Artificial Intelligence and Law (2024) 32:57–86
https://doi.org/10.1007/s10506-022-09341-8

ORIGINAL RESEARCH

Attentive deep neural networks for legal document retrieval

Ha-Thanh Nguyen[1] · **Manh-Kien Phi**[2] · **Xuan-Bach Ngo**[2] · **Vu Tran**[1] · **Le-Minh Nguyen**[1] · **Minh-Phuong Tu**[2]

Accepted: 29 November 2022 / Published online: 27 December 2022

Abstract

Legal text retrieval serves as a key component in a wide range of legal text processing tasks such as legal question answering, legal case entailment, and statute law retrieval. The performance of legal text retrieval depends, to a large extent, on the representation of text, both query and legal documents. Based on good representations, a legal text retrieval model can effectively match the query to its relevant documents. Because legal documents often contain long articles and only some parts are relevant to queries, it is quite a challenge for existing models to represent such documents. In this paper, we study the use of attentive neural network-based text representation for statute law document retrieval. We propose a general approach using deep neural networks with attention mechanisms. Based on it, we develop two hierarchical architectures with sparse attention to represent long sentences and articles, and we name them Attentive CNN and Paraformer. The methods are evaluated on datasets of different sizes and characteristics in English, Japanese, and Vietnamese. Experimental results show that: (i) Attentive neural methods substantially outperform non-neural methods in terms of retrieval performance across datasets and languages; (ii) Pretrained transformer-based models achieve better accuracy on small datasets at the cost of high computational complexity while lighter weight Attentive CNN achieves better accuracy on large datasets; and (iii) Our proposed Paraformer outperforms state-of-the-art methods on COLIEE dataset, achieving the highest recall and F2 scores in the top-N retrieval task.

Keywords Legal text retrieval · Deep neural networks · Hierarchical representation · Global attention

This paper is an improved and extended work of Kien et al. (2020).

Ha-Thanh Nguyen and Manh-Kien Phi have contributed equally to this work.

✉ Ha-Thanh Nguyen
nguyenhathanh@jaist.ac.jp

Extended author information available on the last page of the article

1 Introduction

Social relations arise, develop and change daily, so legal documents also need to be promulgated to keep up with the changes of life. There is apparently an increment in the number of legal cases as well as the number of legal documents in different nations. In 2020, the number of civil and criminal cases in the US reached more than 500 thousands.[1] As a civil-law nation, Vietnam has more than 20 types of legal documents with thousands of new documents being issued every week.[2] From the above situation, it can be seen that the use of automatic systems in finding and retrieving documents that match the needs of users is a mandatory requirement. Because of the importance of correctness in the legal field, the performance of these systems is an important attribute to bring them into real life. In this paper, we propose an effective legal retrieval approach for statute law using novel architectures of attentive deep neural networks.

For a legal retrieval system, given a query q, and a legal corpus $\mathcal{L}$, the system needs to return a set of articles $\mathcal{A} \subseteq \mathcal{L}$ that:

$$Relevance(q, \alpha) \forall \alpha \in \mathcal{A}$$

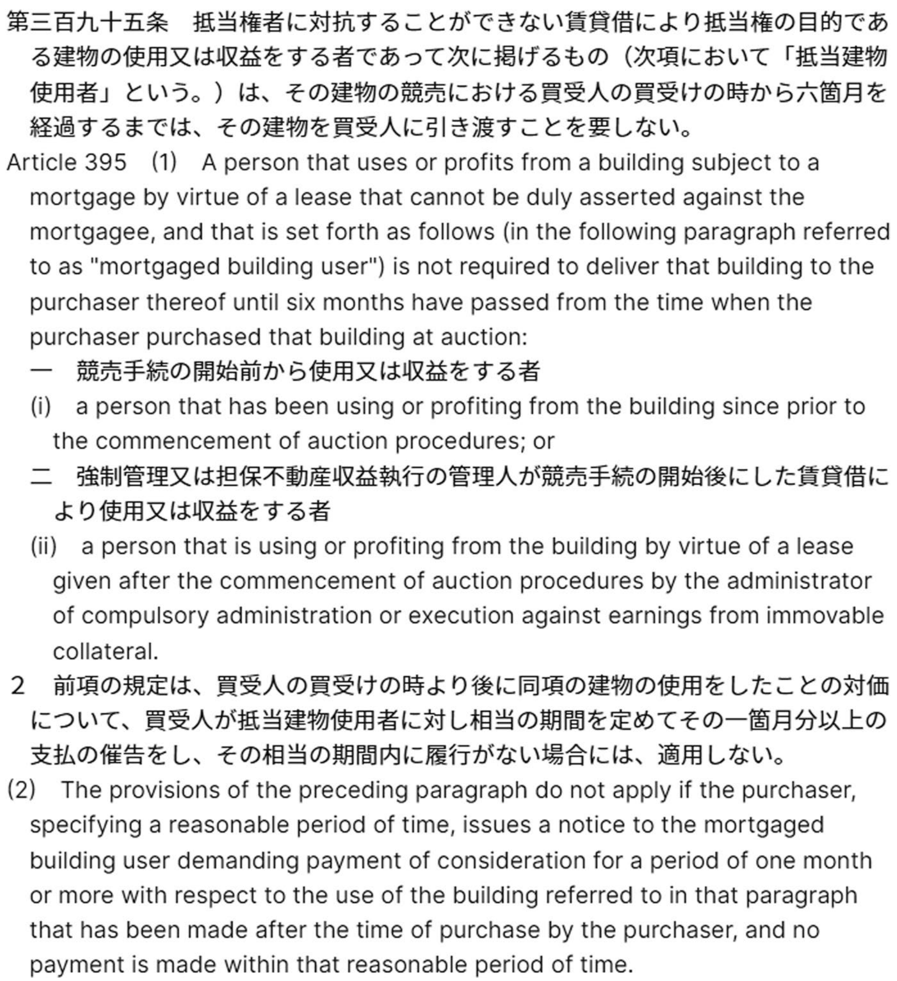
in which *Relevance* is a boolean function that indicates if an article is relevant to the given query.

To define the problem without ambiguity, we first need to clarify the concept of relevance. Dealing with problems in the legal domain requires expert knowledge and understanding in this field. Information retrieval in this field does not simply mean finding all the texts with the most lexical overlapping with the query. A good system also needs to consider the meaning of the query as well as the articles to make reliable alignment between them (Šavelka and Ashley 2021). A relevant article is the one that can be used to answer or validate the lawfulness of a query. Moreover, each article also needs to be interpreted in the appropriate meaning for a specific given query. In turn, queries with non-legal vocabulary also need to be mapped to the corresponding knowledge area in the legal domain.

Merely relying on lexical matching may not be the sufficient approach for this problem. For example, with the purpose of confirming the lawfulness of the query *"Extended parts of the building shall be regarded as appurtenance."*, according to lexical matching result, Article 395 in the Japanese Civil Code (Fig. 1) is the best candidate. This article contains many words in common with the given query. However, the most important word *"appurtenance"* does not appear in Article 395. The correct article to answer this query is Article 87 (Fig. 2), a shorter article that contains fewer words in common with the given query. This article does not mention any *"building"* in its content but can be used to verify the lawfulness of this query. Hence, the better the system understands the semantics of the concepts, the better

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

[2] https://thuvienphapluat.vn/van-ban-moi.



第三百九十五条　抵当権者に対抗することができない賃貸借により抵当権の目的である建物の使用又は収益をする者であって次に掲げるもの（次項において「抵当建物使用者」という。）は、その建物の競売における買受人の買受けの時から六箇月を経過するまでは、その建物を買受人に引き渡すことを要しない。

Article 395 (1) A person that uses or profits from a building subject to a mortgage by virtue of a lease that cannot be duly asserted against the mortgagee, and that is set forth as follows (in the following paragraph referred to as "mortgaged building user") is not required to deliver that building to the purchaser thereof until six months have passed from the time when the purchaser purchased that building at auction:

一　競売手続の開始前から使用又は収益をする者

(i) a person that has been using or profiting from the building since prior to the commencement of auction procedures; or

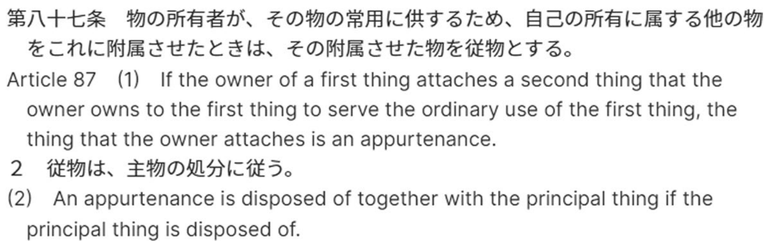
二　強制管理又は担保不動産収益執行の管理人が競売手続の開始後にした賃貸借により使用又は収益をする者

(ii) a person that is using or profiting from the building by virtue of a lease given after the commencement of auction procedures by the administrator of compulsory administration or execution against earnings from immovable collateral.

２　前項の規定は、買受人の買受けの時より後に同項の建物の使用をしたことの対価について、買受人が抵当建物使用者に対し相当の期間を定めてその一箇月分以上の支払の催告をし、その相当の期間内に履行がない場合には、適用しない。

(2) The provisions of the preceding paragraph do not apply if the purchaser, specifying a reasonable period of time, issues a notice to the mortgaged building user demanding payment of consideration for a period of one month or more with respect to the use of the building referred to in that paragraph that has been made after the time of purchase by the purchaser, and no payment is made within that reasonable period of time.

Fig. 1 Article 395 in Japanese Civil Code



第八十七条　物の所有者が、その物の常用に供するため、自己の所有に属する他の物をこれに附属させたときは、その附属させた物を従物とする。

Article 87 (1) If the owner of a first thing attaches a second thing that the owner owns to the first thing to serve the ordinary use of the first thing, the thing that the owner attaches is an appurtenance.

２　従物は、主物の処分に従う。

(2) An appurtenance is disposed of together with the principal thing if the principal thing is disposed of.

Fig. 2 Article 87 in Japanese Civil Code

the performance it can obtain. Building an accurate legal document retrieval system, therefore, depends heavily on good text representation methods.

