Journal of JAITC, Vol. 10, No. 1, pp. 45-56, Jul. 31, 2020
http://dx.doi.org/10.14801/JAITC.2020.10.1.45

Fine-tuning BERT Models for Keyphrase Extraction in Scientific Articles

Yeonsoo Lim[1], Deokjin Seo[2], and Yuchul Jung[3]*

[1,2,3]*Cognitive Intelligence Lab., Department of Computer Engineering, Kumoh National Institute of Technology, Gumi, Korea*
[1]*yslim6168@kumoh.ac.kr, https://orcid.org/0000-0002-8760-2842*
[2]*406023@naver.com, https://orcid.org/0000-0002-3760-9616*
[3]*jyc@kumoh.ac.kr, https://orcid.org/0000-0002-8871-1979*
*(*Corresponding Author)*

Abstract

Despite extensive research, performance enhancement of keyphrase (KP) extraction remains a challenging problem in modern informatics. Recently, deep learning-based supervised approaches have exhibited state-of-the-art accuracies with respect to this problem, and several of the previously proposed methods utilize Bidirectional Encoder Representations from Transformers (BERT)-based language models. However, few studies have investigated the effective application of BERT-based fine-tuning techniques to the problem of KP extraction. In this paper, we consider the aforementioned problem in the context of scientific articles by investigating the fine-tuning characteristics of two distinct BERT models — BERT (i.e., base BERT model by Google) and SciBERT (i.e., a BERT model trained on scientific text). Three different datasets (WWW, KDD, and Inspec) comprising data obtained from the computer science domain are used to compare the results obtained by fine-tuning BERT and SciBERT in terms of KP extraction.

Keywords: *keyphrase extraction, BERT, fine-tuning, embedding, scientific articles*

1. Introduction

Keyphrases (KPs) are short but useful expressions in a document that can convey crucial information about the contents of the document being considered. KPs are usually single (or composite) keywords that are related to the primary topic or core content of a document, and approximately five KPs are usually selected for each document. If identified correctly, KPs find beneficial applications in various processes, such as information retrieval, document summarization, and topic classification. In contrast to its usefulness, keyphrase (KP) extraction is one of the most difficult tasks in Neuro-Linguistic Programming (NLP) that requires both document-side statistical information and comprehensive background knowledge [1].

Received: Jul. 09, 2020 Revised: Jul. 23, 2020 Accepted: Jul. 26, 2020

pISSN 2234-1072/eISSN 2234-0963

Traditional KP extraction methods can be largely divided into two stages. First, candidate KPs are extracted, and subsequently a pruning procedure is executed. During the former stage, potential KPs are extracted from the given document. For this purpose, either supervised learning or unsupervised learning-based methods may be chosen. The details of the pruning stage depend on the method chosen to execute the first stage. If candidate KPs selection is conducted via supervised learning, the pruning process is regarded as a binary classification problem of whether the given candidate is a KP. Meanwhile, if unsupervised learning is used during the first stage, pruning is regarded as a ranking problem; the candidate KPs are ranked based on a specific criterion and removed if they exhibit a score below a predefined threshold [2]. Various graph-based unsupervised learning methods and sequential labeling techniques based on supervised learning have been used over the years to achieve good performances in KP extraction [3]. However, even though NLP researchers have achieved gradual performance enhancement in KP extraction, the current performance is still far from perfect.

A fine-tuning technique based on a deep contextual language model (LM) was introduced [4], which was comparable to the existing state-of-the-art supervised technique. Deep contextual LMs, such as Embeddings from Laguage Models (ELMo) [5] and Bidirectional Encoder Representations from Transformers (BERT) [6], have evolved significantly over the years. These models are capable of providing contextual embeddings for each token to use in downstream architectures corresponding to an input text, or to perform task-specific fine-tuning. Fine-tuning is the technique of applying a pre-trained LM. It reduces the task-specific parameters as much as possible and, instead, changes the pre-trained parameters slightly via downstream task learning. In particular, BERT has been shown to achieve state-of-the-art results for 11 NLP tasks, including document classification, question answering, and dependency parsing [5, 7]. Recent studies have established that contextual embedding models trained on domain or genre-specific corpora generally perform better than general-purpose models trained on Wikipedia data. However, the reasons underlying the state-of-the-art performance of BERT are poorly understood. Therefore, in the case of BERT, "current research has focused on investigating the relationship behind BERT's output as a result of carefully chosen input sequences, analysis of internal vector representations through probing classifiers, and the relationships represented by attention weights[1]."

