Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

The Impact of Preprocessing and Word Embedding in Extreme Multi-label Patent Classification

GUIK JUNG[1], (Student Member, IEEE), JUNGHOON SHIN[2], AND SANGJUN LEE.[3], (Member, IEEE)

[1]Department of Software, Soongsil University, Seoul 07027 South Korea (e-mail: rndlr96@ssu.ac.kr)
[2]Department of Software Convergence, Soongsil University, Seoul 07027, South Korea (e-mail: junghoon.shin@ssu.ac.kr)
[3]Department of Software, Soongsil University, Seoul 07027 South Korea (e-mail: sangjun@ssu@ac.kr)

Corresponding author: Sangjung Lee (e-mail: sangjun@ssu.ac.kr)

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2018-0-01419) supervised by the IITP(Institute for Information communications Technology Promotion).

ABSTRACT The patent classification is an essential task to efficiently process the patent data and make the data available to information users effectively. Various algorithms and deep learning-based methods have been proposed to address the inefficiency of the patent classification. However, the research on the impact of preprocessing, word embeddings, and Data field on the patent classification is relatively insufficient. In this paper, we examined three general situations to compare and analyze the effects of the generalization of words by Stemming on the performance when considering the characteristics of patent data. In addition, in this paper, a comparison experiment between the pre-trained word embedding model and the embedding model learned with patent dataset. As a result, it was found that the CBoW embedding model learned through patent dataset could best represent the words of the patent documents and that the IPC classification code had a great impact on the performance. In addition, the relationship between the number of embedding words and the classification performance was confirmed. Finally, we conducted experiments with different data fields and classification models. The performance was greatly improved when IPC was included and the classification accuracy was high when the classification model considering the relationship between labels and words was used. The experiment was to classify about 1 million patent data in the United States, Europe, and Japan into 1383 classes. The final performance was P@1 = 71.896%, P@3 = 36.697%, and P@5 = 24.301% by using two simple ensembles of LAHA models.

INDEX TERMS Preprocessing, Word Embedding, Extreme Multi-label Classification, Patent Classification, Deep Learning

I. INTRODUCTION

TOday, as we enter a knowledge-based society that generates value through knowledge and information, the importance of patents, a representative form of intellectual property rights(IP), is increasing very much. These patents are regarded as a kind of strategic resource for managing information and knowledge, and are used in Research and Development activities such as the development of new products [1]. Because patent documents contain the latest technology and knowledge, it is possible to reduce the time consumption and the cost for research through prior investigation and analysis of related patents [2]. However, since a patent documents contain specialized content for a specific field, anyone other than patent experts and those in the field have a lot of difficulty in analyzing and selecting data they need. In addition, the number of patents filed every year is rapidly increasing across all fields. According to statistics released by the World Intellectual Property Organization (WIPO), the number of patent applications in eight of the top 10 countries increased [3]. This rapidly increasing large amount of patent data makes it difficult for information users to use patent data. To address this problem, many alternatives have proposed to efficiently process, analyze, classify, and store patent data [4]-[6]. While studies on how to classify patents have been actively conducted, studies on the effects of preprocessing and word embedding have not been conducted.

To the best of our knowledge, this is the first study of the impact of preprocessing and word embedding on the patent classification. There are several problems with the patent data classification. First, patent data is a set of documents corresponding to multiple domains. For that reason, patent data must be classified into a large number of classes to be classified. IPC, an international standard classification, classifies patents from 8 sections to more than 60,000 subgroups depending on the classification level [7]. Second, patent documents are multi-label data in which several classes can exist in one document, and the number of labels designated in a document is not determined. This creates scalability problems and makes the classification task computationally difficult. Third, patent data is not evenly distributed by class. Classes, which have been actively researched for a long time, have a very large amount of accumulated data. On the other hand, the number of patents in recently emerging fields is relatively small. Finally, patent documents are lengthy or ambiguous, and contain many technical and specific terms. In this paper, various experiments were conducted for patent data classification with such problems. The experiment was conducted in the order of Data sampling, Text preprocessing, word embedding, Data field comparison, and Deep learning networks for classification.

Experimentation is a very important part of artificial intelligence research. The first step in the Supervised learning experiment is to properly partition the dataset into a learning dataset and an experimental dataset, or a learning dataset and a validation/test dataset. In general, Stratified data sampling is mainly used for data segmentation in classification tasks [8]. However, when multi-label data is randomly classified using Stratified data sampling, it may cause problems in that a sample for that class may not exist in the case of a class with a small number of data in the test dataset. This problem does not make the experiment impossible. However, it can be a factor that degrades the reliability of the experiment. For example, in this study, the top 500 classes account for 80% of the total data. In such case, when Stratified data sampling is carried out, the test class may not have data corresponding to the rare class. When the data set is made in this way, the experiment shows higher accuracy. Prior to the experiment, this paper established a dataset through sampling that reflected the characteristics of multi-label data [9].

In general, Text Classification Task consists of preprocessing, Word Representation, and Classification Stage. It has been shown through several experiments and verifications that preprocessing in the Classification Task affects the performance [10]-[12]. The detailed processes included in the preprocessing should be appropriately used depending on the characteristics of each data and Task. In most NLP Task experiments, Stop-word Removal improved performance [11]. Stemming, which reduces complexity by generalizing words based on algorithm rules, has multiple algorithms depending on rules. Several experiments have shown the impact of the Stemming algorithm on the performance [19],[20]. In this paper, preprocessing including Non-ASCII Removal, Tag Removal, Stop-word Removal, and Lemmatize was applied to all experimental data. We thought that reducing the complexity of Features through the generalization of words would reduce the performance, considering the characteristics of patent data including data from multiple domains. To confirm this, the performance was compared through an experiment using data to which two representative Stemming algorithms were applied and the data not applied.

