Application of Mix-Up Method in Document Classification Task Using BERT

Naoki Kikuta and Hiroyuki Shinnou
Faculty of Engineering, Ibaraki University, Japan
{21nm720n, hiroyuki.shinnou.0828}@vc.ibaraki.ac.jp

Abstract

The mix-up method (Zhang et al., 2017), one of the methods for data augmentation, is known to be easy to implement and highly effective. Although the mix-up method is intended for image identification, it can also be applied to natural language processing. In this paper, we attempt to apply the mix-up method to a document classification task using bidirectional encoder representations from transformers (BERT) (Devlin et al., 2018). Since BERT allows for two-sentence input, we concatenated word sequences from two documents with different labels and used the multi-class output as the supervised data with a one-hot vector. In an experiment using the livedoor news corpus, which is Japanese, we compared the accuracy of document classification using two methods for selecting documents to be concatenated with that of ordinary document classification. As a result, we found that the proposed method is better than the normal classification when the documents with labels shortages are mixed preferentially. This indicates that how to choose documents for mix-up has a significant impact on the results.

1 Introduction

The high cost of constructing training data is always a problem when solving natural language processing (NLP) tasks using machine learning approaches. Several attempts have been made to solve this problem. One of the most recent methods for constructing training data is data augmentation (Shorten and Khoshgoftaar, 2019). Data augmentation methods can be divided into two types: processing and generation. For image identification, even if an image in the training data is flipped or cropped, the image label does not change. This means that the training data can be increased by adding such processed images to the training data. Alternatively, the method for generating artificial data using generative adversarial network can be considered as a type of data augmentation. The mix-up method is one of the methods for generating data augmentation. It is highly effective and can easily be implemented. Although the mix-up method is used for image identification, it can also be useed for NLP.

2 Related topic

2.1 Bidirectional encoder representations from transformers (BERT)

BERT is a high-performance, pre-trained model that has been widely used since its creation by Google (2018) (Devlin et al., 2018). It can be used for classification, word prediction and context determination. In this study, we improve the accuracy of the BERT-based document classification task using the mix-up method.

2.2 Mix-up method

Mix-up is a data augmentation method in the field of image proposed by Hongyi Zhang (2017) (Zhang et al., 2017). The data augmentation is performed using Equations 1 and 2 for image data and labels respectively.

$$x = \lambda x_i + (1 - \lambda)x_j \tag{1}$$

$$y = \lambda y_i + (1 - \lambda)y_j \tag{2}$$

x is a vector of image data , y is a one-hot vector of labels, and λ is the mixing ratio.

3 Previous studies using mix-up method for NLP

Hongyu Guo (2019) conducted a study using the mix-up method for NLP (Guo et al., 2019). The method is based on Equations 1 and 2, as in the previous section on the image field. For NLP, x is a word or sentence embedding. The following is an example of mixing in the ratio of 6:4.

Proceedings of Recent Advances in Natural Language Processing, pages 679–683
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_077

Vector of document 1

[0.2, −0.3, 0.5, ...]

Vector of document 2

[0.4, 0.1, −0.5, ...]

Vector of mixed document

[0.2, −0.3, 0.5, ...] × 0.6 + [0.4, 0.1, −0.5, ...] × 0.4 = [0.28, 0.22, 0.1, ...]

4 Proposed method

If we adopt the mix-up method of the previous study for BERT, we will have a problem. In the methods of the previous study (Guo et al., 2019), it is necessary to create feature vectors of the documents before learning the neural network (NN) (Figure 2). However, document classification using BERT obtains the feature vectors of the documents during the NN learning process (Figure 1), which is a different order from the methods used in the previous study. If we were to adopt the method of the previous study, the calculation of feature vectors by BERT would be done outside the learning process. Therefore, high classification accuracy cannot be expected.

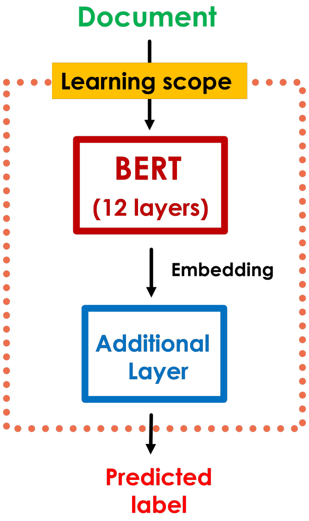

Figure 1: Classification with BERT

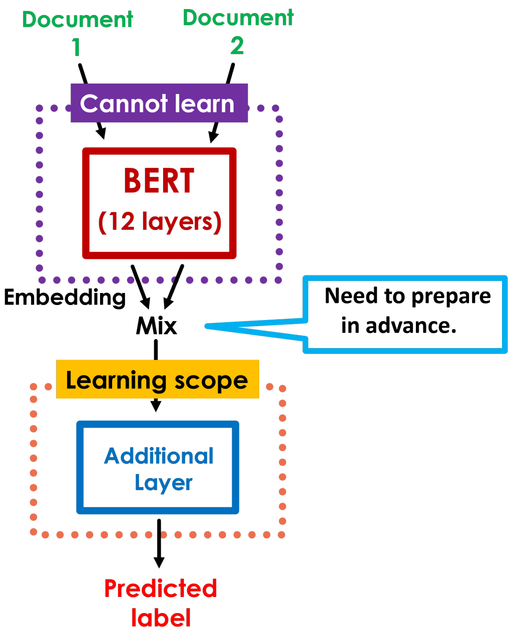

Figure 2: When to use prior research methods in BERT

In this paper, we propose the following method.

