, extent fall within, fall within scope, within scope purpose, scope purpose apply, purpose apply minor, apply minor dispose, minor dispose property, dispose property legal, property legal representative, legal representative permit, representative permit minor, permit minor dispose, minor dispose without, dispose without specify, without specify purpose

d) Query Expansion:

- Query : countermand, causa, venial, link, typeface, life history, authority, self-confidence, living, vacate, press, eventide, casing, may, overturn, casual, abridge, resolve, paternal, potency, Crataegus oxycantha, compact, evening, day by day, life story, reverse, associate, prevail, sales event, English hawthorn, level, flush, nipper, condense, cut-rate sale, bear on, bureau, whitethorn, "compositors case", "typesetters case", shorten, undertake, lifetime, connect, dominance, sales agreement, everyday, grammatical case, federal agency, modest, office, pillowcase, slip, guinea pig, receive, touch, life, instance, sign up, hold, Crataegus laevigata, shell, day-by-day, sale, sign, touch on, pertain, tike, display case, government agency, yet, colligate, close, lawsuit, encase, not, get, narrow, go for, still, reason out, day-to-day, lifetime, liveliness, suit, authorisation, have-to doe with, case, obtain, incur, link up, cut, event, face, underage, fount, sprightliness, concern, assurance, agency, contract bridge, eccentric, even, sheath, caseful, authorization, even out, nestling, sureness, declaration, nonaged, minor, lift, subject, daily, small, May, kid, rescind, find, character, revoke, child, pocket-sized, aliveness, reason, small fry, fry, incase, shrink, tiddler, accept, regular, small-scale, example, concentrate, youngster, squeeze, annul, tyke, lifespan, compress, foreshorten, tied, fifty-fifty, non, eve, consent, come to, say-so, pocket-size, refer, self-assurance, conclude, constrict, reduce, maternal, spirit, tie in, day-after-day, vitrine, contract, parental, showcase, font, shaver, sign on, cause, relate, repeal, take, pillow slip, type, biography, confidence, animation, sanction, interrelate, life sentence, abbreviate
- Relevant article : juridical, acts, minors, article_5, 1, minor, must, obtain, consent, minor, "s", legal, representative, perform, juridical, act, provided, however, not, apply, juridical, act, merely, acquiring, right, released, obligation, 2, juridical, act, contravention, provisions, preceding, paragraph, voidable, 3, notwithstanding, provisions, paragraph, 1, minor, may, freely, dispose, property, legal, representative, permitted, minor, dispose, specified, purpose, extent, falls, within, scope, purpose, applies, minor, disposes, property, legal, representative, permitted, minor, dispose, without, specifying, purpose

e) Combined feature set:

- Query : countermand, causa, venial, link, typeface, life history, authority, self-confidence, living, vacate, press, eventide, casing, may, overturn, casual, abridge, resolve, paternal, potency, Crataegus oxycantha, compact, evening, day by day, life story, reverse, associate, prevail, sales event, English hawthorn, level, flush, nipper, condense, cut-rate sale, bear on, bureau, whitethorn, "compositors case", "typesetters case", shorten, undertake, lifetime, connect, dominance, sales agreement, everyday, grammatical case, federal agency, modest, office, pillowcase, slip, guinea pig, receive, touch, life, instance, sign up, hold, Crataegus laevigata, shell, day-by-day, sale, sign, touch on, pertain, tike, display case, government agency, yet, colligate, close, lawsuit, encase, not, get, narrow, go for, still, reason out, day-to-day, lifetime, liveliness, suit, authorisation, have-to doe with, case, obtain, incur, link up, cut, event, face, underage, fount, sprightliness, concern, assurance, agency, contract bridge, eccentric, even, sheath, caseful, authorization, even out, nestling, sureness, declaration, nonaged, minor, lift, subject, daily, small, May, kid, rescind, find, character, revoke, child, pocket-sized, aliveness, reason, small fry, fry, incase, shrink, tiddler, accept, regular, small-scale, example, concentrate, youngster, squeeze, annul,