Recently, deep neural network models are very successful in text representation in a wide range of tasks. In their development, there are various architectures proposed such as convolutional neural networks (CNNs) (Kim 2014; Shen et al. 2014; Severyn and Moschitti 2015; Vaswani et al. 2017), recurrent neural networks (RNNs) (Mikolov et al. 2011), LSTMs (Wang et al. 2016; Palangi et al. 2016; Mueller and Thyagarajan 2016; Chen et al. 2017; Bach et al. 2019a, b) and gated recurrent units (GRUs) (Tang et al. 2015). Most notably, Transformers (Vaswani et al. 2017) leveraging attention mechanism becomes a well-known approach, its pretrained variants like BERT (Devlin et al. 2019), BART (Lewis et al. 2019), GPTs (Radford et al. 2018, 2019; Brown et al. 2020) achieve impressive results in a wide range of natural language processing tasks.

Although there are differences among legal systems, they can be classified and generalized into two main theoretical constructs, common law and civil law (Husa 2016). In the context of civil law tradition, the legal retrieval problem can be done at the document level, article level, or sentence level. Through surveying legal consulting activities in civil law nations like Japan, Germany and Vietnam, we found that retrieval at the article level is a popular approach to answer a legal question. This survey was conducted through consultation with law professors, attorneys, and investigating scholarly materials (Shao et al. 2020; Rabelo et al. 2019; Yoshioka et al. 2018; Nguyen et al. 2017; Thanh et al. 2021) and legal consultant websites in civil law nations like Vietnam,[3] Japan[4] and Germany.[5] In a real situation of legal question answering, the legal consultant often refers to a specific article, neither a whole document nor only a single sentence. From the technical viewpoint, article-level retrieval has its own challenges. As can be seen in Table 7 which demonstrates a legal retrieval-based question answering example, just a few sentences in an answer article contain the necessary information to answer the question. This observation inspires us to design an architecture using an attention mechanism to focus on the necessary part of an article for a more effective retrieval system.

In this paper, we focus on the task of retrieving legal documents at the article level, which serves for question answering in civil law systems. We study on exploiting deep neural networks with attention mechanisms to solve the task. For attention mechanisms, we investigate two recent advanced architectures, i.e., attentive CNNs and self-attention with Transformer, which achieved state-of-the-art results on many NLP tasks. Our contributions can be summarized in the following points:

1. We design a general framework for legal document retrieval using deep neural networks with attention mechanisms. Based on this framework, we develop two attentive deep learning models: Attentive CNN and Paraformer, where the latter represents legal **para**graphs using Trans**former**. Our approach allows encoding long text by letting the model focus on only the important parts of the text.

[3] https://thuvienphapluat.vn.

[4] https://keiji.vbest.jp.

[5] https://www.anwalt.de.

Compared to previous works, we model legal articles as a hierarchical structure to encode them into the vector space.

2. We introduce a Vietnamese dataset for the task, which is much larger than the existing ones. Our dataset is crucial to verify the effectiveness of retrieval models in different languages as well as compare the models' behavior in different corpus sizes. The dataset is also a good resource for the research community in related problems.
3. We conduct an empirical study on proposed models using three datasets, including our Vietnamese dataset, and the English and Japanese datasets from COLIEE.[6] Experimental results show that our models outperform existing methods, both non-deep learning and deep learning ones. Although both Attentive CNN and Paraformer are effective for the task, each model is superior to the other in specific situations. Our results also indicate that using transformer-based pre-trained models can improve the performance of retrieval models, especially when we only have a relatively small training dataset.

The rest of this paper is structured as follows. Sect. 2 describes related work. Section 3 presents three datasets used in our experiments, i.e., Vietnamese, English, and Japanese. In Sect. 4, we introduce our general framework for legal text retrieval and two retrieval models. Experimental results and discussions are described in Sect. 5. Finally, Sect. 6 concludes the paper and discusses future work.

2 Related work

Before the application of neural networks became widespread, there were approaches in classical NLP to solve information retrieval tasks (Cooper 1971; Luhn 1957; Salton and Buckley 1988). These methods are mainly based on different lexical matching techniques. These authors propose logical models as well as statistical models to calculate the similarity between queries and candidates. The methods have their own advantages such as fast computation speed and applicability to many problems. Non-neural methods, however, mainly rely on morphology in the text to make decisions. In natural languages, morphological similarity does not guarantee semantic similarity, so it is difficult to guarantee correctness in semantic similarity using these approaches. Therefore, these approaches have limited performance in the case that the document-query pairs contain many overlapped texts but no relation in the semantic aspect.

The legal language can be translated into logical language (Kowalski and Datoo 2021). One of the most well-known systems using logical models to perform legal retrieval and reasoning for statute law is PROLEG (PROlog based LEGal reasoning support system) (Satoh et al. 2010). This system is empowered by the Japanese Presupposed Ultimate Fact Theory (Ito 2008). PROLEG is based on the idea of the *burden of proof* (i.e. if a fact is failed to be proved as true, it is considered as

[6] https://sites.ualberta.ca/rabelo/COLIEE2021/

false). The relevant rules of the reasoning process can be called out automatically to make reasoning for a query. This system, however, requires the queries and legal documents to be formatted in a logical form. For that reason, the system is not suitable for lay users.

Overcoming the challenge of the semantic morphology difference and the burden of logical representation, several neural approaches in information retrieval in both the general domain and legal domain are proposed (Palangi et al. 2016; Shen et al. 2014; Huang et al. 2013; Šavelka and Ashley 2021; Nguyen et al. 2018). Most of the systems use classical neural network architecture like CNN or LSTM to handle the task.

For legal text, Sugathadasa et al. (2018) and Tran et al. (2020) propose to use neural networks and achieve impressive results. The authors observe the structure of the legal documents and base on their characteristics to propose novel representation methods. Through their experimental results, the author demonstrates that their proposals effectively work for the legal domain. Kien et al. (2020) introduce the neural network architecture that combines CNN and attention mechanisms. With a lightweight design, our model achieves state-of-the-art results on the Vietnamese legal question-answering dataset. These works also reveal that the combination between the semantic vectors and the lexical features can boost the overall performance of the systems.

Pretrained neural approaches construct the models in two phases. In the pretraining phase, the models are trained with general tasks to abstract the relationships between units in the sentences. After that, the models are finetuned with the specifically designed tasks. This family of approaches has been demonstrated to be effective in a wide range of natural language processing as well as legal document processing.

The earliest form of pretrained models is the pretrained word embeddings (Word2Vec Mikolov et al. 2013, GloVe Pennington et al. 2014 or FastText Mikolov et al. 2018). With these pretrained embeddings, we can easily find the semantic relationship between words (e.g. verify the equation $king = queen + man - woman$). In the legal domain, authors of Law2Vec (Chalkidis and Kampas 2019) introduce a variant of word embedding trained on legal corpus and demonstrate its effectiveness. Recently, pretrained models based on Transformer architecture (Vaswani et al. 2017) achieve state-of-the-art results on many benchmark data, both in the general domain (Devlin et al. 2019; Lewis et al. 2019; Radford et al. 2018, 2019; Reimers and Gurevych 2019; Brown et al. 2020) and in the legal domain (Yilmaz et al. 2019; Nguyen et al. 2020; Yoshioka et al. 2021; Nguyen et al. 2021). Pretrained approaches are useful in the case that the training data is limited in quantity.

3 Datasets

To test the proposed approach, we conduct the experiments on the datasets in three languages: Vietnamese, Japanese, and English. The Japanese and English datasets are the different versions of the dataset provided by COLIEE.

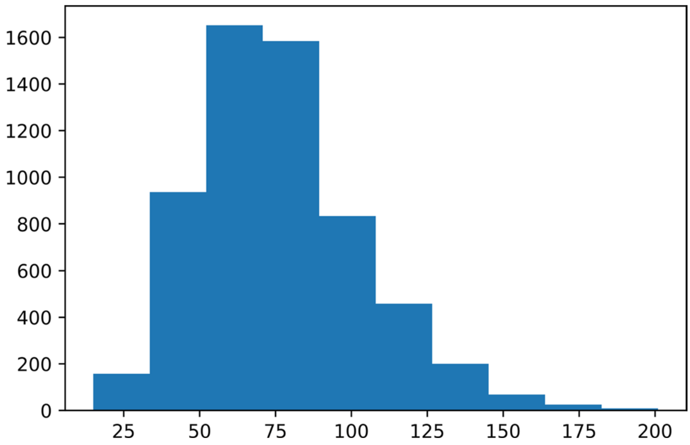


Fig. 3 Query length distribution in character in the Vietnamese dataset

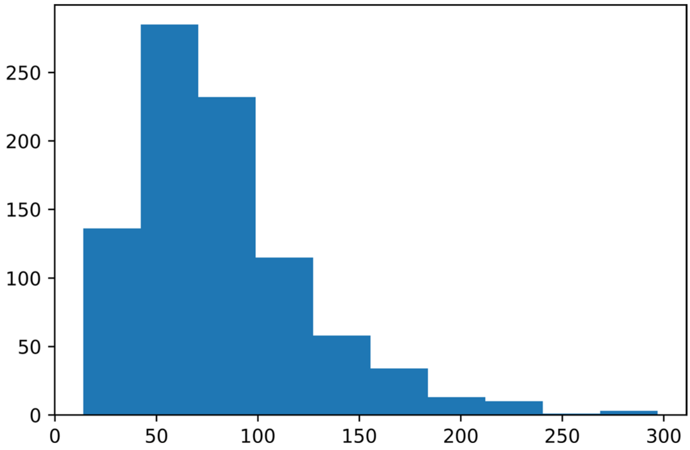


Fig. 4 Query length distribution in character in the Japanese dataset

To build this Vietnamese dataset, we crawled the raw legal documents from the official legal websites[7,8] and the queries from the legal consulting websites.[9,10,11] The raw data to build the corpus of Vietnamese legal documents contains multiple versions of each law and regulation. We removed the redundant old versions and remapped the new relevant articles with the corresponding query in the question-answering dataset. To obtain a good question-answering dataset, we corrected

[7] http://vbpl.vn/tw/pages/home.aspx.

[8] https://thuvienphapluat.vn.

[9] https://hdpl.moj.gov.vn/Pages/home.aspx.

[10] http://hethongphapluat.com/hoi-dap-phap-luat.html.

