Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 45 No. 7
July 2024

Document Classification System Using BERT

Seung-Yeon Hwang[1], Jeong-Joon Kim[2,*]
[1]Dept. of Computer Engineering, Anyang University, Anyang-si, Gyeonggi-do, Republic of Korea
[2,*] Corresponding Author, Dept. of Software, Anyang University, Anyang-si, Gyeonggi-do, Republic of Korea

Abstract: Research is actively being conducted to create meaningful value using big data generated across society. Accordingly, domestic and international papers, patents, e-books, etc., have been databased and are being used for various studies on research and technology trends. This paper has designed and developed a document classification system based on BERT. For experimental performance evaluation, abstracts from the four most prestigious conferences in the field of information security were collected and used to compare a document classification system using BERT with one based on SBERT. The classification results showed that the BERT-based document classification system was about 11.7% superior, and this paper presents ways to develop a better document classifier.
Keywords: AI, Big data, BERT, document classification, natural language processing

1. Introduction

As big data emerges as a global IT concern, the volume of data generated throughout society in sectors including healthcare, finance, production, and culture is rapidly increasing, highlighting the role of big data in addressing various societal issues faced by modern society. Initially focused on big data infrastructure and analytics technologies, recent efforts are now concentrated on creating significant value from big data utilization. Notably, the accessibility and volume of scholarly materials such as domestic and international papers, patents, and e-books have been enhanced by their database integration offered through various web services. As the collection of such academic materials has increased, there has been a surge in research activities analyzing research and technology trends. Researchers attempt to shed light on the evolving interests of scholars and the patterns of academic trends through multidimensional analysis of these trends [1-3]. Long-term strategic planning and adapting to uncertain environmental changes have made the analysis of research trends a longstanding critical subject for corporations and researchers alike. Although trend analysis in research and technology primarily involved quantitative analysis and thematic content analysis, researchers often classify documents based on varying criteria, which can lead to inconsistencies in methods and difficulties in interpreting research outcomes from different documents. Additionally, the subjective values and personal opinions of researchers might influence the results, posing a risk and making it challenging to process large volumes of data. Given the rapid changes in IT research and technological trends and the increasing uncertainty in research directions, corporations and research institutions must create new values through the integration of existing and emerging technologies to secure a global competitive edge and future growth potential [4]. Identifying and predicting promising technologies to prepare for uncertainties is becoming a crucial factor. Therefore, this paper aims to develop a document classification system to identify research and technology trends using various collected paper data. The structure of this paper is as follows: Chapter 2 discusses related technologies, Chapter 3 explains the document classification system proposed in this paper, Chapter 4 conducts experimental performance evaluations of the document classification system, and Chapter 5 concludes the paper.

2. Related Technologies

This chapter introduces and explains natural language processing technologies for text classification.

2.1 BERT (Bidirectional Encoder Representations from Transformers)

Released by Google in 2018, this pre-trained model achieved state-of-the-art (SOTA) results in numerous NLP tasks upon its introduction [5]. BERT is implemented using the Transformer architecture and is a pre-trained language model utilizing unlabeled text data from sources like Wikipedia (2.5 billion words) and BooksCorpus (800 million words). The model is initially pre-trained on vast amounts of unlabeled data and then fine-tuned on various labeled tasks to achieve high performance. BERT's novel pre-training methods include two key approaches: Masked Language Model (MLM) and Next Sentence Prediction (NSP) [6]. MLM

involves randomly masking some tokens in the input and using the Transformer architecture to predict these masked words based solely on their context. For Next Sentence Prediction, during pre-training, BERT is given pairs of sentences and must predict whether the second sentence logically follows the first. This training includes a balanced approach where 50% of the inputs are pairs of actual consecutive sentences, and the other 50% are randomly paired sentences, enabling BERT to learn from both coherent and non-coherent examples.

2.2 SBERT (Sentence-BERT) [7]

SBERT, or Sentence-BERT, significantly enhances the sentence embedding capabilities of BERT. It fine-tunes BERT specifically for the generation of sentence embeddings, making it more effective for certain applications. SBERT's training methods are twofold: the first involves training solely on Semantic Textual Similarity (STS) data, while the second, a continuation learning method, involves further training on STS data after fine-tuning on Natural Language Inference (NLI) data. The primary and most effective approach for training SBERT is through fine-tuning on STS data, guided by a regression objective function. The datasets used for training SBERT include the STS dataset, which measures the similarity between sentences, and the NLI dataset, which discerns the relationships between sentences.

3. Research Content

This chapter details the architecture and various modules of the document classification system proposed in this research.

3.1 System Architecture

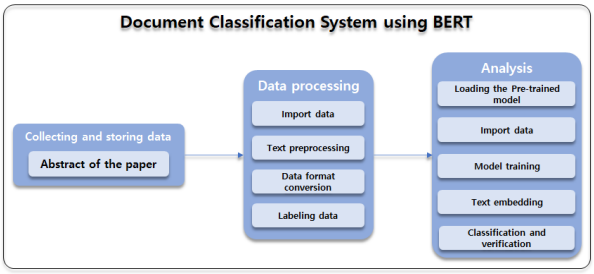


Figure 1. Architecture of the Document Classification System Using BERT

Figure 1 represents the architecture of the document classification system proposed in this paper. First, to collect the necessary papers for document classification, a crawler is developed to gather papers available on the web. Next, a preprocessing module is developed to prepare the collected original data for classification. The preprocessing module consists of data loading, text cleaning, data format conversion, and data labeling modules. In the analysis module, the preprocessed data is used to train the pre-trained model, and this model is then used for classification and verification. The analysis module consists of pre-trained model loading, data loading, model training, text embedding, classification, and verification modules.

3.2 Data Collection and Storage

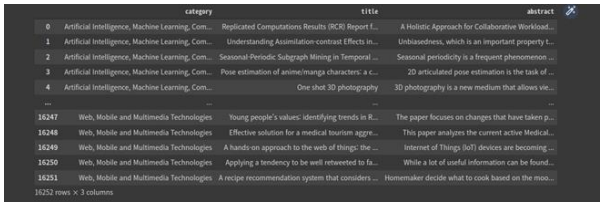
For this paper, a crawler was developed to collect abstracts of papers from 2010 to 2020 from the four most prestigious conferences in the field of information security (ACM CCS, Usenix Security, NDSS, IEEE Security & Privacy). The category information for each journal is as shown in Table 1.

Table 1. Classes of Collected Papers

Journal	Category
ACM CCS	Cryptography, Formal methods and theory of security, Security services, Intrusion/anomaly detection and malware mitigation, Security in hardware, Systems security, Network security, Database and storage security, software and application security, Human and societal aspects of security and privacy
Usenix Security	System security, Network security, Security analysis, Data-driven security and measurement studies, Privacy-enhancing technologies and anonymity, Language-based security, Hardware security, Social issues and security, Applications of cryptography
NDSS	Cyber-crime defense and forensics, Security and privacy for blockchains and cryptocurrencies, Security for cloud/edge computing, Security and privacy of mobile/smartphone platforms, Security for cyber-physical systems, Security and privacy of systems based on machine learning and AI
IEEE Security & Privacy	Application security, Attacks and defenses, Authentication, Blockchains and distributed ledger security, Cloud security, Cyber physical systems security, Distributed systems security, Embedded systems security, Forensics

Using the crawler, approximately 1,600 papers from the ACM CCS conference, about 800 papers from the

Usenix Security conference, around 600 papers from the NDSS conference, and about 1,200 papers from the IEEE Security & Privacy journal were collected. The collected paper information is stored in CSV format, consisting of the attributes 'Category', 'title', and 'abstract'. The 'Category' attribute represents the classification class of the paper, 'title' refers to the title of the paper, and 'abstract' contains the abstract data of the paper. Figure 2 shows a portion of the collected papers.



	category	title	abstract
0	Artificial Intelligence, Machine Learning, Com...	Replicated Computations Results (RCR) Report f...	A Holistic Approach for Collaborative Workload...
1	Artificial Intelligence, Machine Learning, Com...	Understanding Assimilation-contrast Effects in...	Unbiasedness, which is an important property t...
2	Artificial Intelligence, Machine Learning, Com...	Seasonal-Periodic Subgraph Mining in Temporal ...	Seasonal periodicity is a frequent phenomenon ...
3	Artificial Intelligence, Machine Learning, Com...	Pose estimation of anime/manga characters: a c...	2D articulated pose estimation is the task of ...
4	Artificial Intelligence, Machine Learning, Com...	One shot 3D photography	3D photography is a new medium that allows vie...
...	...	...	...
16247	Web, Mobile and Multimedia Technologies	Young people's values: identifying trends in R...	The paper focuses on changes that have taken p...
16248	Web, Mobile and Multimedia Technologies	Effective solution for a medical tourism aggre...	This paper analyzes the current active Medical...
16249	Web, Mobile and Multimedia Technologies	A hands-on approach to the web of things: the ...	Internet of Things (IoT) devices are becoming ...
16250	Web, Mobile and Multimedia Technologies	Applying a tendency to be well retweeted to fa...	While a lot of useful information can be found...
16251	Web, Mobile and Multimedia Technologies	A recipe recommendation system that considers ...	Homemaker decide what to cook based on the moo...

16252 rows × 3 columns

Figure 2. Sample of the Original Data from the Collected Papers

3.4 Data Processing

In the data processing module, the Python Pandas library was utilized to load the collected data in CSV format. A text cleaning module was developed to remove special symbols and meaningless text included in the abstract data of the papers. To divide the original data into training and testing datasets, the train_test_split function provided by the Python sklearn library was used. The ratio of the training dataset to the testing dataset was set at 7:3. Finally, the abstract data in the training dataset was segmented into sentences and labeled accordingly. For example, assuming the presence of abstract data as shown on the left side of Figure 3, the abstract data would be divided into sentence units, and each sentence would be labeled according to the class to which the abstract data belongs.

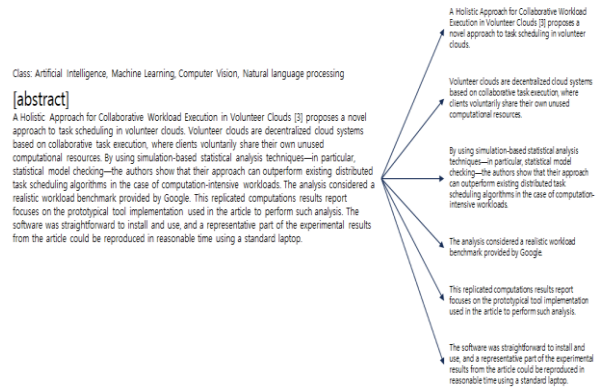


Figure 3. Example of Training Data Labeling Method

In this instance, the labeling of classes is performed using numbers instead of text. Through sentence-level labeling, a total of 81,896 training data instances were generated. Figure 4 illustrates a sample from the training dataset.

	category	title	abstract
0	0	Boosting Steganalysis with Explicit Feature Maps	Explicit non-linear transformations of existin...
1	0	Boosting Steganalysis with Explicit Feature Maps	The non-linear transformations are learned fr...
2	0	Boosting Steganalysis with Explicit Feature Maps	The best performance is achieved with the exp...
3	0	Boosting Steganalysis with Explicit Feature Maps	Since the non-linear map depends only on the ...
4	0	Boosting Steganalysis with Explicit Feature Maps	The map can also be used to significantly red...
...	...	...	...

Figure 4. Sample from the Training Dataset

3.5 Data Analysis

The document classification system proposed in this paper was implemented in Google's Colab development environment using a Pytorch-based BERT model. Initially, the "bert-base-multilingual-cased" model was loaded from Hugging Face (https://huggingface.co/), a platform where various pre-trained models are shared. Since the original model is designed for two classes, the output feature was set to 5 to match the number of classes in the training dataset. The preprocessed dataset was loaded using the Python pandas library, and each sentence was transformed into the format “[CLS] sentence [SEP]” to suit BERT's input specifications. Figure 5 illustrates the sentences converted to match BERT's input format.

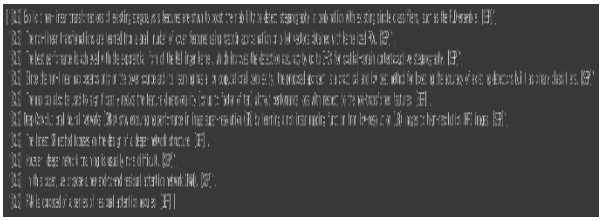


Figure 5. Input Format for BERT

Using BERT's tokenizer, each sentence was segmented into tokens. Figure 6 shows a sample of the sentences segmented into tokens.

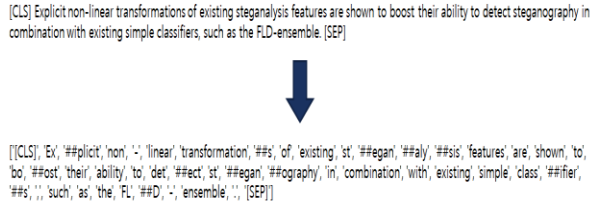
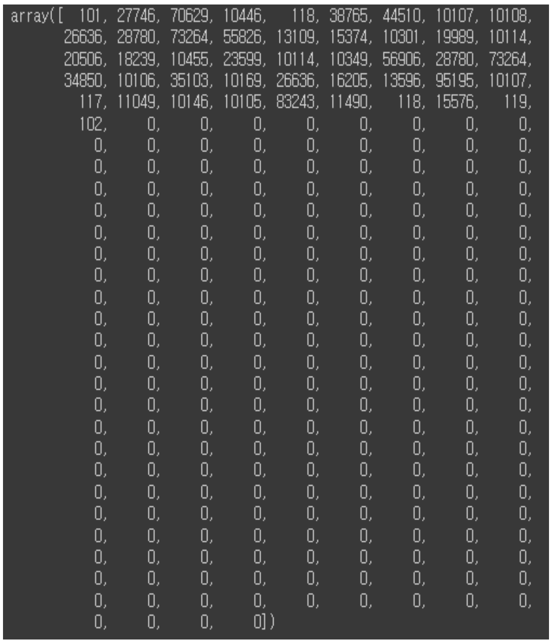


Figure 6. Sample of Sentence Tokenization Using BERT's Tokenizer

The maximum length for input tokens was set to 256, and the tokens were converted into numeric indices. Sentences were split to match this maximum length, and areas without data were padded with zeros. Figure 7 shows a sample of the padded sentences.



```
array([  101, 27746, 70629, 10446,   118, 38765, 44510, 10107, 10108,
       26636, 28780, 73264, 55826, 13109, 15374, 10301, 19989, 10114,
       20506, 18239, 10455, 23599, 10114, 10349, 56906, 28780, 73264,
       34850, 10106, 35103, 10169, 26636, 16205, 13596, 95195, 10107,
         117, 11049, 10146, 10105, 83243, 11490,   118, 15576,   119,
         102,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0])
```

Figure 7. Sample of Sentences Padded to Maximum Length

The model's optimizer is Adam, and the learning rate was set to 0.00002. Additionally, to prevent division by zero during the training process, the epsilon value was set to 0.00000001. The number of epochs for training was set to 12. During the training process, 10% of the training dataset was allocated as a validation dataset to verify the model's performance.

4. Experimental Performance Evaluation

To measure the performance of the document classification system using BERT proposed in this paper, a comparison was conducted with a document classification system based on SBERT. Since SBERT is trained using unsupervised learning methods, its output is not the predicted class of the documents directly. Therefore, to classify documents using the SBERT model, cosine similarity is calculated for the entire set of documents.

The class assigned to a document is determined by the class most frequently appearing among the top 30 documents with the highest cosine similarity scores. For instance, if the first document and the entire set of documents' cosine similarities are calculated, and the most frequent class among the top 30 documents is Class A, then the first document is classified as Class A. Table 2 shows the results of classifying journal papers using SBERT and BERT, and Figure 8 illustrates these results diagrammatically.

Table 2. Results of Document Classification Using SBERT and BERT

Doc. Classifier	ACM CCS	Usenix Security	NDSS	IEEE Security & Privacy	Average
SBERT	50.4 %	48.1%	49.98 %	49.59 %	49.5%
BERT	61.98 %	60.31 %	61.61 %	60.89 %	61.2%

Figure 8. Visualization of Document Classification Results Using SBERT and BERT

Using SBERT for document classification yielded an average accuracy of 49.5%. The document classification system based on BERT proposed in this research achieved an average result of 61.2%. Although the BERT-based system was approximately 11.7% more effective, neither the SBERT nor the BERT-based document classification systems demonstrated strong performance in classifying documents. The likely reasons for the suboptimal performance include a lack of sufficient training data and imperfect preprocessing of the data, such as incomplete removal of special characters and stopwords. It is anticipated that securing more training datasets and utilizing perfectly preprocessed data would improve performance.

5. Conclusion

This research proposed and developed a document classification system using BERT, which is actively researched in the field of natural language processing. During the collection phase, approximately 5,300 abstracts per category were gathered from the ACM Digital Library across five categories. Various preprocessing modules were implemented to prepare the document data for analysis, and document classification was performed using the preprocessed

data. The experimental performance evaluation compared document classification systems using SBERT and BERT, with the BERT-based system showing better performance. However, the performance of the BERT-based document classification system is still not considered excellent, so further research involving the collection of more data and perfect preprocessing of that data is expected to yield better results.

Acknowlegement
This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1F1A1062953).

References

[1] Jahyun Park, Min Song, “A Study on the Research Trends in Library & Information Science in Korea using Topic Modeling”, Journal of the Korean Society for Information Management, Vol. 30, No. 1, pp. 7-32
[2] Jong-Do Park, “A Study on Issue Tracking on Multi-cultural Studies Using Topic Modeling”, Journal of the Korean Library and Information Science, Vol. 53, No. 3, pp. 273-289
[3] KimMiae, Chang-Kyo Suh, “SCM Patent Analysis Using Topic Modeling: 1997~2016”, Journal of the Korean Society of Supply Chain Management, Vol. 17, No. 2, pp. 19-29
[4] Jinbaek Lee, Choongkwan Lee, Kyungjin Cha, “An Analysis of IT Trends Using Tweet Data”, Journal of Intelligence and Information Systems, Vol. 21, No. 1, pp. 143-159
[5] Sojin Oh, Moonkyoung Jang, Heeseok Song, “A BERT-based Transfer Learning Model for Bidirectional HR Matching”, Journal of Information Technology Applications & Management, Vol. 28, No. 4, pp. 33-43
[6] Junghoon Lee, Donghwa Kim, Youngbin Ro, Pilsung Kang, “Improving Korean Emotion Classification via Colloquial-Adaptive Pretraining”, Journal of the Korean Institute of Industrial Engineers, Vol. 47, No. 4, pp. 342-350
[7] N. Reimers, I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks." arXiv preprint arXiv:1908.10084 (2019).