tyke, lifespan, compress, foreshorten, tied, fifty-fifty, non, eve, consent, come to, say-so, pocket-size, refer, self-assurance, conclude, constrict, reduce, maternal, spirit, tie in, day-after-day, vitrine, contract, parental, showcase, font, shaver, sign on, cause, relate, repeal, take, pillow slip, type, biography, confidence, animation, sanction, interrelate, life sentence, abbreviate, contract sale, sale conclude, conclude minor, minor may, may not, not rescind, rescind relate, relate daily, daily life, life even, even case, case consent, consent parental, parental authority, authority not, not obtain, contract sale conclude, sale conclude minor, conclude minor may, minor may not, may not rescind, not rescind relate, rescind relate daily, relate daily life, daily life even, life even case, even case consent, case consent parental, consent parental authority, parental authority not, authority not obtain

- Relevant article : juridical, act, minor, article 5, 1, minor, must, obtain, consent, minor, ”s”, legal, representative, perform, juridical, act, provide, however, not, apply, juridical, act, merely, acquire, right, release, obligation, 2, juridical, act, contravention, provision, precede, paragraph, voidable, 3, notwithstanding, provision, paragraph, 1, minor, may, freely, dispose, property, legal, representative, permit, minor, dispose, specified, purpose, extent, fall, within, scope, purpose, apply, minor, dispose, property, legal, representative, permit, minor, dispose, without, specify, purpose, juridical act, act minor, minor article 5, article 5 1, 1 minor, minor must, must obtain, obtain consent, consent minor, ”minor s”, ”s legal”, legal representative, representative perform, perform juridical, juridical act, act provide, provide however, however not, not apply, apply juridical, juridical act, act merely, merely acquire, acquire right, right release, release obligation, obligation 2, 2 juridical, juridical act, act contravention, contravention provision, provision precede, precede paragraph, paragraph voidable, voidable 3, 3 notwithstanding, notwithstanding provision, provision paragraph, paragraph 1, 1 minor, minor may, may freely, freely dispose, dispose property, property legal, legal representative, representative permit, permit minor, minor dispose, dispose specified, specified purpose, purpose extent, extent fall, fall within, within scope, scope purpose, purpose apply, apply minor, minor dispose, dispose property, property legal, legal representative, representative permit, permit minor, minor dispose, dispose without, without specify, specify purpose, juridical act minor, act minor article 5, minor article 5 1, article 5 1 minor, 1 minor must, minor must obtain, must obtain consent, obtain consent minor, ”consent minor s”, ”minor s legal”, ”s legal representative”, legal representative perform, representative perform juridical, perform juridical act, juridical act provide, act provide however, provide however not, however not apply, not apply juridical, apply juridical act, juridical act merely, act merely acquire, merely acquire right, acquire right release, right release obligation, release obligation 2, obligation 2 juridical, 2 juridical act, juridical act contravention, act contravention provision, contravention provision precede, provision precede paragraph, precede paragraph voidable, paragraph voidable 3, voidable 3 notwithstanding, 3 notwithstanding provision, notwithstanding provision paragraph, provision paragraph 1, paragraph 1 minor, 1 minor may, minor may freely, may freely dispose, freely dispose property, dispose property legal, property legal representative, legal representative permit, representative permit minor, permit minor dispose, minor dispose specified, dispose specified purpose, specified purpose extent, purpose extent fall, extent fall within, fall within scope, within scope purpose, scope purpose apply, purpose apply minor, apply minor dispose, minor dispose property, dispose property legal, property legal representative, legal representative permit, representative permit minor, permit minor dispose, minor dispose without, dispose without specify, without specify purpose