Despite the excellence achieved by the fine-tuning technique in BERT, little is known about the effects of fine-tuning on KP extraction. Therefore, in this paper, we aim to study KP extraction using BERT Models based on scientific articles. The main contributions of this paper are two folds.

1) We performed an in-depth comparison between the fine-tuning results on KP extraction obtained via two different BERT models (i.e., BERT and SciBERT).
2) For the evaluation, we employed three different KP extraction test collections (i.e., WWW, KDD, and Inspec) in computer science domain.

We hope that this study will reveal a crucial research direction that will enhance the approach of pre-training models with respect to KP extraction. The rest of this paper is organized as follows. Related works on KP extraction and recent developments in BERT-based techniques, also known as deep contextualized LMs, have been briefly summarized

[1] https://en.wikipedia.org/wiki/BERT_(language_model)

pISSN 2234-1072/eISSN 2234-0963

Journal of JAITC, Vol. 10, No. 1, pp. 45-56, Jul. 31, 2020
http://dx.doi.org/10.14801/JAITC.2020.10.1.45

in Section 2. Following this, the basics of the BERT model and its corresponding textual resources have been discussed in detail in Section 3. Experiments using BERT and SciBERT and additional in-depth discussion have been presented in Section 4. Finally, the conclusions have been presented in Section 5, alongside possible directions for future work.

2. Related Works

In this section, we first summarize existing KP extraction techniques. KP extraction methods can be largely categorized into supervised and unsupervised approaches. Following this discussion, we present a brief introduction to popular LMs that can be used for KP extraction.

2.1 Unsupervised Approaches to KP Extraction

Unsupervised methods have the advantage of not requiring any training data and are capable of producing results in any domain. They include statistics-based, graph-based, embedding-based, and LM-based approaches [8]. Statistics-based methods select candidate KPs by combining appropriate scores, such as, Term Frequency (TF), Inverse Document Frequency (IDF), and co-occurrences [9]. For example, TF-IDF is used as the common baseline in KP extraction. The fundamental idea in graph-based ranking is to create a graph based on the contents of a document by using the candidate KPs as vertices. Each pair of related candidate KPs corresponds to an edge between the two corresponding vertices. After the adoption of the PageRank algorithm [10] in KP extraction, TextRank [11], LexRank [12], TopicRank [13], SGRank [14], and SingleRank [15] directly leverage the graph-based ranking algorithm PageRank, with a combination of other heuristics based on TF-IDF scores, word co-occurrence measures, extraction of specific lexical patterns, and clustering [16] to combine semantic similarity clustering with knowledge graph structure and expedite the discovery of semantic relations hidden in the input document.

2.2 Supervised Approaches to KP Extraction

Like other machine learning-based approaches, supervised learning methods require prior knowledge - training data to aid learning. The training data for KP extraction includes the corpus and the corresponding pre-labeled KPs. Supervised learning methods can be divided into basic machine learning techniques and applications of recently developed deep learning techniques.

In this case, KP extraction is regarded as a binary classification problem, with annotated keyphrases serving as positive examples and all other phrases serving as negative examples. In general, supervised approaches employ a machine learning-based model to determine if a given candidate phrase is a KP based on analysis of its textual features, such as term frequencies [17], syntactic properties [18], and location information [19]. Further, [20] used conditional random fields (CRF) equipped with multiple textual features, such as TF*IDF of the terms, orthographic information, POS tags, and positional information.

With recent developments in deep learning, especially Long-Short Term Memory (LSTM) [21, 22], deep learning-based approaches have also been adopted in KP extraction.

pISSN 2234-1072/eISSN 2234-0963

[23] proposed a KP extraction approach for Twitter-like corpora and sites. The authors used a recurrent neural network (RNN)-based model to exploit contextual information among keywords in order to retrieve appropriate KPs. [24] proposed another deep learning-based model for KP retrieval. More recently, [25] used Bi-directional Long-short Term Memory (BiLSTM) – Conditional Random Fields (CRFs), in which words were represented by fixed word embeddings like Global Vectors for Word Representation (Glove) embeddings [26]. [27] further explored a bi-directional LSTM-CRF-based sequence labeling technique for KP extraction using different word embedding models.

Recent research has corroborated the significant improvement that can be effected on the performance of deep learning-based LMs by training them on sufficiently large corpora with respect to multiple NLP tasks as well as in transfer learning [5, 6, 28]. Our experiments are based on the adoption of pre-trained large scale contextual language models in terms of a kind of transfer learning.