[11] https://hoidapphapluat.net.

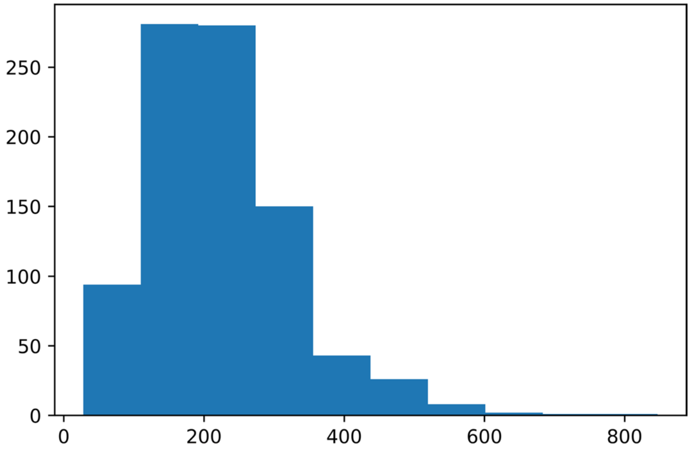


Fig. 5 Query length distribution in character in the English dataset

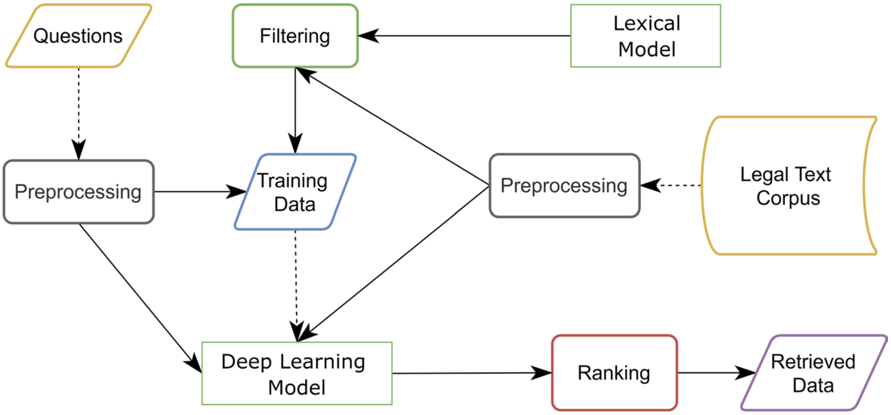


Fig. 6 The pipeline of our proposed approach

spelling, formatting, grammar errors and filtered out the contents which are confusing, uninformative, or low quality. The process of reviewing and editing was done with the support of lawyers. The final version contains 8586 documents (117,545 articles) and 5922 legal queries.

The English and Japanese data provided by COLIEE are of high quality. Though, the number of training samples is relatively small compared to the Vietnamese dataset, which is an interesting challenge for the deep learning approach.

The total number of samples to train the model is 806. The formal test set contains 81 samples. The limitation in the amount of data makes it a practical situation to compare the performance of training-from-scratch models and pretrained models.

Figures 3, 4, and 5 demonstrate the length distribution in characters of the queries in the Vietnamese, Japanese and English datasets respectively. The Vietnamese dataset contains the largest number of queries and almost all of them are shorter than 200 characters. The distribution suggests this dataset is suitable for training deep learning models from scratch. The Japanese and English datasets contain not only fewer but also longer samples. The longest sample is in the English dataset with more than 800 characters. Datasets in multiple languages containing samples of varying lengths are useful for analyzing the characteristics of different models.

4 Retrieval methods

4.1 General approach

The pipeline of our general approach is shown in Fig. 6. There are two phases in the process (i.e. training and inference). In the training phase, from the given question set and the legal text corpus, we preprocess the raw text into a proper form. To obtain the training data, we use the lexical model to filter out non-lexical-matched articles. This process may also remove the relevant candidates from the data; however, this is the trade-off we have to take due to computational resource limitations. After that, the deep learning model is trained by the negative sampling paradigm. In the inference phase, we combine the score from the trained model and the lexical score to rank the candidates to obtain the final relevant articles.

We propose two different architectures of deep neural networks with the general idea of *divide-and-conquer*. The first architecture uses convolutional networks without pretraining, which is named Attentive CNN, the second architecture leverages the power of the Transformer-based pretrained language model, which is named Paraformer. Both architectures contain two main components, namely sentence encoder and paragraph encoder. The sentence encoder is designed to encode legal sentences (i.e. articles and queries) into vectors. The paragraph encoder aggregates the signal from the sentence encoder to obtain the final representation. Finally, this representation is used to calculate the relevance between the query and the candidate article (paragraph).

To build the training data, we apply a negative sampling paradigm. With each query, along with the P positive articles given by the ground truth, we sample N negative articles from the corpus. The model needs to predict the labels of each candidate in the set of $P + N$ articles. In making training data for Attentive CNN, we combine both negative sampling using lexical matching and random negative sampling. For Paraformer, we only sample negative candidates with high lexical overlapping with the query.

In the remaining part of this section, we introduce the detailed architecture of Attentive CNN and Paraformer and the way to train them to rank candidates given a query. Considering that query has important information for the model to interpret the candidates in an appropriate aspect, in both designs, we inject the representation of the query as an input to construct the final article representation.

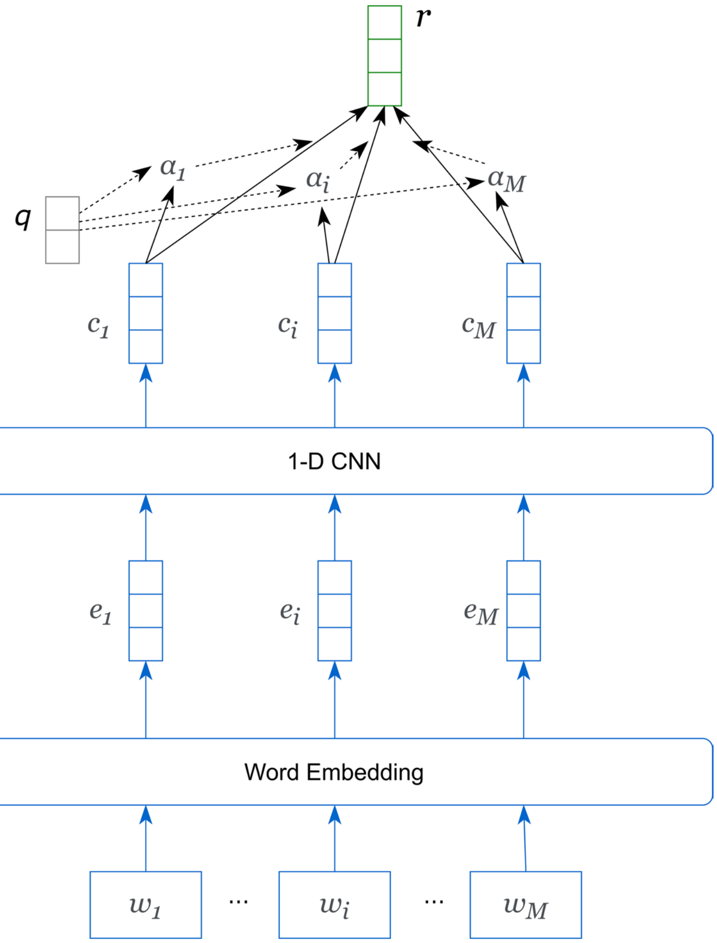


Fig. 7 Sentence encoder component in Attentive CNN architecture

4.2 Attentive CNN

4.2.1 Sentence encoder

Figure 7 shows the architecture of our sentence encoder component in Attentive CNN. This component contains three layers: word embedding, convolution, and attention layers. With M be the length of the input, word embedding is a mapping matrix from the index of the words $(w_1, w_2, \dots, w_M)$ into corresponding vectors $(e_1, e_2, \dots, e_M)$. The convolution layer aggregates the outputs of word embeddings to produce a more abstract vector c_i for each position i in the input considering the context formed by the surrounding words (e.g. “river *bank*” should be distinguished from “financial *bank*”).

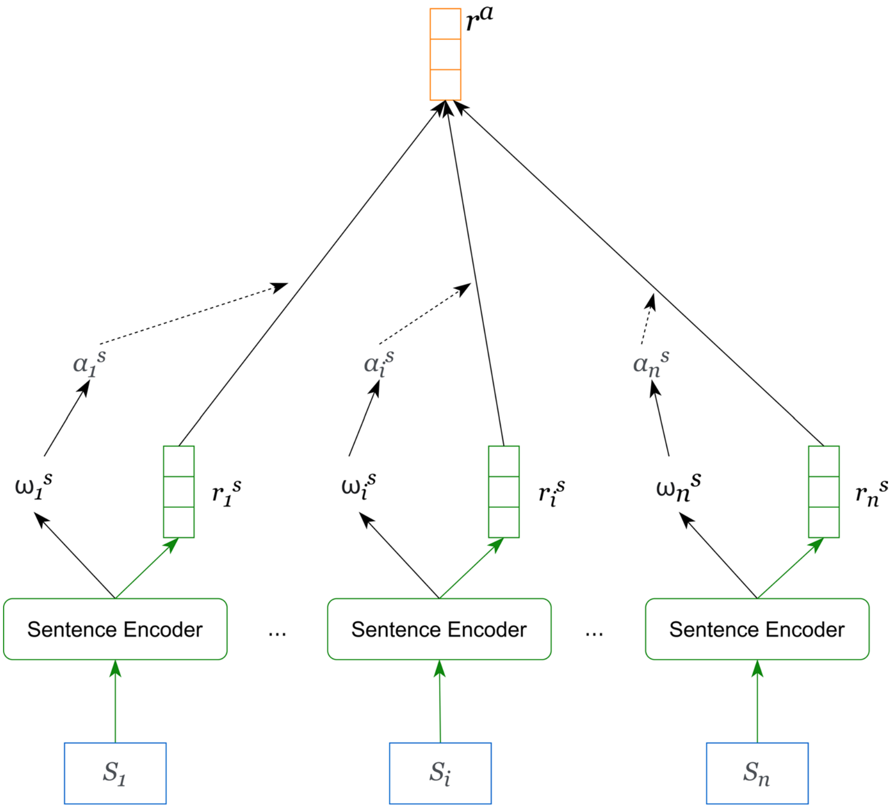


Fig. 8 Paragraph encoder component in Attentive CNN architecture

With $e_{(i-K):(i+K)}$ be the vector at the positions from $(i-K)$ to $(i+K)$, $F \in \mathbb{R}^{N_f \times (2K+1)D}$ and $b_t \in \mathbb{R}^{N_f}$ be the kernel and the bias of the convolutional layer, N_f be the number of filters, $2K + 1$ be the window size, D be the vector dimension, the formula calculates the context c_i of the word i is as in Eq. 1.

