KICS
The Korean Institute of
Communications and
Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express xxx (xxxx) xxx

www.elsevier.com/locate/icte

TABAS: Text augmentation based on attention score for text classification model

Yeong Jae Yu[a], Seung Joo Yoon[b], So Young Jun[b], Jong Woo Kim[a,*]

[a] *School of Business, Hanyang University, Seoul, Republic of Korea*
[b] *Department of Business Informatics, Hanyang University, Seoul, Republic of Korea*

Received 18 August 2021; received in revised form 12 October 2021; accepted 3 November 2021
Available online xxxx

Abstract

To improve the performance of text classification, we propose text augmentation based on attention score (TABAS). We recognized that a criterion for selecting a replacement word rather than a random selection was necessary. Therefore, TABAS utilizes attention scores for text modification, processing only words with the same entity and part-of-speech tags to consider informational aspects. To verify this approach, we used two benchmark tasks. As a result, TABAS can significantly improve performance, both recurrent and convolutional neural networks. Furthermore, we confirm that it provides a practical way to develop deep-learning models by saving costs on making additional datasets.

Keywords: Attention mechanism; Data augmentation; Natural language processing; Text classification

1. Introduction

Text classification, such as sentiment analysis and topic classification, is one of the main tasks in natural language processing. Since it can be used in many industrial areas, many researchers have recently proposed methodologies to complement and improve deep learning models [1]. It is necessary to make these models robust by reducing model overfitting. Robust models generally require the support of a large quantity of high-quality data for the training process. However, it is not easy to obtain such highly suitable datasets for a specific task in practice. Further, it is challenging to get enough quality data in supervised learning since a labeled dataset is required for model training. Thus, it is costly and time-consuming to acquire a proper dataset for model development. Data augmentation can help to build effective models by solving these problems.

In particular, the techniques are mainly used for image augmentation. This is because an image can be altered easier than text, by flipping, rotating, etc. [2–4]. However, it is difficult to directly apply these methods to text data because the meaning may change when a letter or word is erased, or the position is changed. Therefore, most text data augmentation techniques use several methods that preserve the meaning of the data. Related studies have shown that a model's performance is improved by selecting words randomly and then modifying them using a thesaurus and WordNet [5–7]. However, such an augmentation method has a probability to modify words that have less influence on predicting the target label. If it happens, the results are like learning the original data twice from the model's perspective. Therefore, we identify that a criterion is necessary, rather than randomly picking out words to be modified when augmenting data.

We take advantage of the fact that each word has a different influence on label prediction. For the augmented data to be differentiated from the original data and have meaning for training as new data, informative words for predicting the label should be changed. To realistically modify sentences by selecting an informative word for label prediction, we use the attention mechanism and word dictionary with part-of-speech (POS) and named-entity recognition (NER) tags. TABAS amplifies data by selecting only words with more influence on label prediction based on the attention score. Then, to preserve the whole meaning, it changes the words using the dictionary made of the tokens with named entities and the part-of-speech tags. This approach differs from previous

[*] Correspondence to: School of Business, Hanyang University, 222 Wangsimni-ro, Seongdong-gu, Seoul 04763, Republic of Korea.

E-mail addresses: uyeongjae@hanyang.ac.kr (Y.J. Yu), thingjoo@hanyang.ac.kr (S.J. Yoon), thdud1282@hanyang.ac.kr (S.Y. Jun), kjw@hanyang.ac.kr (J.W. Kim).

Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2021.11.002
2405-9595/

Please cite this article as: Y.J. Yu, S.J. Yoon, S.Y. Jun et al., TABAS: Text augmentation based on attention score for text classification model, ICT Express (2021), https://doi.org/10.1016/j.icte.2021.11.002.

studies in that the attention mechanism and taggers are utilized to exquisitely consider the informative effect of words on the target label. We empirically confirmed that TABAS effectively improves the performance of text classification models by data augmentation. Our proposed technique is evaluated with two benchmark classification tasks. In addition to using the entire dataset, we verify how much enhancement is achieved by assuming insufficient data. For the experiments, we apply different types of benchmark datasets to two deep learning models: Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN).

2. Literature review

2.1. Attention mechanism

Attention mechanism is a technique introduced when performing machine translation using sequence-to-sequence (Seq2Seq) model [8,9]. Seq2Seq is a model using LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Unit) derived from RNN (Recurrent Neural Network) model for encoder and decoder. However, there is a problem in that information is lost when a sentence becomes long in the process of implying information of a context vector for a sentence through nodes of the RNN. Attention mechanism is utilized to overcome the limitation. The basic principle of attention is to obtain a context vector that reflects the weight by paying attention to the relationship between all words for a sentence from the encoder and each word coming in at every timestep from the decoder and then connect the existing context vector. Instead of giving attention to all the input sentences at the same rate, it focuses on the input sequence most similar to the predicted feature at that point in time. It showed improved performance in the field of machine translation by preventing a sharp drop in accuracy even if the input sequence is long [10,11]. Attention mechanism is utilized not only in machine translation but also in tasks such as document classification [12], image captioning [13], syntax analysis [14], and question answering [15].

2.2. Text augmentation

Data Augmentation techniques are frequently applied to improve the performance of deep learning models. Earlier studies presented techniques mainly targeting image data [2–4]. This is because the methods can quite effectively transform a given image by flipping, cropping, rotating, etc. [16]. However, these techniques are not suitable for application to text data because the original meaning can change depending on the existence and position of a word. Despite the intrinsic limitations, recent studies have suggested techniques that can preserve text characteristics. The research can be divided into two approaches: Modification and Generation.

Wei and Zou (2019) allow for easy data augmentation (EDA) with four operations, namely synonym replacement (SR), random insertion (RI), random swap (RS), and random deletion (RD). Xie et al. (2019) suggested a method for calculating the term frequency-inverse document frequency (TF-IDF) value for each token and then altering the words based on the TF-IDF values [17]. Kobayashi (2018) presented the technique of contextual augmentation or replacing words with a paradigmatic relation based on a synonym dictionary [18]. Kumar et al. (2020) proposed a technique using bi-directional encoder representations from transformers (BERT) [19]. This form of modification is resolved by randomly masking words and then altering the words predicted by the language model.

Sennrich et al. (2016) first proposed a back-translation technique [20], and Edunov et al. (2020) analyzed this technique in detail, aiming to minimize data loss [21]. This technique is processed by translating the target language corpus to a source language, creating a parallel corpus with the synthetic machine translation, and thus increasing the number of training data entries. Anaby-Tavor et al. (2020) proposed the language-model-based data augmentation (LAMBADA) technique [22]. This technique is executed by fine-tuning the generative pre-training-2 (GPT-2) [23] model based on a dataset and then labeling through the classifier the sentences, which are generated from the original dataset.

We propose the TABAS framework to overcome the dependent deficiencies as a word modification-based method. Related studies do not have a specific criterion for selecting the words to be modified. However, this approach is not efficient because not all the words have the same influence on the prediction of labels. Furthermore, it is not easy to significantly enhance the capacity of the model when the less informative token is predominately altered. In other words, arbitrarily augmented data is likely to be ineffective as training data. Therefore, we propose the TABAS method to efficiently change only the words with explanatory power for the target label.

3. Methodology

We present the text augmentation based on the attention score (TABAS) model as a novel technique to improve the performance of text classifiers (see Fig. 1). It is a new method of text data augmentation that combines an attention mechanism and the existing taggers of NER and POS. In this methodology, the tokens are modified using the measured attention score rather than randomly changing words.