The Word Representation for converting natural language into a vector, which can be understood by a computer, is carried out through word embedding. word embedding is useful for the vector representation of documents and words. There are count-based methods and deep learning-based methods, and have drawn attention in various NLP tasks such as Text Classification, Keyword Extraction, and Similarity Analysis. Simple count-based methods cannot express the relationship and similarity between words and words, but can achieve a certain level of classification performance [21]. word embedding based on Deep Learning learns words through Shallow Deep Learning network, through which it expresses the relationship and similarity between words and words. There are two typical deep learning-based word embeddings, Word2Vec and GloVe. There are models that have learned many words in advance through a very large dataset [13]-[15]. Most of all, there are the Word2Vec pre-trained model, which learned about 3 million words using GoogleNews as a dataset, and the GloVe pre-trained model, which learned about 2.2 million words through the Common Crawl dataset [22],[23]. We expected that word embedding learned through our new dataset would perform better than using pre-trained word embedding models because of the characteristics of patent data where technical/ specific terms frequently appear. To confirm this, we conducted an experiment using both the pre-trained word embedding model and the word embedding model that learned with 300 dimensions vectors based on the new dataset, and tried to find the word embedding model with the highest performance.

Patent documents are highly structured documents that consist of several fields such as IPC, Title, Abstract, and Claim. Title and Abstract mean the title and the abstract of the patent. Claim clause is a representative field that differentiates patent documents and describes the value of their ideas for new discoveries and inventions. This is likely to have a negative impact on classification performance, because this reflects the contents of a new idea the most and shows the usefulness of the patent. However, it is necessary to consider the claims that contain the differentiation of patent documents to classify patent data into subdivided technology and industrial fields [24]. We thought IPC, a hierarchical international standard classification code, could be an important element for classification. To confirm this, we made a dataset with different combinations of patent fields and compared the performance by the dataset.

The final step, Classification State, is composed of machine learning and deep learning model that receive text data converted into vectors and output class predictions. There are

various success models for text classification, including the deep learning network based on Convolution Neural Network (CNN) and the deep learning network based on Recurrent Neural Network (RNN) [16]-[18]. Patent data classification belongs to the Extreme multi-label text classification (XMTC) task because there can be multiple labels in one document and the number of labels is as many as about 1,400. There are many successful models based on deep learning for XMTC [25],[26]. We classified patent documents by using AttentionXML and LAHA, which are typical XMTC models, in order to find a classification model with higher accuracy.

We aimed for the effective automatic classification of documents considering the characteristics of patent documents for about 1 million patent data and about 1,400 classes. we 1) compared the effect of word normalization by Stemming algorithm on performance in the preprocessing, 2) conducted comparison experiments to find the word embedding model that makes the best performance in patent classification, and compared 3) the performance depending on the combination of patent data fields and 4) the performance depending on the deep learning network models for XMTC. This paper is as follows: Section 2 describes the background of this study, and related works. Section 3 describes the experiments conducted in this study. Section 4 shows the experimental results and explains the analysis of the results. Section 5 deals with conclusions and future studies.

II. BACKGROUND AND RELATED WORKS

This section first presents a brief overview of text data preprocessing and word embedding. Next, we summarize the documents related to the patent classification algorithm, and the documents related to the XMTC algorithm predicting more than one label when a large number of labels exist.

A. PREPROCESSING AND WORD EMBEDDING

akwl

B. PATENT CLASSIFICATION

With the increasing importance of patents and the technological advances in text classification, a number of studies on the classification of patent documents have been conducted. For the patent classification, various methods such as methods through the most common algorithms such as ANN, SVM, and kNN and methods through deep learning have been proposed. Because patent analysis has no standard dataset associated with it, the standards for the performance of existing studies are not clear. Because the performance depends on data distribution by data sampling and by class, and depends on the average number of labels per document, it is very difficult to compare between studies. Reference [6], a paper published in 2017, classified IPC subclasses through the CBOW word embedding model and Long Short Term Memory (LSTM) network. The experiment was conducted with different the dimensions and input lengths of the word embedding model. In this case, the USPTO dataset showed an accuracy of 63% when using 200-dimensional word embedding and 150-length input. In this experiment, the basic LSTM was used, resulting in a vanishing gradient problem when the input was lengthened, which rather reduced the performance. In this paper, to prevent this, we used a network based on Bi-LSTM, which is less affected to vanishing gradient than basic LSTM. Reference [29] proposed a hierarchical feature extraction model (HFEM) for the multi-label classification of patents, which performs classification through Bi-LSTM after N-Gram Feature Extraction based on CNN. They argued that performing Feature Extraction through CNN shows higher performance than when word embedding was only performed. As a result of the experiment, when only 96 IPCs corresponding to the machine area were classified, the accuracy of 80.54% was shown in the Precision top 1 and 31.69 in Precision top 5. Reference [30] proposed an IPC patent classification model using a Convolutional Neural Network (CNN) and a word embedding-based network. They conducted learning and evaluation using USPTO-2M Datasets. In the experiment using USPTO-2M, 637 IPC classification showed an accuracy of 73.88% in Precision top 1 and more than 30% in Precision top 5. [31] classified the CPCs of patent documents through Fine-Tuning of BERT which achieved SOTA in many NLP Tasks. They showed high performance improvements with more than 80% accuracy through simple Fine-Tuning. We thought that the inclusion of IPC and CPC in the dataset has a significant impact on the performance improvement, so we tried to check the performance depending on whether IPC is included or not.

III. EXPERIMENTS

This section provides a brief discussion about the composition of our experiments, the dataset used for experiment, data sampling for build dataset, preprocessing methods, word embedding methods, and Classification Algorithms for Extreme Multi-label Classification. Fig.1 is a picture of the experimental process and the techniques and methods involved in each process.

