J Inf Process Syst, Vol.19, No.6, pp.830~841, December 2023
https://doi.org/10.3745/JIPS.04.0296

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Word-Level Embedding to Improve Performance of Representative Spatio-temporal Document Classification

Byoungwook Kim[1] and Hong-Jun Jang[2,*]

Abstract

Tokenization is the process of segmenting the input text into smaller units of text, and it is a preprocessing task that is mainly performed to improve the efficiency of the machine learning process. Various tokenization methods have been proposed for application in the field of natural language processing, but studies have primarily focused on efficiently segmenting text. Few studies have been conducted on the Korean language to explore what tokenization methods are suitable for document classification task. In this paper, an exploratory study was performed to find the most suitable tokenization method to improve the performance of a representative spatio-temporal document classifier in Korean. For the experiment, a convolutional neural network model was used, and for the final performance comparison, tasks were selected for document classification where performance largely depends on the tokenization method. As a tokenization method for comparative experiments, commonly used Jamo, Character, and Word units were adopted. As a result of the experiment, it was confirmed that the tokenization of word units showed excellent performance in the case of representative spatio-temporal document classification task where the semantic embedding ability of the token itself is important.

Keywords

Spatio-temporal Document Classification, Tokenization, Word-Level Embedding

1. Introduction

Recently, with the development of internet technology and the spread of smart devices, a vast amount of text data is being produced digitally. Text data is being generated exponentially through various media such as online news, personal blogs, and social media. As such a large amount of information can be easily acquired, the need for document classification for more efficient document management has rapidly increased. Document classification means classifying text documents by defining them into two or more categories or classes. A representative algorithm used for document classification is the k-nearest neighbor (KNN) [1]. Naive Bayes algorithm is a type of probability classifier that applies Bayes' theorem, and is a popular classification method that classifies a document into one of several categories [2]. Support vector machine (SVM) is a classification model that finds a hyperplane that best separates data into two categories as a learning technique proposed to solve the binary pattern recognition problem [3].

Manuscript received September 28, 2022; first revision December 19, 2022; accepted January 25, 2023.

* **Corresponding Author:** Hong-Jun Jang (hongjunjang@jj.ac.kr)
[1] Dept. of Computer Science and Engineering, Dongshin University, Naju, Korea (bwkim@dsu.ac.kr)
[2] Dept. of Computer Science and Engineering, Jeonju University, Jeonju, Korea (hongjunjang@jj.ac.kr)
Current affiliation for first author, Byungwook Kim, is Department of Computer Engineering, Gangneung-Wonju National University, Wonju, Korea
Current affiliation for second author, Hong-Jun Jang, is Department of Data Science in Kangwon National University, Chuncheon, Korea

A decision tree is a classification model where rules for classifying objects are expressed in tree form, a method widely used in the field of machine learning [4].

Developing core technologies is necessary for the real-time analysis and future prediction of global online trends, ensuring information on various web and social media issues is not distorted. In the era of global competition, online trend analysis and prediction technology has the advantage of obtaining the most effective and quick results compared to the cost of replacing expensive and low-efficiency manual work traditionally done by expert groups [5]. In data analysis, including data mining, spatio-temporal information is becoming increasingly important. Existing research on spatio-temporal information extraction focuses on accurately extracting spatio-temporal information included in documents. Documents contain a variety of spatio-temporal information, some of which are important while others are insignificant [6–8]. In our previous study, we newly defined information that expresses the core topic of a document, among the various temporal and spatial details it includes, as representative spatio-temporal information [9]. It is important to extract only representative spatio-temporal information because spatio-temporal information that is far from the core content of the document only reduces the accuracy of data analysis. In addition, in order to extract representative spatio-temporal information, a document having representative spatio-temporal information should first be classified among a large amount of documents. In our previous study, we proposed a representative spatio-temporal document classifier based on a character-level convolutional neural network (CNN) model, developed for the first time, but did not consider the tokenization method for the Korean language.