The TABAS framework can be divided into two steps. The first step is the model preparation for the entire dataset. We train an attention score model through the dataset and then build a word dictionary, which includes tuples of words with a tag and label $(word_n, tag_m, label_i)$. In the second step, for each sentence in the dataset, we tokenize the sentence and decide whether to modify each token, using the attention model and the word dictionary, which are both generated in Step I.

3.1. Model preparation

The exemplar process is shown in Fig. 2, from the original sentence to the new sentence. Before the model preparation,

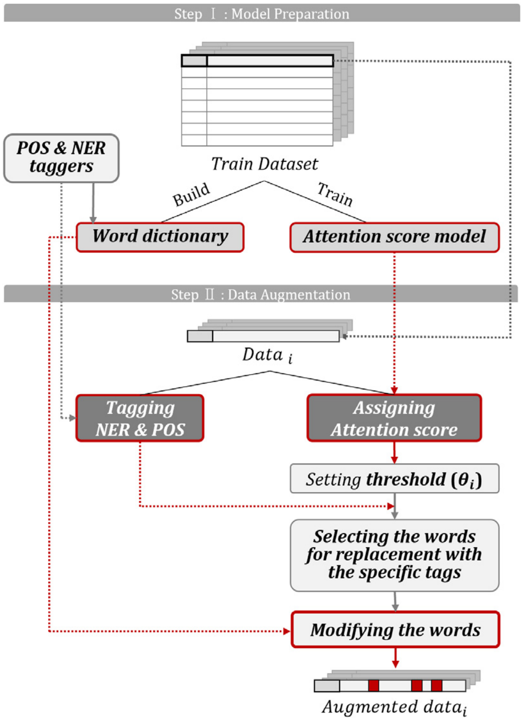


Fig. 1. TABAS framework using attention mechanism.

pre-processing is necessary, including the removal of special characters in the dataset. Attention score tagging proceeds through text classification model of a bi-directional recurrent neural network structure using attention mechanism for words in input sequence. In addition, part-of-speech and entity name tagging is performed through POS tagger[1] and NER[2] built into the Python NLTK library.

Train Attention score model. We train the attention score model. A bi-directional GRU (Gated Recurrent Unit) [24] is utilized to extract the attention scores. It allows influential tokens to be given higher weight to their target label prediction. Fig. 3 shows the architecture of the model for classification. Among the various attention mechanisms, this study utilized the adaptive attention [9]. This is a mechanism developed by Bahdanau et al. (2014), which is known to be designed to be slightly more complex than the dot-product attention [25] proposed by Luong et al. (2015).

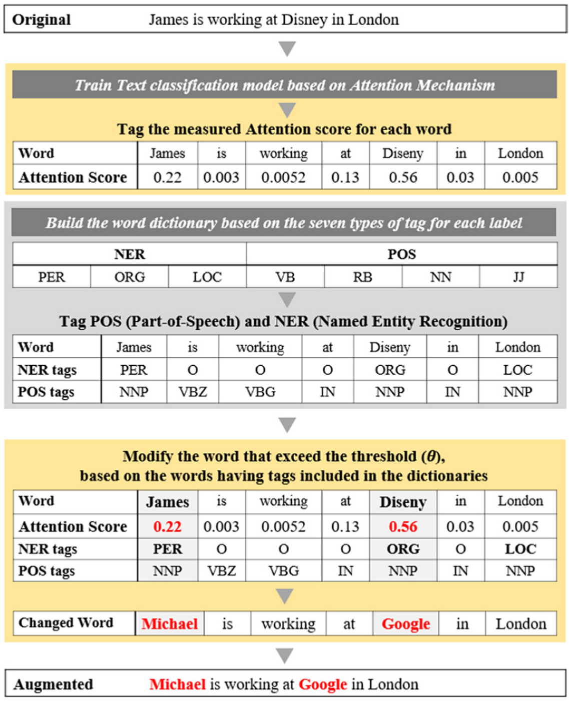
Consider an ith input sequence of length T on the entire training dataset. Through the feed-forward neural network the *attention vector* for the jth input token is:

$$e_{i,j} = score\left(h_{i,T}, h_{i,j}\right) = W_a^{\top} \tanh\left(W_b h_{i,T} + W_c h_{i,j}\right), \quad (1)$$

where $h_{i,j}$ is the hidden state of jth token input at that point as the output of the GRU-based encoder. W_a, W_b, W_c are the weight matrix to be trained.

The corresponding *attention weight* is:

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^{T} \exp(e_{i,k})}, \quad (2)$$



Original	James is working at Disney in London

Train Text classification model based on Attention Mechanism

Tag the measured Attention score for each word

Word	James	is	working	at	Diseny	in	London
Attention Score	0.22	0.003	0.0052	0.13	0.56	0.03	0.005

Build the word dictionary based on the seven types of tag for each label

NER			POS			
PER	ORG	LOC	VB	RB	NN	JJ

Tag POS (Part-of-Speech) and NER (Named Entity Recognition)

Word	James	is	working	at	Diseny	in	London
NER tags	PER	O	O	O	ORG	O	LOC
POS tags	NNP	VBZ	VBG	IN	NNP	IN	NNP

Modify the word that exceed the threshold (θ), based on the words having tags included in the dictionaries

Word	**James**	is	working	at	**Diseny**	in	London
Attention Score	0.22	0.003	0.0052	0.13	0.56	0.03	0.005
NER tags	PER	O	O	O	ORG	O	LOC
POS tags	NNP	VBZ	VBG	IN	NNP	IN	NNP

Changed Word	Michael	is	working	at	Google	in	London

Augmented	Michael is working at Google in London

Fig. 2. Exemplar process based on TABAS method.

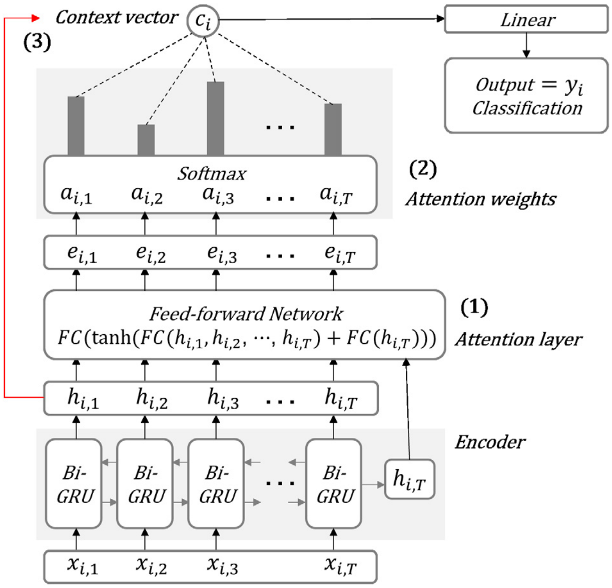

Context vector
c_i
(3)
Linear
Output = y_i Classification
Softmax
$a_{i,1}$ $a_{i,2}$ $a_{i,3}$ ... $a_{i,T}$
(2)
Attention weights
$e_{i,1}$ $e_{i,2}$ $e_{i,3}$ ... $e_{i,T}$
Feed-forward Network
$FC(\tanh(FC(h_{i,1}, h_{i,2}, \cdots, h_{i,T}) + FC(h_{i,T})))$
(1)
Attention layer
$h_{i,1}$ $h_{i,2}$ $h_{i,3}$ ... $h_{i,T}$
Encoder
Bi-GRU
$h_{i,T}$
$x_{i,1}$ $x_{i,2}$ $x_{i,3}$ $x_{i,T}$

Fig. 3. Classification architecture with attention mechanism.

which can be obtained by applying SoftMax function to $e_{i,j}$, and indicated the relationship between the words in the input sequence and the target label.

The *context vector* is represented as:

$$c_i = \sum_{j=1}^{T} \alpha_{i,j} h_{i,j}, \quad (3)$$

which can also be referred to as a weighted sum, the result of multiplying the attention weight and hidden state of each encoder and finally adding them all together. Then, the context

[1] https://nlp.stanford.edu/software/tagger.html.

[2] https://nlp.stanford.edu/software/CRF-NER.html.

vector becomes the input of the softmax and linear layer, which are added to the final decoder output to predict the target label.

Build word dictionary. It is the process of creating a word dictionary for the given dataset. For effective augmentation, we select the tags of NER and POS that seem to have relatively more impact on the target value. The selected tags are the three types of NER, consisting of a person (PER), organization (ORG), and location (LOC), and the four types of POS: verb (VB), adverb (RB), noun (NN) and adjective (JJ). Next, the tagged words are added to the dictionary with the tags and label to which the words belong, and the duplicate words for each tag are removed. The reason labels are put together in the dictionary is because a word may be biased to a specific label and is only used in sentences with that label.

3.2. Detailed process for text augmentation

After the model preparation for the entire dataset, the specific replacement occurs on a word-by-word basis. So, the entire dataset is reconstructed in units of sentences, which in turn is divided into units of words.

Assign attention scores. Through the trained attention model, we assign the score to all tokens according to the importance of the word to the label prediction. Thus, the sum of attention scores of all tokens in a piece of data results in 1, and the gap of scores allows us to identify the difference in importance.

Tag POS and NER. Targeting each isolated word, we tag the NER and POS which are included in the NLTK package in Python. Then, we transform the structure, linking the tags of NER and POS with each word. For example, “James”, “Disney”, and “London” are assigned the NER tags of PER, ORG, and LOC, respectively. And “is”, “working”, “at”, “in” are assigned the POS tags of VBZ, VBG, and IN, respectively.

Modify word for augmentation. To construct relevant alternatives of words, we measure the threshold to decide whether to modify each token. We defined a threshold as a weighted value of the highest attention score tagged to words in each data.

$$\theta_i = w \times \max\left(\alpha_{i,1}, \alpha_{i,2}, \alpha_{i,3}, \ldots, \alpha_{i,T}\right), \tag{4}$$

where w has a constant value for weighting the maximum of attention score. This makes the number of replaced words in each data different. To replace only some words without changing the overall meaning of the sentences, this study sets w as 1/3 because the weight increases the possibility of optimal performance improvement. Wei and Zou (2020) measured the performance gain according to the percentage of words in sentence replaced by each augmentation [5]. As a result, when synonym replacement was performed for the full dataset, changing about 20% of words in a sentence suggested highest performance gain. Although the results differ for each training dataset size, in most cases, stable performance could be guaranteed at a replacement ratio of about 20%. In this study, weights were adjusted for replacement ratio suitable for performance improvement. As a result of applying the weight, about 27% of the TREC and 18% of the IMDb dataset were replaced.

Then, only words with attention scores that exceed the threshold are altered with another word with the same tag and label in the dictionary. For example, in Fig. 2, only “James” and “Disney” are altered because their attention scores exceed the threshold. Also, “James” is changed to “Michael” because they have the same tag, PER. Also, “Disney” is replaced by “Google” because they have the same tag, ORG. By not replacing words that are relatively insignificant on a score basis, we change only words that seem to influence the prediction of the target label. In other words, it is possible to avoid the generation of low-value data for effective learning.

4. Experiments

4.1. Experimental setups

The datasets. We utilize two text classification datasets: (1) TREC (Text REtrieval Conference) is a question-type dataset with six labels. (2) IMDb (Internet Movie Database) is a movie review dataset for binary sentiment analysis with two labels: positive and negative. Each dataset is divided into training data (80%) and testing data (20%): the first one for text classification and applying data augmentation techniques, and the last one to evaluate the enhancement of the model by the augmentation. All training datasets were randomly extracted at the rate of 25%, 50%, 75%, and 100%, to measure the impact of the difference on the improvement of performance according to the size of the training dataset. The model can also gauge the conditions where there is insufficient data.

Baseline for comparison. To evaluate the performance of the proposed model, we utilize the data augmentation methods, which do not consider the influence of the token’s target variable prediction. These methods randomly replace the word based on POS, NER, or with an approach that combines the two methods. First, the POS tag-based approach deals with the word dictionary. If the POS of each word belongs to the selected tags in the dictionary, it is randomly replaced with the words with the same POS. Next, the word replacement based on the NER, and POS tag also reflects the entity name and the POS of all tokens. When the NER or POS of each word is one of the seven tags, it is also replaced by any word in the dictionary.

Text Classification Models. To verify the validity of our proposed approach, we conduct experiments with two models for the text classification: RNN [24] and Text-CNN [26].

The embedding of the text classification with the RNN is 128 dimensions, the hidden state is a single layer with 256 dimensions, with the structure of a fully connected neural network. Next, in the model of CNN, the embedding is 128 dimensions, and the size of the filter is 3 × 128. In addition, the number of filters is 100 with a structure of a fully connected neural network, including max pooling. In the experiments with both models, the learning rate is 0.001, the loss function is a cross-entropy function, and we use Adam as the optimizer.

In the case of IMDb, the maximum length of the data is 500, and the sigmoid function is used for the activation

Table 1
Results of average performance (%) with TREC dataset.

Model		Training size			
		25%	50%	75%	100%
RNN	+(None)	74.46	79.08	83.51	84.34
	+POS	76.34	79.82	84.00	85.32
	+NER	76.40	81.70	84.86	86.76
	+NER&POS	76.04	79.90	84.20	85.62
	+TABAS	**77.90**	**82.96**	**85.42**	**87.60**
CNN	+(None)	77.39	81.97	84.22	86.13
	+POS	79.18	83.10	84.37	86.46
	+NER	80.98	83.84	86.28	87.74
	+NER&POS	79.42	83.04	84.56	87.27
	+TABAS	**81.54**	**84.53**	**86.74**	**88.26**

Table 2
Results of average performance (%) with IMDb dataset.

Model		Training size			
		25%	50%	75%	100%
RNN	+(None)	71.53	80.25	83.48	85.22
	+POS	76.70	82.32	84.11	85.36
	+NER	77.22	82.23	84.21	85.32
	+NER&POS	77.31	82.37	84.25	85.34
	+TABAS	**77.96**	**82.39**	**84.52**	**85.43**
CNN	+(None)	80.66	83.44	84.72	85.55
	+POS	80.75	83.68	85.16	85.85
	+NER	81.19	83.73	85.18	85.70
	+NER&POS	80.76	83.71	85.26	85.72
	+TABAS	**81.79**	**84.45**	**86.08**	**86.13**

Table 3
Results of average performance (%) with BERT.

Dataset	Training size	BERT				
		+(None)	+POS	+NER	+NER&POS	+TABAS
TREC	25%	92.62	93.83	**95.08**	94.49	94.49
	50%	94.88	96.29	**96.95**	95.98	96.56
	75%	96.95	96.95	96.95	96.56	**97.93**
	100%	**97.66**	95.98	96.88	96.17	**97.66**
IMDb	25%	86.61	85.63	**86.84**	85.43	85.77
	50%	87.68	86.34	**88.10**	86.19	87.00
	75%	88.89	86.46	**88.92**	86.74	87.31
	100%	**88.70**	84.67	88.19	86.13	86.08

function in both of models. Meanwhile, for the TREC dataset, the maximum length of a sentence is 200 and the SoftMax activation function is used. To evaluate the performance of models, the criterion is based on accuracy, which is measured as the average of experimental results after ten-fold cross validation.

4.2. Experimental results

The Results of TREC. We ran two models with baseline methods or TABAS. First, using the TREC dataset ($N = 5452$), we measure the average performances (%), shown in Table 1. As a result of comparing the performance, the text classification with TABAS showed the highest performance. In addition, the case of extracting 25% ($N = 1363$) showed the greatest improvement in performance, producing the highest average 3.8% for both models. Of note, the average improvement for all cases was 3.12% using the RNN and 2.84% using the CNN.

The Results of IMDb. With IMDb dataset ($N = 25,000$), we calculate the average performances (%) of text classification (see Table 2). Comparing the results, the proposed method, TABAS, showed the highest performance in all cases. The result when extracting 25% ($N = 6250$) of the training data also showed the greatest improvement, presenting the highest average 3.8% for both of models. Especially, the average improvement when using TABAS for all cases was 2.46% for the RNN and 1.02% for the CNN.

4.3. Discussion

Further research with BERT. We conducted additional experiments based on the pre-trained BERT (base) model [27] with the same dataset. The hidden state has 768 dimensions for representation. Training is proceeded with five epochs using Adam optimizer of 5e-5 learning rate. The result is shown in Table 3. Although the impact of data augmentation is negligible when using BERT, the method using TABAS or NER had more influence on performance. BERT is a pre-trained language model with a myriad of corpus, so the performance gained by data augmentation was lower than analysis using other deep learning models.

Meanwhile, the accuracy is different depending on the dataset. In the case of TREC, since the original dataset is small, data augmentation using TABAS or NER is effective. On the other hand, although data augmentation affects the performance in IMDb, the gain is not as high as in the TREC dataset. However, better accuracy was presented when the data augmentation technique was added, except when the training size was 100%. It seems that a data augmentation method is necessary when there is data scarcity. Furthermore, there is the problem that BERT takes more time and cost even in fine-tuning than other deep learning models. So, we propose TABAS as a method that can effectively improve performance under the constraints of training time and computing resources.

Discussion. We confirmed that TABAS could be an effective method for text augmentation for classification. The improvement differs by the size of the training set and the taggers. Except for the TABAS, NER is more successful than POS tagging in most situations through an experiment using 25%, 50%, 75% of the training datasets. This seems to result from the specific tagging by NER, as words to be tagged as nouns simply by the POS tagger can be segmented into people, places, and organizations as named entities. Meanwhile, the highest performance growth occurred when extracting 25% of the training data out of all cases. The text augmentation in the smallest dataset has a more positive effect against the baseline than it did for larger datasets. We expect that future studies will be encouraged by the implications that TABAS established logical criteria for selecting words to replace.

5. Conclusion

We proposed the TABAS technique with the attention mechanism and examined the efficiency for text augmentation. TABAS employs an attention score and two different tags: POS and NER. With the specific POS and NER tags, the model builds a word dictionary. And the model converts individual words that exceed a threshold into another word with the same label and tag from the dictionary. We found that the proposed method can be an effective strategy in augmenting qualitative text data. It outperformed the other methods with deep learning models: RNN and Text-CNN. Our contributions are as follows: This method transforms a dataset by replacing words based on an attention score model and the POS and NER taggers. TABAS has shown that text data can be augmented regardless of the type of dataset. Consequently, it is practical to utilize this method to efficiently boost the performance of text classification models.

A few issues remain to be explored. First, we did not deal with the various replacement ratio of the input tokens due to time and effort limitations. Although related studies have provided a rationale for specific weights, it is difficult to generalize to all data augmentation studies. If we consider multiple weights for replacement in future studies, better performance can be expected. In addition, we used only the basic NER or POS-based augmentation as the baseline methods. To verify the quality of the word dictionary more reliably, it is necessary to compare the latest data augmentation techniques.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification, in: Proceedings of the National Conference on Artificial Intelligence, 2015.

[2] J. Wang, L. Perez, The effectiveness of data augmentation in image classification using deep learning. arXiv, 2017.