$$c_i = \mathrm{ReLU}\left(F \times e_{(i-K):(i+K)}\right) + b_t \tag{1}$$

The attention layer is designed to calculate how important each word contributes to answering a given query. Let q be the attention query vector, attention weight a_i and normalized attention weight α_i of the word i are calculated by Eqs. 2 and 3 with V and v be the weight matrix and the bias value.

$$a_i = q^T \tanh\left(V \times c_i + v\right) \tag{2}$$

$$\alpha_i = \frac{\exp\left(a_i\right)}{\sum_{j=1}^{M} \exp\left(a_j\right)} \tag{3}$$

The final representation vector r is the weighted sum of c_i, as follows:

$$r = \sum_{i=1}^{M} \alpha_i c_i \tag{4}$$

4.2.2 Paragraph encoder

An article in a legal document is often presented in a paragraph (i.e. a set of sentences). We design a module called *paragraph encoder* whose architecture is demonstrated in Fig. 8. This architecture shows the *divide-and-conquer* paradigm idea as presented. Instead of using a language model to directly encode an article, we encode each sentence of it and combine the signals via a global attention mechanism.

In designing this component, we have an important observation about the semantic contribution in a legal paragraph. No single sentence represents the whole meaning of the paragraph and each sentence contributes an amount of semantics differently to the entire semantics. We can recognize this phenomenon by reading the example given in Table 7. Only several sentences in the highlighted parts contribute most to the necessary information to answer the query. Other parts are not much relevant and may be used to answer other queries. For that reason, we propose to apply sparsemax (Martins and Astudillo 2016) to aggregate the signal from each sentence. If we use a softmax or an average function in this case, the required signal may be incomplete or diluted.

The representation vector r^a of a paragraph is calculated by Eqs. 5, 6, and 7. Let $|s|$ be the number of words in the sentence s, the attention weight ω^s is the average value of the attention weights of the words belonging to that sentence as in Eq. 5.

$$\omega^s = \frac{\sum_i a_i^w}{|s|} \tag{5}$$

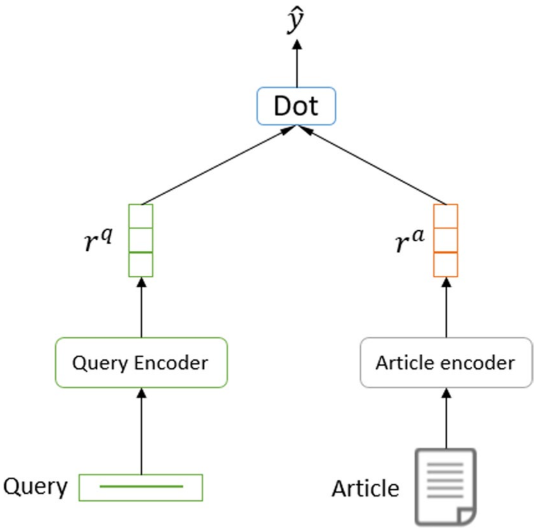
The normalized attention weight α_j^s and the final representation r^a are calculated as in Eqs. 6 and 7 with N being the number of sentences in the paragraph, ω_j^s and r_j^s be the original attention weight and the representation vector of the j^{th} sentence in the paragraph. Sparsemax function (Martins and Astudillo 2016) produces the Euclidean projection of the input vector ω_j^s onto the probability simplex.

$$\alpha_j^s = \text{sparsemax}\left(\omega_j^s\right) \tag{6}$$

$$r^a = \sum_{j=1}^{N} \alpha_j^s r_j^s \tag{7}$$

With the proposed approach, the system learns to focus on the important parts and ignore other irrelevant ones. Besides, with the ability to highlight the important sentences in a lengthy article, the system can benefit the real user experience in its application.

Fig. 9 Training attentive CNN as a similarity function



4.2.3 Model training

We assign the components proposed above as backbones in our Attentive CNN architecture as demonstrated in Fig. 9 and train them using the negative sampling paradigm. In this approach, we encode the query and the article using the sentence encoder component and the paragraph encoder component to get corresponding representation vectors. We then use dot product between the two vectors as the similarity score.

We normalize the similarity score as in Eq. 8. Given a query q, $\hat{y}_i^+$ is the probability that the article i related to q, $\hat{y}_{i,j}^-$ is such probability that the article j in the negative set of the article i related to q, and K is the number of articles in the sampled negative set.

$$p_i = \frac{\exp\left(\hat{y}_i^+\right)}{\exp\left(\hat{y}_i^+\right) + \sum_{j=1}^{K} \exp\left(\hat{y}_{i,j}^-\right)} \tag{8}$$

4.3 Paraformer

4.3.1 Sentence encoder

Attentive CNN's sentence encoder can work effectively with a sufficient amount of data (Kien et al. 2020). However, like other training-from-scratch approaches, this component may struggle with problems with small amounts of data. We confirm this issue in Sect. 5. For the problem with limited data, this component shows severely reduced performance. For that reason, we propose to replace this component with

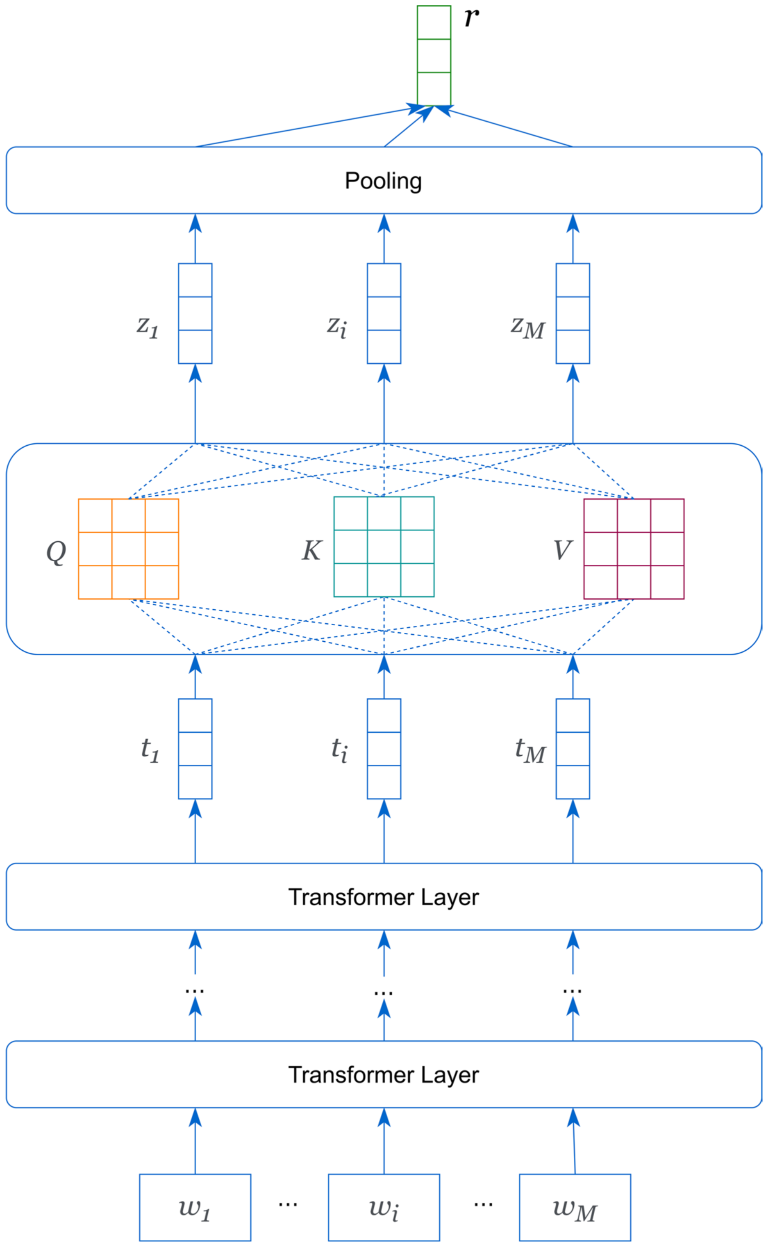


Fig. 10 Sentence encoder component in paraformer architecture

a pretrained language model. As in Fig. 10, the signal of an *M*-token input is transformed using the self-attention mechanism through the transformer layers. After that, the vectors in the final transformer layer are fed through a pooling layer to obtain a sentence-level representation vector.

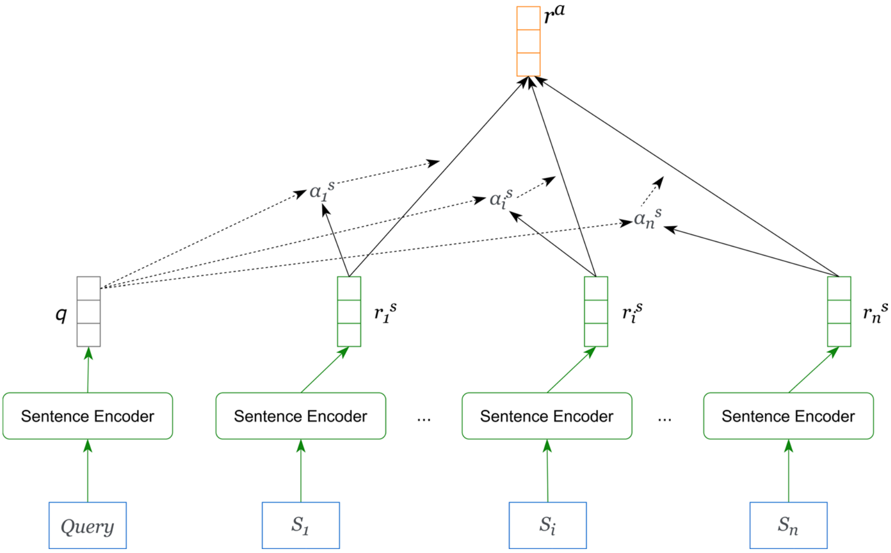


Fig. 11 Paragraph encoder component in paraformer architecture

4.3.2 Paragraph encoder

Unlike the paragraph encoder of Attentive CNN, the paragraph encoder of Paraformer incorporates query information with sentences in the article based on general attention, as in Fig. 11.