Segmentation of text is called tokenization, and various tokenization methods have been proposed for major languages such as English, but they do not show good performance in the Korean language. One of the reasons for this result is that the characteristics of agglutinating words existing in the Korean language were not considered in the tokenization stage. While research has been conducted on Korean language-specific tokenization methods that consider the characteristics of an agglutinating language, it has primarily focused on the performance of tokenization itself, and performance is considered with the final task and application to machine learning in mind. There has been very little exploratory research on what tokenization methods might be helpful for improvement in classification performance.

In this paper, we analyzed the effect of the Korean tokenization method on the performance of representative spatio-temporal document classification. For the comparison, the most basic tokenization units were used: Jamo, Character, and Word. The classification of representative spatio-temporal documents was selected as the final task for the performance comparison study. As the basic model for the experiment, the CNN-based document classifier proposed in the previous study was used, and the tokenization method of the input sentence was applied differently. The experiment revealed that morpheme-based tokenization, which guarantees the ability to retain the meaning of tokens, showed the best performance in the Korean machine reading task.

The main contributions of this study are summarized as follows.

- We developed our own set of 10,000 learning data that can train a representative spatio-temporal document classifier. The learning data were developed using the news article corpus from the National Institute of the Korean Language. Eight workers directly built the learning data by performing the labeling task.
- We verified that the word-level embedding method improves performance of a representative spatio-temporal document classifier. Since Korean language has the characteristic of an agglutinative language, it seems that the word-level tokenization method shows high performance

in classifying Korean documents because it contains a linguistic meaning in a single word rather than a syllable.

This paper is structured as follows. Section 2 describes the embedding methods corresponding to preprocessing tasks in document classification and natural language processing. In Section 3, various tokenization methods used in this study are described. In Section 4, we present the experimental results according to each tokenization method. Section 5 summarizes the results of this study and suggests future research directions.

2. Background

2.1 Document Classification

With the development of the Internet and the widespread adoption of digital data, the amount of computerized documents is increasing, and classification of documents is becoming an important problem. Various studies have been conducted to automatically classify documents into categories through machine learning. Representative machine learning methods used for document classification include Gaussian Naive Bayes (GNB), based on Bayes' theorem, and SVMs.

GNB was first used to classify spam documents [10, 11]. GNB uses the frequency and conditional probability of frequently used words in an email to determine whether a given email is spam. Mitra et al. [12] used SVM to classify documents based on document titles. Random forest (RF) is suitable for classifying high-dimensional text data containing noise. Islam et al. [13] also improved the performance of the random forest classifier by proposing a dynamic ensemble selection method. Machine learning has been used for a long time in document classification. With the adoption of deep learning techniques in document classification, the performance of document classifiers has improved.

Machine learning, along with its components deep learning and neural networks, are detailed subsets of artificial intelligence (AI). AI processes data to make decisions and make predictions. AI processes data using machine learning algorithms, and it becomes more intelligent over time as it learns from the data without the need for additional programming. AI is a superset that encompasses all machine learning-related subsets. The first subset is machine learning, which has deep learning within it and neural networks within deep learning. Deep learning is showing high performance in computer vision as LeCun et al. [14] presented a CNN model as a method for automatically recognizing characters. Deep learning based on CNN is being used in various fields such as sound and text as well as the field of computer vision, and it has been extended and applied to recurrent neural network (RNN) [15].

Deep learning began to draw significant attention in the field of natural language processing with the introduction of Word2vec [16]. Word2vec is a method of quantifying the meaning of words to reflect significant similarity between word vectors. Deep learning demonstrates higher performance than traditional machine learning methods in various natural language processing tasks. These tasks include spam email detection [17], emotion classification [18], question-and-answer systems [19], and detecting the occurrence of infectious diseases [20].

