Data Augmentation through Transformers for Text Classification

José Medardo Tapia-Téllez[1]

Instituto Nacional de Astrofisica Optica y Electronica(INAOE), Puebla, Mexico

Resumen The following work provides methods for Data Augmentation (DA) based on Transformers. Four methods were created: Single Masking, Double Masking, Triple Masking and Augmented Sentence. Methods were tested in four different classification tasks against data without augmentation and, on average, provide better accuracy results for CNN and RNN implementations. Of the four methods, Single Masking provided the best results, but Double and Triple Masking also showed accuracy improvement, thus showing that sentence multimasking is a possible DA method for text classification.
Palabras Clave: Natural Language Processing , Text Classification, Data Augmentation, Transformers, Natural Language Generation

1. Introduction

Data Augmentation (DA) techniques are widely use to increase the size of the training data. Through this process, one is able to reduce overfitting and increase the robustness of the machine learning models when not enough data is available. DA techniques are commonly used in Computer Vision (CV), however in Natural Language Processing (NLP) they have not been thoroughly explored. Since Text classification is a fundamental task in NLP, investigation in this area is pertinent, and thus our exploration of different techniques related to DA with transformers for text classification is of importance.

Within the work done in DA for NLP, recurrent problems come to light. First of all, methods like the one used in [2], though efective, lack the ability to utilize the context of the sentence. Another problem, which is addressed interestingly in [3] and [1], is the capacity to preserve the class labels for the augmented sentence. Finally, in CV we found generalized rules (for example rotation) that are commonly use for data augmentation, in NLP coming up with these rules is challenging, this is mainly because language transformations are not as direct, but is the research in this area that will develop these so needed rules.

In this work, we address two of the main problems previously mentioned. First of all, through the use of Transformers, we will obtain new data that is based in the context of the previous data. Because of the lack of time, we were not able to tackle the issue related to the class labels, but it would definitely be of interest in future work. Finally, the issue related to the exploration of language transformation rules, will be tackled through the analysis of four different DA techniques: single masking, double masking, triple masking and augmented sentence.

2. Related Work

In this section we review three related works that are the base for our research. In [2] they propose Easy Data Augmentation (EDA), which are four different operations to augment data: synonym replacement (Consists in choosing randomly n words from a sentence and replace them by a synonym.), random insertion (Consists on inserting a random synonym of a random word in a random position in the sentence.), random swap (Randomly choose two words in the sentence and swap their positions.) and random deletion (Randomly remove each word in the sentence with probability p.). EDA improves performance on both, convolutional and recurrent neural networks, and is particularly strong on smaller data sets.

The second work that is related to our research is [1]. They propose DA for label sentences, their method is called contextual augmenation. They stochastically replace words with other words that are predicted by a bi-directional language model at the word positions. Also, and it is important to remark this, they retrofit the language model with a label-conditional architecture, which allows the model to augment sentences without breaking the label-compatibility. Their results for six different classification tasks improve the classification, this based on convolutional neural networks.

The final work related to our research is [3]. They study different types of pre-trained transformer based models for what they call: conditional data augmentation. They are able to show that prepending the class labels to text sequences is a simple yet effective way to condition the pre-trained models for DA. Their conclusion is that on three classification benchmarks, pre-trained Seq2Seq model outperforms other models.

Based on the results provided in [2] we have to decided to work with small sets, also, since EDA has been criticized by the lack of context in their augmentation process, we decided to utilize Transformers. On [1] they perform their data augmentation with the use of a bi-directional language model to provide context to the augmentation process, since language model based pre-trained models such as BERT have provided significant gain across different NLP tasks, our approach differs in the use of transformers. The previously mentioned, along with the idea of introducing multiple masking and sentence augmentation are the main differences between the works. Finally, the work done on [3], although it is done with transformers, it is oriented in the comparison of different pre-trained Transformers, ours is oriented in the techniques implemented through transformers. This is done in order to tentatively provide some possible rules of transformation for the language so that data augmentation is viable.