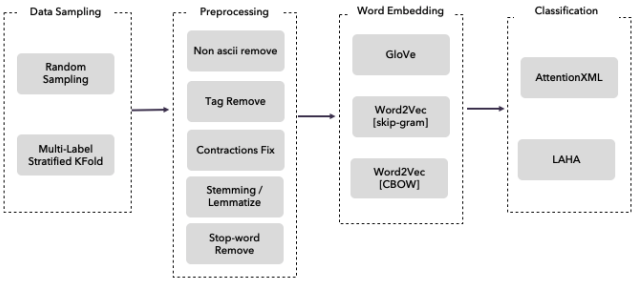


FIGURE 1. Experimental process and the techniques and methods involved in each process

A. DATASET

This section describes the dataset used in this study. The dataset in this study used about 1 million patent data filed in

Europe, the United States, and Japan. Our new dataset is the original document without any preprocessing. The number of classes to be classified are 1383 to cover all technical and industrial fields. The amount of data allocated to each class is not evenly distributed. For the class with the least data, only eight data has the corresponding class, and the class with the most data contains more than 30,000 data. All patent data used in the experiment were classified by patent experts such as patent attorneys and patent analysis researchers. Patent documents contain many ambiguous, specific and technical terms, and is composed of highly structured data fields such as bibliographic written with IPC and application number, title, abstract, drawings, claims, description, etc. This study aims to classify patent documents using text data. To this end, the dataset is composed of various combinations of IPC, title, abstract, and claim data fields containing the contents of the patent. Our new dataset is multi-label data where there can be more than one class in one patent data. For example, 'Method Apparatus Pipe Image Chemical Analysis' patent has four categories: 'laser light parts technology','UI/UX for HDM','water pollution measurement system', and 'manufacturing robot'. Most of them have between 1 and 5 classes, but there are also patent data with more classes. The patent data with the most classes have 37 classes. The average number of classes per document is 1.44, which is very small among the XMTC datasets. The average number of classes per document has a significant impact on performance measurement. The smaller the average number of classes, the more accurate prediction is required, resulting in poor performance in multiple class predictions index.

TABLE 1. Dataset Statistics

N	V	M	D	L	$\hat{L}$	$\check{L}$
791,839	70,000	128,116	768,151	1383	1.44	2086.65

N is the number of training instances, V is the number of validation instances, M is the number of test instances, D is the total number of features, L is the total number of class labels, $\hat{L}$ is the average number of label per document, $\check{L}$ is the average number of documents per label

B. DATA SAMPLING

Since we use a new dataset, we need to divide the dataset into learning data, validation data and experimental data for learning. Data partitioning should ensure that the proportion of each class is approximately equal to that of the entire dataset. For example, to divide a dataset by 5:5, the proportion of each class should also be 5:5. In addition, the learning data and the experimental data should be divided respectively so that at least one data corresponding to the class exists. For this, the most widely used data sampling method is the hierarchical sampling method. If one data has only one class, sampling data using the typical Stratified Cross validators will not cause any problems. However, if typical layered data sampling is used in the Multi-Label Dataset, this can cause problems. For rare classes, the test dataset with a small proportion may not contain more than one data [9]. Such abnormal data sampling does not cause any problem in the experiment, but may degrade the reliability of the experiment. In general, prediction for a class with less data shows relatively low accuracy compared to a class with a lot of data [32]. When the test data for the rare class does not exist due to abnormal data sampling, the experimental evaluation index appears to be high performance. In this study experiment, to solve the problems that may occur in the data sampling process of Multi-Label as shown above, the learning dataset and the test dataset were sampled with an 8:2 ratio through iterative-stratification's Multi-Label Stratified Kfold, which implements the Stratified shuffle split method, In addition, for verification, we chose 70,000 data from the test data and conducted the test using the remaining data. Fig. 2 is the result of segmentation of the train/test data of the rare class with less than 50 data as a result of data sampling. It can be seen that at least one data is included in the learning train and the test dataset even in the rare class with a small number of data.

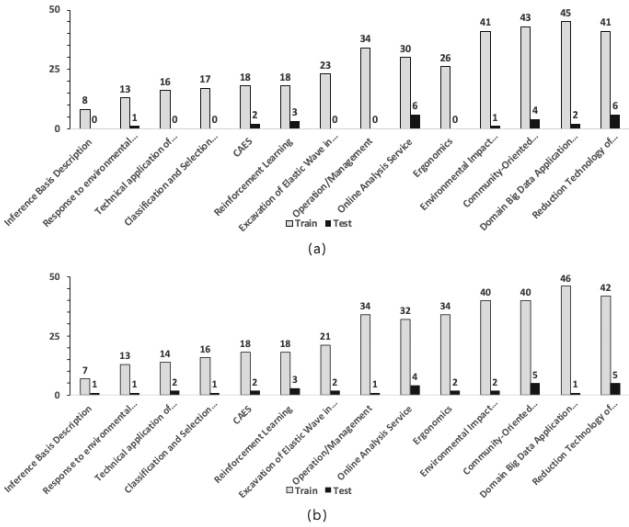


FIGURE 2. Result of segmentation of the train/test data of the rare class with less than 50 data. (a) Not consider multi-label data (b) Consider multi-label data

C. PREPROCESSING

In this paper, we apply five major standard NLP preprocessing steps such as non-ASCII removal, Tag removal, Contractions fix, stop-words removal, and Lemmatize to all datasets, and compare the performance change with or without the Stemming algorithm. Non-ASCII removal and Tag removal are steps to remove the elements that do not have significance and that increase complexity in the text classification. In the case of'-', which means a linking word, if you remove it, the word and the word are connected to become a new word. To prevent this, the connection word was replaced with space. Abbreviations can cause a lot of confusion even for humans.

The NLP model cannot recognize that abbreviations are abbreviations of words, which can intensify the confusion of the model. In the case of patent documents, there are no rules for the use of abbreviations. Therefore, the usage is determined by the author of the document, and the abbreviations are used depending on the document. Such unstructured representation increases the complexity of the model. In order to prevent this increase in complexity, we proceeded with the Contractions fix step to remove the abbreviations. The most commonly used words are called stop-words. Stop-words are words such as prepositions, determiners, and articles, and they are used with high frequency and do not have additional information. Complexity reduction through the removal of stop-words is very helpful in improving performance [33].
Lemmatize and Stemming are the preprocessing steps that serve to generalize words by extracting stems of words with overlapping meanings from the corpus and by integrating them into one. Both have a common function of reducing the word variants, but there is a subtle difference between them. Lemmatizing transforms word forms into prototypes after understanding the Part of Speech (POS) and the context. Stemming finds the root of a word without considering the POS and the context, and transforms it by applying a set of rules. Since Stemming extracts stems using a series of rules, there are several types of algorithms depending on which rules are applied, and the performance also depends on the algorithm.
Considering the characteristics of patent data that include data from multiple domains, we thought that reducing the complexity of Feature through normalization of words would reduce the performance. We considered the difference in Classification performance depending on the application of Lancaster Stemming Algorithm and Snowball Stemming Alogrithm [34],[35].

