Data Augmentation For Chinese Text Classification Using Back-Translation

Data Augmentation For Chinese Text Classification Using Back-Translation

Jun Ma[1st a, *], Langlang Li[2nd, b]

School of Information Science & Engineering, Lanzhou University, Lanzhou, Gansu Province, China

[a*]junma@lzu.edu.cn, [b]lill19@lzu.edu.cn

Abstract—Text classification is a basic task in natural language processing. When the amount of data is insufficient, the classification accuracy will be greatly affected. We propose to use the back-translation method to expand three Chinese data sets used for text classification, and then train and predict the data sets through deep learning classification model. The results prove that using back-translation to expand the data is particularly helpful on a smaller dataset, it also can reduce the unbalanced distribution of samples and improve the classification performance.

1. INTRODUCTION

1.1 Data Augmentation

Data augmentation [1] is a method of data generation based on visual invariance or semantic invariance, and it is the simplest and direct method to improve model performance. When using smaller data sets, the data augmentation technique can make the model show better generalization ability and performance.

Data augmentation is commonly applied for the field of computer vision [2]. In image processing, rotation, translation, and scaling do not change the meaning of the image. Before inputting data to the model, make small adjustments to the images, such as rotation, translation, scale, random cropping, adding noise and so on. Based on these operations, the previously established network model will learn the transformation form of some samples. For example, without data augmentation, the model usually does not pay too much attention to the position information of the target object during the learning process. It may cause classification error after a simple flip operation. However, one of the most important feature of an ideal model is that the classification result is independent of the position of the target.

In text classification [3] tasks, text data plays a very important role in the classification model. To achieve a good performance for a classifier model, abundant labeled data is often required. However, in many cases, the amount of label data is small and the acquisition cost is high [4], such as product reviews.

At present, the application of data augmentation is minimal in NLP [5]. As a common text data augmentation method, the synonym replacement has been used previously [6-8]. From the papers, the best way to do data augmentation is to use human rephrases of sentences, but this is unrealistic and expensive because we have a large number of samples in our datasets. As a result, for us, replacing words or phrases with their synonyms for data augmentation is a good method [6]. It is the application of word similarity to achieve data augmentation [7]. EDA (Easy Data Augmentation) was proposed by Wei and Zou. For a given sentence in the train dataset, one of the following operations is randomly selected and

performed: Synonym Replacement (SR), Random Insertion (RI), Random Swap (RS), Random Deletion (RD) [9].

In order to expand the text data, we imitate the relevant processing methods of image expansion and propose to use back-translation [10]. Back-translation was used in the 1st place solution for the "Toxic Comment Classification Challenge" on Kaggle. The winner leveraged idea of machine translations to augment both train and test data sets using French, German, and Spanish translations translated back to English. Back-translation is also often used in machine translation [11] to check the accuracy of the translation [12].

TABLE 1BACK-TRANSLATION

	Sentence
S	配置**均衡**，性能**强劲**，**做工精致**，散热良好，配件丰富，其中还有一个 **Dell 原装的**鼠标垫。我同事很满意。
S_1	Balanced configuration, strong performance, exquisite workmanship, and good heat dissipation. Abundant accessories. There is also an original Dell mouse pad. My colleague is very satisfied.
S_2	**平衡**的配置，**强大**的性能，**精湛的工艺**以及良好的散热性，配件丰富， 还有一个**原始的戴尔**鼠标垫。我的同事很满意。

Back-translating the source text is usually converted into the original language after two translations, the original sentence S is translated into other languages (such as English) as S_1, and then back-translated to original language as S_2. After back-translation of the target language sentences, the augmentation corpus is composed of S and S_2. Due to differences in translation software and grammar, the back-translation is in many ways different from the source sentence from words to complete grammatical structure, which can be understood as expanding the number of datasets, back-translation can generate diverse paraphrases while preserving the semantics of the original sentences, these include synonym replacement [6-8], syntactic structure substitution, deletion of irrelevant words [13]. As shown in Table 1, take a Chinese computer review as an example to carry out back-translation through Google Translate.

1.2 Related Work

We need a machine translation service to translate to different languages and return to the original language. Google Translate can accurately translate text into many different languages, so the Google Translate API service is very suitable for our translation tasks. We also recommend the use of Google sheets. The Google Sheets provides a convenient translation code. Take Chinese-English-Chinese as an example: GOOGLE TRANSLATE (GOOGLETRANSLATE (A1, "zh-CN", "en"), "en", " zh-CN "). Moreover, with a simple operation, you can apply this formula over the whole column. Therefore, we can also use Google Sheets to achieve the purpose of back-translation.