Experimentation of Legal Text Retrieval								
Model	**Features**	**Total Ret**	**Rel Ret**	**Total Rel**	**Precision**	**Recall**	**F2 Score**	**Remarks**
Classical Approach								
TF-IDF	Tokens	170	51	140	0.300	0.364	0.349	q=2, th=0.31
	Lemma	205	62	140	0.302	0.443	**0.405**	q=2, th=0.27
	Query Expansion	116	49	140	0.422	0.350	0.362	q= 2, th = 0.30
	Bigrams	117	51	140	0.435	0.364	0.376	q=2, th=0.31
	Trigrams	139	50	140	0.359	0.357	0.357	q=2,th=0.15
	Lemma + Bigrams + Trigrams + Query Expansion	192	60	140	0.312	0.428	**0.398**	q=2, th=0.10
BM-25	**Tokens**	112	56	140	0.500	0.400	**0.416**	q=1, th=15
	Lemma	201	64	140	0.318	0.457	**0.420**	q=2, th=15
	Query Expansion	251	68	140	0.271	0.486	**0.419**	q=3, th=7
	Bigrams	190	63	140	0.332	0.450	**0.420**	q=2, th=10
	Trigrams	200	51	140	0.255	0.364	0.336	q=3, th=7
	Lemma + Bigrams + Trigrams + Query Expansion	197	61	140	0.310	0.436	0.403	q=3, th=5
Modern Approach								
WMD	Law2Vec(lemma)	560	10	140	0.0179	0.0714	0.0446	q = 5, th = 0.30
	Law2Vec(Query Expansion)	1344	2	140	0.001 49	0.0143	0.005 25	q= 5 . th = 0.30
	GloVe(lemma)	560	12	140	0.0210	0.0850	0.0530	q= 5, th = 0.30
	GloVe (Query Expansion)	1344	2	140	0.001 50	0.0142	0.005 00	q= 12 , th = 0.30
DistilBERT	100 times oversampled dataset (without class weights)	0	0	140	0	0	0	Epochs = 2. Imbalanced dataset. Most of the instances were classified as 0 (not relevant)
	595 times oversampled dataset (without class weights)	2582	11	140	0.004 26	0.0786	0.0175	Epochs = 2. Imbalanced dataset. Most of the instances were classified as 0 (not relevant)
	Without class weights							
	Top 10 random irrelevant class downsample and oversampling of relevant class dataset	3446	6	140	0.001 74	0.0429	0.007 49	Epochs = 2. Imbalanced dataset. Most of the instances were classified as 0 (not relevant)
	Only top 10 random Irrelevant class downsampled dataset	4084	9	140	0.002 20	0.0643	0.009 69	Epochs = 2. Imbalanced dataset. Most of the instances were classified as 0 (not relevant)
	Top 10 similar irrelevant class downsample and oversampling of relevant class dataset	1323	4	140	0.003 02	0.0286	0.0106	Epochs = 2. Imbalanced dataset. Most of the instances were classified as 0 (not relevant)
	Only top 10 similar Irrelevant class downsampled dataset without class weights	50030	71	140	0.001 42	0.507	0.007 02	Epochs = 2. Imbalanced dataset. Most of the instances were classified as 0 (not relevant)
	With class weights							
	Top 10 random irrelevant class downsample and oversampling of relevant class dataset	2894	5	140	0.001 73	0.0357	0.007 24	Epochs = 2. Imbalanced dataset. Most of the instances were classified as 0 (not relevant)
	Only top 10 random Irrelevant class downsampled dataset	0	0	140	0	0	0	Epochs = 2. Imbalanced dataset. Most of the instances were classified as 0 (not relevant)
	Top 10 similar irrelevant class downsample and oversampling of relevant class dataset	20404	35	140	0.001 72	0.250	0.008 35	Epochs = 2. Imbalanced dataset. Most of the instances were classified as 0 (not relevant)
	Only top 10 similar Irrelevant class downsampled dataset with class weights	86912	139	140	0.001 60	0.993	0.007 95	Epochs = 2. Imbalanced dataset. Most of the instances were classified as 0 (not relevant)

TABLE II: Classical Approach and Modern Approach based Experiment Results