D. WORD EMBEDDING

Word embedding is a method used to effectively express words. In many classification tasks, TF-IDF was used as the traditional word embedding model. The neural network-based word embedding model has drawn attention as an alternative because it allows words to be expressed as vectors more effectively than TF-IDF does. The purpose of word embedding is to learn the vector representation of words by properly mapping the semantic contextual information of the word into the vector space. word embedding works as a core function in many NLP Tasks and has a great impact on performance. There are several ways to learn the word embedding model, and the performance change is specific to the Task.
Both GloVe and Word2Vec conduct learning based on the central word and surrounding words. GloVe makes it easier to measure the similarity between the embedded words by using the number of words in documents and the probability of simultaneous appearance, and reflects statistical information of the entire corpus. For this, in GloVe's learning, the scalar product of the two embedded words is the probability of simultaneous appearance of the entire corpus. There are two ways of learning in the Word2Vec model. The first method is Continuous Bag-of-Words (CBOW), which conducts learning by predicting the masked central word through surrounding words. The second method is Skip-gram, which predicts surrounding words through the central word. According to the author who presented the Word2Vec model, CBOW can express words better syntactically and Skip-gram can express words better semantically [15].
word embedding models are models that pre-trained a large number of words based on a large amount of dataset. When considering the characteristics of patent documents where technical/ specific terms and IPC codes frequently appear, we thought that the word embedding model learned based on the new dataset would show higher performance. To confirm this, an experiment was conducted to compare the pre-trained word embedding model with the word embedding model learned based on our dataset. Since the pre-trained word embedding models used words expressed in 300-dimensional vectors, the newly learned word embedding models also used also 300-dimensional vectors. The word embedding models we created were learned in GloVe, CBOW and Skip-gram, respectively.
The common problem when using word embedding is the out-of-vocabulary (OOV) problem. word embedding is to convert a word learned in advance into a proper vector, so it cannot process a new word. In this experiment, 500,000 words initialized with random vectors were listed, and words in the word embedding model were entered into the vocabulary list. And then, the words that first appeared are mapped with random vectors, and the vocabulary lists could be learned together when learning the classification model. OOV problem was solved by setting all words not included in the vocabulary list to the value of <unk>

TABLE 2. Word Embedding Model Hyperparameters for learning

Dimension Size	300
Window size	15
Min Count	3 or 5
Iteration	15
Negative Sampling	10

E. CLASSIFICATION MODEL

In this study, we classified classes by using AttentionXML model and LAHA model that achieved SOTA in Extreme Multi-label Classification, and compared their performance to find the most suitable model for classification.
AttentionXML. This is an Extreme Multi-Label Text Classification model released in 2019, which consists of Bi-LSTM layer, attention layer, and FC layer. In terms of the overall model structure, there seems to be no difference from Text Classification using LSTM and Attention Mechanism, but

it showed better performance than previous Extreme Multi-Label Text Classification by converting Label into Probabilistic Label Tree (PLT) and utilizing it. Even when using simple AttentionXML without PLT, the performance proved to be better than other models through experiments. In this study, the experiment was conducted using basic AttentionXML without converting the label into PLT.

LAHA. As a model released after AttentionXML in 2019, the network was constructed using Bi-LSTM and Attention Mechanism in the same way as AttentionXML, but there is a difference in that it uses label embedding that utilizes the simultaneous appearance of labels. After drawing a graph of the simultaneous appearance of labels through NetworkX, a label embedding model to show the relationship between labels through Node2Vec is used. The experiments proved that several XMTC dataset could obtain higher accuracy than AttentionXML. Fig.3 shows the network structure for Label Attention added in LAHA.

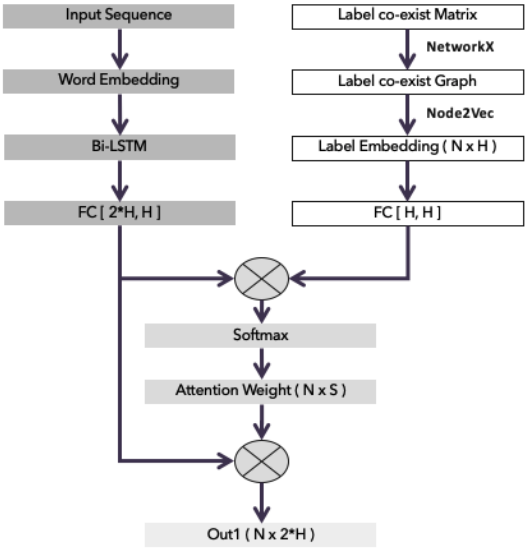


FIGURE 3. , and Network Structure of LAHA Classification Model's Interaction Attention Layer [26]

F. EVALUATION METRICS

The patent classification, the purpose of this study, is a multi-label classification. To measure the performance of each experiment, we used the ranking-based evaluation index Precision at n ($P@n$) and normalized Discounted Cumulative Gain at n ($nDCG@n$). Both evaluation indexes are widely used in multi-label classification and are defined as follows.

$$nDCG@n = \frac{1}{\sum_{i=1}^{min(|R|,k)} * \frac{1}{log_2(i+1)}} \sum_{i=1}^{k} \delta(i \in R) \frac{1}{log_2(i+1)} \quad (1)$$

In $P@n$, $y \in 0, 1^k$ means the ground truth label for each data, and $r_n(\hat{y})$ means the index of the label when n items with the highest prediction scores are extracted. The greater the values of both $P@n$ and $nDCG@n$, the better performance.

IV. RESULT

In this section, we made comparative analysis of the experimental results. All experimental environments are as shown in Table 3.