For EDA, we use an open source python package nlpcda (NLP Chinese Data Augmentation). Install it conveniently through pip install nlpcda and it can help us easily implement Synonym Replacement, Random Swap, Random Deletion, etc. Through the method of synonym replacement and experiments, the following problems were found in the Chinese synonyms replacement:

1) The number of synonym replacements is set manually and not precise enough. For example, short sentences may require 2 keywords with synonym replacement, and long sentences may require 3 or more. For input sentences of different lengths, the synonym replacement algorithm cannot be adjusted the number of substitutions reasonably.

2) The semantics cannot be maintained well after embedding the replaced synonyms in the sentence. It is possible to cause label changes.

In a word, The replacement of each word must be very rigorous, because it may change the meaning of the entire sentence, thus losing all meaning. And we cannot arbitrarily change the order of each character in text processing, because their order represents semantics.

In this paper, the models of text classification we use are LSTM and CNN. LSTM (Long Short-Term Memory) [14] is a special kind of RNN [15] used for the neural network structure of data closely related

to the sequence. LSTM has mature applications in machine translation, text classification, QA(Question Answering) and other fields. As the distance increases, RNN cannot make full use of historical information. By improving the structure of RNN, adding memory cell and three gating units, the historical information is effectively control. Instead of completely washing away the hidden state of the previous moment like RNN, which enhances its ability to handle long text sequences and also solves the problem of vanishing gradient.

CNN(Convolutional Neural Network) [16] is one of the important algorithms of deep learning. It is widely used in the field of computer vision and natural language processing. The hidden layer of CNN model consists of three layers. Convolutional layer is responsible for extracting features in the image. Pooling layer is used to significantly reduce the number of parameter (dimension reduction). The fully connected layer is the part of a traditional neural network that outputs the desired result.

In this paper, we propose Chinese text data augmentation based on back-translation, which is used to generate corpus to enrich the sentence pattern or lexical features of text data, and improve performance on text classification tasks. We apply this method to three Chinese datasets used for text classification. Through training and predicting the datasets with the deep learning classification model, by comparison, the final prediction results prove that this method is effective.

2. EXPERIMENTS

2.1 Datasets

There are three datasets, including binary sentiment classification (positive and negative) and multi-classification. The first dataset ChnSentiCorp_htl_all (Data1) contains 7766 records about hotel reviews, 5322 positive and 2444 negative. The second, waimai_10k (Data2) includes 12,000 labeled Meituan take-away reviews, including 4000 positive and 8000 negative. The third dataset online_shopping_10_cats (Data3) has more than 60,000 reviews, about 30,000 positive and 30,000 negative, 10 categories and detailed statistics are shown in the Fig.1. These datasets are all from github [17] and have a common feature: the amount of data in different categories are unevenly distributed.

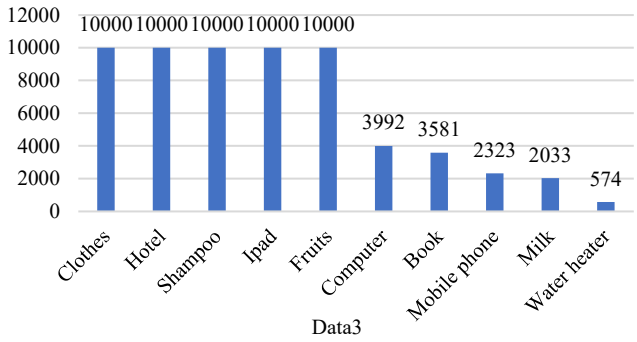

Figure 1. Data3 statistics

2.2 Experiment Steps

2.2.1 Back-translation

Firstly, we analyze the number of each category in original data. For binary classification, if the two categories differ by more than two times in number, the target category for back-translation is the smaller of the two categories. We set α (α represents the percentage of data to be back-translated) for back-translation.

In Data1, compared with the number of Positive, Negative is slightly insufficient, select Negative to translate to English and back translate to Chinese. In order to verify the validity of data augmentation, we select a different α and set α = {0.25, 0.5, 1}. As opposed to Data1, in Data2, the number of Positive was

only half as Negative, so we choose to expand the data of Positive. In Data3, because the number of water heater is very small, we set α=1 and use two language (English and French) for back-translation, that is, each sentence generates two synonymous sentences. The number statistics after data augmentation is shown in the Table 2. After back-translation, combine these data with the original data to form augmentation corpus and are represented with $Data1_{\alpha}$, $Data2_{\alpha}$, $Data3_{\alpha}$, α = {a: 0.25, b: 0.5, c: 1}.