2.2 Korean Language Processing

Human language is characterized by ambiguity in expression and lack of information in sentences, so

the process of changing it so that it can be understood by a computer is a difficult task. In the communication process, language often omits significant information to maximize efficiency. However, since computers have limitations in understanding natural language, as humans do, the omission of more information makes natural language processing increasingly difficult. Because the characteristics of each language are different, the complexity of the natural language processing process also differs from language to language. In the field of natural language processing, Korean is considered one of the most challenging languages to process. This is because Korean belongs to an agglutinative language in which meaning and grammatical function change with suffixes attached to stems. Natural language processing in agglutinative languages like Korean is particularly challenging due to characteristics such as semantic generation by adding affixes, flexible word order, ambiguous spacing rules, and the absence of clear differences between declarative and interrogative sentences.

Therefore, various studies on natural language processing for Korean have been conducted. In order to study natural language processing, first of all, a dataset is essential. It is difficult for researchers to build their own datasets due to time and cost issues. Therefore, many studies use open datasets to conduct research. Ban [21] introduced 15 essential datasets for natural language processing research targeting Korean. Vu et al. [22] suggested data augmentation (DA) as a solution to the scarcity of Korean text data for learning. As a result of applying the data augmentation method using a pre-trained Korean language model, it is suggested that better performance can be obtained in language comprehension tasks and emotion classification. Processing Korean natural language is challenging, not only due to the language's agglutinative nature but also because of the lack of information on technical terminology. Kim et al. [23] collected Korean medical data and presented a Korean medical language model. To address the shortage of Korean datasets, Shin et al. [24] proposed a method that uses machine translation to convert English datasets into Korean, specifically for creating models that detect violence in online text.

2.3 Embedding Methods

To achieve the intended results in natural language processing, data must be tokenized, cleaned, and normalized according to its intended use. The operation of dividing a given corpus into units called tokens is tokenization. The token unit used to divide the corpus influences the size of the word set and the form of the token represented by this set. These factors, in turn, affect the performance of the model. Although the unit of the token varies depending on the situation, a token is usually defined as a meaningful unit. When the standard of the token is a word, it is called word tokenization. In this context, a word may be considered as a phrase or a meaningful character string, in addition to a single word unit.

To date, various tokenization methods have been presented to improve the effectiveness of machine learning, but these methods have mainly been developed for widely-spoken languages, such as English. However, unlike other languages, Korean has its own characteristics. For example, it has unique characteristics such as specific word order, transformations, and agglutinative language properties.

Seo et al. [25] assumed that the tokenizers of pre-trained BERT and RoBERT were not effective for abbreviations and thus modified these tokenizers to better handle abbreviated words, e.g., RERT: i'm = [i, ', m], RoBERT: i'm = [i, 'm], proposed method: i'm = [i, am]. Toraman et al. [26] suggested that language that can generate many words by adding prefixes and suffixes should apply different tokenization methods depending on the language because the performance of the language model differs depending on the tokenization method. Research on tokenization in languages other than English is

ongoing. Alkaoud and Syed [27] proposed a method of dividing words into subwords and embedding them instead of the traditional word embedding model based on the premise that a single Arabic word contains a large amount of information. The sub-word embedding method showed better performance than the existing Word2vec model.

In this study, in order to examine the effect of tokenization method on representative spatio-temporal document classification performance on Korean text data, various tokenization methods are set as a comparison group and comparative experiments are conducted. The tokenization method adopted for the comparative experiment and its description are as follows.

1) Jamo (consonants and vowels): Jamo refers to single characters that can be written by analyzing one syllable into consonants and vowels.
2) Character (syllable): Character (syllable) is a unit of speech that the speaker and listener think of as a group.
3) Word: This basic unit, which composes an article, is defined by spaces that separate one word from another. In this study, a space was used as a delimiter, and text was segmented and used as a word unit.

Table 1 shows examples of results of text segmentation using various tokenization methods used in the comparative experiment.