3. The Application of BERT in the Proposed Analysis

In this section, we briefly introduce an existing BERT model and outline the mode in which BERT-based fine-tuning is implemented based on two contextual LMs, such as BERT [6] and SciBERT [29].

3.1 BERT

BERT stands for Bidirectional Encoder Representations from Transformer. There are two steps during the implementation of BERT: 1) pre-training and 2) fine-tuning. The model is trained on unlabeled data via distinct pre-training tasks during pre-training. The BERT model is first initialized with pre-trained parameters for fine-tuning, and all parameters are fine-tuned using labeled data obtained from the downstream task. Even if each downstream task is initialized with the same pre-trained parameters, a separate fine-tuned model emerges. BERT's features consist of a unified architecture that combines multiple tasks. There is very small difference between the pre-trained model architecture and the final down-stream architecture. The BERT-based model's architecture possesses multiple layers based on a bidirectional transformer encoder, which have been implemented and described in [30].

BERT is designed to pre-train a sentence representation by jointly conditioning both the left and right contexts of a sentence [6]. BERT's base model consists of 12 layers of 768 hidden size, 12 self-attention heads, and 110M total parameters. The large model consists of 24 layers of 1024 hidden size, 16 self-attention heads, and 340M total parameters [6].

As depicted in Figure 1, BERT exhibits two phases ¬— pre-training BERT, depicted on the left, and BERT tuned to correspond to each task, depicted on the right. Based on the previous experiments using BERT [6], task-specific fine-tuning approaches on 11 NLP tasks were performed using the architecture and better performances were observed compared to previously developed supervised methods. However, in the case of KP Extraction, even though the advantage of fine-tuning has been established, little is known about the differences between the performances obtained via fine-tuning works on varying KP test collections and varying pre-trained BERT architectures.

pISSN 2234-1072/eISSN 2234-0963

Journal of JAITC, Vol. 10, No. 1, pp. 45-56, Jul. 31, 2020
http://dx.doi.org/10.14801/JAITC.2020.10.1.45

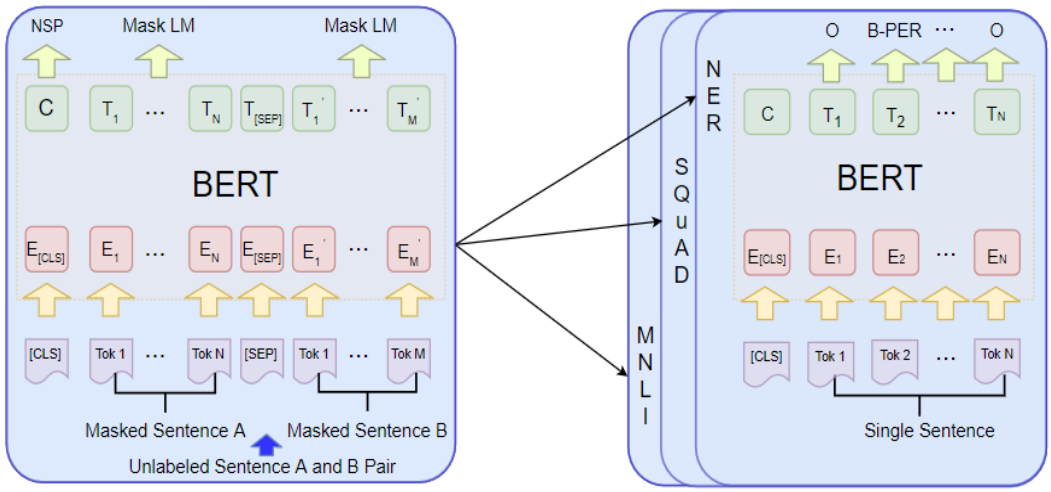

Figure 1. Two steps in the implementation of BERT: (Left) Pre-training and (Right) Fine-tuning

3.2 Fine-tuning BERT (in the context of KP extraction)

A pre-trained BERT model can be easily fine-tuned in the context of a wide range of tasks by simply adding a fully connected (FC) layer without any task-specific modifications on the architecture level. Recently, BERT was used to achieve state-of-the-art performances in 11 NLP tasks (e.g., GLUE task set (consisting of 8 tasks), MultiNLI, SQuAD v1.1, and SQuAD v2.0), thereby outperforming the existing state-of-the-art methods by a large margin [6].

During KP extraction, we employ two pre-training models, BERT and SciBERT, to perform the fine-tuning. Construction of a pre-training model, involves identical architecture, optimization, and hyper-parameters as the ones discussed in [6]. Despite that, we experiment with different vocabularies during embedding. We create a pre-training model by reconstructing the embedding corresponding to each base word. We then fine-tune the models based on a learning rate of 5e-5 and a batch size of 8. This learning rate is selected as it is observed to exhibit the best performance among the ones that are experimented with (including 5e-6). Tokens that pass through the bottom layers and the FC layer, as depicted in Figure 2, are assigned one of the following labels — B-KP, I-KP, or O. “B-KP” stands for begin in KP, and “I-KP” is assigned to any phrase following a B-KP phrase. By allowing the I-KP tag for phrases that follow B-KP phrases to deal with n-gram KPs, it is possible to extract phrases sequentially instead of words in a single context. In our experiments, fine-tuning proceeded up to 5 epochs although only 3 epochs exhibited the best performance according to test collections.

KP Extraction is similar to the previously known named entity recognition (NER) task. In our experiment, we extract KPs using sequence labeling, as presented in [20]. Therefore, we feed the final BERT vector corresponding to each token into a linear classification layer equipped with a soft-max output, like sequence labeling.

pISSN 2234-1072/eISSN 2234-0963

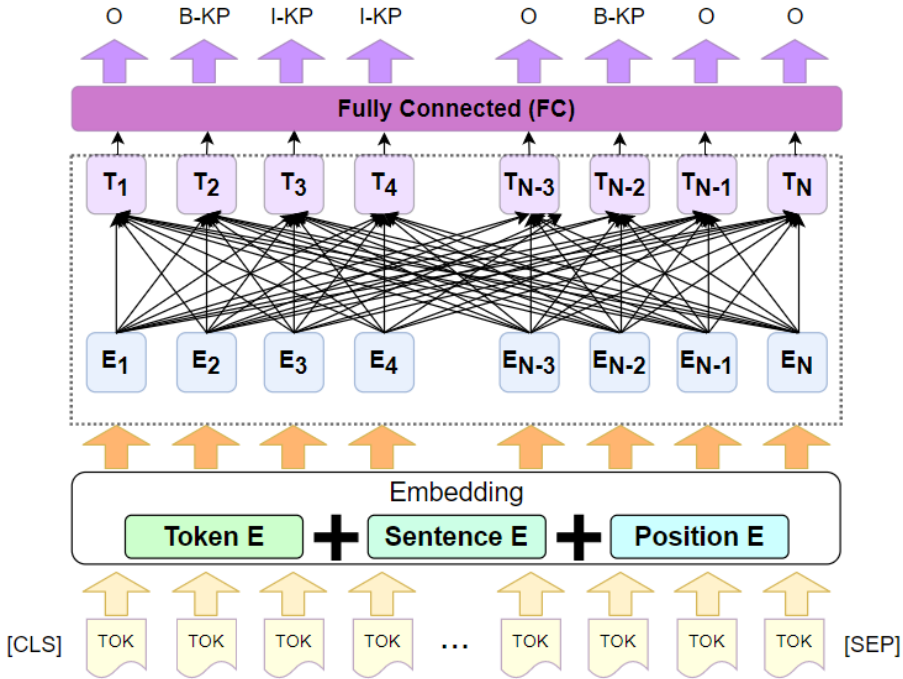

Figure 2. Fine-tuning BERT for the KP extraction task

As depicted in Figure 2, following input tokenization via the tokenizer, three different embeddings (namely token information embedding, sentence information embedding, and position information embedding) are applied to the input. The model learns this embedding information using its bi-directional layer, and finally, the model is used for KP extraction by automatically assigning B-KP, I-KP, and O tags to the given input sentences.

4. Experiments

In this section, we evaluate the influence of BERT fine-tuning on the KP extraction performance by applying different BERT models to different KP test collections. We use the F1-score as the evaluation metric. The F1-score is defined to be the harmonic mean of precision and recall scores[2].

4.1 Datasets

We perform our experiments on three different publicly available KP extraction datasets[3] — KDD, WWW, and Inspec. The detailed statistics of the datasets have been presented in Table 1.

