applied sciences

MDPI

Article

Reliable Classification of FAQs with Spelling Errors Using an Encoder-Decoder Neural Network in Korean

Youngjin Jang and Harksoo Kim *

Program of Computer and Communications Engineering, Kangwon National University, Chuncheon 24341, Korea; dan_yon@kangwon.ac.kr

* Correspondence: nlpdrkim@kangwon.ac.kr; Tel.: +82-33-250-6388

Received: 7 October 2019; Accepted: 3 November 2019; Published: 7 November 2019

Abstract: To resolve lexical disagreement problems between queries and frequently asked questions (FAQs), we propose a reliable sentence classification model based on an encoder-decoder neural network. The proposed model uses three types of word embeddings; fixed word embeddings for representing domain-independent meanings of words, fined-tuned word embeddings for representing domain-specific meanings of words, and character-level word embeddings for bridging lexical gaps caused by spelling errors. It also uses class embeddings to represent domain knowledge associated with each category. In the experiments with an FAQ dataset about online banking, the proposed embedding methods contributed to an improved performance of the sentence classification. In addition, the proposed model showed better performance (with an accuracy of 0.810 in the classification of 411 categories) than that of the comparison model.

Keywords: FAQ classification; encoder-decoder neural network; multi-level word embeddings

1. Introduction

Frequently asked questions (FAQs) in commercial services based on social media (e.g., chatbot for online banking) accommodate both customer needs and business requirements. As a useful tool for information access, most commercial services provide customers with a keyword search. However, sometimes the keyword search does not perform well in FAQ retrieval because of lexical disagreements between users' queries and the predefined questions in an FAQ set, as shown in Figure 1.

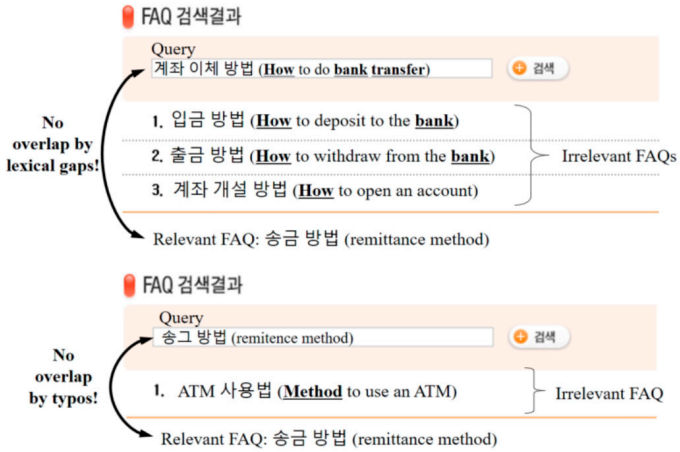

Figure 1. Motivational example.

In Figure 1, the lexical disagreements are caused by using different words with the same meanings (e.g., remittance vs. bank transfer), and by using incorrect words with spelling errors (e.g., remittance

vs. remitence). To resolve these lexical disagreement problems, most FAQ retrieval systems expand keywords by looking up synonym dictionaries and bridge lexical gaps between different words with the same meanings. However, they cannot cope with the lexical agreement problem caused by spelling errors because it is impossible to pre-construct a synonym dictionary containing all misspelled keywords. Recently, FAQ classification models based on deep learning have been proposed because they have the ability to cluster semantically or lexically similar words through various distributed representation schemes like word embeddings and character embeddings. In this paper, we propose an FAQ classification model based on an encoder-decoder neural network with multiple word embedding vectors instead of keyword search methods. To increase FAQ classification performance, the proposed model adopts class embeddings, including domain knowledge of each FAQ category.