TABLE 3. Experimental Environment

OS	Ubuntu 16.04 LTS
CPU	Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz
RAM	192GB
GPU	Tesla V100 32GB 10TFLOP

A. WORD EMBEDDING

In the experiment in this section, datasets that included IPC, Title, Abstract, and Claim and did not apply the Stemming algorithm were used. Table 4 shows the words included in the learning dataset, the number of corpus of each word embedding, and the number of words not included in the corpus. The number of all words included in the learning dataset is about 770,000, including the IPC code. All IPC codes are classified into subgroups, and a total of IPC codes in the learning dataset is 54,580. In the case of the word embedding learning, the learning proceeds except for those that appear less than the minimum number of appearances. So, we thought that the minimum number of appearances was reduced, the higher performance would be achieved. To confirm this, the Embedding models were learned by setting the minimum number of appearances of words to 3 and 5, respectively.

TABLE 4. Corpos Size and Number of Unkown Token according by Word Embedding Model

	Custom Embedding Model		Pre-Trained Embedding Model	
	min_count = 3	min_count = 5	Word2Vec	GloVe
Vocab Size	351,678	249,083	3,000,000	2,200,000
UNK Token	416,473	519,068	703,715	651,288
IPC	39,027	32,011	0	0
Train Dataset Total Token (Include IPC : 54,580) : 768,151				

The patent documents contain a number of specific words and proper nouns including IPC. In about 800,000 learning dataset, the number of words that appear more than 3 times is less than the number of words that appear less than 3 times. We compared the two while considering the characteristics of these patent documents. 1) the performance change depending on the minimum number of appearances when learning an Embedding model based on learning data. 2) the performance change when the complexity is relatively reduced by using only general words in a pre-trained model.

Table 5 shows the difference in performance depending on the minimum number of appearances when learned based on the newly learned dataset in the AttentionXML model. When learning word embedding, words that appear less than the minimum number of appearances are excluded. So we thought that if there are fewer excluded words, words can be better expressed. Almost all the experiment showed that performance improved by more than 2% when the minimum number of appearances was set to 3.

TABLE 5. Accuracy of AttentionXML Patent Classification by min count and Word Embedding Model

Model	Min Count	Evaluation Method	Custom GloVe	Custom Skip-gram	Custom CBoW
AttentionXML	3	P@1	66.152%	66.842%	**70.519%**
		P@3	34.272%	34.308%	**36.168%**
		P@5	23.019%	22.903%	**23.994%**
		nDCG@3	71.985%	72.193%	**76.212%**
		nDCG@5	74.759%	74.744%	**78.652%**
	5	P@1	66.176%	66.112%	68.128%
		P@3	33.907%	34.208%	35.143%
		P@5	22.650%	22.913%	23.480%
		nDCG@3	71.319%	71.800%	73.923%
		nDCG@5	73.878%	74.477%	76.552%

Table 6 shows the classification performance by using the pre-trained word embedding model and Embedding Model we trained. The pre-trained GloVe model contains fewer words than the GloVe model and Skip-gram model we trained. However, as a result of the experiment, the pre-trained GloVe model shows better performance. We know that to enable many words embedded does not have always good performance. Fig.4 is a graph of the performance difference for each word embedding model.

The results of the experiment show that the word embedding model trained with the CBOW method, which is a method of predicting the central word through surrounding words using the learning dataset, by setting a minimum number of appearances as 3, has the highest performance : an accuracy of 70.519%, 36.168%, and 23.994%, respectively in Precision Top n. All subsequent experiments used the corresponding word embedding model based on this. We conducted subsequent experiments by using the CBOW model that showed the best performance in the experiment.

B. DATA FIELD

In order to examine the effects of each field of patent documents on the classification performance, we organized the following four experiments with Title, Abstract, Claim, and IPC. First, the classification performance was compared when only Title and Abstract were used, and when only Claim was used. Next, the case of using the remaining fields except IPC and the case of using all fields including IPC were compared. Table 7 shows the average word length depending on each field and a combination of fields. Data corresponding to Title and IPC fields must exist in all patent documents. Unlike this, in the case of Abstract and Claim fields, there may be no text data in the corresponding field.

We conducted an experiment using a combination of patent documents composed of the following four different fields. Table 8 shows classification performance according to data fields in the same classification model and word embedding model. We thought that using only the Title and IPC fields would not be able to obtain more than a certain level of classification performance because the length of the input data was too short. So, an experiment using only Title and IPC was not conducted. In the experiment using Title and Abstract fields, the P@1 index, which determines whether one prediction is in the correct answer label, showed 64.081% accuracy. On the other hand, when Claim field was only used, it showed the lowest classification performance with an accuracy of 49.160%. Since Claim field mainly describes the differentiation of the corresponding patent, it shows low performance in

TABLE 6. Accuracy of AttentionXML Patent Classification by min count and Word Embedding Model

Model	Word Embedding	Evaluation Method	GloVe	Skip-gram	CBoW
AttentionXML	Pre-Trained	P@1	67.429%	60.925%	-
		P@3	34.242%	32.056%	-
		P@5	22.769%	21.738%	-
		nDCG@3	72.293%	66.769%	-
		nDCG@5	74.708%	69.721%	-
	Custom	P@1	66.176%	66.842%	**70.519%**
		P@3	33.907%	34.308%	**36.168%**
		P@5	22.650%	22.903%	**23.994%**
		nDCG@3	71.319%	72.193%	**76.212%**
		nDCG@5	73.878%	74.744%	**78.652%**

TABLE 7. The average word length per data field

Field	Average word length
Title	7.93
Abstract	63.05
Claim	406.22
IPC	3.26
Title+Abstract	70.98
Title+Abstract+Claim	474.63
IPC+Title+Abstract+Claim	477.90