Table 1. Example of application of tokenization method

Input sentence	나는 학교에 간다
Jamo	[ㄴ,ㅏ,ㄴ,ㅡ,ㄴ, ,ㅎ,ㅏ,ㄱ,ㄱ,ㅛ, ㅇ,ㅖ, ,ㄱ,ㅏ,ㄴ,ㄷ,ㅏ]
Character (syllable)	[나,는, ,학,교,에, ,간,다]
Word (eojeol)	[나는, 학교에, 간다]

3. CNN Model

The purpose of this study is to verify tokenization that improves the performance of a representative spatio-temporal document classifier. Previous studies used and applied only character-based tokenizers to CNN models. We modified the tokenizer in the embedding stage so that the CNN model presented in the previous study can apply various tokenization methods. In Fig. 1, we illustrate how the tokenization method of the input text was adapted for the RepSTDoc_ConvNet model, which is used in existing representative spatio-temporal document classification. RepSTDoc_ConvNet consists of nine convolutional layers (Conv1D and MaxPooling1D) and three fully connected layers (Dense). The difference from previous studies lies in the embedding layer, where tokenization occurs not only at the Character level but also at the Jamo and Word level.

In a CNN model that involves various hyperparameters, finding the values that yield optimal performance is important. However, the purpose of this paper is to investigate the effect of tokenization method on representative spatio-temporal document classification performance. Therefore, we use the optimal hyperparameter values proposed in previous studies for our experiments. Table 2 shows the hyperparameter settings where the CNN model exhibits optimal performance.

In deep learning, overfitting is one of the factors hindering performance improvement. In order to solve the overfitting problem, learning is stopped early if the loss is not reduced in the learning process [28].

We configured the system with Keras and used EarlyStopping and ModelCheckpoint callbacks to monitor the model's performance and stop training when it no longer improves.

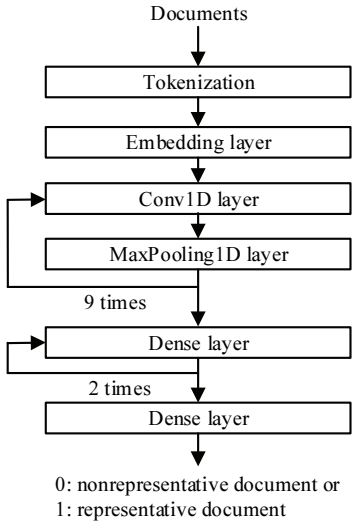

Fig. 1. CNN Model used in experiment.

Table 2. Optimal values of hyperparameters in CNN model

Hyperparameter	Value
Batch size	64
Feature maps	256
Kernel size	5
Stopping patience	200
Epochs	1,000
Dense layer neurons	100
Dropout rate	0.6
Pooling window size	3
Learning rate	0.0001
Activation function	ReLu
Optimizer	Adam
Pooling type	Max pooling

4. Word-Level Embedding

The purpose of this paper is to investigate the effects of the Korean language tokenization method on improving the performance of a representative spatio-temporal document classifier. For comparison, we used the most basic tokenization units: Jamo, Character, and Word. Fig. 2 shows three methods of tokenizing Korean language in this study.

As shown in Fig. 2, even for sentences of the same length, the length of the token is different depending on the tokenization method. In the case of Jamo tokenization, the sequence length is longer due to further division, while with Word tokenization, it's shorter as several syllables combine to form a single word.

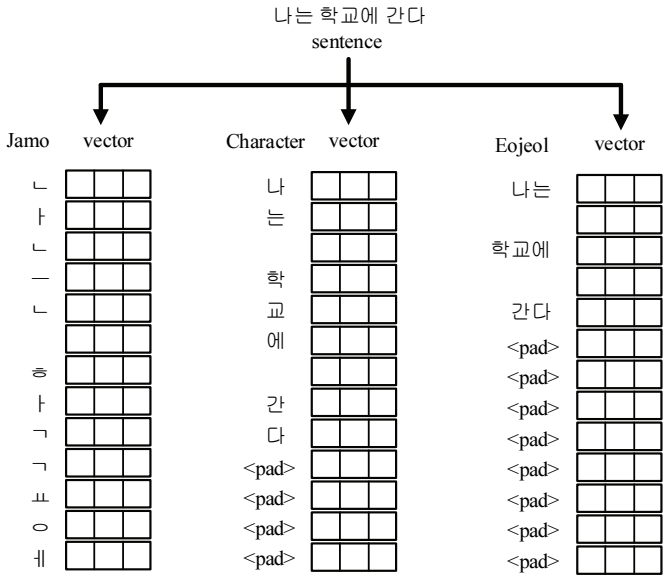

Fig. 2. Embedding method according to token configuration: Jamo, Character and Word.

Therefore, the length of the sequence input to embedding should be different for each method. This is important because a longer base sequence results in excessive padding (<PAD>), affecting accurate performance measurement. Conversely, setting the sequence length too short leads to loss of information. In order to minimize the number of <PAD> in the embedding process of input data, we set the maximum length of the sequence according to the maximum size of training data divided by each tokenization. For embedding, we set the maximum sequence lengths to 6000 for Jamo, 4700 for Character, and 2000 for Word.

5. Experiments

5.1 Dataset

Training data is required to develop a representative spatio-temporal document classification model from a large corpus of text documents. As the concept of a representative spatio-temporal document was first introduced in a previous study, no pre-existing learning data is available. We directly built the training data to train the model. Learning data was constructed using a corpus of news articles written in Korean provided by the National Institute of the Korean Language. Eight workers read news articles and identified their topics. The training data was built for binary classification. An article was labeled as 1 if it contained information about time and space describing the event of the news article topic, and as 0 otherwise. We constructed a dataset of 10,000 training instances over a period of 1 year. Fig. 3 shows an example of training data to be used for a representative spatio-temporal document classification model. The entire body of a news article is used as input data, while the classifications made by workers are used as target data. The input data is decomposed according to the embedding method and fed into the model, as illustrated in Fig. 2. To ensure the accuracy of the constructed learning data, regular cross-checks were conducted among the workers.

Input data	Target data
용산참사의 실상과 함께 희생자들을 추모하는 전시전인 '망루전'이 2일 대구시민회관서..	1
봉화군 물야면 가평리의 계서당(溪西堂)은 계서(溪西) 성이성(成以性) 선생의 집으로 …	0
곽은진은 도시의 밤풍경을 자유로운 시선으로 카메라앵글에 담았다…	0
국민참여당 대구시당 창당대회가 2009년 12월 26일 오후 3시 500여 명의 당원과 …	1

Fig. 3. Examples of training data for the representative spatio-temporal document classification.

5.2 Evaluation Results and Discussion

To investigate the effect of Korean language tokenization on document classifier performance, we conducted experiments using both existing machine learning models and CNN models. Due to weight initialization randomness in deep learning, we conducted 10 experiments for each model and calculated the average as the final result. Table 3 displays the performance of each model, configured with the optimal hyperparameter values. In the experiments, the CNN model (Word) demonstrated the best performance, achieving an accuracy of 0.788 and an F1-score of 0.625. These results indicate that the CNN-based model outperforms traditional machine learning methods, and within the CNN models, the word-unit tokenization method exhibits the best performance.

We investigated the change in performance according to the size of the training data. Fig. 4 compares the performance of different tokenization methods according to data size. In the graph, the x-axis represents the ratio of the data used in the experiment relative to the total data. We performed 10 experiments in each experimental environment. Fig. 4(a) shows the highest performance measurements for each experimental environment, while Fig. 4(b) shows the average of these measurements across 10 repeated experiments. The results indicate that the Jamo method has the lowest performance. The Character method performs well when the data size is less than 60%, whereas the Word method excels with more than 80% of the data. The experimental results suggest that a sufficient amount of data is necessary for word units to be contextually meaningful.

The receiver operating characteristic (ROC) curve is used to evaluate the performance of the classification model for all possible thresholds. In this study, we used the ROC curve to evaluate a binary classification model that determines whether a document is a representative spatio-temporal document or not. Fig. 5 shows the ROC curves for the three tokenization methods (Jamo, Character, and Word) used in the CNN model. Overall, except for the part where the false positive rate is lower than 0.1, the Word tokenization method shows the best performance for document classification.

Table 3. Comparison of performance evaluation by model

Model	Precision	Recall	Accuracy	F1-score
GNB	0.563	0.223	0.751	0.320
Linear SVM	0.626	0.421	0.783	0.503
Random forest	0.729	0.191	0.770	0.303
CNN				
Jamo	0.361	0.488	0.748	0.542
Character	0.538	0.661	0.772	0.593
Word	0.571	0.630	**0.788**	**0.625**

The bold font indicates the best performance in each test.

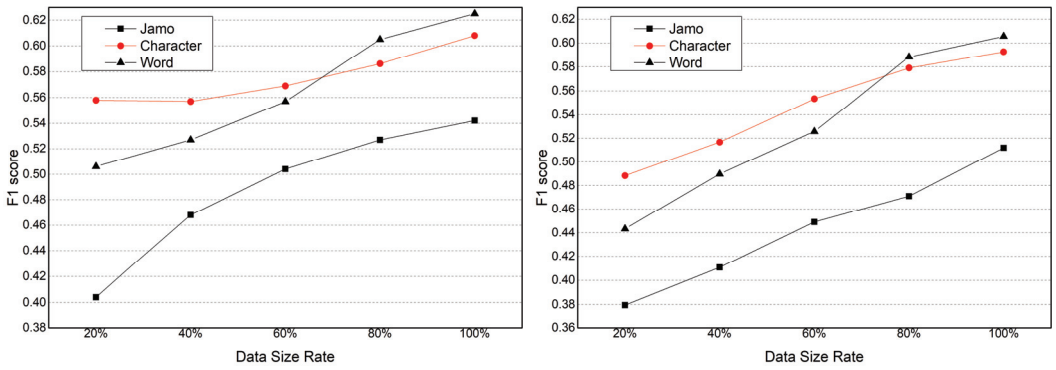

Fig. 4. Comparison of performance of tokenization methods according to data size: (a) highest F1-score and (b) average F1-score.

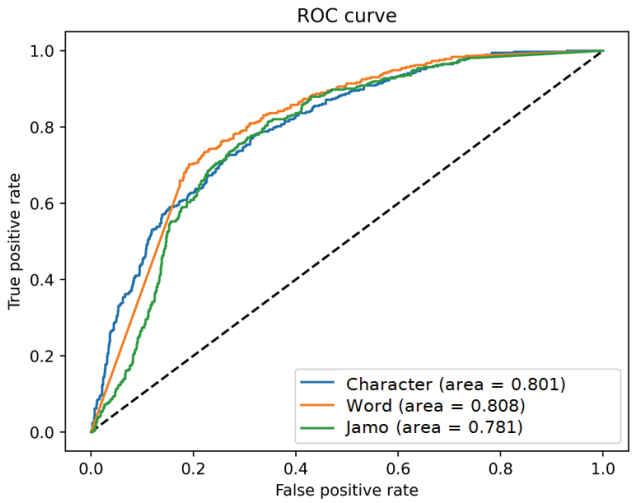

Fig. 5. ROC curves by Jamo, Character and Word methods.

It can be seen that the resulting values of accuracy and F1-score do not come out as well as the performance of general document classifiers. This suggests that classifying representative spatio-temporal documents is a challenging task. This study is an initial study to classify representative spatio-temporal documents, and so far only the body of the article has been used as input data. Future studies should expand to include morphological analysis and NER (named entity recognition) in the model, rather than solely relying on the article text as input data.

6. Conclusion

In this study, a comparative analysis experiment was performed to investigate the effect of the Korean tokenization method on the performance of representative spatio-temporal document analysis. For the experiment, eight workers manually built 10,000 training data. We conducted an experiment using a CNN-based document classifier we developed. The input sentences were tokenized in units of Jamo,

Character, and Word. The experiment revealed that morpheme-based tokenization, which retains the meaning of tokens, performed best in the Korean machine reading task. This is due to its ability to maintain a high level of language model quality and semantic retention. This result is attributed to the proposed method's ability to maintain high language model quality and semantic retention of tokens while reducing the lexicon size.