3. Background

In order to completely understand our work some background is necessary. Let´s start with the notion of Transformers. A Transformer is a simple network architecture that is based entirely on attention mechanisms. Where an attention

function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. In essence, it is a weight assignation based on importance (this is why it "pays attention"). Transformers have increasingly become the state-of-the-art in NLP tasks and, in this research, we specifically utilize two: GPT-2 and BERT.

BERT, actually stands for Bidirectional Encoder Representations from Transformers, and is a language representation model. It is designed to pre-train deep bidirectional representaions from unlabeled text by jointly conditioning on both left and right context in all layers. Since BERT was original trained with a masked language modelling (MLM) objective, it is therefore efficient at predicting masked tokens. The previous, and the fact that we are interested in contextual prediction, are the main reasons for using BERT as our predicting tool.

The second Transformer that we use in our project is GPT-2. Which is a causal unidirectional Transformer pre-trained using language modeling on over a large corpus of 40GB of text data. GPT-2 is trained with a simple objective: predict the next word, given all of the previous words in some text. Based on this, is that we decided that GPT-2 was ideal for the purpose of augmenting a specific sentence.

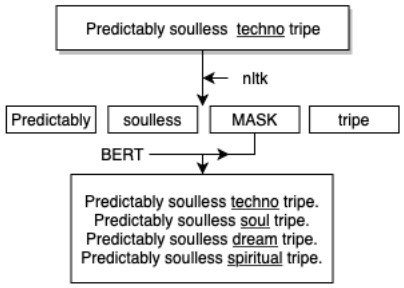
The last element in order to understand our research is the Natural Language Toolkit (NLTK). NLTK is a suit of open source program modules that provide a ready-to-use computational linguistics courseware. NLTK covers symbolic and statistical NLP, and is interfaced to annotated corpora. NLTK was vital in our research, along with BERT and GPT-2.

4. Method

As we've mentioned, in this reserch we've created 4 different augmentation techniques. Three were developed through the use of BERT: Single, Double, and Triple Masking. One was developed with the use of GPT-2: Augmented Sentence. Let's look at them in detail.

4.1. Single Masking Augmentation

The first method corresponds to Single Masking Augmentation. Where an specific sentence is first tokenized. Then a a random word in the sentence is masked, this masked sentence is inserted into BERT and a series of tokens are predicted. Based on the number of sentences to be augmented per sentence in training data, we produce new sentences with the respective tokens in the masked position. For example, if we would like three new sentences, then the first three tokens would be used to create them. 4.1 provides an scheme of the previous explanation.



H

Figura 1. Caption

4.2. Double Masking Augmentation

The second method corresponds to Double Masking Augmentation. The idea is the same as in Single Masking Augmentation but now we will mask two words of the original sentence. To do this we follow the procedure for Single Masking Augmentation, as we can see in 4.2. The obtained sentences are now masked in the second random position and a token is provided based on the order of the respective sentences. The final results are sentences with two words changed based on BERT. We wanted to test if mask augmentation had an impact in the classification, whether it was positive or negative.

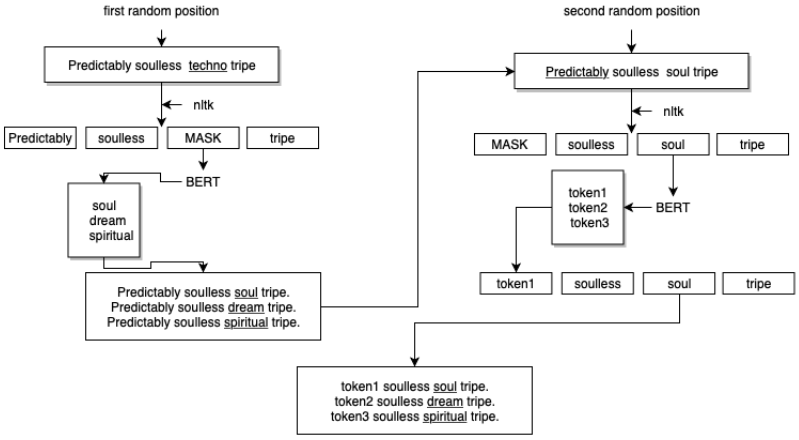


Figura 2. Caption

4.3. Triple Masking Augmentation

The third method corresponds to Triple Masking Augmentation. The idea is the same as in the previous two methods, but we now mask three words of the sentence. As we can see in 4.3, the dynamic is the same as in the previous method, but the sentences obtained from double masking are fed again into the machine with a new random position, thus creating the set of new sentences with three different words in them. This method has a positive and a negative side, in one way the sentence is completely different with three different words, but, in can also be so different that we loose the class label, we hope to see this in the classification results.