2. Previous Works

Initial sentence classification models based on deep learning were n-gram models using convolutional neural networks (CNNs) [1–5]. The authors of [3] proposed a CNN architecture using diverse versions of pre-trained static word vectors and variable size convolution filters. It was shown in [2] that simple convolutions of word n-grams could contribute to improving the performance of sentence classification by fine-tuning pre-trained static word vectors like Word2Vec [6]. These n-gram models were effective in exploring the regional syntax of words, but they could not account for order-sensitive situations where the order of words was critical to the meaning of a sentence. To overcome this problem, [7] proposed a classification model combined with a recurrent neural network (RNN) and a CNN. Then, some studies demonstrated that sub-word units like character n-grams could contribute to improving the performance of downstream natural language processing (NLP) tasks [8–13]. The authors of [12] proposed a part-of-speech tagging model based on an RNN in which each word is represented by a combination of Korean alphabet embeddings for making the model robust to typing errors. The authors of [13] proposed a character-level CNN model for text classification which showed that the character-level CNN model could achieve state-of-the-art or competitive results. In addition, [14] demonstrated that domain embeddings (i.e., embeddings of predefined categories) could contribute to improving the performance of large-scale domain classification. Recently, bidirectional encoder representations from transformers (BERT) was proposed [15], which is deeply bidirectional, unsupervised language representation that is pre-trained using a large amount of plain text corpus. BERT has shown state-of-the-art performance in many downstream NLP tasks such as classification, sequence labeling, and span prediction by learning task-specific vectors through fine-tuning. In sentence classification tasks such as sentiment analysis and semantic textual similarity analysis, BERT also outperformed the previous state-of-the-art models.

3. FAQ Classification Using an Encoder-Decoder Neural Network

Figure 2 shows the overall architecture of the proposed FAQ classification model. As shown in Figure 2, the proposed model consists of an embedding layer, a transformer encoder with attentions, and an RNN decoder.

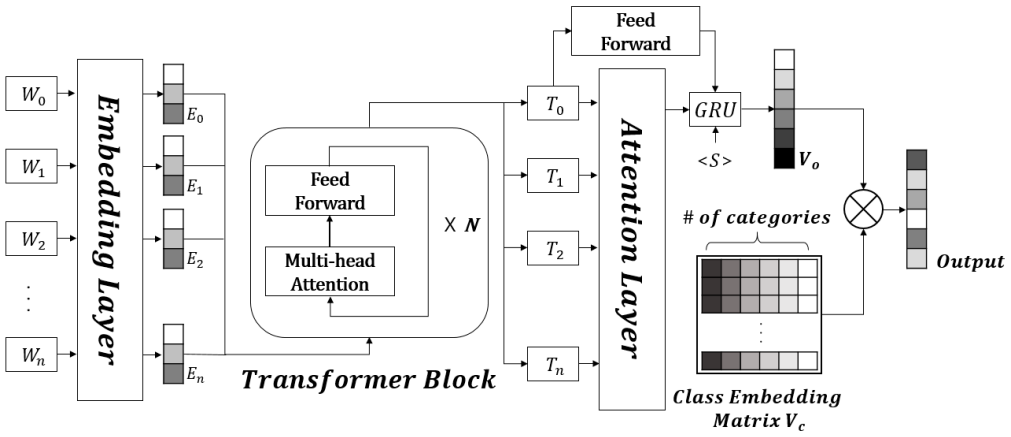

Figure 2. Overall architecture of the proposed model.

To make the proposed model robust to lexical disagreements, the embedding layer consists of three types of embedding vectors: Fixed word embedding vectors, fine-tuned word embedding vectors, and character-level word embedding vectors using a CNN. We expect that the fixed word embedding vectors represent domain-independent meanings of each word, and the fine-tuned word embedding vectors represent domain-specific meanings of each word. For example, we hope that "transfer" has the domain-independent meaning "move something" and the domain-specific meaning "send money" in a banking domain. We also expect that the character-based word embedding vectors alleviate lexical disagreement problems that are raised by spelling errors. For example, we hope that the misspelled word "remitence" has a similar vector representation with "remittance." In Figure 2, W_0 and E_0 are [CLS] (a special symbol added in front of every input example) and an embedding of [CLS], respectively. W_i except W_0 and E_i except E_0 are the i-th word in a sentence, and its embedding vector concatenated with three types of word embedding vectors, respectively. Figure 3 exemplifies three types of word embedding vectors.