A limitation of this study is the representative spatio-temporal document classifier's inferior performance compared to general document classifiers. Therefore, in order to improve the performance of the representative spatio-temporal document classifier, it is necessary to consider various characteristics such as morphological analysis information and word arrangement order of natural language. In addition, natural language is not just a list of words, but also contains contextual information in the order in which words are presented. Therefore, extending the deep learning model to include consideration of word order is also important.

Acknowledgement

This research was funded by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Korean Government (MSIT) (No. 2021R1F1A1049387) and this result was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2021RIS-002).

References

[1] A. Mucherino, P. J. Papajorgji, P. M. Pardalos, A. Mucherino, P. J. Papajorgji, and P. M. Pardalos, “K-nearest neighbor classification,” in *Data Mining in Agriculture*. New York, NY: Springer, 2009, pp. 83-106. https://doi.org/10.1007/978-0-387-88615-2_4

[2] Y. Wang, J. Hodges, and B. Tang, “Classification of web documents using a naive bayes method,” in *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*, Sacramento, CA, USA, 2003, pp. 560-564. https://doi.org/10.1109/TAI.2003.1250241

[3] S. Mayor and B. Pant, “Document classification using support vector machine,” *International Journal of Engineering Science and Technology*, vol. 4, no. 4, pp. 1741-1745, 2012.

[4] W. M. Noormanshah, P. N. Nohuddin, and Z. Zainol, “Document categorization using decision tree: preliminary study,” *International Journal of Engineering & Technology*, vol. 7, no. 4.34, pp. 437-440, 2018. https://doi.org/10.14419/ijet.v7i4.34.26907

[5] J. Kalita, “Detecting and extracting events from text documents,” 2016 [Online]. Available: https://arxiv.org/abs/1601.04012.

[6] A. Badia, J. Ravishankar, and T. Muezzinoglu, “Text extraction of spatial and temporal information,” in *Proceedings of 2007 IEEE Intelligence and Security Informatics*, New Brunswick, NJ, USA, 2007, pp. 381-381. https://doi.org/10.1109/ISI.2007.379527

[7] C. G. Lim, Y. S. Jeong, and H. J. Choi, “Survey of temporal information extraction,” *Journal of Information Processing Systems*, vol. 15, no. 4, pp. 931-956, 2019. https://doi.org/10.3745/JIPS.04.0129

[8] A. Feriel and M. K. Kholladi, “Automatic extraction of spatio-temporal information from Arabic text documents,” *International Journal of Computer Science & Information Technology*, vol. 7, no. 5, pp. 97-107, 2015. https://doi.org/10.5121/ijcsit.2015.7507

[9] B. Kim, Y. Yang, J. S. Park, and H. J. Jang, "A convolution neural network-based representative spatio-temporal documents classification for big text data," *Applied Sciences*, vol. 12, no. 8, article no. 3843, 2022. https://doi.org/10.3390/app12083843

[10] J. Chen, H. Huang, S. Tian, and Y. Qu, "Feature selection for text classification with Naïve Bayes," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5432-5435, 2009. https://doi.org/10.1016/j.eswa.2008.06.054

[11] H. Pavel, "How to build and apply Naive Bayes classification for spam filtering," 2020 [Online]. Available: https://towardsdatascience.com/how-to-build-and-apply-naive-bayes-classification-for-spam-filtering-2b8d3308501.

[12] V. Mitra, C. J. Wang, and S. Banerjee, "Text classification: a least square support vector machine approach," *Applied Soft Computing*, vol. 7, no. 3, pp. 908-914, 2007. https://doi.org/10.1016/j.asoc.2006.04.002

[13] M. Z. Islam, J. Liu, J. Li, L. Liu, and W. Kang, "A semantics aware random forest for text classification," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, Beijing, China, 2019, pp. 1061-1070. https://doi.org/10.1145/3357384.3357891

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[15] Z. Zhong, Y. Gao, Y. Zheng, and B. Zheng, "Efficient spatio-temporal recurrent neural network for video deblurring," in *Computer Vision–ECCV 2020*. Cham, Switzerland: Springer, 2020, pp. 191-207. https://doi.org/10.1007/978-3-030-58539-6_12

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013 [Online]. Available: https://arxiv.org/abs/1301.3781.