[3] M.D. Bloice, C. Stocker, A. Holzinger, Augmentor: An image augmentation library for machine learning arXiv, 2017, http://dx.doi.org/10.21105/joss.00432.

[4] A. Mikołajczyk, M. Grochowski, Data augmentation for improving deep learning in image classification problem, in: 2018 International Interdisciplinary PhD Workshop, IIPhDW 2018, 2018, http://dx.doi.org/10.1109/IIPHDW.2018.8388338.

[5] J. Wei, K. Zou, EDA: Easy data augmentation techniques for boosting performance on text classification tasks, in: EMNLP-IJCNLP 2019-2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference, 2020, http://dx.doi.org/10.18653/v1/d19-1670.

[6] V. Marivate, T. Sefara, Improving short text classification through global augmentation methods, in: International Cross-Domain Conference for Machine Learning and Knowledge Extraction, Springer, Cham, 2020, pp. 385–399.

[7] D. Zhang, T. Li, H. Zhang, B. Yin, On data augmentation for extreme multi-label classification, 2020, arXiv preprint arXiv:2009.10778.

[8] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.

[9] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, 2014, arXiv preprint arXiv:1409.0473.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, ... I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[11] Y. Wu, M. Schuster, Z. Chen, Q.V. Le, M. Norouzi, W. Macherey, . J. Dean, Google's neural machine translation system: Bridging the gap between human and machine translation, 2016, arXiv preprint arXiv:1609.08144.

[12] X. Sun, W. Lu, Understanding attention for text classification, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3418–3428.

[13] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, . Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: In International Conference on Machine Learning, PMLR, 2015, pp. 2048–2057.

[14] O. Vinyals, Ł. Kaiser, T. Koo, S. Petrov, I. Sutskever, G. Hinton, Grammar as a foreign language, Adv. Neural Inf. Process. Syst. 28 (2015) 2773–2781.

[15] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, . R. Socher, Ask me anything: Dynamic memory networks for natural language processing, in: In International Conference on Machine Learning, PMLR, 2016, pp. 1378–1387.

[16] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data (2019) http://dx.doi.org/10.1186/s40537-019-0197-0.

[17] Q. Xie, Z. Dai, E. Hovy, M.T. Luong, Q.V. Le, Unsupervised data augmentation for consistency training. arXiv, 2019.

[18] S. Kobayashi, Contextual augmentation: Data augmentation bywords with paradigmatic relations, in: NAACL HLT 2018-2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 2018, http://dx.doi.org/10.18653/v1/n18-2072.

[19] V. Kumar, A. Choudhary, E. Cho, Data augmentation using pre-trained transformer models. arXiv, 2020.

[20] R. Sennrich, B. Haddow, A. Birch, Improving neural machine translation models with monolingual data, in: 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, 2016, http://dx.doi.org/10.18653/v1/p16-1009, Long Papers.

[21] S. Edunov, M. Ott, M. Auli, D. Grangier, Understanding back-translation at scale, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018, 2020, http://dx.doi.org/10.18653/v1/d18-1045.

[22] A. Anaby-Tavor, B. Carmeli, E. Goldbraich, A. Kantor, G. Kour, S. Shlomov, . N. Zwerdling, Do not have enough data? Deep learning to the rescue! arXiv, 2019, http://dx.doi.org/10.1609/aaai.v34i05.6233.

[23] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, OpenAI Blog 1 (8) (2019) 9.

[24] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: EMNLP 2014-2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 2014, http://dx.doi.org/10.3115/v1/d14-1179.

[25] M.T. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation, 2015, arXiv preprint arXiv:1508.04025.

[26] Y. Kim, Convolutional neural networks for sentence classification, in: EMNLP 2014-2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 2014, http://dx.doi.org/10.3115/v1/d14-1181.

[27] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: NAACL HLT 2019-2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 2019.