We first produce sentence-level representations of the query (q), and n sentences in an article ($r_1^s - r_n^s$) with the sentence encoder component. Then, with general attention, the representation of an article for the given query is calculated by Eqs. 9, 10, and 11, with A being the weight matrix, b being the bias value.

$$a_i^s = q^T \tanh\left(A \times r_i^s + b\right) \tag{9}$$

$$\alpha_i^s = \text{sparsemax}\left(a_i^s\right) \tag{10}$$

$$r^a = \sum_{i=1}^{M} \alpha_i^s r_i^s \tag{11}$$

4.3.3 Model training

As described in the design of this architecture, the sentence encoder is the unit component of the paragraph encoder. In addition, this unit, which contains multi-head attention layers, is already pretrained with a large amount of data. With Paraformer,

Table 1 Value of parameters in Attentive CNN

Parameter	Value
Size of word embedding layer	512
Number of CNN filter	512
Size of attention query vector	200
Dropout rate	0.2

Table 2 Value of parameters in Paraformer

Parameter	Value
Max position embeddings	514
Hidden size	768
Hidden layers	12
Attention heads	12
Dropout rate	0.1

we put one fully connected layer on top of the paragraph encoder and treat the whole model as a binary classifier. We also use the cross-entropy loss in this approach. Training this model is essentially updating the weights of global attention and fine-tuning the pretrained weights for a similarity prediction problem. In the inference phase, we extract the logit value from the fully connected layer as the ranking score of this model.

5 Experiments

5.1 Experimental settings

The experiments are conducted with COLIEE's datasets and the Vietnamese dataset introduced in Sect. 3. In the Vietnamese dataset, we used 90% of the query set for training and validation, and the test set is 10%. For English and Japanese, we use COLIEE 2021 data with the same train/test division as in the official competition. We compare Attentive CNN, Paraformer and the vanilla XLM-RoBERTa, which is a strong multilingual pretrained baseline. On the English dataset, we also experiment with BERT-PLI (Shao et al. 2020), a very successful model for English legal retrieval of common law (Task 1, 2 of COLIEE 2019).

The Attentive CNN is trained from scratch, so it can perform in all three languages. The size of the vocabulary of this model is 31,450. For the backbone of Paraformer's sentence encoder, among pretrained models provided by Reimers and Gurevych (2019), we choose *paraphrase-xlm-r-multilingual-v1* for the multilingual version (including Japanese and Vietnamese), and *paraphrase-mpnet-base-v2* for the English version. The size of the vocabulary in the English version is 30,527 and in the multilingual version is 250,002. Tables 1 and 2 indicate the parameters of our two models (i.e. Attentive CNN and Paraformer). For BERT-PLI, we also finetune

Table 3 Performance of the systems without using the lexical score ($\alpha = 1$). The bold values indicate best performance

Systems	Precision	Recall	F2
English dataset			
Paraformer	**0.3827**	**0.3450**	**0.3498**
XLM-RoBERTa	0.2099	0.1975	0.1989
BERT-PLI	0.1728	0.1543	0.1564
Attentive CNN	0.0864	0.0864	0.0864
Japanese Dataset			
Paraformer	**0.3457**	**0.3148**	**0.3182**
XLM-RoBERTa	0.2940	0.3086	0.3086
Attentive CNN	0.2593	0.2222	0.2263

this model with case law entailment data as suggested by the authors before training the model on article retrieval data. Before conducting the experiment, we did not expect a model designed for the document level of case law to work well at the article level of statute law.

For all systems, we retrieve the articles in two stages: lexical matching and reranking. In the lexical matching stage, for the Vietnamese dataset, because of the huge number of articles, we use ElasticSearch[12] and for English and Japanese datasets, we use a lightweight python package Rank-BM25.[13]

In the reranking stage, we rank the articles using the final score calculated in Eq. 12.

$$S_{final} = \alpha \cdot S_{deep} + (1 - \alpha) \cdot S_{lexical} \tag{12}$$

where lexical score $S_{lexical}$ is obtained from the lexical matching system, and the semantic score S_{deep} is given by the deep learning model. $\alpha \in [0, 1]$, which can be tuned using hyperparameter tuning techniques, determines the weight between the two scores.

We use the same metrics with COLIEE 2021, in which Macro-F2 at top 1 is the main metric to measure the performance of retrieval systems. We also consider Precision and Recall scores for the analysis purpose.

5.2 Experimental results on COLIEE datasets

COLIEE datasets have been used by many research groups. This helps us better validate our methods and compare them with already presented systems. We conduct the experiment in two phases. At first, we compare different deep learning candidates' performances on the datasets without the support of BM25 (i.e. $\alpha = 1$). After that, we apply a grid search optimization to our best candidate to know the highest performance our method can achieve.

[12] https://www.elastic.co/.

[13] https://pypi.org/project/rank-bm25/.

Table 4 Performance of Paraformer* compared with other competitors on COLIEE 2021's official test. The bold values indicate best performance

Run ID	Precision	Recall	F2
Paraformer*	**0.7901**	0.7346	**0.7407**
OvGU (Wehnert et al. 2021)	0.6749	0.7778	0.7302
JNLP (Nguyen et al. 2021)	0.6000	**0.8025**	0.7227
UA (Kim et al. 2022)	0.7531	0.7037	0.7092
TR (Frank et al. 2021)	0.3333	0.6173	0.5226
HUKB (Masaharu et al. 2021)	0.2901	0.6975	0.5224

Table 5 Experimental results on Vietnamese Dataset on top-1 article. The bold values indicate best performance

Systems	Precision	Recall	F2
BM25	0.2395	0.1966	0.2006
XLM-RoBERTa	0.2395	0.1966	0.2006
Attentive CNN	0.5919	0.4660	0.4774
Paraformer	**0.5987**	**0.4769**	**0.4882**

The first phase's results are shown in Table 3.

Paraformer achieves state-of-the-art results in both languages. BERT-PLI, a model proposed for case law retrieval, surprised us with significantly better performance than Attentive CNN on the English dataset. This can be explained by the ability of the deep learning models in transferring knowledge between similar data domains.

From this result, we can observe that pretrained models may be able to overcome situations in which data is not abundant.

Next, we tune the model to reach the optimal configurations in COLIEE 2021's formal dataset. In the first phase, Paraformer achieves state-of-the-art results on the English dataset. We choose this model as the deep learning component to combine with BM25 in the optimized reranking phase. In this paper, the full table of grid-search can be found in Appendix 2.

Table 4 shows the performance of our final system (i.e. *Paraformer**) compared to the state-of-the-art approaches from different teams in COLIEE 2021. *Paraformer** obtains state-of-the-art performance on Precision and Macro-F2. The best Recall performance belongs to the systems of Nguyen et al. (2021) and Wehnert et al. (2021). It could be room for future improvement.

5.3 Experimental results on Vietnamese dataset

Vietnamese dataset is larger than the COLIEE's datasets. Conducting an experiment on this dataset allows us to understand more about the behavior of the models. In this dataset, we compare 4 candidates as follows:

- *BM25*: A well-known retrieval system using only the lexical features.

Table 6 Length in characters of Vietnamese, English and Japanese test sets

Dataset	Query length			Article length		
	Min	Max	Avg.	Min	Max	Avg.
Vietnamese	20	182	78	53	252,955	10,941
English	60	379	214	203	1891	742
Japanese	21	219	90	58	550	224

- *XLM-RoBERTa*: Transformer-based model pretrained on a multilingual dataset in 100 languages (Conneau et al. 2019) including English, Japanese and Vietnamese.
- *Attentive CNN*: The convolutional neural network with the global attention mechanism.
- *Paraformer*: Our novel proposed system taking advantage of the pretrained language model and the global attention.

Table 5 shows the experimental results on the Vietnamese dataset. As we can see in the table, XLM-RoBERTa contributes no significant improvement compared to BM25 in Macro-F2 (0.2006). Our Attentive CNN and Paraformer lead the ranking, Paraformer (0.4882) slightly outperforms Attentive CNN (0.4774) by about 1%. In our experiments, because of computation complexity, the number of articles filtered by lexical matching *N* for Paraformer (from 10 to 150 articles) is significantly smaller than for the Attentive CNN (from 300 to 2000 articles). Curious about this difference, we further measure the performance on the top 20 articles retrieved by the two models, Attentive CNN achieves 0.2220 in Macro-F2@20 and 0.5849 in NDCG@20 while Paraformer achieves only 0.1839 and 0.4464, respectively. This suggests that, for searching many results over a large search space, Attentive CNN might be a more suitable approach.

Despite being a pretrained model, XLM-RoBERTa performs badly in the Vietnamese dataset. Analyzing the dataset, we see that the average length of Vietnamese legal sentences is significantly longer than English and Japanese sentences. In addition, concatenating the query and articles to construct the input for the system makes more burden on this model. Even a powerful model can perform badly if they do not have full information for inference. This strengthens the usefulness of the models proposed in this paper with the idea of *divide-and-conquer*.