[2] F1-score: https://en.wikipedia.org/wiki/F1_score
[3] Datasets of Automatics Keyword Extraction: https://github.com/LIAAD/KeywordExtractor-Datasets

pISSN 2234-1072/eISSN 2234-0963

Journal of JAITC, Vol. 10, No. 1, pp. 45-56, Jul. 31, 2020
http://dx.doi.org/10.14801/JAITC.2020.10.1.45

Table 1. Statistics of the processed datasets used in our experiments

Dataset	KDD	WWW	Inspec
# of Docs (Train/Dev/Test)	755 (604/76/75)	1330 (1064/133/133)	2000 (1000/500/500)
Avg. # of Keyphrases	4.10	4.82	4.48
Max length of Tokens in Keyphrases	8	8	5
Min length of Tokens in Keyphrases	1	1	1
# of Tokens in Docs	144162	217747	259803
Avg. # of Tokens in Docs	72.08	108.87	129.90
Max # of Tokens in Docs	377	588	549
Min # of Tokens in Docs	3	3	11
uni-gram	788	2201	3088
n-gram (n >= 2)	2305	4204	6566

1) KDD: The KDD collection is based on the abstracts of papers collected from the ACM Conference on Knowledge Discovery and Data Mining (KDD) published during the period of 2004-2014, comprising a total of 755 documents. The gold-keywords of these papers are the author-labeled terms. The total data (755) were divided into 604, 76, and 75 points according to the ratio 8: 1: 1 for training, validation, and testing, respectively.
2) WWW: The WWW collection is based on the abstracts of papers collected from the World Wide Web Conference (WWW) published during the period of 2004-2014, comprising a total of 1330 documents. The gold-keywords of these papers are the author-labeled terms. The total data (1330) were divided into 1064, 133, and 133 by a ratio of 8: 1: 1, respectively.
3) Inspec: The Inspec collection is based on the abstracts of scientific journal papers from the CS domain collected during 1998-2002, comprising a total of 2000 documents. Each document is assigned two sets of keywords - the controlled keywords, which are manually controlled assigned keywords that appear in the Inspec thesaurus but may not appear in the document, and the uncontrolled keywords which are freely assigned by the editors, i.e., which are not restricted to the thesaurus or to the document. However, in our experiment, we use only the keywords that have been assigned by the authors in order to maintain consonance with the experiments on other datasets.

4.2 Experimental Results

1) Fine-tuning using (base) BERT:

Table 2 presents the KP extraction results on three different test collections (i.e., KDD, WWW, and Inspec) in the case in which pre-trained BERT is fine-tuned using the target data collections. The original BERT model [6] was pre-trained using WordPiece tokenization [32] and a vocab consisting of 30,000 tokens. In our experiments, we trained them using a batch size of 16, a learning rate of 5e-5, and 5 epochs were performed during the fine-tuning.

pISSN 2234-1072/eISSN 2234-0963

Table 2. KP extraction results using fine-tuned BERT

	1 epochs	2 epochs	3 epochs	4 epochs	5 epochs	Avg.	Max.	SOTA
KDD	29%	**41%**	38%	35%	40%	36.6%	41%	**41.08%**
WWW	36%	30%	**37%**	32%	32%	33.4%	37%	**39.43%**
Inspec	38%	42%	**48%**	44%	47%	43.8%	**48%**	42.80%

**The last column "SOTA" denotes state-of-the-art results for the designated data set. The SOTA results of KDD and WWW are obtained from Bidirectional LSTM+CRF Sequential labeling technique [20] and that of Inspec is obtained from the Bidirectional LSTM RNN sequential labeling approach [31].*

The best performances on each test collection were observed within 3 epochs of fine-tuning. We obtain 41% after 2 epochs on KDD data, 37% after 3 epochs on WWW data, and 48% after 3 epochs on Inspec data, respectively. Due the differences between the sizes and characteristics of the test collections, the trends of performance gain are observed to be dissimilar to each other. However, when the results of our experiments are compared with the SOTA ones obtained from previous experiments, the accuracies on KDD and Inspec are observed to be higher than their SOTA counterparts, while the WWW accuracy is observed to be lower than its SOTA counterpart. The experiment on Inspec data, in particular, exhibited an accuracy improvement by approximately 6% in terms of the F1-score via general domain contextual LM (i.e., BERT)-based fine-tuning.

2) Fine-tuning using SciBERT:

Table 3 depicts the results obtained by fine-tuning the pre-trained SciBERT on three different test collections (KDD, WWW, and Inspec). According to [29], as in the case of BERT, SciBERT uses the WordPiece tokenization introduced in [32] during pre-training to construct a model of 30K scientific vocab to match the number of BERT vocabs. The fine-tuning settings are almost identical to the BERT fine-tuning in the previous experiment. Following the trends of KP extraction via fine-tuned BERT in the previous experiment, the best performances obtained via the fine-tuning of SciBERT are also observed within 3 epochs. However, the improvement enhancements are much more substantial in this case. On KDD and WWW data, the KP extraction accuracies are observed to surpassed 45%. Further, almost 50% accuracy is achieved on the Inspec data. Unlike in the case of BERT-based fine-tuning, the maximum accuracy on each test collection is observed to surpass those obtained via existing state-of-the-art methods by significant intervals.

Table 3. KP extraction results obtained using fine-tuned SciBERT

	1 epoch	2 epochs	3 epochs	4 epochs	5 epochs	Avg.	Max.	SOTA
KDD	**45%**	43%	44%	44%	44%	44%	**45%**	41.08%
WWW	**48%**	43%	47%	46%	44%	45.6%	**48%**	39.43%
Inspec	45%	**50%**	**50%**	49%	48%	48.4%	**50%**	42.80%

**The last column "SOTA" scores are identical to those presented in Table 2.*

pISSN 2234-1072/eISSN 2234-0963

Journal of JAITC, Vol. 10, No. 1, pp. 45-56, Jul. 31, 2020
http://dx.doi.org/10.14801/JAITC.2020.10.1.45

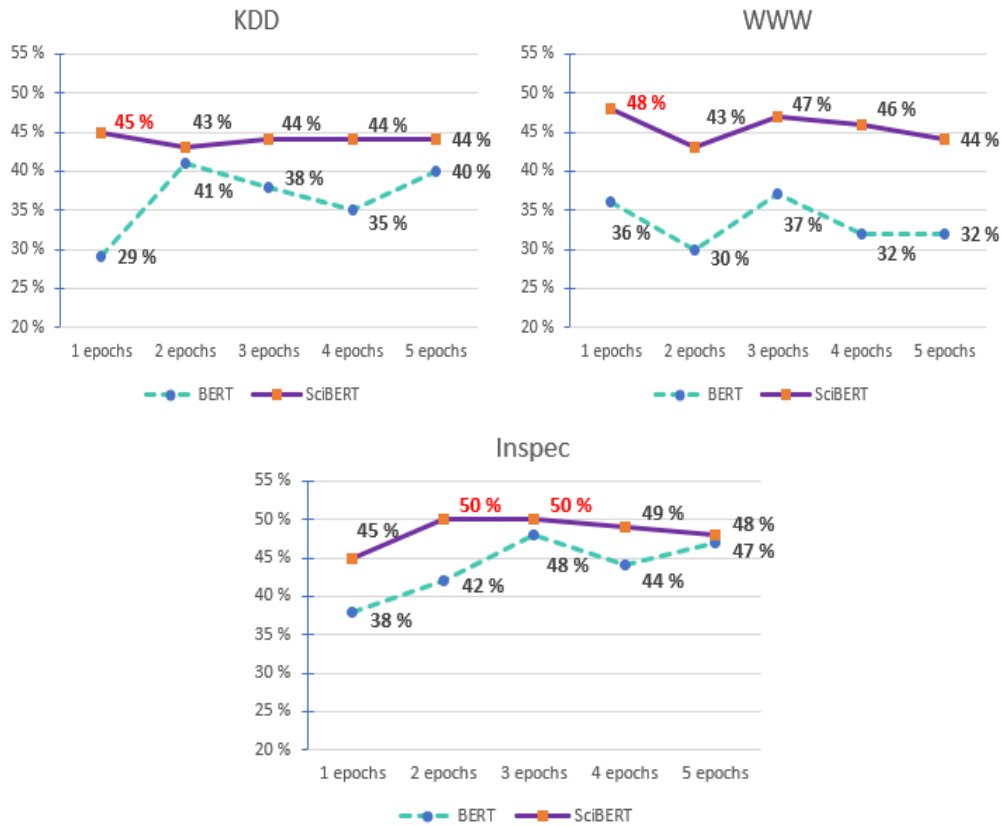

Figure 2. Comparison between BERT-based fine-tuning and SciBERT based fine-tuning

A comparison of the results obtained via BERT-based fine-tuning and SciBERT-based fine-tuning in terms of KP extraction reveals that the latter exhibited better performances on all three test datatests, even though both fine-tunings were performed using identical datasets. As depicted in Figure 2, corresponding to the KDD, WWW, and Inspec datasets, the maximum differences between the accuracies achieved by BERT-based fine-tuning and SciBERT-based fine-tuning are observed to be approximately 4%, 11%, and 2%, respectively. It is thus evident that SciBERT-based fine-tuning outperforms BERT-based fine-tuning.

Considering that the datasets (i.e., KDD, WWW, and Inspec) originated from the domain of CS and that SciBERT's original corpus also contained papers from the same domain, it is likely that the embeddings composed of SciBERT's vocab can aid in the understanding of the documents from the CS domain more significantly than the embeddings composed of BERT's vocab. We assume that the differences between the performances of the two architectures can be attributed to the aforementioned difference between the embeddings based on SciBERT's vocab and embeddings based on BERT's vocab.

pISSN 2234-1072/eISSN 2234-0963

5. Conclusion and Future Work

In this paper, we evaluated the performance of the BERT models with respect to KP extraction via fine-tuning. For this purpose, a series of experiments on BERT and SciBERT models were carried out on three different KP test collections obtained from the domain of CS. Based on the results of the experiments, SciBERT-based fine-tuning was verified to have carried over its exceptional performance in other NLP tasks to the task of KP extraction in scientific domain. In our future work, we intend to extend our analysis by considering additional KP resources and the KP 20k dataset [24] with respect to fine-tuning. The KP 20k datasets consists of 567,830 high quality computer science articles, thus it may allow us to examine various settings of fine-tuning. In particular, we are keen to identify the optimal BERT-based fine-tuning strategy for target data collection by analyzing the performances corresponding to varying amounts of training data and different BERT models.

Acknowledgements

This work was supported by Kumoh National Institute of Technology.

References

[1] R. Wang, W. Liu, and C. McDonald, "Corpus-independent generic keyphrase extraction using word embedding vectors", in Proc. of Software Engineering Research Conference, 1999, Vol. 39, pp. 1-8.

[2] X. Jiang, Y. Hu, and H. Li, "A ranking approach to keyphrase extraction", in Proc. of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Boston MA USA, July 2009, pp. 756–757.

[3] J. Mothe, F. Ramiandrisoa, and M. Rasolomanana, "Automatic keyphrase extraction using graph-based methods", in Proc. of the 33rd Annual ACM Symposium on Applied Computing, Pau France, April 2018, pp. 728-730.

[4] J. Howard, "Universal Language Model Fine-tuning for Text Classification", *Association for Computational Linguistics*, January 2018. [Online]. Available: http://arxiv.org/abs/ 1801.06146.

[5] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations", in Proc. of the North American Chapter of the Association for Computational Linguistics, March 2018, [Online]. Available: http://arxiv.org/ abs/ 1802.05365.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", in Proc. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, Vol. 1, pp. 4171-4186.

[7] Z. Zhang et al., "Semantics-aware BERT for Language Understanding", in Proc. of the Association for the Advancement of Artificial Intelligence, 2019, [Online]. Available: http://arxiv.org/abs/1909.02209.

[8] E. Papagiannopoulou and G. Tsoumakas, "A review of keyphrase extraction", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Vol. 10, No. 2, Sep. 2019.

[9] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval", *Journal of Documentation*, 1972. [Online]. Available: http://danigayo.info/teaching/SIW/PDF /sparckjones1972-bis.pdf.

pISSN 2234-1072/eISSN 2234-0963

Journal of JAITC, Vol. 10, No. 1, pp. 45-56, Jul. 31, 2020
http://dx.doi.org/10.14801/JAITC.2020.10.1.45