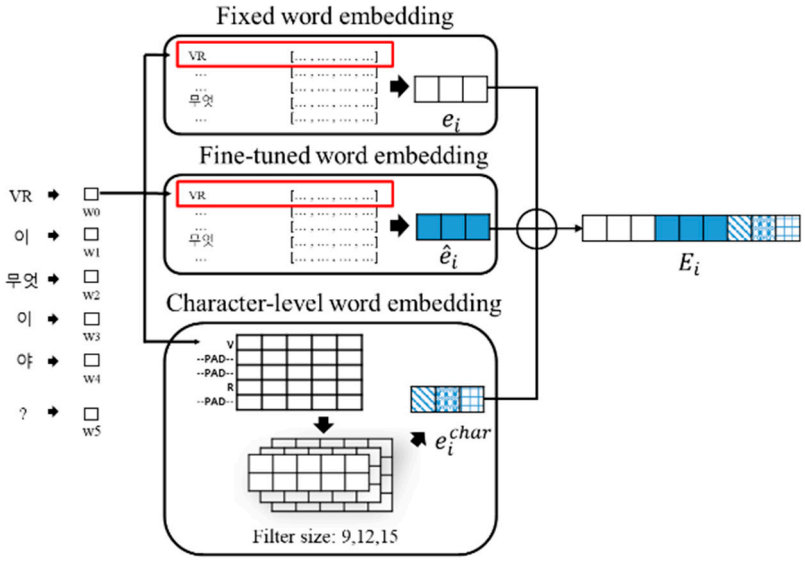

Figure 3. Embedding layer of the proposed model. The Korean sentence "VR 이 무엇이야?" means "What is VR?" in English.

In Figure 3, e_i, $\hat{e}_i$, and e_i^{char} are a fixed word embedding vector, a fine-tuned word embedding vector, and a character-level word embedding vector of the i-th one among n words in an input sentence

S (i.e., an input query or a predefined question in a FAQ set), respectively. e_i^{char} is generated by a CNN, as shown in the following equation.

$$e_i^{char} = CNN(c_1, c_2, \ldots, c_j, \ldots, c_l), \quad (1)$$

where c_j is the j-th one among l characters in a word w_i. In this paper, a character refers to the Korean characters called *jamo*. A final word embedding vector E_i is represented by the concatenation of e_i, $\hat{e}_i$, and e_i^{char}, as shown in the following equation:

$$E_i = \left[e_i; \hat{e}_i; e_i^{char}\right]. \quad (2)$$

To supplement word embedding vectors with contextual information, we adopt an encoder-decoder neural network in which word embedding vectors are encoded by a transformer's encoder [16]. The output T_i of the transformer's encoder is represented by a multi-head scaled dot-product self-attention mechanism, as shown in the following equations.

$$Q = K = V = E \quad (3)$$

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (4)$$

where E is one among n word embedding vectors, and Q, K, and V are a query, a key, and a value for calculating attentions, respectively. Then, d_k is the size of E for scaling dot-products. In Equation (4), the query, key, and value are the same vectors according to Equation (3). This case is called self-attention, relating different positions of a single sequence $E_1, E_2, \ldots, E_n$, to compute a representation of the sequence. Self-attention has been successfully used in various NLP tasks, such as machine translation, machine reading comprehension, abstractive document summarization, etc. The query, key, and value are first linearly transformed into N heads. Then, each head is entered into Equation (4). Therefore, the self-attention is calculated N times, making it so-called multi-headed.

The first output T_0 (the final output vector of the special [CLS] token) of the transformer's encoder is input as an initial value of the RNN decoder, implemented by a gated recurrent unit (GRU) [17] with Luong's encoder-decoder attention mechanism [18], after passing through a fully connected neural network (FNN). Figure 4 shows the RNN decoder with Luong's encoder-decoder attention mechanism in detail.