[17] T. Huang, "A CNN model for SMS spam detection," in *Proceedings of 2019 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, Hohhot, China, 2019, pp. 851-85110. https://doi.org/10.1109/ICMCCE48743.2019.00195

[18] S. Liu and I. Lee, "Sequence encoding incorporated CNN model for Email document sentiment classification," *Applied Soft Computing*, vol. 102, article no. 107104, 2021. https://doi.org/10.1016/j.asoc.2021.107104

[19] E. Mutabazi, J. Ni, G. Tang, and W. Cao, "review on medical textual question answering systems based on deep learning approaches," *Applied Sciences*, vol. 11, no. 12, article no. 5456, 2021. https://doi.org/10.3390/app11125456

[20] M. Kim, K. Chae, S. Lee, H. J. Jang, and S. Kim, "Automated classification of online sources for infectious disease occurrences using machine-learning-based natural language processing approaches," *International Journal of Environmental Research and Public Health*, vol. 17, no. 24, article no. 9467, 2020. https://doi.org/10.3390/ijerph17249467

[21] B. Ban, "A survey on awesome Korean NLP datasets," in *Proceedings of 2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, South Korea, 2022, pp. 1615-1620. https://doi.org/10.1109/ICTC55196.2022.9952930

[22] D. T. Vu, G. Yu, C. Lee, and J. Kim, "Text data augmentation for the Korean language," *Applied Sciences*, vol. 12, no. 7, article no. 3425, 2022. https://doi.org/10.3390/app12073425

[23] Y. Kim, J. H. Kim, J. M. Lee, M. J. Jang, Y. J. Yum, S. Kim, et al., "A pre-trained BERT for Korean medical natural language processing," *Scientific Reports*, vol. 12, article no. 13847, 2022. https://doi.org/10.1038/s41598-022-17806-8

[24] J. Shin, H. Song, H. Lee, and J. Park, "Constructing Korean abusive language dataset using machine translation," in *Proceedings of the Korea Computer Congress (KCC)*, Jeju, South Korea, 2022.

[25] J. Seo, S. Lee, L. Liu, and W. Choi, "TA-SBERT: token attention sentence-BERT for improving sentence representation," *IEEE Access*, vol. 10, pp. 39119-39128, 2022. https://doi.org/10.1109/ACCESS.2022.3164769

[26] C. Toraman, E. H. Yilmaz, F. Sahinuc, and O. Ozcelik, "Impact of tokenization on language models: an analysis for Turkish," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 4, article no. 116, 2023. https://doi.org/10.1145/3578707

[27] M. Alkaoud and M. Syed, "On the importance of tokenization in Arabic embedding models," in *Proceedings of the 5th Arabic Natural Language Processing Workshop*, Virtual Event (Barcelona, Spain), 2020, pp. 119-129.

[28] S. Li, J. Hu, Y. Cui, and J. Hu, "DeepPatent: patent classification with convolutional neural networks and word embedding," *Scientometrics*, vol. 117, pp. 721-744, 2018. https://doi.org/10.1007/s11192-018-2905-5

Byoungwook Kim **https://orcid.org/0000-0001-5755-7510**

He received B.S. and Ph.D. degrees in computer science education from Korea University, Seoul, South Korea. He worked at Dongguk University Gyeongju Campus from 2018 to 2021. He worked at Dongshin University from 2021 to 2023. Since 2023, he has been with the Department of Computer Engineering, Gangneung-Wonju National University, Wonju, Republic of Korea. His research interests include data mining, machine learning, and deep learning.

Hong-Jun Jang **https://orcid.org/0000-0001-9979-6851**

He received B.S. at Department of Computer Science Education in Korea University, Korea. He received Ph.D. degrees at Department of Computer Science and Engineering in Korea University. He worked at Jeonju University from 2021 to 2023. Since 2023, he is currently an assistant professor in Department of Data Science in Kangwon National University. He was a senior researcher at Korea Institute of Science and Technology Information. His research interests include spatial databases, data mining and machine learning.