[10] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web", *Stanford InfoLab*, 1999. [Online]. Available: http://ilpubs.stanford.edu:8090/422.
[11] R. Mihalcea and P. Tarau, "TextRank: Bringing Order Into Texts", in Proc. of the Empirical Methods in Natural Language Processing, 2004, pp. 404-411.
[12] G. Erkan and D. R. Radev, "LexRank: Graph-based lexical centrality as salience in text summarization", *Journal of Artificial Intelligence Research*, Vol. 22, pp. 457–479, 2004.
[13] A. Bougouin and F. Boudin, "TopicRank : Graph-Based Topic Ranking for Keyphrase Extraction", in Proc. of the International Joint Conference on Natural Language Processing, 2013, pp. 543–551.
[14] S. Danesh, T. Sumner, and J. H. Martin, "SGRank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction", in Proc. of the fourth joint conference on lexical and computational semantics, 2015, pp. 117–126.
[15] X. Wan and J. Xiao, "Single document keyphrase extraction using neighborhood knowledge", in Proc. of the Association for the Advancement of Artificial Intelligence, Vol. 2, July 2008, pp. 855–860.
[16] W. Shi, W. Zheng, J. X. Yu, H. Cheng, and L. Zou, "Keyphrase Extraction Using Knowledge Graphs", *Data Science and Engineering*, Vol. 2, No. 4, pp. 275–288, November 2017.
[17] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge", in Proc. of the Empirical Methods in Natural Language Processing, Philadelphia, Pa. USA, July 2003, pp. 216–223.
[18] S. N. Kim and M. Y. Kan, "Re-examining automatic keyphrase extraction approaches in scientific articles", in Proc. of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications, Singapore, August 2009, pp. 9-16.
[19] T. D. Nguyen and M. Y. Kan, "Extraction in scientific publications", in Proc. of the International Conference on Asian Digital Libraries, 2007, pp. 317–326.
[20] S. D. Gollapalli and X. Li., "Keyphrase Extraction using Sequential Labeling", *Journal of Arxiv*, 2016. [Online]. Available: http://arxiv.org/abs/1608.00329.
[21] A. Graves, N. Jaitly, and A. R. Mohamed, "Hybrid speech recognition with Deep Bidirectional LSTM", in Proc. of the Institute of Electrical and Electronics Engineers, Olomouc, Czech Republic, December 2013, pp. 273-278.
[22] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks", in Proc. of the Institute of Electrical and Electronics Engineers, Montreal, Que., Canada , August 2005, pp. 2047-2052.
[23] Q. Zhang, Y. Wang, Y. Gong, and X. Huang, "Keyphrase extraction using deep recurrent neural networks on twitter", in Proc. of the Empirical Methods in Natural Language Processing, Austin, Texas, November 2016, pp. 836–845.
[24] R. Meng, S. Zhao, S. Han, D. He, P. Brusilovsky, and Y. Chi. (2017, Apri). "Deep keyphrase generation", *Annual Meeting of the Association for Computational Linguistics* [Online]. Available: https://arxiv.org/abs/1704.06879.
[25] R. Alzaidy, C. Caragea, and C. L. Giles, "Bi-LSTM-CRF sequence labeling for keyphrase extraction from scholarly documents", in Proc. of the World Wide Web Conference, San Francisco CA USA, May 2019, pp. 2551–2557.
[26] J. Pennington, R. Socher, "GloVe: Global Vectors for Word Representation", in Proc. of the Empirical Methods in Natural Language Processing, Doha, Qatar, October 2014, pp. 1532-1543.
[27] D. Sahrawat, et al, "Keyphrase Extraction from Scholarly Articles as Sequence Labeling using Contextualized Embeddings", in Proc. of the European Conference on Information Retrieval, October 2019, pp. 328-335.

pISSN 2234-1072/eISSN 2234-0963

[28] A. Radford and T. Salimans, "Improving Language Understanding by Generative Pre-Training", June 2018, [Online]. Available: https://s3-us-west-2.amazonaws.com/openai-assets/ research-covers/language-unsupervised/language_understanding_paper.pdf.
[29] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A Pretrained Language Model for Scientific Text", in Proc. of the Empirical Methods in Natural Language Processing, Hong Kong, China November 2019, pp. 3615–3620.
[30] A. Vaswani, et al., "Attention is all you need", in Proc. of the Advances in Neural Information Processing Systems, 2017, pp. 5998–6009.
[31] M. Basaldella, et al, "Bidirectional lstm recurrent neural network for keyphrase extraction", in Proc. Italian Research Conference on Digital Libraries, Udine, Italy, January 2018, pp. 180-187.
[32] Y. Wu, et al., "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation", *Arxiv*, October 2016. [Online]. Available: http://arxiv.org/abs/1609.08144.

Authors

Yeonsoo Lim

He is currently an undergraduate student of the Department of Computer Engineering at Kumoh National Institute of Technology. His research interests include NLP, Speech Recognition, and Deep Learning.

Deokjin Seo

He is currently an undergraduate student of the Department of Computer Engineering at Kumoh National Institute of Technology. His research interests include NLP, Speech Recognition, and Deep Learning.

Yuchul Jung

Received his B.E. degree in Computer Science from Ajou University, South Korea in 2001 and his M.E. in Information & Communication Engineering and Ph.D. degree in Computer Science from KAIST (Korea Advanced Institute of Science and Technology) in 2005 and 2011, respectively.
He joined the faculty of the Department of Computer Engineering at Kumoh National Institute of Technology (KIT), Gumi, as an assistant professor, in 2017. Prior to joining KIT, he worked as a senior researcher at Korea Institute of Science and Technology Information (KISTI) ('13~'17) and Electronics and Telecommunications Research Institute (ETRI) ('09~'13), Daejeon, South Korea. His research interests include AI, NLP, Speech Recognition, and Medicine 2.0.

pISSN 2234-1072/eISSN 2234-0963