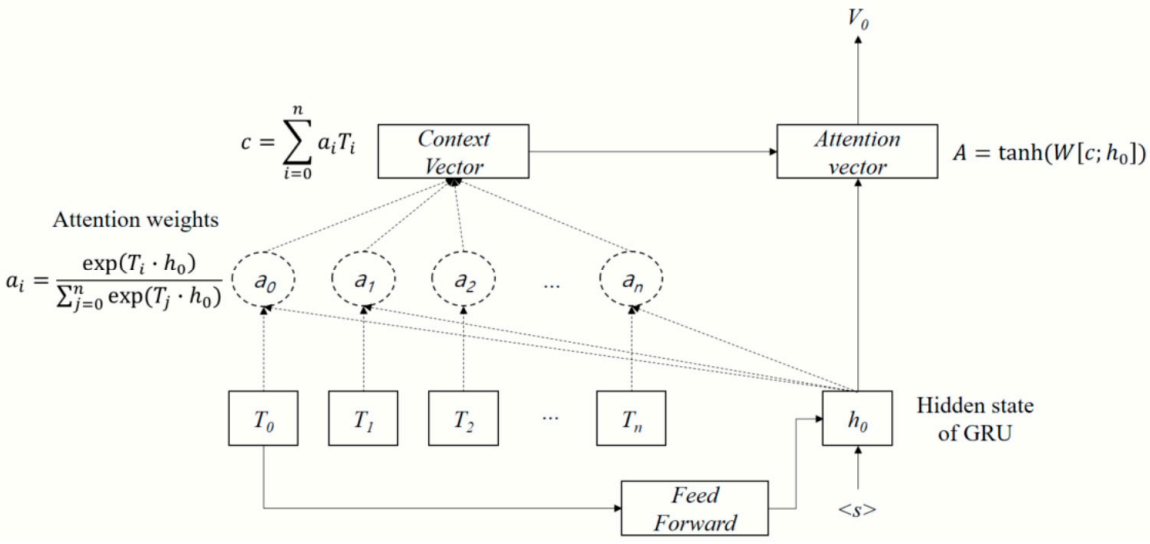

Figure 4. Recurrent neural network (RNN) decoder with Luong's encoder-decoder attention mechanism.

As shown in Figure 4, each attention weight a_i is induced by inner products between each output T_i of the transformer's encoder and the first hidden state h_o of the RNN decoder. The attention weights mean how much each output T_i is associated with the hidden state h_o. Then, the context vector c is constructed by the weighted sum of a_i and T_i. Finally, the RNN decoder generates an output vector V_o using the FNN-encoded input sentence $FNN(T_0)$, the start symbol <*S*>, and the context vector c, as shown in the following equation:

$$V_o = Dec(FNN(T_0), < s >, c). \tag{5}$$

To supplement the output vector V_o with domain-specific knowledge, we adopt a domain embedding scheme proposed by [14]. We define one class embedding vector per FAQ category, as shown in the following equation.

$$V_{C_t} = mean(\sum_k e_k), \tag{6}$$

where V_{C_t} is a class embedding vector that is calculated as an average of the word embedding vectors, e_k's, in sentences belonging to the t-th FAQ category. Finally, to classify input sentences into FAQ categories, we use an FNN. The vector of inner products between the output vector V_o of the RNN decoder and the class embedding matrix V_C is used as an input vector of the FNN.

4. Evaluation

4.1. Data Sets and Experimental Settings

We collected an FAQ dataset (10,495 pairs of FAQs about online banking). The FAQ dataset is a set of users' queries manually annotated with FAQ categories. The queries had many spacing errors and spelling errors because they were collected from a real mobile app service. An FAQ in the dataset consists of, on average, 23.3 *eumjeol*s (Korean syllables) and contains, on average, 0.7 typo-like spelling errors and spacing errors. The number of FAQ categories was 411. Table 1 shows a sample of the FAQ dataset.

Table 1. Sample of the FAQ dataset.

Sentence (Korean)	Sentence (English)	ID of FAQ Category
간편 이체	Easy bank transfer	3
쉽게 송금하는 방법	How to easily send money	3
비밀 번호 변경	How to change the password	7
비번 바꾸는 법	How to change PW	7

Figure 5 shows a full histogram of the data distribution over the full 411 categories. Figure 6 shows the distribution of FAQ categories according to the number of queries included in each FAQ category.

As shown in Figures 5 and 6, 63% of FAQ categories included less than six queries. To evaluate the proposed model, we divided the FAQ dataset into a training set, a validation set, and a test set by a ratio of 8:1:1 according to a random sampling scheme. As an evaluation measure, we used an accuracy calculation.