TABLE 2 DATA AUGMENTATION STATISTICS

Data and category	**α**	**Translation language**	**Data size**
Data1_Negative	0.25	English	2444+611
	0.5	English	2444+1222
	1	English	2444+2444
Data2_Positive	0.25	English	4000+1000
	0.5	English	4000+2000
	1	English	4000+4000
Data3_water heater	1	English	574+574
	1	English, French	574+1122

2.2.2 Text data processing

Firstly, delete punctuation except letters, numbers and Chinese. Then use jieba to split the words and remove the Chinese stop words. Next, the Tokenizer function is used to vectorize a text corpus, by turning each text into either a sequence of integers, each integer represents a word index. Before building the model, we analyze the data to obtain the number distribution of reviews' length, and calculate the average length of reviews to set the suitable max_length in the model, which can ensure that a uniform text length is input to the model. The data in the document is arranged in order by category. Before split training dataset and test dataset, we should randomly shuffling it. For all datasets, we split 10% of the testing dataset and keep the remaining 90% as the training dataset.

2.2.3 Build the model

We implemented the model through Keras. LSTM layer with 128 cells, dropout layer with 0.5, the first dense layer with ReLU activation and the second with softmax. We used categorical_crossentropy loss function with adam optimizer. CNN model starting from a embedding layer. The model convolves the text matrix with filters of different lengths through 3 convolutional layers. Then 2 max-pooling layers is used to operate the vectors extracted from each filter. Finally, each filter corresponds to a number, and these filters are spliced together to extract a vector representing the sentence. We set the dropout rate to 0.5 between BatchNormalization and fully connected layers, and it can prevent overfitting.

2.3 Evaluation Metrics

Before the experiments, we first introduce the following evaluation metrics and calculation formulas of text classification. These evaluation metrics we implement through Classification_Report from skit-learn.

1) *TP:* predict positive class as positive class
2) *TN:* predict negative class as negative class
3) *FP:* predict negative class as positive class
4) *FN:* predict positive class as negative class
5) *Support:* the number of test samples for each category

Precision: $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$ (1)

Recall: $\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$ (2)

Accuracy: $\text{Accuracy} = \frac{(\text{TP+TN})}{(\text{TP+TN+FP+FN})}$ (3)

F1 score: $\text{F1 score} = \frac{\text{Precision}\times\text{Recall}\times 2}{\text{Precision+Recall}}$ (4)

3. RESULTS AND COMPARISON

In all tables, Precision, Recall, F1 score, Support represent the metrics of the category which is be back-translated. In Table 3, when use $Data1_c$ to train, we could find that CNN has the highest Precision, Recall, F1 score and Test Accuracy, LSTM also has a better performance. The experimental data of the two models indicate that the classification performance has been greatly improved.

As can be seen from Table 4, in the CNN model, the Test Accuracy showed the best score in $Data2_a$. This shows that the accuracy does not necessarily increase with the increase in the number of data expansion. By the Table 5, Precision decreased significantly, but Recall, F1 score and Test Accuracy increased.

TABLE 3 RESULTS OF DATA1

Training data	LSTM					CNN				
	Precision	*Recall*	*F1score*	*Support*	*Test Accuracy*	*Precision*	*Recall*	*F1score*	*Support*	*Test Accuracy*
Data1	0.78	0.72	0.75	514	0.841	0.82	0.71	0.76	514	0.855
$Data1_a$	0.79	0.82	0.81	584	0.863	0.77	0.88	0.82	584	0.863
$Data1_b$	0.81	**0.89**	0.85	708	0.877	0.80	**0.89**	0.84	708	0.863
$Data1_c$	**0.87**	**0.89**	**0.88**	**998**	**0.883**	**0.91**	0.87	**0.89**	**998**	**0.893**

TABLE 4 RESULTS OF DATA2

Training data	LSTM					CNN				
	Precision	*Recall*	*F1score*	*Support*	*Test Accuracy*	*Precision*	*Recall*	*F1score*	*Support*	*Test Accuracy*
Data2	0.75	0.83	0.79	388	0.856	0.81	0.74	0.77	388	0.862
$Data2_a$	**0.88**	0.81	0.82	497	0.868	**0.86**	0.81	0.83	497	**0.871**
$Data2_b$	0.84	0.85	0.85	576	0.873	0.83	0.83	0.84	576	0.863
$Data2_c$	0.84	**0.88**	**0.86**	**789**	**0.874**	0.84	**0.87**	**0.86**	**789**	0.856

TABLE 5 RESULTS OF DATA3

Training data	LSTM					CNN				
	Precision	*Recall*	*F1score*	*Support*	*Test Accuracy*	*Precision*	*Recall*	*F1score*	*Support*	*Test Accuracy*
Data3	**0.86**	0.35	0.50	123	0.872	0.95	0.34	0.50	123	0.862
$Data3_{c,\ English}$	0.72	0.67	0.69	242	0.877	**0.98**	0.48	0.65	242	0.874
$Data3_{c,\ English,\ French}$	0.85	**0.74**	**0.79**	**334**	**0.886**	0.87	**0.63**	**0.73**	**334**	**0.876**

We think it may be that the quality of the enhanced corpus is not good enough, resulting in some additional noise. Although it's not enough to just use two language, we should pay attention to the simplicity and implementability of back-translation. Just for the balance of datasets in Data3, some noise will be added to the training data, even many repeated word features will appear, and we can't get the expected effect.