4.1 How to mix data

We don't use Equation 1 to mix data. Our method is to concatenate the word sequences of the two documents when they are entered into BERT. With this method, it is possible to learn the BERT part to obtain the feature vector of the document (Figure 3). We present the following examples. This time, we used Japanese document as target and Japanese BERT as model. Compared to English, there is no clear separation between words in Japanese. Therefore, when processing Japanese, it is necessary to divide it by tokenizer into words, characters, and other parts by toke. Then, each devided word is assigned an ID. The ID 2 indicates the beginning of the sentence and is not necessary for the following sequence, so it is excluded. Additionally, since the maximum input length for BERT is 512, we limited the first and second halves of the word sequences' length to 252 each to avoid exceeding this value. If the sentence length exceeded 252, we discarded the remainder.

The first word sequence

[2, 6259, 9, 12396, 14, 3596, 3]

The second word sequence

[2, 11475, 9, 3741, 5, 12098, 75, 3]

Mixed word sentences

[2, 6259, 9, 12396, 14, 3596, 3,

11475, 9, 3741, 5, 12098, 75, 3]

4.2 How to mix labels

First, each label was represented by a one-hot vector consisting of 0 and 1. A vector consisting two 0.5 and seven 0 was created. The mixed labels contain 0.5; it indicates that the two documents are mixed equivalently. We have the following gexamples.

Label 3
[0, 0, 0, 1, 0, 0, 0, 0, 0]

Label 6
[0, 0, 0, 0, 0, 0, 1, 0, 0]

Mixing of labels 3 and 6
[0, 0, 0, 0.5, 0, 0, 0.5, 0, 0]

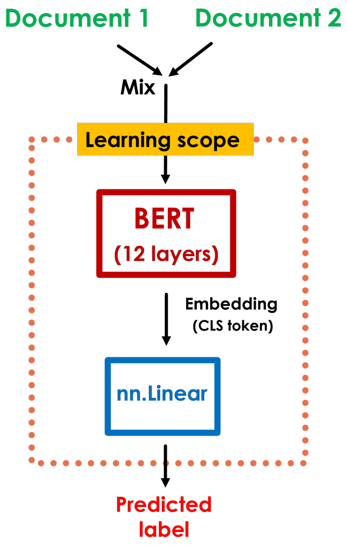

Figure 3: This research method

5 Experiment

5.1 Conditions

5.1.1 Execution environment

The experiment was conducted using the graphics processing unit environment of Google Colaboratory.

5.1.2 BERT model we used

We used bert-base-japanese-whole-word-masking[1], one of the pre-training BERT models for Japanese. It was developed by Inui and Suzuki Lab of Tohoku university.

5.1.3 Corpus we used

We used the livedoor news corpus[2] , which is Japanese to classify documents into the following nine labels.

- label 0 : dokujo-tsushin
- label 1 : IT lifehack
- label 2 : Home Appliances Channel
- label 3 : livedoor HOMME
- label 4 : MOVIE ENTER
- label 5 : Peachy
- label 6 : smax
- label 7 : Sports Watch
- label 8 : topic news

5.2 Experimental procedure

5.2.1 Preparing data

In this experiment, we extracted 6623 articles (texts) from the livedoor news corpus and sorted them as shown in Table 1.

label	train	val	test	sum
0	87	128	566	781
1	87	125	571	783
2	86	111	581	778
3	51	72	335	458
4	87	114	582	783
5	84	106	565	755
6	87	106	590	783
7	90	131	589	810
8	77	107	508	692
sum	736	1000	4887	6623

Table 1: Breakdown of data used

5.2.2 Mix-up of training data

For the training data, we used the mix-up method to expand the data. We used the following two methods for selecting the documents to be mixed.

Selection method 1
The first method is to mix the documents of all labels randomly. We randomly sorted 736 documents

[1] https://github.com/cl-tohoku/bert-japanese

[2] https://www.rondhuit.com/download.html#ldcc

using a random number, and mixed two adjacent documents (and their labels) in order. We generated 735 extended data usind this method. As a result, the number of training data was expanded from 736 to 1471. Additionally, each time the program is run, the selected combination changes.

Selection method 2

The second method of selection is to make up for documents with labels shortages preferentially. In this experiment (Table 1), the training data lacks documents with label 3. Thus, we select label 3 documents to mix. Specifically, we randomly selected one document from 51 label 3 documents. Then, we randomly selected one document from 685 non-label 3 documents, and repeated the procedure of mixing the two documents. For the order of concatenation, the documents with and without label 3 form the first and second halves, respectively. As a result, the number of training data was expanded from 736 to 1501. Similar to the selection method (1), the combination changes every time the program is run.

