IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 21, 2022, accepted March 25, 2022, date of publication April 1, 2022, date of current version May 5, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3164098

Domain-Specific Language Model Pre-Training for Korean Tax Law Classification

YEONG HYEON GU[1], XIANGHUA PIAO[1,2], HELIN YIN[1], DONG JIN[1,2], RI ZHENG[1,2], AND SEONG JOON YOO[1]

[1]Department of Computer Science and Engineering, Sejong University, Seoul 05006, South Korea
[2]Department of Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, South Korea

Corresponding author: Seong Joon Yoo (sjyoo@sejong.ac.kr)

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00755, Dark data analysis technology for data scale and accuracy improvement).

Yeong Hyeon Gu and Xianghua Piao contributed equally to this work.

ABSTRACT Owing to their increasing amendments and complexity, most taxpayers do not have the required knowledge of tax laws, which results in issues in everyday life. To use tax counseling services through the internet, a person must first select a category of tax laws corresponding to their tax question. However, a layperson without prior knowledge of tax laws may not know which category to select in the first place. Therefore, a model capable of automatically classifying the categories of tax laws is needed. Recently, a model using BERT has been frequently used for text classification; however, it is generally used in open-domains, and often experiences a degraded performance due to domain-specific technical terms, such as tax laws. Furthermore, a significant amount of time is required to train the model, since BERT is a large-scale model. To address these issues, this study proposes Korean tax law-BERT (KTL-BERT) for the automatic classification of categories of tax questions. For the proposed KTL-BERT, a new pre-trained language model was constructed by performing learning from scratch, to which a static masking method was applied based on DistilRoBERTa. Subsequently, the pre-trained language model was fine-tuned to classify five categories of tax law. A total of 327,735 tax law questions were used to verify the performance of the proposed KTL-BERT. The F1-score of the proposed KTL-BERT was approximately 91.06%, which is higher than that of the benchmark models by approximately 1.07%-15.46%, and the training speed was approximately 0.89%-56.07% higher.

INDEX TERMS BERT, domain-specific, Korean tax law, pre-trained language model, text classification.

I. INTRODUCTION

Taxation is one of the tools ensuring the economic development of a country [1]. Tax laws are continuously amended, and thus become more complicated, as a country develops and insufficient parts are supplemented or new policies are created. Accordingly, a majority of taxpayers do not have knowledge of the relevant laws related to tax issues that they encounter in everyday life.

The National Tax Service of Korea, Home Tax, provides phone counseling and internet counseling services for the convenience of taxpayers. Phone counseling is a service where a person inquires about tax matters through a phone call with an agent. However, phone counseling, in general, has a long wait time due to a limited number of available agents, and a caller is given an insufficient amount of time for their questions due to the limited service time. Internet counseling is a service where a person poses a tax question on a website and a tax law analyst responds after a certain period. However, the person must select the category of tax law corresponding to their question before registering their question. If an incorrect category is selected, a different agent may need to be assigned, or a less appropriate response be given, which will delay the response. The majority of individuals find it difficult to select the proper category for their tax issues. Thus, a model is needed that can automatically classify the category of tax questions.

Text classification can be largely divided into methods using a deep learning model or a pre-trained language model. A Bidirectional Encoder Representations from

The associate editor coordinating the review of this manuscript and approving it for publication was Jolanta Mizera-Pietraszko.

Transformers (BERT)-based pre-trained language model uses a transformer bidirectionally to perform better than a conventional unidirectional language model, which insufficiently learns contextual information [3]. Therefore, a BERT-based pre-trained language model is more widely used than a deep learning model for text classification. BERT is a general model trained using BooksCorpus [4] and English Wikipedia, and demonstrates excellent performance in open-domain fields [5], [6].

Natural language processing (NLP) can be divided into open-domain, which includes ordinary terms, and domain-specific, which includes technical or academic terminology, such as tax law, medicine, and biomedical field terms. However, since BERT is a general model, its ultimate performance is degraded due to the technical terminology when solving domain-specific tasks. Further pre-training, fine-tuning, and learning from scratch can improve the performance of the model in domain-specific tasks.

Further pre-training involves updating a pre-trained language model by adding a domain-specific dataset, since open-domain and domain-specific data distributions vary [7]. Fine-tuning improves the performance of a model by finely tuning the fixed parameters of a pre-trained model [8]. Unlike fine-tuning and further pre-training, learning from scratch builds a new vocabulary and retrains the model with domain-specific data to construct a new pre-trained language model [9]. A BERT-based pre-trained language model has errors in result values when the vocabulary changes since the model calculates the softmax for the entire vocabulary [3]. Since the token and index values of the vocabulary are also used during the training process, the model needs to be retrained if these values are changed. Therefore, fine-tuning and further pre-training, which use the vocabulary of an existing model, are highly convenient, since the step of building a pre-trained language model can be skipped. Tokenizer methods, such as WordPiece and SentencePiece, which are used when building a model, divide a domain-specific word into multiple tokens. However, the divided tokens cannot accurately convey the meaning of a domain-specific [9], because words from other domains contain the corresponding token. Therefore, using fine-tuning and further pre-training techniques for domain-specific words still results in a degraded performance. In contrast, learning from scratch can most effectively reduce the probability of performance degradation occurring since a new vocabulary and retraining are built from the beginning.

Static masking and dynamic masking are commonly used when pre-training a model by learning from scratch. Static masking involves masking certain words and then learning the masked words once, while dynamic masking changes the position of the words being masked to avoid learning the same words repeatedly [10]. Therefore, using dynamic masking requires a greater number of datasets or a larger batch size [11]. Robustly Optimized BERT Pretraining Approach (RoBERTa), which used dynamic masking, required a 160 GB dataset, while the dataset used in this study is only 188 MB in size. Since domain-specific tasks use significantly less data for learning compared to open-domain tasks, static masking is more suitable in preventing overfitting during the learning process.

Solving the effect of technical terminology on the performance problem is crucial to improving the performance of a model. However, cost is another factor that needs to be taken into consideration when actually applying the model. A BERT-base pre-trained language model is a large-scale model; thus, an extensive amount of time is required when actually applying it to industries [12]. Furthermore, there is a high demand on the device used for training due to restrictions in computational training and long training times [13]. Therefore, a lightweight model is needed to overcome these drawbacks.

The contributions of this study are as follows:

1) This study proposes KTL-BERT, which can automatically classify five categories of tax questions.
2) Tax law counseling data were collected from the National Tax Service Home Tax website, and a tax question dataset in Korean was constructed in 23 categories including ''Withholding tax'', ''Value added tax'', ''Capital gains tax'', ''Aggregate income tax'', and ''Corporate tax''; based on this dataset, a vocabulary containing 345,859 tokens was built by employing the byte-level BPE tokenizer technique.
3) To address the degraded performance due to technical terms, this study proposes learning from scratch, to which static masking is applied. The F1-score of the proposed KTL-BERT was 91.06%, which was approximately 1.07%–15.46% higher than that of the benchmark models; the training speed of KTL-BERT was approximately 0.89%–56.07% faster than that of the benchmark models.

The remaining contents of this paper are organized as follows. Related works are introduced in Section 2, while the datasets used in the study and the proposed model are outlined in Section 3. The performance evaluation of the proposed model is presented in Section 4. Lastly, Section 5 concludes the work.

II. RELATED WORKS

BERT, which was developed based on a transformer, has been widely applied in NLP due to its outstanding performance compared to conventional models. However, BERT exhibits degraded performance in domain-specific tasks [7], since it was developed to be applied generally. Therefore, researchers studying natural language processing carry out domain-specific tasks using the following methods: 1) Fine-tuning a pre-trained language model; 2) A pre-trained language model is updated by adding a domain-specific dataset; and 3) Learning a pre-trained language model from scratch to create a new pre-trained language model.

A. FINE-TUNING PRE-TRAINED LANGUAGE MODEL

Since high performance cannot be achieved in domain-specific tasks if only a pre-trained language model is used, research is continuously conducted into improving performance by fine-tuning a model. Li *et al.* [19] analyzed the sentiments of stock investors by combining attention and BERT. Their experimental results showed that the proposed BERT+Attention outperformed Long Short Term Memory (LSTM)+Attention and the support vector machine. Tong *et al.* performed named entity recognition (NER) using BERT-BiGRU-CRF in Spacecraft domain data [20]. In addition, hate speech detection experiments were conducted by combining a genetic algorithm and BERT [21].

B. FURTHER PRE-TRAINING LANGUAGE MODEL

Owing to the limitation of solving a domain-specific task by fine-tuning a pre-trained language model, Lee *et al.* [7] used further pre-training. A new pre-trained language model, BioBERT, was constructed by further pre-training on a biomedical-domain dataset. Their experimental results showed that BioBERT outperformed BERT in named entity recognition (NER), relation extraction, and question and answering tasks. CT-BERT [22], which was developed by additionally learning data related to COVID-19 on Twitter, exhibited a 10%–30% improvement in performance compared with BERT-Large. Zhang *et al.* [23] conducted an experiment on identifying antimicrobial peptides. Italian BERT [24], which was developed to classify Italian Twitter data, also exhibited outstanding performance. Low *et al.* [25] conducted a study on classifying fake news and satire based on the DistilBERT-base-uncased model, where the F1-score and accuracy were improved by 5.2% and 6.4%, respectively, when compared with the DistilBERT-base-uncased [13] model. Their study also demonstrated an outstanding performance in offensiveness identification, which is a Twitter NLP task, and in NER, when compared to CamemBERT, FlauBERT [26], and BARThez [27], which are models related to French language. However, fine-tuning or further pre-training a pre-trained language model is based on an existing model and vocabulary, thus it is incapable of achieving high performance because it is still affected by words from other domains.

C. LEARNING PRE-TRAINED LANGUAGE MODEL FROM SCRATCH

A model constructed by applying the learning from scratch method can achieve a high performance as the approach can solve the problem of degraded performance due to technical terms in a domain-specific task. The results of examining studies that apply the learning pre-trained language model from scratch are shown in Table 1.

The learning method and architecture of BERT are outstanding; therefore, most studies do not revise either aspect. GREEK-BERT [28] demonstrated an outstanding performance in part-of-speech tagging, named recognition, and natural language inference in GREEK-domain tasks. Fang *et al.* [29] solved the automated text classification of new-misses from safety reports. FBERT [30], which is a model that identifies offensive content, demonstrated a better performance than HateBERT [31], which is a BERT-based abusive language detection model.

Certain researchers revise the learning methods of existing models to improve their performance. For example, BERTweet [32] is based on the architecture of BERT, but the model is retrained using the learning method of RoBERTa. This model achieved the state-of-the-art (SOTA) performance in part-of-speech tagging, NER, and three tweet nlp tasks of text classification, and performed better than both RoBERTa-base and XLM-RoBERTa-base [15]. PurMedBERT [9] is a BERT-based model that has been retrained through whole-word masking using abstracts of PubMed papers in a biology-related domain. Experimental results showed that PurMedBERT outperformed both BioBERT and SCIBERT [33].

A BERT-base-multilingual-uncased model supports more than 100 languages and performs particularly well in the tasks of widely-used languages, such as English. However, for a language for which there is insufficient data in the wiki corpus used during the pre-training process, high performance cannot be achieved due to insufficient training. Research is being conducted on developing pre-trained language models for solving tasks in these types of languages [28], [34].

Since the data used in this study are in Korean, studies that developed a model for the Korean domain have been examined. As shown in Table 2, researchers studying the natural language processing of the Korean language have developed a pre-trained language model specialized for the Korean language in order to enhance the performance of the model in Korean-related tasks. SKT developed KoBERT[1] based on BERT by learning on the Korean wiki dataset. Beomi developed KcBERT [18], based on BERT, by collecting wiki data and news comments on the Naver website. Soongsil-BERT[2] is a model based on RoBERTa, pre-trained using news comments on the Naver website; the data of college communities and notices were utilized to develop a model reflecting the characteristics of college communities. KR-BERT [35] employed the Bidirectional WordPiece tokenizer to capture the characteristics of the Korean language. This model exhibited a similar level of performance to that of KoBERT, but with a smaller model size.

Learning from scratch is the optimal method for solving domain-specific tasks. When solving tasks related to the Korean language, a pre-trained language model built using Korean corpus outperforms a pre-trained language model that supports multi-languages. According to related studies, there is a large number of pre-trained language models that have been developed to solve tasks in Korean, but there has not been a pre-trained language model developed for the Korean

[1] https://github.com/SKTBrain/KoBERT
[2] https://github.com/jason9693/Soongsil-BERT

TABLE 1. Re-training model from scratch.

Model	Domain	Corpus Size	Tokenizer
GREEK-BERT [28]	Greek-specific	29GB	SentencePiece
BERT [29]	Near-misses from safety reports	10,500M words	WordPiece
FBERT [30]	Identifying offensive content	20GB	WordPiece
HateBERT [31]	Abusice language detection	12.5GB	SentencePiece
BERTweet [32]	English Twitter	80GB	TweetTokenizer
PurMedBERT [9]	Biomedical	21GB	WordPiece

TABLE 2. Korean-specific pre-trained language model.

Model	Domain	Corpus Size	Tokenizer
KoBERT(SKT)	Korean Wiki	5M sentences	SentencePiece
KcBERT [18]	Korean Wiki, Naver News Comments	12.5GB	WordPiece
Soongsil-BERT	Korean Wiki, Naver News Comments, Soongsil University data	20GB	Byte-Level BPE
KR-BERT [35]	Korean Wiki	2.47GB	BidirectionalWordPiece

tax law-domain; this is what is being addressed in this study. The Korean-specific pre-trained language models, KoBERT, Soongsil-BERT-base, and KcBERT-base, have better performances than those of XLM-RoBERTa-base or BERT, which supports multiple languages in classifying Korean Tax Law questions. However, their performance is still limited due to terminology related to the domain. Therefore, in this paper, by building a new pre-trained language model that has learned the Korean Tax Law-Domain Dataset, we plan to solve the problem of performance degradation under the influence of the relevant terminology.

III. PROPOSED METHOD

This study proposes the KTL-BERT model, which is capable of classifying the category of tax questions; a flowchart of the overview of the model is illustrated in Fig. 1 First, the cases of tax law counseling on the National Tax Service Home Tax website were collected and preprocessed to build a vocabulary. Then, the vocabulary and preprocessed tax questions were used to perform learning from scratch, which was applied with static masking, based on DistilRoBERTa. The model was then fine-tuned using the pre-trained KTL-BERT. When a tax question is input into the fine-tuned KTL-BERT, the model outputs the category of the tax question.

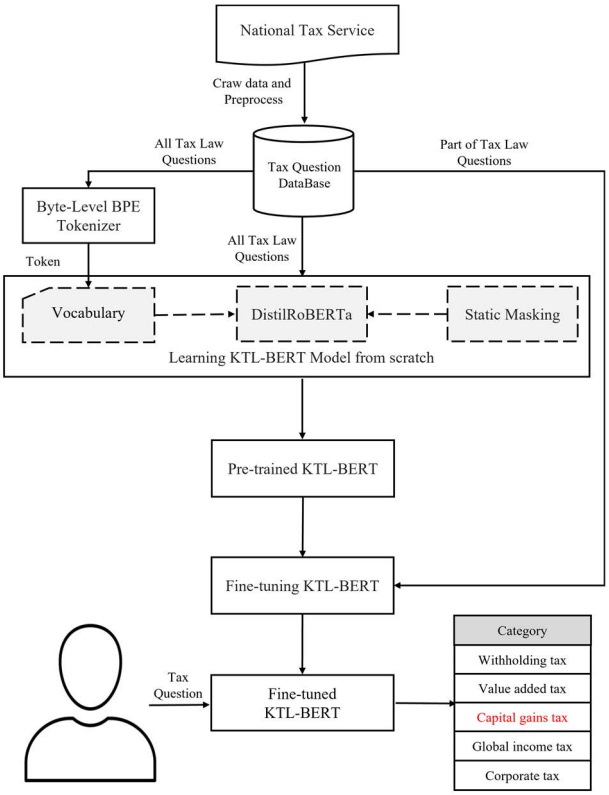

FIGURE 1. Flowchart to build KTL-BERT.

A. DATASET AND DATA PREPROCESSING

To construct the tax law dataset, tax law counseling data were collected from the second category ''Internet counseling cases'' of ''Search counseling cases'' in the menu banner ''Counseling/Information'' on the website of the National Tax Service Home Tax.[3] A total of 387,539 cases were collected from January 2008 to January 2021.

[3]https://www.hometax.go.kr/websquare/websquare.html?w2xPath=/ui/pp/index.xml

Since some questions regarded calculations, special characters were not removed in order to retain as much data content as possible. Similar to the English language, there is a space between each token in the Korean language. Some questions may contain spelling or grammar errors. The performance of a model can be affected if tokenizing cannot be accurately carried out due to errors in the sentences used for training. Therefore, a spelling-correction program[4] was used on the collected data to revise incorrectly written questions. From the collected data, data including null values for ''category'', ''title'', and ''question'' were deleted.

TABLE 3. Before data preprocessing vs after data preprocessing.

Category	Before	After
Withholding tax	82,041	72,700
Value added tax	73,846	60,661
Capital gains tax	60,824	48,454
Aggregate income tax	50,603	43,843
Corporate tax	44,095	34,584
Other	32,817	29,449
Inheritance and gift tax	21,077	16,252
Individual consumption tax	4,270	3,487
International tax	3,682	2,683
The basic law for national taxes	2,501	2,084
Year-end settlement	2,277	1,960
Comprehensive Real Estate Holding Tax	2,117	1,800
Work, child support	1,646	1,497
Education tax	1,163	1,065
Stamp duty	1,024	942
The gas price tax rebate	992	797
The National Tax Collection Law	891	721
Securities transaction tax	702	616
Liquor tax law	665	575
Unfair profit tax	157	120
Traffic tax	73	65
Asset revaluation tax	49	41
Excessively increased valuable land tax law	27	25
Total	387,539	327,735

The number of data after preprocessing was 327,735, and the distribution of data in each category before and after preprocessing is shown in Table 3. Taxpayers choose a category of tax law and leave a question when using the ''internet counseling'' service; however, some taxpayers select ''Other'' because they do not know the category of their tax question. When the tax laws belonging to the ''Other'' category were analyzed, all categories except ''Unfair profit tax,'' ''Asset revaluation tax,'' and ''Land excess income tax'' were included. It can thus be concluded that a large number of individuals do not know the proper category of their tax question. The number of data in the fifth and the seventh categories varied significantly, as can be seen in Table 3. Therefore, only the data of the top five categories, ''Withholding tax'', ''Value added tax'', ''Capital gains tax'', ''Aggregate income tax'', and ''Corporate tax'', were used in this study.

The process of building a vocabulary in the learning from scratch step is as follows. First, the byte-level BPE tokenizer was used for tokenizing the entire preprocessed tax question dataset. Subsequently, tokens appearing only once were removed to build a vocabulary containing 345,859 tokens.

B. LEARNING KTL-BERT FROM SCRATCH

In this section, we outline how learning from scratch was carried out by outlining the steps of applying static masking based on the newly built vocabulary, the entire tax question dataset, and the DistilRoBERTa model. In the learning from scratch step, all 327,735 preprocessed tax questions were used in the dataset. The learning rate and batch size were set to 1e-4 and 24, respectively, and the model was retrained for 40 epochs. Five to six hours per epoch were required to retrain the KTL-BERT model, and a total of nine days were spent to retrain the 40 epochs.

This study used DistilRoBERTa, which is a lightweight model, to shorten the training time. DistilRoBERTa consists of six transformer encoders, and each encoder has multi-head attention, (position-wise) fully-connected feed-forward network, and add and normalize. The architecture of the DistilRoBERTa model is illustrated in Fig. 2.

As the input of the model, input embedding and positional encoding were combined in order to record the relative positions of each token.

Multi-head attention obtains the token vector, or the Q (query: variable representing the affected words), K (key: variable representing the words having effects), and V (value: weights of the effects) matrices by projecting the embedding dimension matrix of the sequence length. Afterward, Q, K, V are divided by the number of heads on the multi-head to parallel process the scaled dot-product attention.

Scaled dot-product attention is similar to self-attention, however it prevents the inner product from enlarging through $\sqrt{d_k}$, which is for scaling. d_k is the K matrix value representing a vector dimension. The attention weights distribution is the value to which softmax is applied, and it is obtained by dividing the dot product of Q and K by the scaling factor $(1/\sqrt{d_k})$. The attention value is obtained by multiplying V with the attention weights distribution. The formula for calculating the scaled dot-product attention is shown in (1), and the calculation process is shown in Fig. 3. The attention value suggests which words the model must pay close attention to among the input sentences.

$$Attention\,(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (1)$$

[4]https://github.com/ssut/py-hanspell

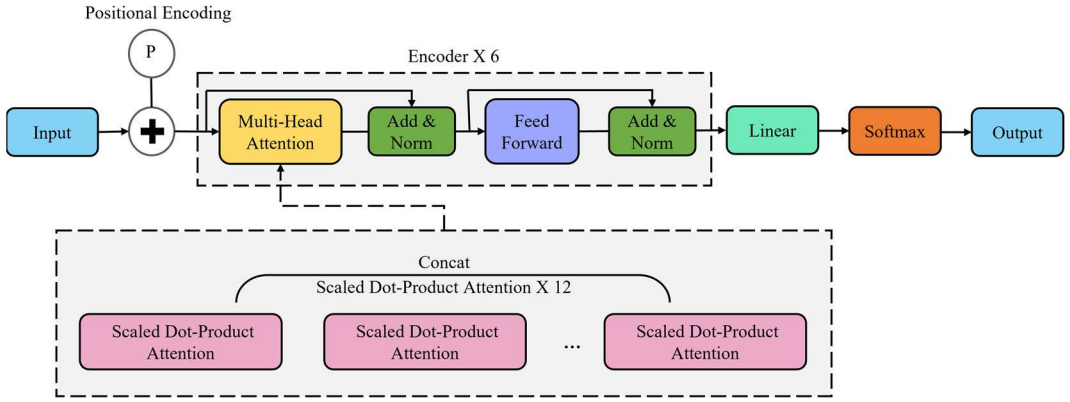

FIGURE 2. Architecture of DistilRoBERTa.

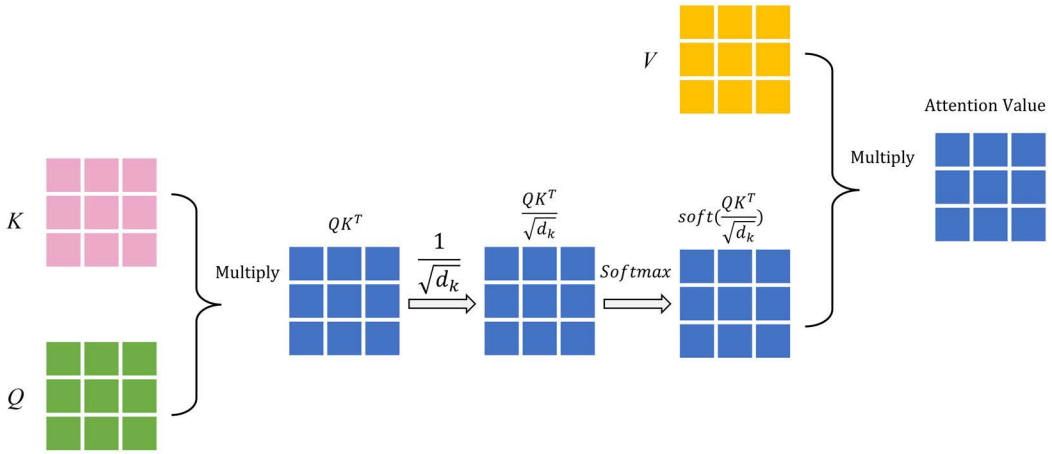

FIGURE 3. Scaled dot-product attention.

The results $h_1, h_2, h_3, \ldots, h_m$ obtained from the scaled dot-product attention are concatenated and multiplied with the weighted matrix W^o; the multi-head attention output matrix can be obtained using (2).

$$Multihead\ (Q, K, V) = Concat(h_1, h_2, h_3, \ldots, h_n)W^o \quad (2)$$

Multi-head attention allows the model to focus on information at each location simultaneously [36].

Subsequently, a fully-connected feed-forward network process was performed as shown in (3). Linear transformation was applied to the output matrix obtained in the multi-head attention step, and then linear transformation was applied once again through Rectified Linear Unit function. The same parameters W and b were applied to each position, but different parameters were applied if the layers changed. This process is also referred to as position-wise since it applies to each position.

$$FFN(x) = Max\ (0, xW_1 + b_1)\ W_2 + b_2 \quad (3)$$

C. FINE-TUNING KTL-BERT

In this section, the process of fine-tuning KTL-BERT is briefly explained. BERT [3] is known to exhibit excellent performance if it has a batch size of {16, 32} during fine-tuning, an epoch of {2, 3, 4} and a learning rate of {2e-5, 3e-5, 5e-5}. A total of 170,000 tax questions in the top five categories, "Withholding tax," "Value added tax," "Capital gains tax," "Aggregate income tax," and "Corporate tax," were used to fine-tune the model. The distribution of the tokenized sentence lengths of the dataset is visualized in Fig. 4, which shows that the length of tokenized sentences slowly increased in the beginning, but the rate of change of sentence lengths accelerated toward the end. If the sequence length is set to be excessively long, the parts exceeding the sentence length are filled with zeros, thus affecting the overall performance of the model [17]. Therefore, the sequence length was set to 100 in this study. For the parameters used to fine-tune the pre-trained KTL-BERT model, the training epoch was {1, 2, 3, 4}; the batch size was {16, 32}; and the learning rate was {1e-5, 2e-5, 3e-5, 5e-5}.

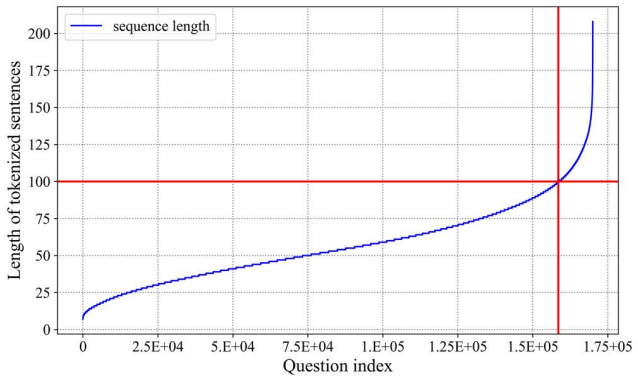

FIGURE 4. Set sequence length.

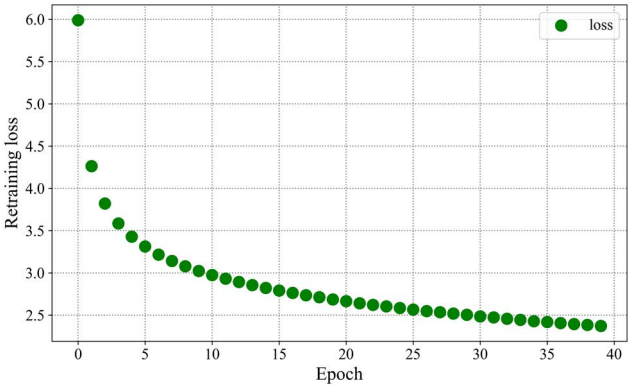

FIGURE 5. Retraining loss by epoch.

IV. EXPERIMENT AND RESULTS

A. ENVIRONMENTAL SETTING

The experiment in this study was conducted on the Centos (v7.9.2009) system, Intel Xeon CPU E5-2630 v3 @2.40GHz, and GeForce RTX 3090 GPU, on the Pytorch (v1.8.0) Deep Learning framework. The byte-Level BPE tokenizer provided in the transformers (v3.0.0) package was used to build the vocabulary in the learning from scratch step.

B. PERFORMANCE METRICS

The F1-score, precision, and recall were used as performance metrics for the proposed model, and their equations are as follows.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6}$$

True positive (TP) in the equations represents the number of correct predicted data in the current category, false positive (FP) represents the number of data in the another category, but predicted to be in the current category, while false negative (FN) is the number of data in the current category but predicted to be in another category.

C. PRE-TRAINED KTL-BERT MODEL

We outline the pre-trained KTL-BERT model in this section. In the learning from scratch step, a model and a loss were saved per model when retraining the model for 40 epochs. The loss of the model gradually decreased along the epoch, as shown in Fig. 5. The loss was the smallest at the 40th epoch and, therefore, the 40th pre-trained KTL-BERT model was used in this study.

D. EXPERIMENTAL RESULTS ON HYPERPARAMETER SETTINGS OF THE MODEL

In this section, the process of fine-tuning the pre-trained KTL-BERT model is explained in detail. Devlin *et al.* [3] proposed that setting the batch size to 16 or 32 would be helpful for improving the performance of the model. Accordingly, Fang *et al.* [29] and Zhang *et al.* [23] set the batch size to 32 in a downstream task, while Le *et al.* [26] set the batch size to 16. Since both values were applied in this study, the batch size appropriate for the tax question was selected through parameter-tuning. Training for two or four epochs would suffice if there were a large number of data, since the model is not sensitive to parameters [3].

KTL-BERT was trained for four epochs, and the performance from the test dataset was recorded for each epoch. The learning rate was set to 1e-5, 2e-5, 3e-5, or 5e-5, and the performance of KTL-BERT is shown in Fig. 6 for all learning rates.

The performance was optimal when the batch size and learning rate were set to 16 and 2e-5, respectively, and the pre-trained KTL-BERT model was fine-tuned twice are shown in Table 4; the F1-score, precision, and recall were 91.06%, 91.05%, and 91.06%, respectively.

The confusion matrix of the proposed KTL-BERT for the test dataset is shown in Fig. 7. The confusion matrix shows that the model can accurately classify questions regarding "Corporate tax," "Value added tax," "Capital gains tax," and "Withholding tax," but cannot accurately classify questions about "Global income tax".

A morphological analysis was conducted on the questions that belonged to the "Global income tax" category but were not classified correctly in the test dataset. Since the nouns "Withholding tax" and "Capital gains tax" appeared frequently, it was discovered that the source text was associated with two or more categories. Recall refers to the rate of actually true values being predicted as true by the model, and therefore, a low recall was recorded because the model could not accurately predict the correct category. This is because names of other categories were included in the sentences.

E. EXPERIMENTAL RESULTS ON DIFFERENT MODELS

In this section, we describe the proceedings of the experiments conducted to compare the performance of KTL-BERT to that of the other models. The first experiment compared

TABLE 4. Relationship between batch size, learning rate, epoch and F1-score. The value in bold indicate the best performance.

Batch size	Learning rate	Epoch			
		0	1	2	3
16	1e-5	90.47	90.98	90.81	90.18
	2e-5	90.63	**91.06**	90.77	90.45
	3e-5	90.65	90.93	90.26	90.02
	5e-5	90.36	90.28	89.41	88.95
32	1e-5	90.43	90.86	90.83	90.54
	2e-5	90.73	91.04	90.76	90.66
	3e-5	90.84	90.99	90.50	90.20
	5e-5	90.84	90.42	89.61	89.39

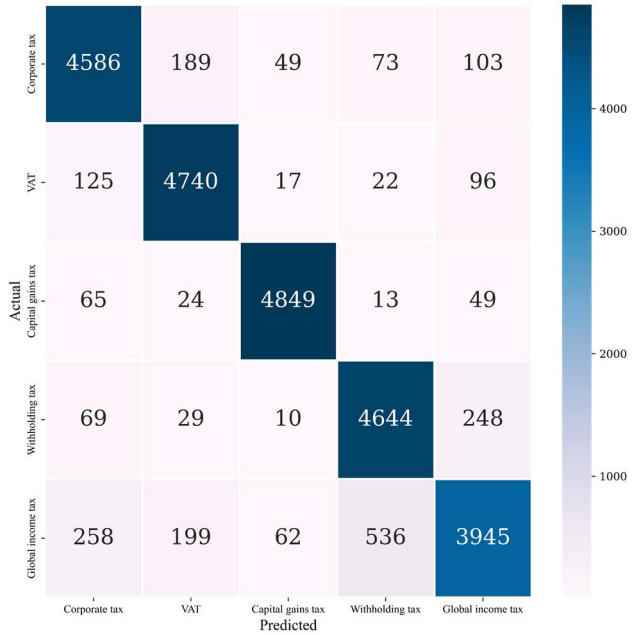

FIGURE 6. Confusion matrix.

the performances of KTL-BERT and a deep learning model, and the second experiment compared the performances of KTL-BERT and the BERT-based pre-trained language models developed in previous studies.

1) KTL-BERT VS DEEP LEARNING MODEL

In this section, we delineate the experiment conducted to compare the performance of KTL-BERT with that of Word2vec+LSTM [16] and Attention-based Bidirectional Gated Recurrent Unit-Convolutional Neural Network (Attention-based BiGRU-CNN) [2], both of which exhibited excellent performance in previous studies.

- Word2vec+LSTM [16]: The minimum word count and window value were set to five and 10, respectively, using Skip-gram provided by Gensim, and the sentence features were extracted. Subsequently, the extracted features were applied to the LSTM model for classification. Since input size, epoch, and batch size were not specified in the previous study by Noguti *et al.*, they were set to 100, 4, and 16, respectively, as in the case of fine-tuning the proposed KTL-BERT.
- Attention-based BiGRU-CNN [2]: As mentioned in a study by Liu *et al.*, words were vectorized using Continuous Bag Of Word, and both the input and output of BiGRU were set to 256. Moreover, epoch and batch size were set to 50 and 128, respectively. The weight of each word was calculated by applying the attention mechanism to the output of BiGRU. Afterward, a convolutional layer with a kernel height of two, three, and four was connected. Finally, the model was constructed by connecting the pooling layer, dropout layer, and fully-connected layer. The epoch and batch size were set to 50 and 128, respectively.

Table 5 presents the time required for the model to train for one epoch, as well as the maximum F1-score, precision, and recall in the comparative experiment conducted with the test dataset. The training time of the proposed KTL-BERT was not the fastest, but the F1-score was 9.64%–10.36% higher than those of both the Word2vec+LSTM and Attention-based BiGRU-CNN.

2) KTL-BERT VS PRE-TRAINED LANGUAGE MODEL

In this section, we describe the experiment conducted to compare the performance of KTL-BERT with benchmark models, which can be broadly classified into four types: general models using a byte-level BPE tokenizer for different languages, specific models using a byte-level BPE tokenizer for different languages but similar domains, general models using multilingual (including Korean) datasets, and Korean-specific models.

The KTL-BERT is based on distilled pre-trained language model, which is quicker than the conventional size model,

TABLE 5. Performance comparison with deep learning model.

Method	F1-score	Precision	Recall	Time/Epoch(s)
Word2vec+LSTM [16]	80.70	80.69	80.69	2,101
Attention-based BiGRU-CNN [2]	81.42	81.53	81.42	**379**
KTL-BERT(Ours)	**91.06**	**91.05**	**91.06**	1,129

TABLE 6. Performance comparison with distilled pre-trained language model.

Method	F1-score	Precision	Recall	Time/Epoch(s)
DistilRoBERTa [10]	75.98	75.88	75.98	620
DistilBERT-base-multilingual-uncased [3]	87.20	87.17	87.20	872
DistilKoBERT	88.63	88.64	88.64	**392**
Soongsil-BERT-small	89.58	89.58	89.59	813
KTL-BERT(Ours)	**91.06**	**91.05**	**91.06**	1,129

TABLE 7. Performance comparison with pre-trained language model.

Method	F1-score	Precision	Recall	Time/Epoch(s)
RoBERTa-base [10]	75.60	75.54	75.60	1,507
LegalRoBERTa [14]	76.33	76.32	76.33	1,203
BERT-base-multilingual-uncased [3]	87.48	87.45	87.48	1,182
KR-BERT [35]	89.35	89.37	89.35	1,586
XLM-RoBERTa-base [15]	89.52	89.52	89.52	1,762
Soongsil-BERT-base	89.85	89.82	89.85	1,139
KoBERT	89.90	89.94	89.90	1,652
KcBERT-base [18]	89.99	89.97	89.99	1,535
KTL-BERT(Ours)	**91.06**	**91.05**	**91.06**	**1,129**

but when the model size is reduced, performance is deteriorated. To demonstrate the performance of KTL-BERT, it was compared to both a distilled model and a normal-sized model. The experimental results of KTL-BERT and the conventional models when all parameters are set identically are presented from Tables 6–7.

Performance comparison between KTL-BERT and distilled pre-trained language models are shown in Table 6. Although the F1 score of KTL-BERT is higher than that of other distilled models, the training speed is slower. DistilKoBERT has three hidden layers, whereas other distilled benchmark models have six. As a consequence, DistilKoBERT outperforms other models in terms of speed. The structure of DistilRoBERTa is identical to that of KTL-BERT, but its training speed is half. We attempted to investigate the differences between the two models under the same experimental settings. The size of the vocabulary in

TABLE 8. Record the vocabulary size of each model.

Method	Vocabulary size
DistilRoBERTa [10]	**50,265**
DistilBERT-base-multilingual-uncased [3]	119,547
DistilKoBERT	8,002
Soongsil-BERT-small	16,000
KTL-BERT(Ours)	**345,859**

KTL-BERT is found to be excessive when compared to other models, as shown in Table 8. As a result, it can be deduced that the vocabulary size influenced KTL-BERT training speed.

The results in Table 7 show that the proposed KTL-BERT outperformed normal-sized benchmark models. KcBERT-base was developed for Korean-specific applications and outperformed the other benchmark models, with an F1-score that was just 1.07% lower than that of KTL-BERT. Even though XLM-RoBERTa-base supported multi-languages, its performance was virtually identical to that of the Korean-specific model, with an F1-score of over 89%.

F. DISCUSSION

Selecting appropriate parameters plays a crucial role in the performance of a model when fine-tuning a pre-trained language model for a domain-specific task. In this study, when the learning rate was set to {1e-5, 2e-5, 3e-5, 5e-5} and KTL-BERT was fine-tuned, the best performance was recorded with a learning rate was 2e-5; this confirms the finding of a previous study, which reported that setting the learning rate to 2e-5 [33] would result in good performance.

In a study on solving a domain-specific task using a BERT-based pre-trained language model, the performance of the model was compared against that of a model that recorded SOTA performance in the respective domain [7], [9], [30]. However, since no model has been developed to solve a Korean tax law-domain task, this study examined a pre-trained language model based on a legal domain. Legal-RoBERTa, which is a pre-trained language model based on a legal domain, was constructed using English language data. Although LegalRoBERT and RoBERTa-base were not developed using the data in the Korean language, Table 5 shows that LegalRoBERTa exhibited a better performance than RoBERTa-base, which is a general model that only uses English data. Thus, it can be concluded that using a byte-level BPE tokenizer causes a model to be influenced by the domain, as demonstrated by the two above models. Not surprisingly, the F1-scores of BERT-base-multilingual-uncased and XLM-RoBERTa-base, which support multiple languages, are higher than that of LegalRoBERTa.

Even though XLM-RoBERTa-base is a multilingual pre-trained model, it performs similarly to the Korean-specific model. As seen in Figure 8, we tokenized terminology with each model. KTL-BERT splits terms into one token, whereas the Korea-specific model and XLM-RoBERTa-base split three to four tokens. The term was broken into seven parts using BERT-base-multilingual-uncased. The length of terms tokenized by our proposed KTL-BERT is the shortest. It can also be demonstrated that the proposed KTL-BERT can handle the problem of model performance degradation in the field of tax law owing to professional terminology. Higher performance is aided by shorter tokenized word lengths, whether in the open-domain or domain-specific.

Performance comparison experiments with both the distilled model and normal-sized model were conducted to establish the advantages of the proposed KTL-BERT.

Model	Tax law term tokenization
	Aggregate income tax(English)/종합소득세(Korean)
RoBERTa-base[10]	['ì', 'ę', 'h', 'ìķ', '©', 'ì', 'Ì', 'Ţ', 'ë', 'j', 'Ł', 'ì', 'Ħ', ',']
LegalRoBERTa[14]	['ì', 'ę', 'h', 'ìķ', '©', 'ì', 'Ì', 'Ţ', 'ë', 'j', 'Ł', 'ì', 'Ħ', ',']
BERT-base-multilingual-uncased[3]	['ᄌ', '##소ᄋᆞᆸ', '##스', '##드ᅳ', '##ᄀᆡ', '##ᄉᆡᆨ']
KR-BERT[35]	['종합', '##소득', '##세']
XLM-RoBERTa-base[15]	['▁종합', '소득', '세']
Soongsil-BERT-base	['ìɕh', 'ìķ©', 'ìÎļëjŁ', 'ìĦ,']
KoBERT	['▁종합', '소득', '세']
Ours	['ìɕhíķ©ìÎļëjŁìĦ,']

FIGURE 7. Tokenization example.

The F1-score of KTL-BERT was 1.48%-15.08% and 1.07%-15.46% higher than those of the distillation and regular size models, respectively. KTL-BERT exhibited a better performance than the benchmark models in experiments, even when it was based on a distilled model.

V. CONCLUSION

In this paper, we propose KTL-BERT, which is the first pre-trained language model of Korean Tax Law-Domain. KTL-BERT was built by applying static masking to the DistilRoBERTa model; a pre-trained model learned from scratch and then was fine-tuned. As a result, the performance of the model was improved by solving the problem of performance degradation due to technical terminology, and the training speed was increased because it was based on the distilled pre-trained language model. As a result of performance measurement using tax questions collected by the Korea National Tax Service, the F1-score of KTL-BERT was 91.06%, and when compared with the deep learning models Word2vec+LSTM and Attention-based BiGRU-CNN, it was 9.64-10.36% higher. As a result of comparing the performance with the existing BERT-based pre-trained language model, KTL-BERT is better than RoBERTa-base, Legal-RoBERTa, BERT-base-multilingual-uncased, XLM-RoBERTa-base, Soongsil-BERTa-base, KoBERT, and KcBERT-base. The F1-score was about 1.07-15.46% higher, and the training speed was 0.89-56.07% higher than that of the existing models. In a study with the distilled pre-trained language model, we found that more time is required to train the model due to the large size of the KTL-BERT vocabulary. However, when tokenizing terminology, the size of the vocabulary increases because the terminology must be included in the vocabulary so that it is not divided into multiple subwords. We think future work can refer to DistilKoBERT. Although DistilKoBERT was built with three hidden layers, there was no significant difference in performance. In the future, we will conduct an experiment to improve the training speed by lowering the model complexity while maintaining the vocabulary size and performance of the model.

REFERENCES

[1] C. J. Ihenyen and E. G. Mieseigha, "Taxation as an instrument of economic growth (The Nigerian Perspective),' *Inf. Knowl. Manage.*, vol. 4, no. 12, pp. 49–53, 2014.
[2] J. Liu, Y. Yang, S. Lv, J. Wang, and H. Chen, "Attention-based BiGRU-CNN for Chinese question classification," *J. Ambient Intell. Hum. Comput.*, vol. 4, pp. 1–12, Jun. 2019, doi: 10.1007/s12652-019-01344-9.
[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Associat. Comput. Linguist.*, 2019, pp. 4171–4186.
[4] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 19–27.
[5] P. Nie, Y. Zhang, X. Geng, A. Ramamurthy, L. Song, and D. Jiang, "DC-BERT: Decoupling question and document for efficient contextual encoding," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1829–1832.
[6] C. Liang, Y. Yu, H. Jiang, S. Er, R. Wang, T. Zhao, and C. Zhang, "BOND: BERT-assisted open-domain named entity recognition with distant supervision," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 1054–1064.
[7] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, Feb. 2020, doi: 10.1093/bioinformatics/btz682.
[8] S. Paul and S. Saha, "CyberBERT: BERT for cyberbullying identification," *Multimedia Syst.*, vol. 2020, pp. 1–8, Nov. 2020, doi: 10.1007/s00530-020-00710-4.
[9] Y. R. H. M. N. X. Gu Tinn Cheng Lucas Usuyama Liu and H. Poon, "Domain-specific language model pretraining for biomedical natural language processing," *ACM Trans. Comput. Healthcare*, vol. 2021, vol. 3, no. 1, pp. 1–23.
[10] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
[11] Z. Zhao and H. Wang, "MaskGEC: Improving neural grammatical error correction via dynamic masking," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 1226–1233.
[12] X. Ren, R. Shi, and F. Li, "Distill bert to traditional models in Chinese machine reading comprehension (student abstract)," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 10, pp. 13901–13902.
[13] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.
[14] S. Geng, R. Lebret, and K. Aberer, "Legal transformer models may not always help," 2021, *arXiv:2109.06862*.
[15] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," 2019, *arXiv:1911.02116*.
[16] M. Y. Noguti, E. Vellasques, and L. S. Oliveira, "Legal document classification: An application to law area prediction of petitions to public prosecution service," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.
[17] M. Li, L. Chen, J. Zhao, and Q. Li, "Sentiment analysis of Chinese stock reviews based on BERT model," *Int. J. Speech Technol.*, vol. 51, no. 7, pp. 5016–5024, Jul. 2021, doi: 10.1007/s10489-020-02101-8.
[18] J. Lee, "Kcbert: Korean comments bert," in *Proc. Conf. Hum. Lang. Technol.*, 2020, pp. 437–440.
[19] M. Li, W. Li, F. Wang, X. Jia, and G. Rui, "Applying BERT to analyze investor sentiment in stock market," *Neural Comput. Appl.*, vol. 33, no. 10, pp. 4663–4676, May 2021.
[20] B. Tong, J. Pan, L. Zheng, and L. Wang, "Research on named entity recognition based on bert-BiGRU-CRF model in spacecraft field," in *Proc. IEEE Int. Conf. Comput. Sci., Electron. Inf. Eng. Intell. Control Technol.*, Sep. 2021, pp. 747–753.
[21] K. J. Madukwe, X. Gao, and B. Xue, "A GA-based approach to fine-tuning bert for hate speech detection," in *Proc. IEEE Symp. Series Comput. Intell. (SSCI)*, Dec. 2020, pp. 2821–2828.
[22] M. Müller, M. Salathé, and P. E Kummervold, "COVID-Twitter-BERT: A natural language processing model to analyse COVID-19 content on Twitter," 2020, *arXiv:2005.07503*.
[23] Y. Zhang, J. Lin, L. Zhao, X. Zeng, and X. Liu, "A novel antibacterial peptide recognition algorithm based on BERT," *Briefings Bioinf.*, vol. 22, no. 6, Nov. 2021, Art. no. bbab200, doi: 10.1093/bib/bbab200.
[24] M. Pota, M. Ventura, R. Catelli, and M. Esposito, "An effective BERT-based pipeline for Twitter sentiment analysis: A case study in Italian," *Sensors*, vol. 21, no. 1, p. 133, 2021, doi: 10.3390/s21010133.
[25] J. F. Low, B. C. Fung, F. Iqbal, and S. C. Huang, "Distinguishing between fake news and satire with transformers," *Expert Syst. Appl.*, vol. 187, Dec. 2022, Art. no. 115824, doi: 10.1016/j.eswa.2021.115824.
[26] H. Le, L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L. Besacier, and D. Schwab, "FlauBERT: Unsupervised language model pre-training for French," 2019, *arXiv:1912.05372*.
[27] M. Kamal Eddine, A. J.-P. Tixier, and M. Vazirgiannis, "BARThez: A skilled pretrained French sequence-to-sequence model," 2020, *arXiv:2010.12321*.
[28] J. Katsikas, I. Chalkidis, P. Malakasiotis, and I. Androutsopoulos, "Greek-BERT: The greeks visiting sesame street," in *Proc. 11th Hellenic Conf. Artif. Intell.*, 2020, pp. 110–117.
[29] W. Fang, H. Luo, S. Xu, P. E. Love, Z. Lu, and C. Ye, "Automated text classification of near-misses from safety reports: An improved deep learning approach," *Adv. Eng. Informat.*, vol. 44, Dec. 2020, Art. no. 101060.
[30] D. Sarkar, M. Zampieri, T. Ranasinghe, and A. Ororbia, "FBERT: A neural transformer for identifying offensive content," in *Proc. Findings Assoc. Comput. Linguistics*, 2021, pp. 1792–1798.
[31] T. Caselli, V. Basile, J. Mitroviá, and M. Granitzer, "HateBERT: Retraining BERT for abusive language detection in English," 2020, *arXiv:2010.12472*.
[32] D. Q. Nguyen, T. Vu, and A. Tuan Nguyen, "BERTweet: A pre-trained language model for English tweets," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, 2020, pp. 9–14.
[33] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A pretrained language model for scientific text," 2019, *arXiv:1903.10676*.
[34] A. Virtanen, J. Kanerva, R. Ilo, J. Luoma, J. Luotolahti, T. Salakoski, F. Ginter, and S. Pyysalo, "Multilingual is not enough: BERT for Finnish," 2019, *arXiv:1912.07076*.
[35] S. Lee, H. Jang, Y. Baik, S. Park, and H. Shin, "KR-BERT: A small-scale Korean-specific language model," 2020, *arXiv:2008.03979*.
[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

YEONG HYEON GU received the B.S., M.E., and Ph.D. degrees in computer science and engineering from Sejong University, Seoul, South Korea, in 2004, 2006, and 2014, respectively. He is currently a Research Professor with the Department of Computer Science and Engineering, Sejong University.

XIANGHUA PIAO received the B.S. degree in computer science and engineering from Yanbian University, China, in 2019, and the M.S. degree in computer science and engineering from Sejong University, Seoul, South Korea, in 2021, where she is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering.

HELIN YIN received the B.S. degree in computer science and engineering from Yanbian University, China, in 2015, and the M.S. degree in computer science and engineering from Sejong University, Seoul, South Korea, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering.

DONG JIN received the B.S. degree in computer science and engineering from Yanbian University, China, in 2018, and the M.S. degree in computer science and engineering from Sejong University, Seoul, South Korea, in 2020, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering.

RI ZHENG received the B.S. degree in computer science and engineering from Yanbian University, China, in 2018. He is currently pursuing the Ph.D. degree with the Department of computer science and engineering with Sejong University, Seoul, South Korea.

SEONG JOON YOO received the M.E. degree in electrical engineering from Korea University and the Ph.D. degree in computer and information science from Syracuse University. He is currently a Professor with the Department of Computer Science and Engineering, Sejong University, Seoul, South Korea. He is the Director of the Advanced Bigdata Center for Research and Collaboration (ABRC) and works on research of big data and artificial intelligence.