ISSN 2395-8618

Development of Methods and Algorithms for Augmenting the Texts with Additional Information

Zhamilya Bimagambetova, Arukyz Sundetulla, and Syrym Moldash

***Abstract*—We have here explored different ways of text augmentation to explain each of them. The purpose of the article is to show methods of augmentation and calculate which one shows the best result in terms of the amount of new data created and the similarity of this data with the original. To do this, we use the subtitles for the movie as data and run our algorithm on each phrase in these subtitles.**

***Index Terms*—Natural language processing, data augmentation, deep neural networks, text classification.**

I. Introduction

Augmentation is the construction of additional data from the source data when solving machine learning problems. Usually, during augmentation, transformations of the original objects are used, which do not change their labels, but change (sometimes significantly) the descriptions. For example, if, while training a neural network that should distinguish photos of cats from photos of dogs, we rotate, stretch, change the brightness and contrast of the original images, this will not change what is depicted on them, but will give the opportunity to learn the network on "bad", deformed photos, as well as on angles that can be in short supply in the training sample.

Text classification is a fundamental task in natural language processing (NLP). Machine learning and deep learning have achieved high accuracy in tasks leading up to emotion analysis Tang et al., 2015 [1] to topic classification Tong and Koller, 2002 [2], but high performance often depends on the size and quality of training data, so collection is often boring. Automatic data magnification is commonly used in computer vision and speech Simard et al., 1998 [3]. It helps to develop more reliable models, especially when using small data sets. However, since it is difficult to develop generalized language transformation rules, universal methods of data augmentation in NLP have not been fully studied.

Text augmentation is a bit more complicated than image augmentation. Firstly, converting the text is more likely to distort its meaning (or even get meaningless text). Secondly, here the transformations are "less automatic". For example, to rotate a photo you don't need to be a photographer or know the laws of optics, but to rephrase some sentence you need to be at least a native speaker (and also know synonyms, context, etc.)

This paper is a review article that collects and synthesizes up-to-date information about several universal data augmentation methods for NLP. Such as, a set of Random swap (RS) [4] , Back translation [5], Replacement with synonyms [6]. We used subtitles from the Harry Potter movie as our dataset for each of the above methods. We compare each of these methods and calculate the pros and cons of each of them.

Manuscript received on 23/03/2022, accepted for publication on 13/02/2024.

II. Data

As data, we took the subtitles for the movie "Harry Potter and the Philosopher's Stone". The data consists of the phrases of the characters in the film. There are 1,241 phrases in the dataset, each containing more than 7 words on average. For each phrase, we will use the algorithm separately, so that later we can calculate the average similarity value.

III. Description of the Methods

Below we describe each of the listed types of augmentation.

A. Replacing with a synonym

The easiest way to rephrase is to replace words with synonyms (Synonym Replacement). The usual substitutions using a dictionary of synonyms were considered in the work of Zhang et al. Character-level Convolutional Networks for Text Classification [7]. Usually, when replacing, stop words are not chosen (articles, prepositions, conjunctions and other very common words that do not convey the main meaning of the text).

B. Contractions

You can both apply some accepted contractions (since = because, so on = td), and "disclose these contractions". There are lists of similar accepted contractions. For example, for English, there is such a list on the Wiki. Not all contractions can be unambiguously disclosed, for example, in English, "He's" can mean "Not is", or maybe "He has". There is a library for such augmentations.

C. Back Translation

If there are good automatic translators, the text is often translated into another language, and then translated "back" to the original one. It is clear that this just turns out to be a paraphrase of the original phrase. This method was used, for example, in the work of Xie et al. Unsupervised Data Augmentation [8], as well as by the winner of the Kaggle competition Toxic Comment Classification Challenge [9].

There are several techniques used in reverse translation that increase the number of possible augmentations. The first is that translation can be carried out into different languages. The second is that you can play with setting up the language model that forms the translation text (generating slightly less likely, from the point of view of LM, texts that can be successful paraphrases).

The authors are with the Kazakh-British Technical University, Almaty, Kazakhstan (syrym@seikolabs.kz).

ISSN 2395-8618

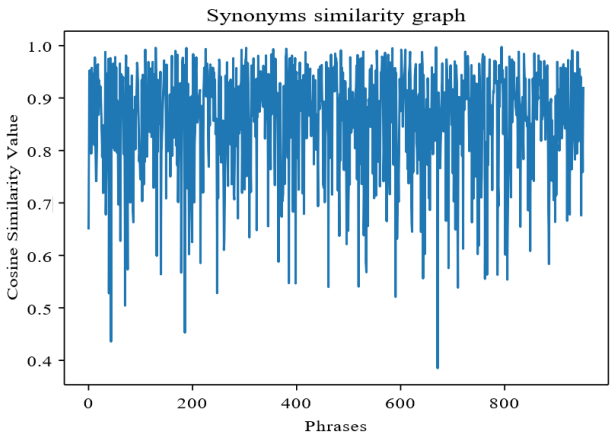

Fig. 1. Graph of newly created phrases and their similarity to the original by cosine metric

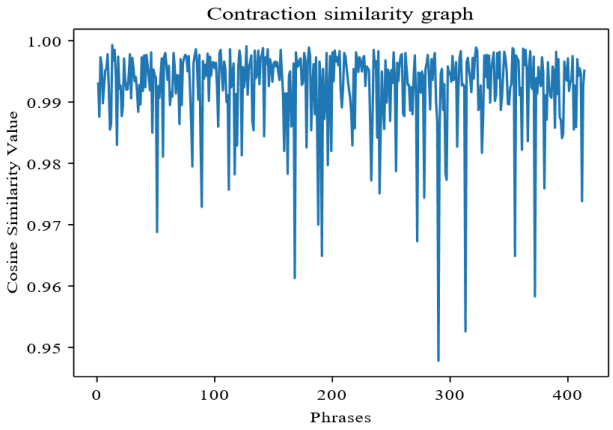

Fig. 2. Graph of newly created phrases and their similarity to the original by cosine metric