5.4 Further discussions

5.4.1 Impact of content length

Table 6 indicates the length in characters of the Vietnamese, English and Japanese testing sets. Note that, since each model has a different way of tokenizing input sentences, in this paper, we use the number of characters as a common unit to measure

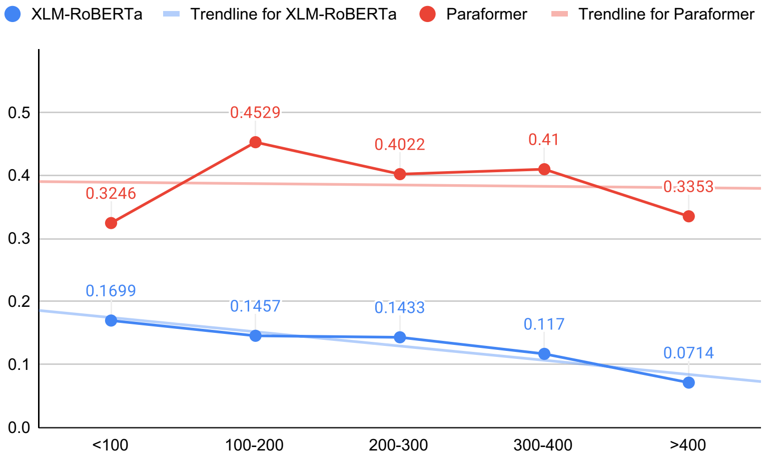
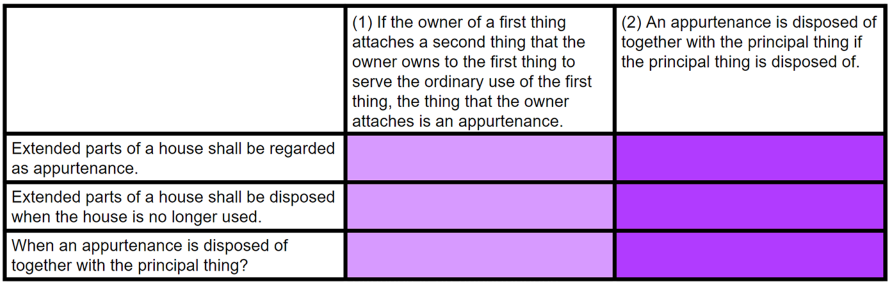
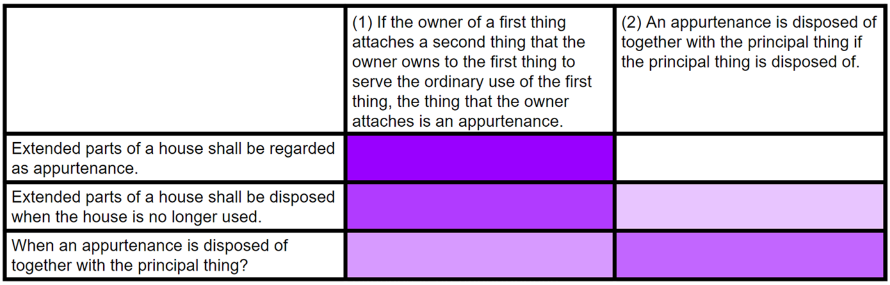


Fig. 12 Performance of XLM-RoBERTa and Paraformer when working on different lengths of queries. The x-axis represents the length of the query chunk in characters, the y-axis represents the performance of the models in Macro F2



	(1) If the owner of a first thing attaches a second thing that the owner owns to the first thing to serve the ordinary use of the first thing, the thing that the owner attaches is an appurtenance.	(2) An appurtenance is disposed of together with the principal thing if the principal thing is disposed of.
Extended parts of a house shall be regarded as appurtenance.	▒	█
Extended parts of a house shall be disposed when the house is no longer used.	▒	█
When an appurtenance is disposed of together with the principal thing?	▒	█

Fig. 13 Weight visualization of Attentive CNN for the example in Sect. 1. The more important the content, the darker the color



	(1) If the owner of a first thing attaches a second thing that the owner owns to the first thing to serve the ordinary use of the first thing, the thing that the owner attaches is an appurtenance.	(2) An appurtenance is disposed of together with the principal thing if the principal thing is disposed of.
Extended parts of a house shall be regarded as appurtenance.	██	
Extended parts of a house shall be disposed when the house is no longer used.	█	░
When an appurtenance is disposed of together with the principal thing?	▒	▓

Fig. 14 Weight visualization of Paraformer for the example in Sect. 1. The more important the content, the darker the color

the length of samples. In the Vietnamese dataset, the length of articles varies greatly, the longest article is about 250K characters, the shortest article is 53 characters. The pretrained models have a limit of 514 tokens. This creates a significant challenge for vanilla XLM-RoBERTa with the approach of treating an entire article as a sentence. Looking at Tables 3, 5 and 6, we have the observation that XLM-RoBERTa may obtain poor results with too lengthy articles.

Figure 12 shows the performance of the XLM-RoBERTa and Paraformer along with their trendlines on different chunks of query length in the English dataset. It can be seen that the longer the query, the worse the performance of both models. However, we can see that Paraformer is the winner in all chunks and its trendline reduces slower.

5.4.2 Global attention visualization

Although sharing a common *divide-and-conquer* idea with Attentive CNN, the architecture of Paraformer allows us to represent the relevance between the queries and the articles more flexibly. After being trained, while the Attentive CNN generates only one article representation regardless of the query, Paraformer's paragraph encoder allows us to derive information about the relevance between queries and each sentence in an article through its attention weights. Figures 13 and 14 demonstrate the attention weights of Attentive CNN and Paraformer for the same example mentioned in Sect. 1.

As we can see in the figure, Paraformer focuses differently on the contents of Article 87 depending on the given query while Attentive CNN produces the same attention weights for all queries. This also opens up interesting research directions in explainable AI where we can debug what information the models are paying attention to instead of accepting their results as black-box output.

6 Conclusions

In this paper, we investigate and solve the problem of information retrieval for the legal domain by using deep learning models with the attention mechanism to represent the query and article for the ranking purpose. The general idea of our approach, *divide-and-conquer*, is to break down articles to represent them individually and then combine them back using global attention. We propose two new architectures named Attentive CNN and Paraformer based on this idea. In our experiment, we demonstrate the effectiveness of this method compared to strong baselines in reliable legal datasets in three different languages, i.e., English, Japanese, and Vietnamese. We also analyze the strengths and weaknesses of each model with each specific data condition for a clear insight in designing the models for this problem. In addition, our large Vietnamese dataset for this problem enables us to perform detailed analysis as well as to contribute to the research community. In future work, we intend to extend this work by introducing more legal domain-specific pretrained methods for this architecture.

Appendix 1 Data examples

See Tables 7, 8, 9.

Table 7 A sample in the Vietnamese dataset with highlighted parts

Question	Con riêng có được hưởng di sản thừa kế của người cha đã mất khi không để lại di chúc không?
Answer	Article 651 from the Code of Civil law of Vietnam (2015)
Article content	Điều 651. ***Người thừa kế theo pháp luật*** ***1. Những người thừa kế theo pháp luật được quy định theo thứ tự sau đây:*** ***(a) Hàng thừa kế thứ nhất gồm: vợ, chồng, cha đẻ, mẹ đẻ, cha nuôi, mẹ nuôi, con đẻ, con nuôi của người chết;*** ***(b) Hàng thừa kế thứ hai gồm: ông nội, bà nội, ông ngoại, bà ngoại, anh ruột, chị ruột, em ruột của người chết; cháu ruột của người chết mà người chết là ông nội, bà nội, ông ngoại, bà ngoại;*** ***(c) Hàng thừa kế thứ ba gồm: cụ nội, cụ ngoại của người chết; bác ruột, chú ruột, cậu ruột, cô ruột, dì ruột của người chết; cháu ruột của người chết mà người chết là bác ruột, chú ruột, cậu ruột, cô ruột, dì ruột; chắt ruột của người chết mà người chết là cụ nội, cụ ngoại.*** ***2. Những người thừa kế cùng hàng được hưởng phần di sản bằng nhau.*** ***3. Những người ở hàng thừa kế sau chỉ được hưởng thừa kế, nếu không còn ai ở hàng thừa kế trước do đã chết, không có quyền hưởng di sản, bị truất quyền hưởng di sản hoặc từ chối nhận di sản.***

Table 8 A sample in the Japanese dataset

Question	未成年者がした売買契約は、親権者の同意を得ないでした場合であっても、その契約が日常生活に関するものであるときは、取り消すことができない。
Answer	Article 5 from Japanese Civil Code.
Article content	第五条　未成年者が法律行為をするには、その法定代理人の同意を得なければならない。ただし、単に権利を得、又は義務を免れる法律行為については、この限りでない。 2 x前項の規定に反する法律行為は、取り消すことができる。 3 第一項の規定にかかわらず、法定代理人が目的を定めて処分を許した財産は、その目的の範囲内において、未成年者が自由に処分することができる。目的を定めないで処分を許した財産を処分するときも、同様とする。

Table 9 A sample in the English dataset

Question	A contract of guarantee concluded by a person under curatorship may not be rescinded in cases the consent of the curator is obtained
Answer	Article 13 from Japanese Civil Code
Article content	Article 13 (1) A person under curatorship must obtain the consent of the curator in order to perform any of the following acts;provided, however, that this does not apply to an act provided for in the proviso of Article 9: (i) receiving or using any property producing civil fruit (ii) borrowing money or guaranteeing an obligation; (iii) performing an act with the purpose of acquiring or losing any right regarding immovables or other significant property (iv) suing any procedural act (v) giving a gift, reaching a settlement, or entering into an arbitration agreement (meaning an arbitration agreement as provided in Article 2, paragraph (1) of the Arbitration Act (Act No. 138 of 2003)) (vi) accepting or renouncing a succession or dividing an estate (vii) refusing an offer of a gift, renouncing a legacy, accepting an offer of gift with burden, or accepting a legacy with burden (viii) constructing a new building, renovating, expanding, or undertaking major repairs (ix) granting a lease for a term that exceeds the period set forth in Article 602; or (x) performing any of the acts set forth in the preceding items as a legal representative of a person with qualified legal capacity (meaning a minor, adult ward, or person under curatorship or a person under assistance who is subject to a decision as referred to in Article 17, paragraph (1); the same applies hereinafter) (2) At the request of a person as referred to in the main clause of Article 11or the curator or curator's supervisor, the family court may decide that the person under curatorship must also obtain the consent of the curator before performing an act other than those set forth in each of the items of the preceding paragraph;provided, however, that this does not apply to an act provided for in the proviso to Article 9 (3) If the curator does not consent to an act for which the person under curatorship must obtain the curator's consent even though it is unlikely to prejudice the interests of the person under curatorship, the family court may grant permission that operates in lieu of the curator's consent at the request of the person under curatorship (4) An act for which the person under curatorship must obtain the curator's consent is voidable if the person performs it without obtaining the curator's consent or a permission that operates in lieu of it