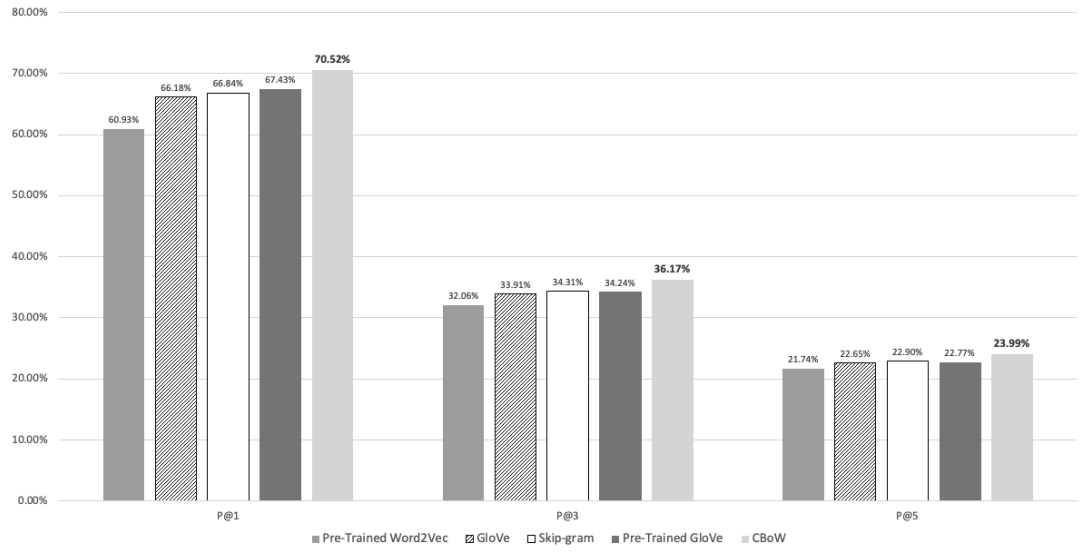


FIGURE 4. Result of segmentation of the train/test data of the rare class with less than 50 data. (a) Not consider multi-label data (b) Consider multi-label data

the classification model that predicts a label based on a word. However, the low performance of Claim-only classification does not mean that Claim is not necessary for classification. According to the third experiment, the experiment using Title, Abstract, and Claim show higher performance than the experiment using only Title+Abstract. It can be seen that Claim is helpful for the classification. Most noteworthy is the performance change in the experiment using all fields including IPC. Only the shortest IPC field of the fields was added, which resulted in a performance improvement of about 5%, on average. Since IPC is a stratified standard classification code, it can be seen that IPC plays as an important index for the classification. Because the pre-trained model does not include the IPC code, it can be considered to have classified through the remaining fields excluding IPC. In the fields excluding IPC, CBOW is 65.584%, which is less accurate than pre-trained GloVe. This shows that the reason why CBOW showed higher accuracy than pre-trained GloVe is that it was able to learn and embed IPC. When learning CBOW word embedding, higher performance can be expected by adding a dataset other than the current ones.

C. DATA LENGTH

Table 9 shows the performance change when the length of the input data is changed. Both dataset utilized all fields including IPC, and used CBOW models based on learning data. As a result, when the length of the input for learning and evaluation is set to 256, the accuracy is 69.104% decreased by about 1.4% compared to the case of 512. This result can be seen from the performance change depending on the presence or absence of Claim above. The average length of Title+Abstract is 70.98, and the part that is removed from the length change is Claim field. Because Claim field represents the specificity of a patent, it shows low performance when used alone. However, it acts as a factor to improve performance when used together with Title and Abstract. Therefore, when the length is set to 512, more Claim data is included, resulting in this performance difference.

TABLE 8. Accuracy of Custom CBoW word Embedding + AttentionXML Patent Classification by Data Field

Evaluation Method	Title+Abstract	Claim	Title+Abstract Claim	IPC+Title +Abstract +Claim
P@1	64.081%	49.160%	65.584%	**70.519%**
P@3	33.478%	26.937%	34.349%	**36.168%**
P@5	22.513%	18.740%	23.055%	**23.994%**
nDCG@3	69.830%	56.033%	71.856%	**76.212%**
nDCG@5	72.563%	59.393%	74.619%	**78.652%**

TABLE 9. Accuracy of Patent Classification by input data length

Model	Word Embedding	Evaluation Method	len = 256	len = 512
AttentionXML	Custom CBoW	P@1	69.104%	**70.519%**
		P@3	35.519%	**36.168%**
		P@5	23.653%	**23.994%**
		nDCG@3	74.716%	**76.212%**
		nDCG@5	77.251%	**78.652%**

D. PREPROCESSING

In this study, we tried to compare the performance of the Stemming algorithm with or without application through experiments. The Stemming algorithm generalizes words by extracting stems. The effect of this generalization on the performance was confirmed. We used typical Stemming algorithms such as the Lancaster algorithm and Snowball algorithm. Table 10 shows the total number of words when with or without each Stemming algorithm applied, and show the number of Unique Words when the minimum number of appearances is 3 when learning word embedding. It can be seen that the Lancaster algorithm generalizes words better than the Snowball algorithm does. As a result of the actual application of algorithms, in the case of Snowball, there was no significant change in words, when compared to the case Stemming was not applied. However, in the case of Lancaster, there was a big change. For example, when applying Stemming algorithms to the adjective 'organic', the Lancaster Algorithm converted it to 'org' and the Snowball Algorithm converted it to 'organ'.

TABLE 10. Total number of words and unique words when with or without each Stemming algorithm applied

	Lancaster	SnowBall	No Stemming
Total words	640,198	684,693	768,151
Total Unique Words	289,501	310,653	351,678

Table 11 is the result of experimental analysis with or without the application of Stemming Algorithms. Since patent data contains data from various domains and has many labels for the classification, it was thought that the generalization of words by Stemming would degrade the performance. The actual experiment showed the lowest accuracy of 67.166% when applying the Lancaster algorithm with the highest degree of normalization, and showed the highest performance of 70.519% when applying the Stemming algorithm, This shows that considering the characteristics of patent data, the text generalization through the Stemming algorithm degrades performance.

TABLE 11. Accuracy of Custom CBoW word Embedding + AttentionXML Patent Classification by Stemming Algorithm

Model	Word Embedding	Evaluation Method	Lancaster	SnowBall	No Stemming
AttentionXML	Custom CBoW	P@1	67.466%	70.264%	**70.519%**
		P@3	34.913%	36.048%	**36.168%**
		P@5	23.372%	23.936%	**23.994%**
		nDCG@3	73.281%	75.957%	**76.212%**
		nDCG@5	75.982%	78.411%	**78.652%**

E. MODEL

We conducted experiments using AttentionXML and LAHA models, which are SOTA models for XMTC, and compared the performance of the two models. Table 12 shows the performance change depending on the classification model when the dataset and word embedding model with the highest accuracy in the experiments above were applied. As a result of the experiment, the LAHA classification model showed higher accuracy in all evaluation indexes than the AttentionXML model. Experiments with our new dataset did not show as much performance difference as published in the LAHA paper. We determined that this was due to the average number of labels per document. The average number of labels per document in our dataset is 1.44, which is smaller than the dataset for other XMTCs with an average number of labels of 2 or more. For this reason, label embedding-based Attention did not bring much effect.