5.2.3 The classifier we created

The model of the NN used as the classifier is BERT with an additional nn.Linear layer. We input a sequence of words of length 512 or less into BERT and obtain a feature vector of 768 dimensional documents from final layer as output. Then, we input it to nn.Linear layer Pytorch has and obtain the prediction for each label in nine dimensions as output. The detailed settings are shown below.

Loss function

Cross entropy : When training a classification problem with classes, nn.CrossEntropyLoss is usually used in Pytorch. However, this time, the labels are in one-hot representation and cannot be input directly into nn.CrossEntropyLoss. Therefore, we used LogSoftmax Pytorch has to calculate the loss according to the definition of cross-entropy (Equation 3). Since we used batch in this experiment, the loss is Equation 4 which is the batch average of Equation 3.

$$E = -\sum_k t_k \log y_k \qquad (3)$$

$$E = -\frac{1}{B}\sum_b \sum_k t_k \log y_k \qquad (4)$$

Here, t_k is a correct answer, y_k is a predicted value, and B is a batch size.

Optimization function

Stochastic gradient descent (SGD): Using the validation data, we set the learning rate to 0.01, considering both classification accuracy and learning efficiency (Figure 4).

Batch size of the training data

The batch size was set to 10, which was the maximum value possible in Google Colaboratory, the execution environment.

Number of epochs

Considering the range of increase in classification accuracy in the validation data (Figure 4), it was determined that the accuracy reached a convergence value after ten epochs of training.

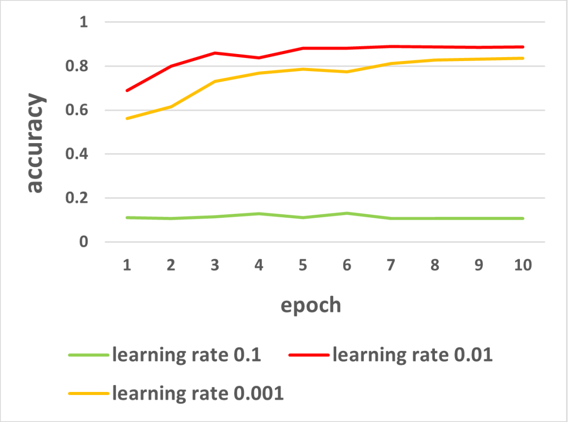

Figure 4: Percentage of correct answers in the validation data

6 Result

For the normal BERT that does't use mixup, the BERT with mix-up of selection method 1, and the the BERT with mix-up of selection method 2, we prepared ten models trained with ten epochs of the training data for each of the methods. Then, the accuracy rate for the test data was calculated. Figure 5 show the box plots comparing each model. The comparison of the mean values of accuracy rate is presented in Table 2.

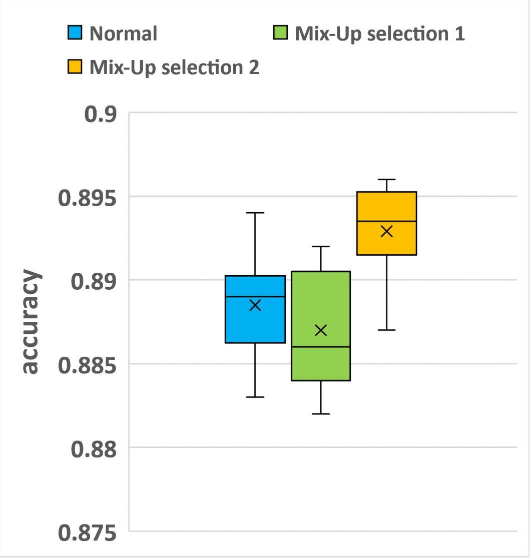

Figure 5: Result

	Accuracy rate (mean)
Normal	0.889
Mix-up selection 1	0.887
Mix-up selection 2	0.893

Table 2: Comparison of mean

In the order of increasing accuracy, there are mix-up for selection method 2, normal BERT, and mix-up of selection method 1.

7 Consideration

Given the result, we obtained that the selection method of the documents to be mixed has a great influence on the accuracy. In this experiment, it is effective to prioritize mixing documents with labels shortages. We would try different methods and conclude. In this experiment, we used two documents with equivalent values (ratio 0.5 : 0.5). However, we think that it is worthwhile to try a method for varying the length of the concatenated words. Mix-up method is easier to implement than other data augmentation methods in NLP, and its accuracy has been improved. It is expected to become a mainstream method in the future.

8 Conclusion

In this paper, we applied the mix-up method to a document classification task using BERT. Since BERT allows for two-sentence input, two documents with different labels were combined and input. Then, the labels were mixed by creating two 0.5 elements in a one-hot vector. In an experiment using the livedoor news corpus, which is Japanese, we found that the proposed method is better than the normal classification when the documents with labels shortages are mixed preferentially. Therefore, it indicates that the accuracy varies depending on the method of selecting documents to be mixed.

Acknowledgment

The research was supported by JSPS KAKENHI Grant Number JP19K12093 and ROIS NII Open Collaborative Research 21FC05.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers forlanguage understanding. *arXiv preprint arXiv:1810.04805*.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*.

Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.