Appendix 2 Grid search table for tuning paraformer*

α	Validation			Test		
	P	R	F2	*P*	R	F2
Top_BM25=10						
0.1	0.5077	0.4692	0.4735	0.6790	0.6481	0.6516
0.2	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.3	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.4	0.5846	0.5462	0.5504	0.6914	0.6543	0.6584
0.5	0.6000	0.5615	0.5658	0.6914	0.6543	0.6584
0.6	0.6308	0.5923	0.5966	0.7160	0.6790	0.6831
0.7	0.6462	0.6000	0.6051	0.7531	0.7099	0.7147
0.8	0.6154	0.5692	0.5744	0.7654	0.7160	0.7215
0.9	0.6154	0.5615	0.5675	0.7901	0.7346	0.7407
1.0	0.5231	0.4462	0.4547	0.3827	0.3457	0.3498
Top_BM25=20						
0.1	0.5077	0.4692	0.4735	0.6790	0.6481	0.6516
0.2	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.3	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.4	0.5846	0.5462	0.5504	0.6914	0.6543	0.6584
0.5	0.6000	0.5615	0.5658	0.6914	0.6543	0.6584
0.6	0.6308	0.5923	0.5966	0.7160	0.6790	0.6831
0.7	0.6462	0.6000	0.6051	0.7654	0.7222	0.7270
0.8	0.6154	0.5692	0.5744	0.7778	0.7284	0.7339
0.9	0.5846	0.5385	0.5436	0.7654	0.7160	0.7215
1.0	0.4154	0.3462	0.3538	0.2840	0.2593	0.2620
Top_BM25=30						
0.1	0.5077	0.4692	0.4735	0.6790	0.6481	0.6516
0.2	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.3	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.4	0.5846	0.5462	0.5504	0.6914	0.6543	0.6584
0.5	0.6000	0.5615	0.5658	0.6914	0.6543	0.6584
0.6	0.6308	0.5923	0.5966	0.7160	0.6790	0.6831
0.7	0.6462	0.6000	0.6051	0.7654	0.7222	0.7270
0.8	0.6154	0.5692	0.5744	0.7778	0.7284	0.7339
0.9	0.5692	0.5308	0.5350	0.7654	0.7160	0.7215
1.0	0.3077	0.2538	0.2598	0.1605	0.1543	0.1550
0.1	0.5077	0.4692	0.4735	0.6790	0.6481	0.6516
0.2	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.3	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.4	0.5846	0.5462	0.5504	0.6914	0.6543	0.6584
0.5	0.6000	0.5615	0.5658	0.6914	0.6543	0.6584
0.6	0.6308	0.5923	0.5966	0.7160	0.6790	0.6831

α	Validation			Test		
	P	R	F2	*P*	R	F2
0.7	0.6462	0.6000	0.6051	0.7654	0.7222	0.7270
0.8	0.6154	0.5692	0.5744	0.7778	0.7284	0.7339
0.9	0.5692	0.5205	0.5256	0.7778	0.7284	0.7339
1.0	0.2308	0.1821	0.1871	0.1481	0.1420	0.1427
Top_BM25=50						
0.1	0.5077	0.4692	0.4735	0.6790	0.6481	0.6516
0.2	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.3	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.4	0.5846	0.5462	0.5504	0.6914	0.6543	0.6584
0.5	0.6000	0.5615	0.5658	0.6914	0.6543	0.6584
0.6	0.6308	0.5923	0.5966	0.7160	0.6790	0.6831
0.7	0.6462	0.6000	0.6051	0.7654	0.7222	0.7270
0.8	0.6154	0.5692	0.5744	0.7778	0.7284	0.7339
0.9	0.5692	0.5205	0.5256	0.7778	0.7284	0.7339
1.0	0.2462	0.1974	0.2025	0.1481	0.1420	0.1427
Top_BM25=60						
0.1	0.5077	0.4692	0.4735	0.6790	0.6481	0.6516
0.2	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.3	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.4	0.5846	0.5462	0.5504	0.6914	0.6543	0.6584
0.5	0.6000	0.5615	0.5658	0.6914	0.6543	0.6584
0.6	0.6308	0.5923	0.5966	0.7160	0.6790	0.6831
0.7	0.6462	0.6000	0.6051	0.7654	0.7222	0.7270
0.8	0.6154	0.5692	0.5744	0.7654	0.7160	0.7215
0.9	0.5692	0.5205	0.5256	0.7778	0.7284	0.7339
1.0	0.2308	0.1821	0.1871	0.1358	0.1296	0.1303
Top_BM25=70						
0.1	0.5077	0.4692	0.4735	0.6790	0.6481	0.6516
0.2	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.3	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.4	0.5846	0.5462	0.5504	0.6914	0.6543	0.6584
0.5	0.6000	0.5615	0.5658	0.6914	0.6543	0.6584
0.6	0.6308	0.5923	0.5966	0.7160	0.6790	0.6831
0.7	0.6462	0.6000	0.6051	0.7654	0.7222	0.7270
0.8	0.6154	0.5692	0.5744	0.7654	0.7160	0.7215
0.9	0.5692	0.5205	0.5256	0.7778	0.7284	0.7339
1.0	0.2154	0.1846	0.1880	0.1358	0.1296	0.1303
Top_BM25=80						
0.1	0.5077	0.4692	0.4735	0.6790	0.6481	0.6516
0.2	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.3	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.4	0.5846	0.5462	0.5504	0.6914	0.6543	0.6584

α	Validation			Test		
	P	R	F2	*P*	R	F2
0.5	0.6000	0.5615	0.5658	0.6914	0.6543	0.6584
0.6	0.6308	0.5923	0.5966	0.7160	0.6790	0.6831
0.7	0.6462	0.6000	0.6051	0.7654	0.7222	0.7270
0.8	0.6154	0.5692	0.5744	0.7654	0.7160	0.7215
0.9	0.5692	0.5205	0.5256	0.7778	0.7284	0.7339
1.0	0.2000	0.1692	0.1726	0.1111	0.1049	0.1056
Top_BM25=90						
0.1	0.5077	0.4692	0.4735	0.6790	0.6481	0.6516
0.2	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.3	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.4	0.5846	0.5462	0.5504	0.6914	0.6543	0.6584
0.5	0.6000	0.5615	0.5658	0.6914	0.6543	0.6584
0.6	0.6308	0.5923	0.5966	0.7160	0.6790	0.6831
0.7	0.6462	0.6000	0.6051	0.7654	0.7222	0.7270
0.8	0.6154	0.5692	0.5744	0.7654	0.7160	0.7215
0.9	0.5692	0.5205	0.5256	0.7778	0.7284	0.7339
1.0	0.1538	0.1308	0.1333	0.1111	0.1049	0.1056
0.1	0.5077	0.4692	0.4735	0.6790	0.6481	0.6516
0.2	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.3	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.4	0.5846	0.5462	0.5504	0.6914	0.6543	0.6584
0.5	0.6000	0.5615	0.5658	0.6914	0.6543	0.6584
0.6	0.6308	0.5923	0.5966	0.7160	0.6790	0.6831
0.7	0.6462	0.6000	0.6051	0.7654	0.7222	0.7270
0.8	0.6154	0.5692	0.5744	0.7654	0.7160	0.7215
0.9	0.5692	0.5205	0.5256	0.7654	0.7160	0.7215
1.0	0.1385	0.1231	0.1248	0.0988	0.0926	0.0933
Top_BM25=110						
0.1	0.5077	0.4692	0.4735	0.6790	0.6481	0.6516
0.2	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.3	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.4	0.5846	0.5462	0.5504	0.6914	0.6543	0.6584
0.5	0.6000	0.5615	0.5658	0.6914	0.6543	0.6584
0.6	0.6308	0.5923	0.5966	0.7160	0.6790	0.6831
0.7	0.6462	0.6000	0.6051	0.7654	0.7222	0.7270
0.8	0.6154	0.5692	0.5744	0.7654	0.7160	0.7215
0.9	0.5692	0.5205	0.5256	0.7654	0.7160	0.7215
1.0	0.1385	0.1231	0.1248	0.0741	0.0679	0.0686
Top_BM25=120						
0.1	0.5077	0.4692	0.4735	0.6790	0.6481	0.6516
0.2	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.3	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516

α	Validation			Test		
	P	R	F2	*P*	R	F2
0.4	0.5846	0.5462	0.5504	0.6914	0.6543	0.6584
0.5	0.6000	0.5615	0.5658	0.6914	0.6543	0.6584
0.6	0.6308	0.5923	0.5966	0.7160	0.6790	0.6831
0.7	0.6462	0.6000	0.6051	0.7654	0.7222	0.7270
0.8	0.6154	0.5692	0.5744	0.7654	0.7160	0.7215
0.9	0.5692	0.5205	0.5256	0.7654	0.7160	0.7215
1.0	0.1231	0.1154	0.1162	0.0741	0.0679	0.0686
0.1	0.5077	0.4692	0.4735	0.6790	0.6481	0.6516
0.2	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.3	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.4	0.5846	0.5462	0.5504	0.6914	0.6543	0.6584
0.5	0.6000	0.5615	0.5658	0.6914	0.6543	0.6584
0.6	0.6308	0.5923	0.5966	0.7160	0.6790	0.6831
0.7	0.6462	0.6000	0.6051	0.7654	0.7222	0.7270
0.8	0.6154	0.5692	0.5744	0.7654	0.7160	0.7215
0.9	0.5538	0.5154	0.5197	0.7654	0.7160	0.7215
1.0	0.1231	0.1154	0.1162	0.0741	0.0679	0.0686
Top_BM25=140						
0.1	0.5077	0.4692	0.4735	0.6790	0.6481	0.6516
0.2	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.3	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.4	0.5846	0.5462	0.5504	0.6914	0.6543	0.6584
0.5	0.6000	0.5615	0.5658	0.6914	0.6543	0.6584
0.6	0.6308	0.5923	0.5966	0.7160	0.6790	0.6831
0.7	0.6462	0.6000	0.6051	0.7654	0.7222	0.7270
0.8	0.6154	0.5692	0.5744	0.7654	0.7160	0.7215
0.9	0.5538	0.5154	0.5197	0.7654	0.7160	0.7215
1.0	0.1231	0.1154	0.1162	0.0741	0.0679	0.0686
Top_BM25=150						
0.1	0.5077	0.4692	0.4735	0.6790	0.6481	0.6516
0.2	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.3	0.5231	0.4846	0.4889	0.6790	0.6481	0.6516
0.4	0.5846	0.5462	0.5504	0.6914	0.6543	0.6584
0.5	0.6000	0.5615	0.5658	0.6914	0.6543	0.6584
0.6	0.6308	0.5923	0.5966	0.7160	0.6790	0.6831
0.7	0.6462	0.6000	0.6051	0.7654	0.7222	0.7270
0.8	0.6154	0.5692	0.5744	0.7654	0.7160	0.7215
0.9	0.5538	0.5154	0.5197	0.7654	0.7160	0.7215
1.0	0.1231	0.1154	0.1162	0.0741	0.0679	0.0686

Acknowledgements This work was supported by JSPS Kakenhi Grant Number 20K20406. The research also was supported in part by the Asian Office of Aerospace R &D(AOARD), AirForce Office of Scientific Research (Grant No. FA2386-19-1-4041). The work would not be complete without valuable data from COLIEE.