TABLE 12. Classification performance according to XMTC model

Word Embedding	Evaluation Method	AttentionXML	LAHA
Custom CBoW	P@1	70.519%	**70.615%**
	P@3	36.168%	**36.187%**
	P@5	23.994%	**24.020%**
	nDCG@3	76.212%	**76.276%**
	nDCG@5	78.652%	**78.724%**

Our ensemble consisted of the following Classifiers and conducted an experiment. 1) Non Stemming preprocessing DataSet, CBOW word embedding, and Attention XML 2) Non Stemming preprocessing Dataset, CBOW word embedding, and LAHA. The ensemble technique we used is an ensemble method using a very simple summation. The top 100 predictions of each Classifier are extracted and the prediction scores of the same label are added. Through a simple linear ensemble process of two Classifiers, the performance improvements of about 1.3% at P@1, 0.6% at P@3, and 0.3% at P@5 were obtained.

TABLE 13. Comparison table of classification performance between single and ensemble models

Word Embedding	Evaluation Method	Single Model	Ensemble	Change
Custom CBoW	P@1	70.615%	**71.896%**	+1.281%
	P@3	36.187%	**36.697%**	+0.510%
	P@5	24.020%	**24.301%**	+0.281%
	nDCG@3	76.276%	**77.474%**	+1.198%
	nDCG@5	78.724%	**79.852%**	+1.128%

V. CONCLUSION

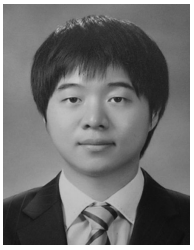
In this paper, we studied the effects of the Stemming strategy, the word embedding, the data field, and the classification model on the classification of patent documents. We examined three Stemming strategies: Lancaster algorithm, snowball algorithm, and no stemming. Then, we looked at the word embedding models for the following five word representations: pre-trained models such as Word2Vec, GloVe, and Skip-gram, CBoW, and GloVe models which trained based on patent dataset. In addition, when learning the word embedding model, we examined the performance change accordingly by varying the minimum number of appearances. Finally, the effect of each data field on the patent classification was checked by using different combinations of data fields. The AttentionXML classification model based on Self-Attention mechanism and Bidirectional learning, and the LAHA classification model, which added layers to AttentionXML to consider the relationship between labels and words, were used for experiments. For the experiment, we divided about 1 million patent data into about 800,000 dataset for learning, 70,000 validation dataset, and 130,000 test dataset considering the characteristics of multi-label data. In addition, the results of the experiment were proven by using the evaluation index of Precision and Normalized Discounted Cumulative Gain. The Stemming algorithm degraded the performance in various deep learning-based text classification, and it was thought that it would work the same for the patent document classification. To confirm this, an experiment was conducted with two typical algorithms and data that were not applied. As a result, the unapplied data showed the highest performance, and the algorithm with the highest degree of generalization of words was applied showed the lowest performance. We compared the classification performance depending on the word embedding model, and we obtained the highest classification accuracy of 70.519% when using the word embedding model trained through CBOW. Through the additional performance comparison depending on the combination of data fields, it was found that the high classification accuracy of the CBOW model was largely due to the IPC classification codes. In addition, considering the characteristics of patent data, in which rare words appear frequently and the word embedding learning, which proceeds with learning except for words that appear less than the minimum number of appearances, we thought when the minimum number of appearances is smaller, it would lead to higher performance. we confirmed that a smaller minimum number of appearances can result in higher classification accuracy through experiments. We also confirmed the influence of the Stemming strategy, the word embedding, the data field and the classification model in the patent classification through the experiments with several conditions. As a result, when stemming was not applied, when the CBoW embedding model learned with a smaller minimum number of appearances was applied, and when the LAHA classification model was used, the highest classification accuracy was recorded. It was also confirmed that the IPC data field has a great influence on performance improvements. Finally, we obtained an accuracy of 71.896% at P@1, 36.697% at P@3, and 24.301% at

REFERENCES

[1] Li Q., Maggitti P. G., Smith K. G., Tesluk P. E., Katila R. "Top management attention to innovation: The role of search selection and intensity in new product introductions,"Academy of Management Journal, 2013, 56(3), pp 893—916.
[2] Wagner, Stefan and Wakeman, Simon. "What do patent-based measures tell us about product commericalization? Evidence from the pharmaceutical industry," Research Policy, 2016, 45(5), pp 1091–1102
[3] WIPO, "World Intellectual Property Indicators 2019," Geneva: World Intellectual Property Orgamization, 2019
[4] Yuen-Hsien Tseng, Chi-Jen Lin and Yu-I Lin. "Text mining techniques for patent analysis," Information Processing&Management, 2007, 43(5), 1216–1247.
[5] Noh, Heeyong and Jo, Yeongran and Lee, Sungjoo, "Keyword selection and processing strategy for applying text mining to patent analysis," Expert Systems with Application, 2015, 42(9), 4348–4360
[6] M. F. Grawe and C. A. Martins and A. G. Bonfante, "Automated Patent Classification Using Word Embedding," 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), 2017, 408–411
[7] WIPO, "International Patent Classification," Guide, Geneva : World Intellectual Property Organization, 2020
[8] Taherdoost, Hamed, "Sampling Methods in Research Methodology; How to Choose a Sampling Technique for Research," International Journal of Academic Research in Management, 2016, 5, 18–27
[9] Sechidis, Konstantinos and Tsoumakas, Grigorios and Vlahavas, Ioannis, Hamed, "On the Stratification of Multi-label Data," Machine Learning and Knowledge Discovery in Databases, 2011, 145–158
[10] Uysal, Alper Kursat and Gunal, Serkan, "The Impact of Preprocessing on Text Classification," Inf. Process. Manage, 2014, 50(1), 104–112
[11] Wael Etaiwi and Ghazi Naymat, "The Impact of applying Different Preprocessing Steps on Review Spam Detection," Procedia Computer Science, 2017, 113, 273–279
[12] K. Dharavath and G. Amarnath and F. A. Talukdar and R. H. Laskar, "Impact of image preprocessing on face recognition: A comparative analysis," 2014 International Conference on Communication and Signal Processing, 2014, 631–635
[13] Salton, G. and Wong, A. and Yang, C. S., "A Vector Space Model for Automatic Indexing," Communications of the ACM, 1975, 18(11), 613–620
[14] Pennington, Jeffrey and Socher, Richard and Manning, Christopher D, "Glove: Global vectors for word representation," Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, 1532–1543
[15] Mikolov, Tomas and Chen, Kai and Corrado, Greg and Dean, Jeffrey, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013
[16] Zhou, Chunting and Sun, Chonglin and Liu, Zhiyuan and Lau, Francis, "A C-LSTM neural network for text classification," arXiv preprint arXiv:1511.08630, 2015
[17] Yao, Liang and Mao, Chengsheng and Luo, Yuan, "Graph convolutional networks for text classification," Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33, 7370–7377
[18] Yu, Shanshan and Su, Jindian and Luo, Da, "Improving BERT-based text classification with auxiliary sentence and domain knowledge," IEEE Access, 2019, 7, 176600–176612
[19] Almuzaini, Huda Abdulrahman and Azmi, Aqil M, "Impact of Stemming and Word Embedding on Deep Learning-Based Arabic Text Categorization," IEEE Access, 2020, 8, 127913–127928
[20] Bounabi, Mariem and Moutaouakil, Karim El and Satori, Khalid, "A comparison of text classification methods using different stemming techniques," International Journal of Computer Application in Technology, 2019, 60(5), 298–306
[21] Amin, Samina and Uddin, M Irfan and Hassan, Saima and Khan, Atif and Nasser, Nidal and Alharbi, Abdullah and Alyami, Hashem, "Recurrent neural networks with TF-IDF embedding technique for detection and classification in tweets of dengue disease," IEEE Access, 2020, 8, 131522–131533

[22] Caliskan, Aylin and Bryson, Joanna J and Narayanan, Arvind, "Semantics derived automatically from language corpora contain human-like biases," Science, 2017, 356(6334), 183–186
[23] Cerisara, Christophe and Kral, Pavel and Lenc, Ladislav, "On the effects of using word2vec representations in neural networks for dialogue act recognition," Computer Speech&Language, 2018, 47, 175–193
[24] Benzineb, Karim and Guyot, Jacques, "Automated Patent Classification. Current Challenges in Patent Information Retrieval," Current challenges in patent information retrieval, 2011, 239–261
[25] You, Ronghui and Zhang, Zihan and Wang, Ziye and Dai, Suyang and Mamitsuka, Hiroshi and Zhu, Shanfeng, "AttentionXM: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification," Advances in Neural Information Processing Systems, 2019, 5820–5830
[26] Huang, Xin and Chen, Boli and Xiao, Lin and Jing, Liping, "Label-aware Document Representation via Hybrid Attention for Extreme Multi-Label Text Classification," arXiv preprint arXiv:1905.10070, 2019
[27] Yang, Xiao and Macdonald, Craig and Ounis, Iadh, "Using word embeddings in twitter election classification," Information Retrieval Journal, 2018, 21(2-3), 183–207
[28] Aydoğan, Murat and Karci, Ali, "Improving the accuracy using pre-trained word embeddings on deep neural networks for Turkish text classification," Physica A: Statistical Mechanics and its Applications, 2020, 541, 123288
[29] Hu, Jie and Li, Shaobo and Hu, Jianjun and Yang, Guanci, "A hierarchical feature extraction model for multi-label mechanical patent classification," Sustainability, 2018, 10(1), 219
[30] Li, Shaobo and Hu, Jie and Cui, Yuxin and Hu, Jianjun, "DeepPatent: patent classification with convolutional neural networks and word embedding," Scientometrics, 2018, 117(2), 721–744
[31] Lee, Jieh-Sheng and Hsiang, Jieh, "PatentBERT: Patent classification with fine-tuning a pre-trained bert model," arXiv preprint arXiv:1906.02124, 2019
[32] Thabtah, Fadi and Hammoud, Suhel and Kamalov, Firuz and Gonsalves, Amanda, "Data imbalance in classification: Experimental evaluation," Information Sciences, 2020, 513, 429–441
[33] Silva, Catarina and Ribeiro, Bernardetei, "The importance of stop word removal on recall values in text categorization," Proceedings of the International Joint Conference on Neural Networks, 2003., 2003, 3, 1661–1666
[34] Hooper, R. and Paice, C., "The Lancaster Stemming Algorithm.," Available at: http://www.comp.lancs.ac.uk/computing/research/stemming/
[35] Porter, M.F., "Snowball: a language for stemming algorithms," Available at: http://www.snowball.tartarus.org/texts/introduction.html.

GUIK JUNG received the B.S. degree in Department of Software from Soongsil University, in 2019, where he is currently pursuing the master's degree with the Graduate School of Software. His research interests include Natural Language Processing, Lightweight Deep learning and Cloud Computing.



JUNGHOON SHIN received the B.S. degree in Information and Communication Engineering from National Institute for Lifelong Education, Seoul, Korea, in 2007 and the M.S. and Ph.D degree in System Software from Soongsil University, Seoul, Korea, in 2010 and 2014, respectively. He is currently an Adjunct professor with the Department of Convergence Software, Soongsil University, Seoul, Korea. He is also the Director of the Wert Intelligence Co. Ltd. His current research interests include BigData, Database System, AI, Data Mining, Cloud Computing and Patent Analysis System.

THIRD C. AUTHOR, JR. received his MS and BS degrees in the Department of Computer Engineering from Seoul National University, Seoul, Korea, in 1996 and 1998, respectively and PhD in the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea in 2004. He is currently a professor of the School of Software, Soongsil University, Seoul, Korea. His current research interests include database systems, multimedia systems, and cloud computing.