To implement the proposed model, we pre-trained GloVe [19] by using 20 GB of Korean news articles. Then, we used the GloVe as the word embedding vectors in Equation (2). The vocabulary size of the GloVe was 210,867. We initialized the character-level embedding vectors with random values. The vocabulary size of the character-level embedding vectors was 132. We set the sizes of embedding vectors (i.e., e_i, $\hat{e}_i$, and e_i^{char}) to 100, 100, and 300, respectively. We set the sizes of the class embedding matrix (i.e., V_C) to 100×411. We set the hidden size, the attention head size, and the number of layers in the transformer's encoder to 500, 12, and 6, respectively. We set the hidden size of the GRU neural network to 100. The model optimization was done with Adam [20] at a learning rate of 0.00005, and

the learning rate was halved if the performance of the validation set did not improve. The dropout rate was set to 0.2, and the mini-batch size was set to 64 sentences, respectively. We empirically set the learning rate, the dropout rate, and the mini-batch size in order to obtain the best performances.

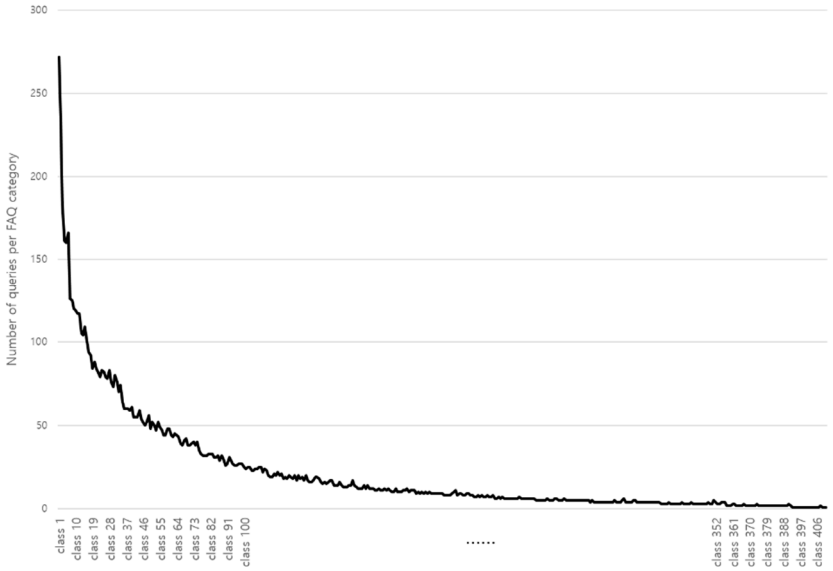

Figure 5. Distribution of the number of queries over the full FAQ categories.

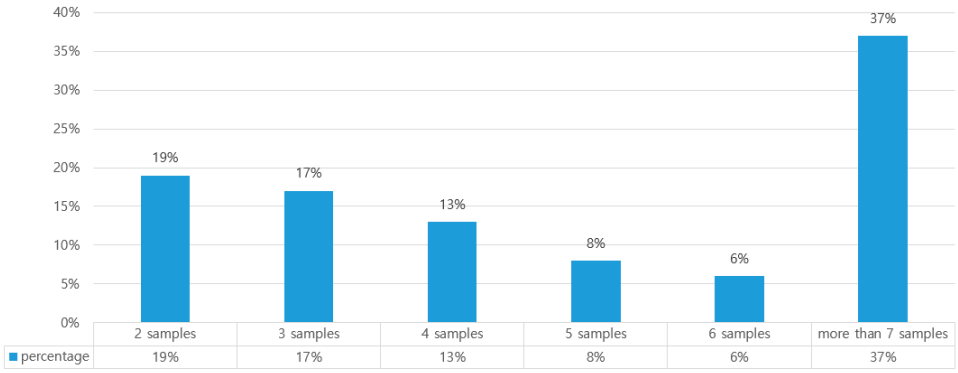

Figure 6. Distribution of FAQ categories according to the number of queries.

4.2. Experimental Results

The first experiment was to evaluate the effectiveness of the proposed embedding methods by comparing the performance changes, as shown in Table 2.