References

Bach NX, Duy TK, Phuong TM (2019) A POS tagging model for Vietnamese social media text using BiLSTM-CRF with rich features. In: Proceedings of the 16th pacific rim international conference on artificial intelligence (pricai), part iii, pp 206–219

Bach NX, Thuy NTT, Chien DB, Duy TK, Hien TM, Phuong TM (2019) Reference extraction from Vietnamese legal documents. In: Proceedings of the 10th international symposium on information and communication technology (soict), pp 486–493

Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P et al. (2020). Language models are few-shot learners. arXiv:2005.14165

Chalkidis I, Kampas D (2019) Deep learning in law: early adaptation and legal word embeddings trained on large corpora. Artif Intell Law 27(2):171–198

Chen Q, Zhu X, Ling ZH, Wei S, Jiang H, Inkpen D (2017) Enhanced lstm for natural language inference. In: Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers), pp 1657–1668

Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, Stoyanov V (2019) Unsupervised cross-lingual representation learning at scale. arXiv:1911.02116

Cooper WS (1971) A definition of relevance for information retrieval. Inf Storage Retr 7(1):19–37

Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), pp 4171–4186. Minneapolis, Minnesota Association for Computational Linguistics

Frank S, Dhivya C, Kanika M, Jinane H, Andrew V, Hiroko B, John H (2021) A pentapus grapples with legal reasoning. Coliee workshop in icail, pp 78–83

Huang PS, He X, Gao J, Deng L, Acero A, Heck L (2013) Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd acm international conference on information & knowledge management, pp 2333–2338

Husa VJM (2016) Future of legal families. Oxford handbooks online: scholarly research reviews. Oxford University Press, Oxford

Ito S (2008) Lecture series on ultimate facts. Shojihomu (in Japanese)

Kien PM, Nguyen HT, Bach NX, Tran V, Nguyen ML, Phuong TM (2020) Answering legal questions by learning neural attentive text representation. In: Proceedings of the 28th international conference on computational linguistics. Barcelona, Spain (Online) International Committee on Computational Linguistics, pp 988–998. https://aclanthology.org/2020.coling-main.86https://doi.org/10.18653/v1/2020.coling-main.86

Kim MY, Rabelo J, Okeke K, Goebel R (2022) Legal information retrieval and entailment based on bm25, transformer and semantic thesaurus methods. Rev. Socionetw. Strateg. 16(1):157–174

Kim Y (2014) Convolutional neural networks for sentence classification. In: Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp), pp 1746–1751

Kowalski R, Datoo A (2021) Logical english meets legal english for swaps and derivatives. Artif Intell Law 30:163–197

Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, Zettlemoyer L (2019) Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv:1910.13461

Luhn HP (1957) A statistical approach to mechanized encoding and searching of literary information. IBM J Res Dev 1(4):309–317

Martins A, Astudillo R (2016) From softmax to sparsemax: a sparse model of attention and multi-label classification. International conference on machine learning, pp 1614–1623

Masaharu Y, Youta S, Yasuhiro A (2021) Bert-based ensemble methods for information retrieval and legal textual entailment in coliee statute law task. Coliee workshop in icail, pp 78–83

Mikolov T, Grave E, Bojanowski P, Puhrsch C, Joulin A (2018) Advances in pre-training distributed word representations. In: Proceedings of the international conference on language resources and evaluation (lrec 2018)

Mikolov T, Kombrink S, Burget L, Černockỳ J, Khudanpur S (2011) Extensions of recurrent neural network language model. 2011 ieee international conference on acoustics, speech and signal processing (icassp), pp 5528–5531

Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems, 26. https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html

Mueller J, Thyagarajan A (2016) Siamese recurrent architectures for learning sentence similarity. In: thirtieth aaai conference on artificial intelligence

Nguyen HT, Nguyen PM, Vuong THY, Bui QM, Nguyen CM, Dang BT, Satoh K (2021) Jnlp team: deep learning approaches for legal processing tasks in coliee 2021. arXiv:2106.13405

Nguyen HT, Nguyen VH, Vu VA (2017) A knowledge representation for vietnamese legal document system. In: 2017 9th international conference on knowledge and systems engineering (kse), pp 30–35

Nguyen HT, Tran V, Nguyen PM, Vuong THY, Bui QM, Nguyen CM, Satoh K (2021) Paralaw nets–cross-lingual sentence-level pretraining for legal text processing. arXiv:2106.13403

Nguyen HT, Vuong HYT, Nguyen PM, Dang BT, Bui QM, Vu ST, Nguyen ML (2020). Jnlp team: deep learning for legal processing in coliee 2020. arXiv:2011.08071

Nguyen TS, Nguyen LM, Tojo S, Satoh K, Shimazu A (2018) Recurrent neural network-based models for recognizing requisite and effectuation parts in legal texts. Artif Intell Law 26(2):169–199

Palangi H, Deng L, Shen Y, Gao J, He X, Chen J, Ward R (2016) Deep sentence embedding using long short-term memory networks: analysis and application to information retrieval. IEEE/ACM Trans Audio Speech Lang Process 24(4):694–707

Pennington J, Socher R, Manning CD. (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp), pp 1532–1543

Rabelo J, Kim MY, Goebel R, Yoshioka M, Kano Y, Satoh K (2019) A summary of the coliee 2019 competition. In: Jsai international symposium on artificial intelligence, pp 34–49

Radford A, Narasimhan K, Salimans T, Sutskever I (2018) Improving language understanding by generative pre-training. The University of British Columbia Repository

Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2019) Language models are unsupervised multitask learners. OpenAI Blog 1(8):9

Reimers N, Gurevych I (2019) Sentence-bert: sentence embeddings using siamese bert-networks. arXiv:1908.10084

Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Inf Process Manag 24(5):513–523

Satoh K, Asai K, Kogawa T, Kubota M, Nakamura M, Nishigai Y, Takano C (2010) Proleg: an implementation of the presupposed ultimate fact theory of Japanese civil code by prolog technology. In: Jsai international symposium on artificial intelligence, pp 153–164

Šavelka J, Ashley KD (2021) Legal information retrieval for understanding statutory terms. Artif Intell Law 30:245–289

Severyn A, Moschitti A (2015) Learning to rank short text pairs with convolutional deep neural networks. In: Proceedings of the 38th international acm sigir conference on research and development in information retrieval, pp 373–382

Shao Y, Mao J , Liu Y, Ma W, Satoh K, Zhang M, Ma S (2020) Bert-pli: modeling paragraph-level interactions for legal case retrieval. Ijcai, pp 3501–3507

Shen Y, He X, Gao J, Deng L, Mesnil G (2014) A latent semantic model with convolutional-pooling structure for information retrieval. In: Proceedings of the 23rd acm international conference on conference on information and knowledge management, pp 101–110

Sugathadasa K, Ayesha B, de Silva N, Perera AS, Jayawardana V, Lakmal D, Perera M (2018) Legal document retrieval using document vector embeddings and deep learning. In: Science and information conference, pp 160–175

Tang D, Qin B, Liu T. (2015) Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp 1422–1432

Thanh NH, Quan BM, Nguyen C, Le T, Phuong NM, Binh DT et al. (2021) A summary of the alqac 2021 competition. In: 2021 13th international conference on knowledge and systems engineering (kse), pp 1–5

Tran V, Le Nguyen M, Tojo S, Satoh K (2020) Encoded summarization: summarizing documents into continuous vector space for legal case retrieval. Artif Intell Law 28:441–467

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L , Gomez AN, Polosukhin I (2017) Attention is all you need. Advances in Neural Information Processing Systems, 30. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Wang Y, Huang M, Zhu X, Zhao L (2016) Attention-based lstm for aspect-level sentiment classification. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp 606–615

Wehnert S, Sudhi V, Dureja S, Kutty L, Shahania S, De Luca EW (2021) Legal norm retrieval with variations of the bert model combined with tf-idf vectorization. In: Proceedings of the eighteenth international conference on artificial intelligence and law, pp 285–294

Yilmaz ZA, Wang S, Yang W, Zhang H, Lin J (2019) Applying BERT to document retrieval with birch. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp): system demonstrations, pp 19–24

Yoshioka M, Aoki Y, Suzuki Y (2021) Bert-based ensemble methods with data augmentation for legal textual entailment in coliee statute law task. In: Proceedings of the eighteenth international conference on artificial intelligence and law, pp 278–284

Yoshioka M, Kano Y, Kiyota N, Satoh K (2018) Overview of Japanese statute law retrieval and entailment task at coliee-2018. In: Twelfth international workshop on juris-informatics (jurisin 2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Ha-Thanh Nguyen[1] · Manh-Kien Phi[2] · Xuan-Bach Ngo[2] · Vu Tran[1] · Le-Minh Nguyen[1] · Minh-Phuong Tu[2]

Manh-Kien Phi
kienpm2205@gmail.com

Xuan-Bach Ngo
bachnx@ptit.edu.vn

Vu Tran
vu.tran@jaist.ac.jp

Le-Minh Nguyen
nguyenml@jaist.ac.jp

Minh-Phuong Tu
phuongtm@ptit.edu.vn

[1] School of Information Science, Japan Advanced Institute of Science and Technology, Nomi, Ishikawa, Japan

[2] Department of Computer Science, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam