J Inf Process Syst, Vol.15, No.5, pp.1119~1130, October 2019
https://doi.org/10.3745/JIPS.04.0142

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

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Fake News Detection Using Deep Learning

Dong-Ho Lee*, Yu-Ri Kim**, Hyeong-Jun Kim***, Seung-Myun Park****, and Yu-Jun Yang*****

Abstract

With the wide spread of Social Network Services (SNS), fake news—which is a way of disguising false information as legitimate media—has become a big social issue. This paper proposes a deep learning architecture for detecting fake news that is written in Korean. Previous works proposed appropriate fake news detection models for English, but Korean has two issues that cannot apply existing models: Korean can be expressed in shorter sentences than English even with the same meaning; therefore, it is difficult to operate a deep neural network because of the feature scarcity for deep learning. Difficulty in semantic analysis due to morpheme ambiguity. We worked to resolve these issues by implementing a system using various convolutional neural network-based deep learning architectures and "Fasttext" which is a word-embedding model learned by syllable unit. After training and testing its implementation, we could achieve meaningful accuracy for classification of the body and context discrepancies, but the accuracy was low for classification of the headline and body discrepancies.

Keywords

Artificial Intelligence, Fake News Detection, Natural Language Processing

1. Introduction

Since 2010, Social Network Services (SNSs) such as Facebook and Twitter have become widespread and fake news, which is a form of false information disguised as media, has started spreading. It had a significant impact on voting decisions in the 2016 US Presidential Election and became a hot topic [1]. Fake news on Facebook during the election was mainly used in support of a certain candidate [2]. Mainstream media around the world united to provide readers a confidence index for articles and employed people to monitor for fake news to prevent its spread [3]. In addition, there have been various attempts to solve this problem by taking a technical approach. For example, there are artificial intelligence (AI)-based detection methods and methods that detect the abnormal diffusion pattern of fake news propagation [4]. AI-based detection methods use models that have been trained on data; this method is classified as a natural language processing (NLP) task based on machine learning. Several previous works have garnered >80% accuracy using this method such as neural network models or decision trees [5,6].

However, Korean has two issues that cannot apply these works: (1) Korean can be expressed in shorter

Manuscript received June 22, 2018; first revision September 20, 2018; accepted October 31, 2018.
Corresponding Author: Dong-Ho Lee (danny911kr@naver.com)
* Dept. of Computer Education, Sungkyunkwan University, Seoul, Korea (danny911kr@naver.com)
** Dept. of Industrial & Management Engineering, Hansung University, Seoul, Korea (dbflek620@naver.com)
*** Dept. of Computer Science, Yonsei University, Seoul, Korea (hjkim2246@gmail.com)
**** Dept. of Information System, Hanyang University, Seoul, Korea (psm5moto@gmail.com)
***** Dept. of Software, Gachon University, Seongnam, Korea (defr5623@gmail.com)

sentences than English even with the same meaning; therefore, it is difficult to operate a deep neural network because of the lack of features for deep learning. (2) Difficulty in semantic analysis due to morpheme ambiguity. We resolve these issues and proposed a suitable fake news detection model for Korean by implementing a system that uses various convolutional neural network (CNN)-based deep learning architecture and "Fasttext" which is a word embedding model learned by syllable unit. Among the various types of fake news, we detect so-called "Click-bait" articles. In this paper, mission1 is the case in which the headline and body are inconsistent and mission2 is the case where the content of the body is irrelevant to the context.

2. Related Work

In this paper, we apply and transform various mechanisms based on "Fasttext" [7] and "Shallow-and-wide CNN" [8] to implement a model for detecting fake news. This section introduces previous related works that we use to implement models for fake news detection.

2.1 Word Embedding

Word embedding is a method of mapping words or phrases to vectors of real numbers. The traditional method, "discrete representation" has a "one-hot vector" representation that consists of 0 second in all dimensions with the exception of a single 1 in only one dimension that is used to represent the word. However, "discrete representation" does not reflect the context and has problems handling synonyms and antonyms. Recently, "distributed representation" has emerged as a way to represent words in a continuous vector space where all dimensions are required to represent the word. This paper introduces and applies "Word2vec" and "Fasttext" among various representations.

2.1.1 Word2vec

"Word2vec" represents word embedding using a neural network; it has two model architectures for learning distributed representations of words: continuous bag-of-words (CBOW) and Skip-gram. The Skip-gram architecture is widely used because it works better on semantic tasks than the CBOW model [9]. The Skip-gram architecture uses each current word as an input (w_t) for the model and predicts words within a certain range before and after the current word ($w_{t-k} \sim w_{t+k}$). It maximizes the classification of a word based on another word in the same sentence, so similar words have similar vectors and their similarity increases [10]. Given a sequence of training words ($w_1 \sim w_t$) and the size of training context (c), the objective of the Skip-gram model is to maximize the average log probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0}\log p(w_{t+j}|w_t) \tag{1}$$

The basic Skip-gram formulation defines $p(w_{t+j}|w_t)$ using the softmax function as follows:

$$p(w_O|w_I) = \frac{\exp({v'_{w_O}}^{\top} v_{w_I})}{\sum_{j=1}^{W}\exp({v'_j}^{\top} v_{w_I})} \tag{2}$$

where v_w and v'_w are the "input" and "output" vector representations of w, and W is the number of words in the vocabulary [11].

2.1.2 Fasttext

"Fasttext" is a method of adding the concept of "Sub-word" to "Word2vec." Each word is represented as the sum of n-gram vectors and the word vector itself. Taking the word *apple* and $n = 3$ as an example. It will be represented by the character n-grams: <*ap, app, ppl, ple, le*> and the word itself <*apple*>. The reason why it has 2-gram vector <*ap, le*> is that it adds special boundary symbols <*and*> at the beginning and end of words to distinguish prefixes and suffixes from other character sequences. The formulation is as follows: suppose that you are given a dictionary of n-grams of size G. Given a word w, let us denote by $G_w \subset \{1, \ldots, G\}$ the set of n-grams appearing in w. z_n is a vector representation which is associated to each n-gram g. Thus, v_w in Eq. (2), which is the "input" vector representation of input word can be represented as follows [7]:

$$v_w = \sum_{g \in G_w} z_n \tag{3}$$

2.2 Shallow-and-Wide CNN

The model architecture as shown in Fig. 1, is the "Shallow-and-wide CNN" architecture of Kim [8]. The first layer is the look-up table that is the set of k-dimensional word vectors that each corresponds to the i^{th} word in the sentence. Then, a convolution operation is applied with multiple filter widths and a max-overtime pooling operation. Finally, these features are passed to a fully connected layer and the prediction is made with the softmax layer.

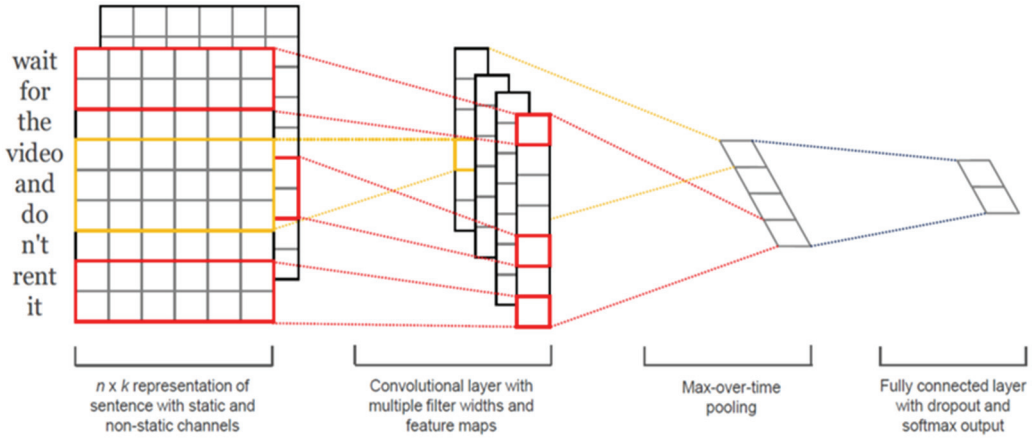

Fig. 1. Shallow-and-wide CNN architecture [8].

It has two channels of word vectors: one named "Static" that is kept static throughout the training and one named "Non-static" that is fine-tuned via backpropagation. Previous works have conducted sentimental analysis with a dataset that consists of short sentences and the "Static" and "Non-static" results were comparable, but "Non-static" allows the words to attain more meaningful representations

[8]. However, if only "Non-static" is used, new words can be over-fitted in this model. Therefore, both channels are used to secure the generality of the meaning of words.

2.3 Attentive Pooling

The model architecture shown in Fig. 2 is the "Attentive-pooling" architecture of Santos et al. [12]. Recently, attention mechanisms have been successfully used for image captioning [13] and machine translation [14]. However, there were no further studies of applying the attention mechanism to NLP tasks with two inputs such as pair-wise ranking or text classification. Meanwhile, "attentive pooling" has contributed to the improvement of performance in these tasks by effectively representing the two inputs' similarity [12]. Although there is the Term Frequency-Inverse Document Frequency (TF-IDF) method that statistically measures the similarity by the frequency of words in a document, this model measures the similarity by increasing the weight for words that have the same or similar meanings to the two inputs.

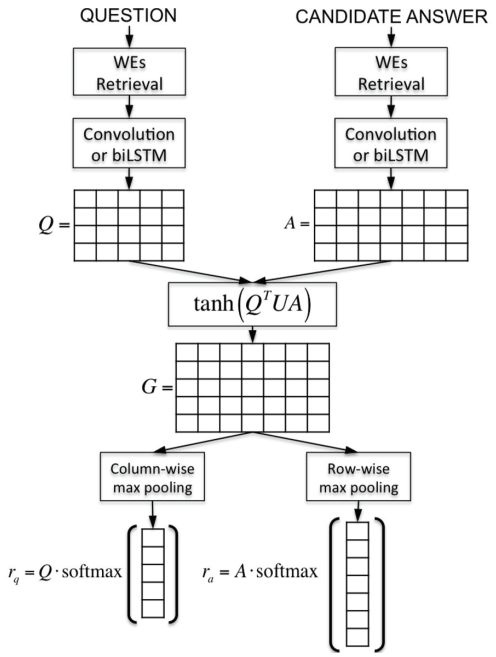

Fig. 2. Attentive pooling networks for answer selection [12].

2.4 Bi-LSTM

Long short-term memory (LSTM) is a structure that learns how much of the previous network state to apply when input data is received. It resolves the long-term dependency problem of conventional recurrent neural network (RNN) using both the hidden state and the cell state, which is a memory for storing past input information and the gates that are used to regulate the ability to remove or add information to the cell state. The multiplicative gates and memory are defined for time t [15]:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{4}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{5}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

$$C_t = f_t * C_{t-1} + i_t * tanh(W_c \cdot [h_{t-1}, xt] + b_c) \tag{7}$$

$$h_t = o_t * \tanh(C_t) \tag{8}$$

where $\sigma(\cdot)$ is the sigmoid function and f_t, i_t, o_t, C_t, and h_t are the vectors of the forget gate, input gate, output gate, memory cell, and hidden state, respectively. All of the vectors are the same size. Moreover, W_f, W_i, W_o, and W_c denote the weight matrices of each gates and b_f, b_i, b_o, and b_c denote the bias vectors of each gates. Another shortcoming of conventional RNN is that they are only able to make use of previous context [16]. To resolve this, bidirectional-RNN (Bi-RNN) stacks two RNN layers. If the existing RNN is the forward RNN that only forwards previous information, Bi-RNN stacks backward RNN that can receive subsequent information, as shown in Fig. 3. Combing Bi-RNN with LSTM gives Bidirectional-LSTM (Bi-LSTM), which can handle long-range context in both input directions [16].

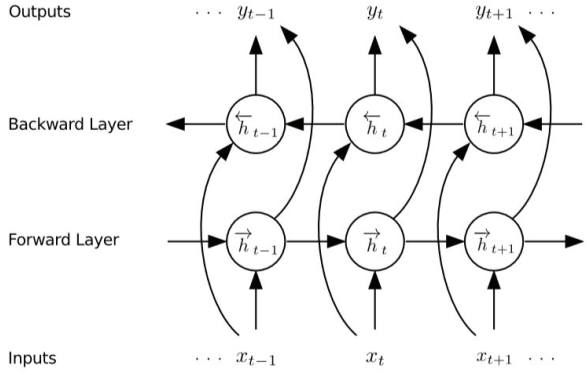

Fig. 3. Bidirectional-RNN. Adapted from Graves et al., "Speech recognition with deep recurrent neural networks," *Proceedings of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645-6649, with the permission of IEEE [16].

3. Model Architecture

This paper modifies and combines "Fasttext" and "Shallow-and-wide CNN" to implement a fake news detection model. To detect so-called "Click-bait" articles among the various types of fake news, we need to understand the consistency and relevance between the headline and body of article. To do this, we extract the global feature vectors from the headline and body, respectively and compare the vectors. For extracting method, there are several methods such as TF-IDF and RNN, but since the overall meaning of text is determined by a few key words in the text, we use CNN which can extract the most salient local features to form fixed-length global feature vector [17]. Then, we pass these features to a fully connected layer and make the prediction with softmax layer. We call this model "BCNN (Bi-CNN)" because for the headline and the body which are two inputs of model, the convolution and the pooling have been used. Moreover, we try to improve the accuracy by implementing new models by applying LSTM/Bi-LSTM and attentive pooling to BCNN; in this section, we first apply "Word2vec" and "Fasttext", which are representative word embedding techniques, to Korean and compare the accuracy. Then, we introduce several BCNN models with better performance word embedding technique.

3.1 Word Embedding

We train 100K articles with "Word2vec" and "Fasttext" to find suitable word embedding for Korean; the results are as shown in Table 1.

This paper uses "Fasttext" because its performance is better in terms of accuracy.

Table 1. Test results for "Word2vec" and "Fasttext"

		한국 : 문재인 ? : 김정은	한국 : 서울 ? : 도쿄
Word2vec	Batch: 5000 Epochs: 50	미국, 데니스, 로드먼	일본 0.71
Fasttext	Epochs: 5	미국, **북한**, 중국	**일본 0.76**

For the first question, geometry of words should show that not only "문재인" and "김정은" can be clustered each other, but they can each have similar distances in vectors space to the corresponding countries which are "한국" and "북한". As "Fasttext" gives "북한" and "일본" for above each question, "Fasttext" shows more comprehensive geometry of words.

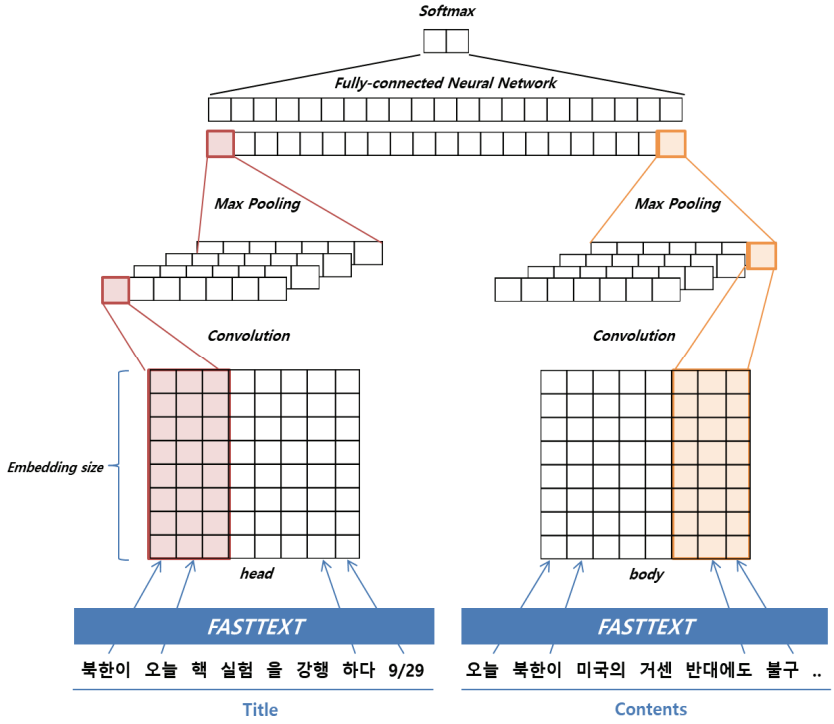

Fig. 4. BCNN architecture.

3.2 BCNN

BCNN is a CNN with two inputs and pre-trained word embedding "Fasttext" as shown in Fig. 4. It extracts feature maps from headlines and bodies using 3-g filters through the convolution layer. The number of filters is proportionally set as 256 filters for the headline and 1024 filters for the body considering the huge difference in the amount of text between them. Then, it makes each feature map to one vector through the max-pooling layer. This is the process of forming fixed-length global vector for

the headline and the body. Finally, classification is performed through the fully connected layer. We use Rectified Linear Unit (ReLU) as an activation function and the softmax function as an output function. We use the "Static" channel that keeps word embedding static with pre-trained word embedding "Fasttext" throughout training.

3.3 (Bi-)LSTM + BCNN

(Bi-)LSTM + BCNN applies the context information to existing word embedding by training it with (Bi-)LSTM as shown in Fig. 5. We expect an improved performance because each word vector would have both the vector trained by "Fasttext" and the context information in the sentence.

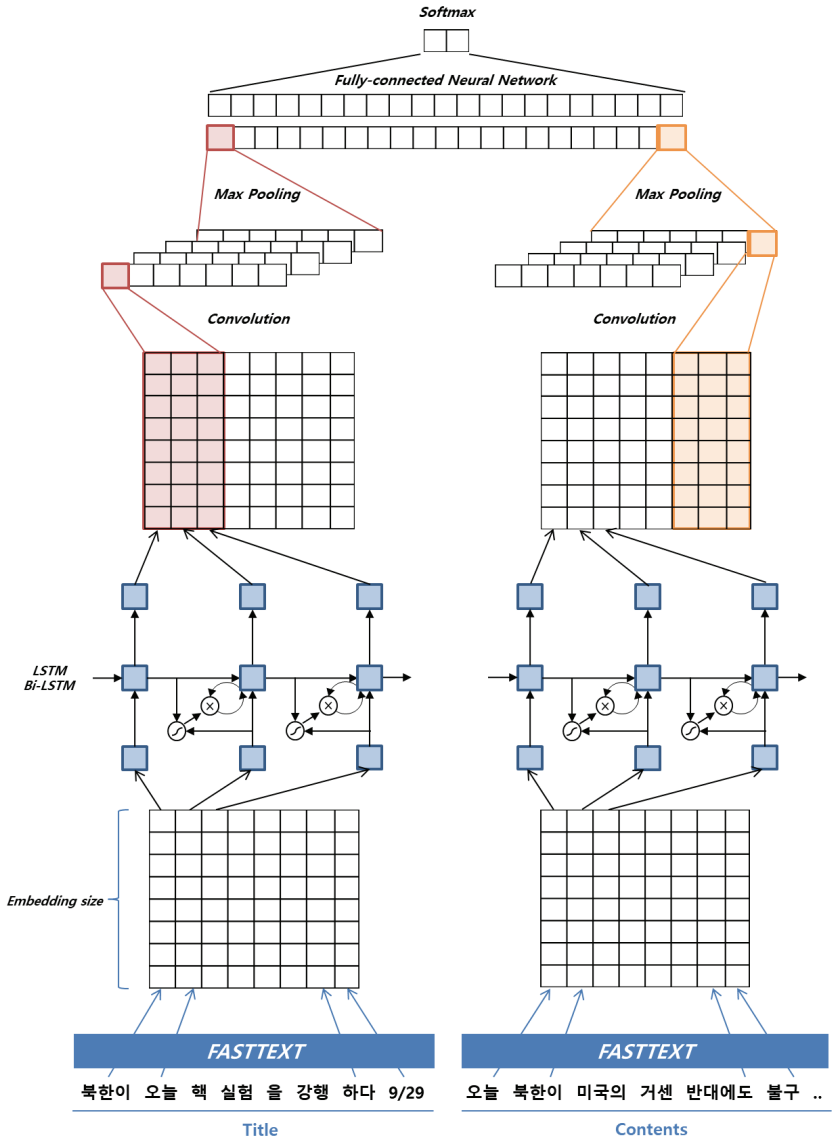

Fig. 5. LSTM/Bi-LSTM + BCNN architecture.

3.4 BCNN with Attentive Pooling Similarity

BCNN with Attentive Pooling Similarity (APS-BCNN) adds a similarity vector between two inputs represented using attention pooling to a one-dimensional vector matrix that is derived from max-pooling in BCNN, as shown in Fig. 6. We expect improved performance because the similarity vector has been added.

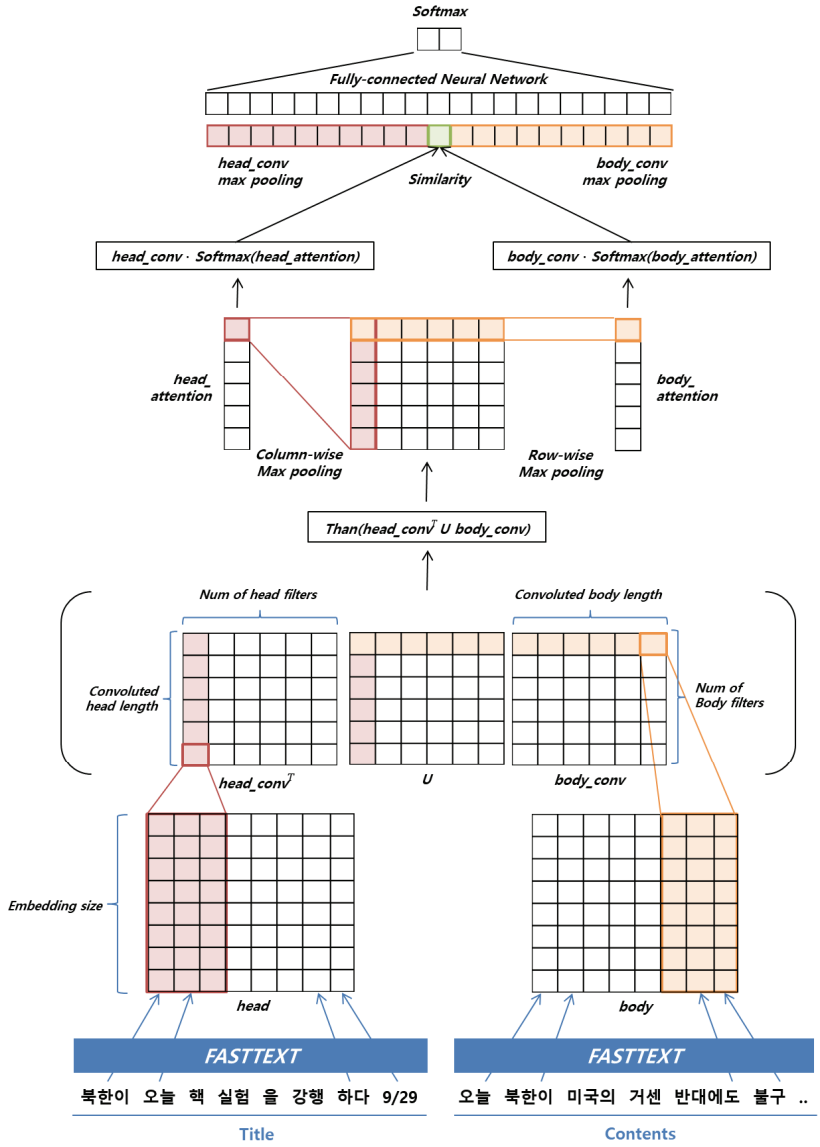

Fig. 6. BCNN with attention pooling similarity architecture.

3.5 Hyperparameters

Hyperparameters are as shown in Table 2. As mentioned above, the number of filters is proportionally set considering the huge difference in the amount of text between the headline and the body.

Table 2. Hyperparameters

Label	Description	Optimized
filter_size	Size of filters	3
num_filters	Number of filters	256 (headline), 1024 (body)
dropout	Dropout rate	0.5
L2_alpha	L2 regularization lambda	0.1
batch_size	Mini-batch size	64
embedding_dim	Word embedding dimension	128

4. Experiments

This paper detects so-called "click-bait" articles among the various types of fake news. We define mission1 as where the headline and body are inconsistent and mission2 as where the content of the body is irrelevant to the context (Table 3).

Table 3. Example of fake news for each mission (Korean)

	Mission 1	Mission 2
Headline	카타르 축구팀 '도하 참사' 한국에 33년 만에 패배	'해리포터' 작가 사인회가 열린다.
Body	한국 축구대표팀은 카타르에 결승골을 내주며 2-3으로 무릎을 꿇었다.	'해리포터'의 작가 J.K Rowling의 사인회가 마련됐다. ... 한편 배우 김지연이 패션쇼에 참석해 과감한 노출 패션으로 시선을 모았다.

4.1 Dataset

We use 100K articles that were crawled from Joongang Ilbo, Dong-A Ilbo, Chosun Ilbo, Hankyoreh, and Maeil businesses as a dataset. For each press, we categorize the news into economic, society, politics, entertainment, and sports, and then collect articles in the same proportion for each category. Of these, we use 31K for mission1 and 68K for mission2. The real news and fake news are in the same proportion for each mission and the training and validation data are composed at a ratio of 9:1. We measure the model's accuracy with test data that includes 350 recent articles (as of March 2018) that are not included in the training and validation data and are composed of real news and fake news in the same proportion.

4.2 Experiment Results

We measure the accuracy using the model with the lowest validation loss among multiple steps of the model and the results are shown in Table 4; AUROC (area under receiver operating characteristic curve) is used as a measurement technique [18].

Table 4. Fake news classification accuracy results

Model	Mission 1	Mission 2
BCNN	0.528	0.720
LSTM – BCNN	0.413	0.718
Bi-LSTM – BCNN	0.417	0.707
APS – BCNN	0.454	0.726

5. Conclusions

This paper implements a deep learning model for fake news detection and measures the accuracy; its main contributions are as follows:

(1) The accuracy of classification for mission2, which consists of fake news that is irrelevant to the article context, is the highest with APS-BCNN at an AUROC score of 0.726. It can be concluded that the similarity vector between the headline and body contributes to detecting the content that is irrelevant to the context.

(2) The accuracy of classification for mission1, which consists of fake news where the headline and body are inconsistent, is the highest with a BCNN in AUROC score of 0.52; however, this accuracy cannot be used to detect real fake news. We can deduce the causes of low accuracy as follows: (a) as CNN uses the local information of texts to classify, mission2 would have achieved high accuracy due to the large amount of perturbed local information. However, as mission1 has a relatively small amount of perturbed local information, it would have been difficult to classify it. (b) The difference in the amount of training data between mission1 and mission2 would have caused a difference in the accuracy between missions; we were able to acquire a large amount of fake news data for mission2 by mixing parts of the bodies of several articles. However, since we had to make the fake news data for mission1 individually, it was difficult to acquire a large amount of fake news data like for mission2.

(3) CNN with LSTM has low classification accuracy. Although there is a previous work of LSTM-CNN with high accuracy in the text classification of one input [19], the application of LSTM in the text classification of two inputs as shown in this paper had low accuracy. We can deduce the cause of the low accuracy as follows: for example, if we assume that both the headline and body have the same word "apple" as shown in Table 5.

Table 5. Example article

	Example
Headline	This **apple** is so tasty.
Body	The red **apple** on the desk seems so tasty.

Before the LSTM has been applied, the word "apple" in both the headline and body would have had the same vector trained by "Fasttext." However, after LSTM has been applied, each word is influenced by the preceding words and has different vectors. This reduces the association of "apple" in the headline and "apple" in the body even though they are the same word.

(4) "Fasttext" performs better than "Word2vec" in terms of Korean word similarity. We can deduce the cause of the better performance as follows: unlike other languages, the syllables that form Korean words have their own meaning. For example, the word "대학" is composed of the syllables "대" that means "big" and "학" that means "learn." This would have made "Fasttext" which is trained in syllable unit, perform better in word similarity than "Word2vec" which is trained in word unit.

This paper proposes a meaningful deep learning model for fake news detection. The limitation of this study is that we could achieve meaningful accuracy for classification in the case where the content of the body is irrelevant to the context, but the accuracy was low when the headline and body were inconsistent.

In future work, we will implement a big-data system to collect and make good-quality fake news for training data and retrain our model to improve the accuracy.

Acknowledgement

This paper is recommended from the 2018 Korea Information Processing Society Spring Conference. All code and data is available on our Github (https://github.com/2alive3s/Fake_news).

References

[1] Y. Yoon, T. Eom, J. Ahn, H. Lee, and J. Heo, "Survey of fake news detection technology," *IITP Weekly Trend*, vol. 1816, pp. 12-23, 2017.

[2] C. Silverman, "This analysis shows how viral fake election news stories outperformed real news On Facebook," 2016 [Online]. Available: https://www.buzzfeednews.com/article/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook#.sgKVv8V32q.

[3] The Trust Project, "News with integrity," [Online]. Available: https://thetrustproject.org/.

[4] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent features of rumor propagation in online social media," in *Proceedings of 2013 IEEE 13th International Conference on Data Mining*, Dallas, TX, 2013, pp. 1103-1108.

[5] W. Largent, "Talos targets disinformation with fake news challenge victory," 2017 [Online]. Available: https://blog.talosintelligence.com/2017/06/talos-fake-news-challenge.html.

[6] A. Hanselowski, "Team Athene on the fake news challenge," 2017 [Online]. Available: https://medium.com/@andre134679/team-athene-on-the-fake-news-challenge-28a5cf5e017b.

[7] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," 2016 [Online]. Available: https://arxiv.org/abs/1607.04606.

[8] Y. Kim, "Convolutional neural networks for sentence classification," 2014 [Online]. Available: https://arxiv.org/abs/1408.5882.

[9] 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.

[10] W. J. Kim, D. H. Kim, and H. W. Jang, "Semantic extention search for documents using the Word2vec," *The Journal of the Korea Contents Association*, vol. 16, no. 10, pp. 687-692, 2016.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111-3119, 2013.

[12] C. D. Santos, M. Tan, B. Xiang, and B. Zhou, "Attentive pooling networks," 2016 [Online]. Available: https://arxiv.org/abs/1602.03609.

[13] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: neural image caption generation with visual attention," in *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, 2015, pp. 2048-2057.

[14] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, 2015.

[15] M. Maimaiti, A. Wumaier, K. Abiderexiti, and T. Yibulayin, "Bidirectional long short-term memory network with a conditional random field layer for Uyghur part-of-speech tagging," *Information*, vol. 8, no. 4, article no. 157, 2017.

[16] A. Graves, A. R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, Canada, 2013, pp. 6645-6649.

[17] W. T. Yih, X. He, and C. Meek, "Semantic parsing for single-relation question answering," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Baltimore, MD, 2014, pp. 643-648.

[18] K. A. Spackman, "Signal detection theory: Valuable tools for evaluating inductive learning," in *Proceedings of the 6th International Workshop on Machine Learning*, Ithaca, NY, 1989, pp. 160-163.

[19] P. M. Sosa, "Twitter sentiment analysis using combined LSTM-CNN models," 2018 [Online]. Available: http://konukoii.com/blog/2018/02/19/twitter-sentiment-analysis-using-combined-lstm-cnn-models/.

Dong-Ho Lee https://orcid.org/0000-0001-8749-9833

He received B.S. degree from the Department of Computer Education at Sungkyunkwan University in 2018. Since August 2018, he has been with the School of Computer Science at the University of Southern California as an M.S. candidate. His current research interests include natural language processing, machine learning, and computer-supported learning.

Yu-Ri Kim https://orcid.org/0000-0002-5801-4056

She is an undergraduate student in Department of Computer Science and Industrial & Management Engineering from Hansung University since 2013. Her current research interests include AI and NLP.

Hyeong-Jun Kim https://orcid.org/0000-0002-2257-6655

He is an undergraduate student in Department of Computer Science in Yonsei University since 2013. His current research interests include AI, computer vision and data analysis.

Seung-Myun Park https://orcid.org/0000-0001-9317-9262

He received B.S. degree in Department of Information System in Hanyang University. He now works at CV Corporation as an Android developer.

Yu-Jun Yang https://orcid.org/0000-0002-9061-3374

He is an undergraduate student in Department of Software in Gachon University since 2018. His current research interests include AI, natural language processing, and semiconductor engineering.