Table 2. Performance changes according to the use of embedding methods.

Model	Accuracy
WordEmbed (baseline)	0.705
WordEmbed + CharEmbed	0.756
WordEmbed + Char & TunedEmbed	0.784
WordEmbed + Char & TunedEmbed + ClassEmbed	0.810

In Table 2, the baseline model (WordEmbed) uses fixed GloVe embeddings as input vectors. CharEmbed, TunedEmbed, and ClassEmbed refer to the character-level word embeddings, the fine-tuned word embeddings, and the class embeddings that are proposed in this paper, respectively. As shown in Table 2, the proposed embedding methods contributed to increasing the performance of FAQ classification.

The second experiment was to compare the proposed model with the previous models, as shown in Table 3.

Table 3. Performance comparison.

Model	Accuracy
CNN	0.638
OKAPI	0.705
BERT-Multilingual	0.779
Proposed Model	0.810

In Table 3, CNN is the sentence classification model based on a CNN [2] in which pretrained word vectors are converted into feature maps by convolution operations based on multiple filters. OKAPI is the Okapi BM25 retrieval model [21] which is a state-of-the-art ranking function used in document retrieval. BERT-Multilingual is a multilingual version of BERT [15] that is pretrained using a large multilingual text corpus, including Korean. In our experiments, BERT-Multilingual was fine-tuned for 15 epochs by using the FAQ dataset. As shown in Table 3, the proposed model outperformed both the well-known sentence classification model and the keyword search model.

The last experiment was to compare the performance changes of the proposed model according to the size of training data, as shown in Figure 7.

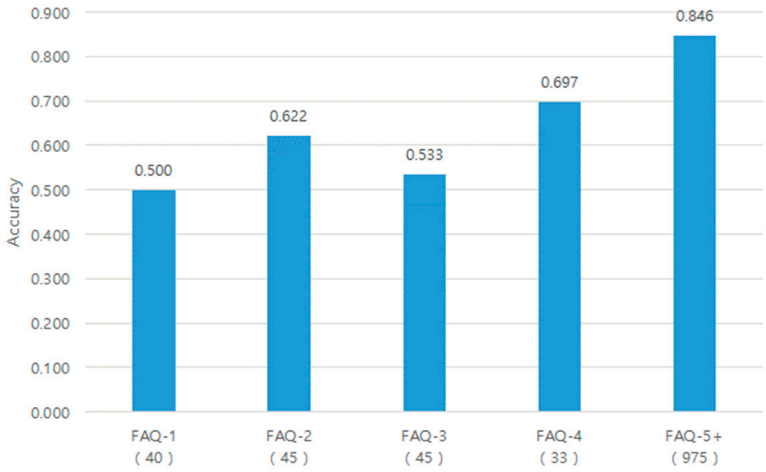

Figure 7. Performance changes according to the size of training data. "5+" means five or more.

In Figure 7, FAQ-*n* indicates FAQ categories in which *n* queries (i.e., *n* training data) are contained. The parenthesized values indicate the number of FAQ categories associated with each FAQ-*n* in the test data. It can be seen from the figure that the proposed model needed at least five training data per FAQ category in order to obtain an accuracy of more than 0.8.

5. Conclusions

We proposed a high-performance sentence classification model based on an encoder-decoder model with an attention mechanism. For bridging the lexical gaps between users' queries and FAQs, we used three types of word embeddings (fixed word embeddings, fine-tuned word embeddings, and character-level word embeddings) as inputs to the transformer's encoder. For supplementing domain knowledge associated with categories, we added class embeddings to the outputs of the RNN decoder. In the experiments with the FAQ dataset, the proposed model outperformed the comparison models. We found that the proposed embedding methods contributed to improving the performance of sentence classification. The proposed model showed low performances in FAQ categories containing a small number of training data. To reduce this problem, we need to adopt pre-trained language models

like BERT and XLNet [22] as encoders. In the future, we will try to combine the proposed model with a chatbot model for assisting online banking customers. Therefore, we will study a method to return a nil category to make the chatbot model generate proper responses when users' queries are not associated with any one of the predefined FAQ categories.

Author Contributions: Conceptualization, H.K. and Y.J.; methodology, H.K. and Y.J.; software, Y.J.; validation Y.J.; formal analysis, Y.J.; investigation, Y.J.; data curation, Y.J.; writing—original draft preparation, Y.J.; writing—review and editing, H.K.; visualization, H.K.; supervision, H.K.; project administration, H.K.; funding acquisition, H.K.

Funding: This work was supported by Shinhan Bank. It was also supported by the National Research Foundation of Korea (NRF) and grant funded by the Korea government (MSIP) (No.2016R1A2B4007732).

Acknowledgments: The authors would like to thank the members of the NLP laboratory in Kangwon National University for their technical support. We would specially like to thank Sebin Kim, Dongho Kang, and Hyunki Jang at Shinhan Bank for their financial and technical support.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A Convolutional Neural Network for Modelling Sentences. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Baltimore, MD, USA, 22–27 June 2014; pp. 655–665.
2. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
3. Yin, W.; Schütze, H. Multichannel Variable-size Convolution for Sentence Classification. In Proceedings of the Nineteenth Conference on Computational Natural Language Learning, Beijing, China, 30–31 July 2015; pp. 204–214.
4. Yu, L.; Hermann, K.; Blunsom, K.; Pulman, S. Deep Learning for Answer Sentence Selection. In Proceedings of the NIPS Deep Learning and Representation Learning Workshop, Montreal, QC, Canada, 12 December 2014.
5. Zhang, Y.; Roller, S.; Byron, C. Mgnc-cnn: A Simple Approach to Exploiting Multiple Word Embeddings for Sentence Classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1522–1527.
6. Mikolov, T.; Sutskever, H.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems (Volume 2), Lake Tahoe, NE, USA, 5–10 December 2013; pp. 3111–3119.
7. Hsu, S.; Moon, C.; Jones, P.; Nagiza, F. A Hybrid CNN-RNN Alignment Model for Phrase-Aware Sentence Classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Valencia, Spain, 3–7 April 2017; pp. 443–449.
8. Kim, Y.; Jernite, Y.; Sontag, D.; Rush, A. Character-Aware Neural Language Models. In Proceedings of the AAAI 2016, Phoenix, AZ, USA, 12–17 February 2016; pp. 2741–2749.
9. Lee, J.; Cho, K.; Hofmann, T. Fully Character-Level Neural Machine Translation without Explicit Segmentation. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 365–378. [CrossRef]
10. Ling, W.; Trancoso, I.; Dyer, C.; Black, A. Character-based Neural Machine Translation. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; pp. 357–361.
11. Park, S.; Byun, J.; Beak, S.; Cho, Y.; Oh, A. Subword-level Word Vector Representations for Korean. In Proceedings of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 2429–2438.
12. Seo, D.; Chung, Y.; Kang, I. A typing error-robust Korean POS tagging using Hangul Jamo combination-based embedding. In Proceedings of the of the HCLT, Daegu, Korea, 13–14 October 2017; pp. 203–208. (In Korean).
13. Zhang, X.; Zhao, J.; LeCun, Y. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems 28(NIPS 2015)*; Courant Institute of Mathematical Sciences: New York, NY, USA, 2015.

14. Kim, Y.; Kim, D.; Kumar, A.; Sarikaya, R. Efficient Large-Scale Domain Classification with Personalized Attention. In Proceedings of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 2214–2224.
15. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805v2.
16. Vaswabu, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
17. Chung, J.; Gulcehre, C.; Cho, K.; Bngio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555v1.
18. Luong, M.; Pham, H.; Manning, C. Effective Approaches to Attention-based Neural Machine Translation. In Proceedings of the Association for Computational Linguistics, Lisbon, Portugal, 17–21 September 2015; pp. 1412–1421.
19. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. In Proceedings of the Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
20. Kingma, D.; Ba, L. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980v9.
21. Robertson, S.; Walker, S.; Jones, S.; Beaulieu, M.; Gatford, M. Okapi at TREC-3. In Proceedings of the TREC-3, Gaithersburg, MD, USA, 2–4 November 1994; pp. 109–126.
22. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; Le, Q.V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv* **2019**, arXiv:1906.08237.

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).