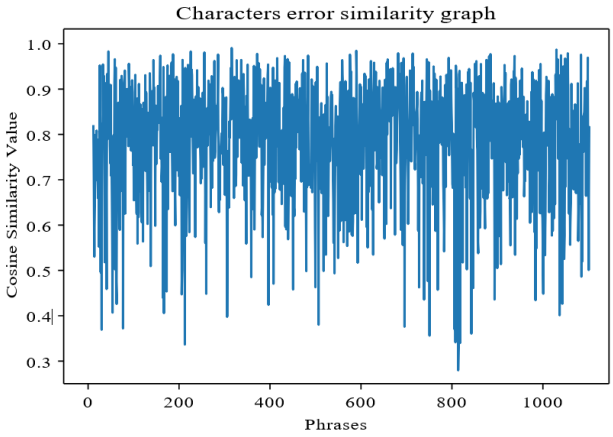

Fig. 3. Graph of newly created phrases and their similarity to the original by cosine metric

D. Random Swap (RS)

RS refers to various ways to spoil the text, which, however, are typical for texts. You can add errors in letters, punctuation marks, and change the case. When adding errors, you can try to make them so that they are similar to those that occur when typing (for example, replace the character with another based on the proximity of the corresponding keys on the keyboard

An interesting technique that is rarely done is random insertion / Random Insertion (RI), when a synonym for a random word of the same sentence is inserted into a sentence at a random place, see EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks (Wei et al.) [10].

IV. Models

Below we describe our experiments and their results.

A. Replacing with a synonym

First, we tested the method of replacing with synonyms. For this, NLTK library was used. First of all, we removed the stop words from our data, then we started looking for synonyms for words from our text. So that the phrases in our dataset do not lose their meaning, we indicated that a maximum of a third of the words in the phrase should be replaced by synonyms. The result of this experiment showed that we received 952 additional phrases in our dataset. The average similarity of each phrase to the original 88 percent (see Fig. 1).

B. Contractions

The next method for the experiment, we took the method of augmentation through contraction. Contraction is a division of stable expressions in the language, for example: I'm - I am. To do this, we use the contractions library. As a result, we received 414 new phrases with 99.4 percent similarity with the original (see Fig. 2).

C. Back Translation

Next, we tested the augmentation method through back translation. A back translation is a translation of a text into some language and a translation back into original. For this we used the googletrans library. In order to get a relatively new phrase, we decided to translate consecutively into 2 languages. German and Japanese were chosen as languages for translation. And as a result, we got 711 new phrases with 99.6 percent similarity with the original.

D. Random Swap (RS)

And lastly, we decided to test the augmentation method with adding errors to phrases. For this method, we have included the nlpaug library. To get natural errors, we used the errors that OCR programs allow. To avoid falling similarities between the feature product and the original, we decided to create only 1 new phrase per original. The result of the test was 1091 new phrases, but with very low similarity with the original in terms of cosine metric, 81 percent (see Fig. 3).

V. Comparisons and Status of the Scientific Elaboration of the Problem

In our experiment, we compared several data augmentation methods. The best result in terms of the amount of new data was shown by the RS method. After running it on our data, we got an 88 percent gain on our existing phrases. And the method of contraction proved to be the worst. With this method, we were only able to generate 33 percent of the new data. And if we measure the similarity of the newly created phrases to the original data, then the best methods will be contraction and back

ISSN 2395-8618

translation, they showed a cosine similarity value of 99.4 and 99.6 percent, respectively. In this component, the RS method showed itself much worse, its indicator at around 81 percent. Thus, we found that there is no one specific method that would be better in everything and it is necessary to approach the increase in the amount of data selectively, depending on the type of original data and on the task at hand.

VI. Conclusion

Augmentation is one of the main tools for improving the quality of networks. Being integrated into the learning process, it adds new properties to it, among which is the greater sensitivity of the network to the transformation parameters, as well as the potential to reduce the architecture while maintaining quality.

Also, we come back to the problem of correct augmentation settings. Since there is no universal criterion of "correctness" in this case, a set of transformations is always set based on the specifics of a specific task. Instead of weighting architecture and other approaches with the complication of the learning process, first of all it is always necessary to analyze the dataset and try to simulate the artifacts found in it – because data plays a major role in obtaining a well-working method.

References

[1] D. Tang, B. Qin, and T. Liu, *Document modeling with gated recurrent neural network for sentiment classification*, 2015.

[2] S. Tong and D. Koller, *Support vector machine active learning with applications to text classification*, 2002.

[3] P. Simard, Y. LeCun, J. S. Denker, and B. Victorri, *Transformation invariance in pattern recognition-tangent distance and tangent propagation*, 1998.

[4] O. Kolomiyets, S. Bethard, and M.F. Moens, *Model portability experiments for textual temporal analysis*, 2011.

[5] S. Edunov, M. Ott, M. Auli, and D. Grangier, *Understanding back-translation at scale*, 2018.

[6] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, *Natural language processing (almost) from scratch*, 2011.

[7] X. Zhang, J. Zhao, and Y. LeCun, *Character-level Convolutional Networks for Text Classification*, 2016.

[8] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, *Unsupervised Data Augmentation for Consistency Training*, 2020.

[9] Kaggle, *Toxic Comment Classification Challenge*, https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification- challenge/data, 2018.

[10] J. Wei and K. Zou, *EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks*, 2019.