TABLE 6 PERFORMANCE IMPROVEMENT

Dataset		Model	
		LSTM	***CNN***
Data1	baseline	0.841	0.855
	augmentation	0.883	0.893
	%improvement	4.9%	4.4%
Data2	baseline	0.856	0.862
	augmentation	0.874	0.871
	%improvement	2.1%	1.0%
Data3	baseline	0.872	0.862
	augmentation	0.886	0.876
	%improvement	1.6%	1.6%

TABLE 7 COMPARISON OF EDA AND BACK-TRANSLATION

Dataset	Data augmentation method	
	EDA for Chinese	***Back Translation***
Data1	0.852	0.883
Data2	0.866	0.874
Data3	0.863	0.886

Table 6 shows the comparison of the best accuracy with the baseline and the performance improvement. Based on all the experimental data, most metrics of the model after back-translation data augmentation have been improved. The average accuracy is 88.1%, which is 2.3% higher than the baseline average accuracy of 85.8%. Although the increase is relatively small, it at least shows that back-translation is effective.

The data augmentation techniques of EDA and back-translation with LSTM are presented in Table 7. The score is the best Test Accuracy of each and EDA is not as good as back-translation. From Data3, it can be seen that EDA may reduce the performance of the model. During the enhancement process, the meaning of the sentence may changed after random operations, but it remains the original label, resulting in a sentence with the wrong label. This does not happen with back-translation.

4. CONCLUSIONS

In this paper, We introduce the data augmentation method of back-translation, and prove that can boost the performance of Chinese text classification, especially when training on smaller datasets. In future work, we will investigate the influence of the corpus quality after back-translation on the text classification effect, and whether this method can improve the effect of some advanced deep learning models.

ACKNOWLEDGMENT

This work is supported by Gansu Province Key Research and Development Project (No.18YF1FA132).

REFERENCES

[1] S. Wong, A. Gatt, V. Stamatescu, and M. D. Mcdonnell, "Understanding Data Augmentation for Classification: When to Warp?," in digital image computing techniques and applications, 2016, pp. 1-6.
[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Commun. ACM, vol. 60, no. 6, pp. 84–90, 2017.
[3] F. Sebastiani, "Machine learning in automated text categorization," ACM Computing Surveys, vol. 34, no. 1, pp. 1-47, 2002.
[4] A. Anabytavor et al., "Do not have enough data? Deep learning to the rescue!," in national conference on artificial intelligence, 2020.
[5] W. Wang, B. Li, D. Feng, A. Zhang, and S. Wan, "The OL-DAWE Model: Tweet Polarity Sentiment Analysis With Data Augmentation," IEEE Access, vol. 8, pp. 40118-40128, 2020.
[6] X. Zhang, J. Zhao, and Y. Lecun, "Character-level convolutional networks for text classification," in neural information processing systems, 2015, pp. 649-657.
[7] W. Y. Wang and D. Yang, "That's So Annoying!!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets," in empirical methods in natural language processing, 2015, pp. 2557-2563.

[8] S. Kobayashi, "Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations," arXiv: Computation and Language, 2018.
[9] J. Wei and K. Zou, "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks," arXiv: Computation and Language, 2019.
[10] A. W. Yu et al., "QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension," in international conference on learning representations, 2018.
[11] R. Sennrich, B. Haddow, and A. Birch, "Improving Neural Machine Translation Models with Monolingual Data," in meeting of the association for computational linguistics, 2016, vol. 1, pp. 86-96.
[12] M. Miyabe and T. Yoshino, "Evaluation of the Validity of Back-Translation as a Method of Assessing the Accuracy of Machine Translation," in international conference on culture and computing, 2015, pp. 145-150.
[13] Q. Xie, Z. Dai, E. Hovy, M. Luong, and Q. V. Le, "Unsupervised Data Augmentation for Consistency Training," arXiv: Learning, 2019.
[14] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.
[15] J. L. Elman, "Finding Structure in Time," Cognitive Science, vol. 14, no. 2, pp. 179-211, 1990.
[16] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in EMNLP, 2014.
[17] https://github.com/SophonPlus/ChineseNlpCorpus/