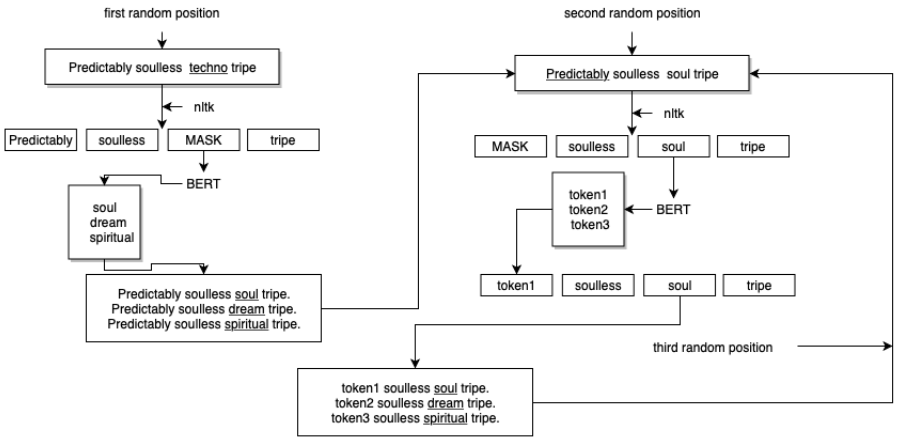


Figura 3. Caption

4.4. Augmented Sentence

Our final method consists in augmenting the size of the sentence. The procedure is pretty straight forward, as input we received a sentence, we introduced it into the model an it produced a sentence or sentences with fifty words added to it. In **??** you can see an scheme with an actual example.

Figura 4. Caption

5. Experiments and Results

5.1. Experiments

We use four different classifications datasets: SST-2 (Stanford Sentiment Treebak), IMDB (Sentiment-related movie reviews), Spam, and Sentence Type (sentences classified as command, question and statement) . Each of our datasets contained 600 elements, we separated 20 % for testing and another 20$ for validation.

For classification we utilize two neural networks: CNN and LSTM-RNN. The CNN and LSTM-RNN are implemented based on [2]. The CNN has the following layers: embedding, convolutional one dimension (activation = relu), dropout, global max pooling one dimension, dense (relu), dense (sigmoid); it is compiled with a binary cross entropy loss and an adam optimizer, and trained for ten epochs. The RNN has the following layers: embedding, LSTM, FC1, activation (relu), dropout, output layer, activation (sigmoid); it is compiled with a binary cross entropy and an RMSprop optimizer and, trained for ten epochs.

For each of the DA methods presented, we realize experiments related to their accuracy. The DA methods related to BERT were tested for an augmentation of one, three, five and ten sentences, this in order to see if there was a relation with the amount of sentences augmented and the accuracy; we expect to obtain a positive correlation. Another important aspect of the BERT related experiments is to see if we can find a relation between the amount of masking and the accuracy, since the sentence changes more and more in relation to the masking, label classification might be lost and thus the accuracy diminished; a negative correlation would be expected. The DA method related to GPT-2 is straight-forward and we did it in order to see if there was, as expected, a positive correlation between accuracy and the method. Finally, we obtained values for classification with and without the DA method for Single Masking with ten sentences augmented, thid from 10 % to 100 % of the data to see the behavior of the curves. We would expect a higher value curve for the classification related to the augmented data.

5.2. Results

Single Masking Augmentation Based on 5.2, we can conclude, that on average, the train data plus the data generated by the DA method Single Masking has a better accuracy on both neural networks. We can also conclude that, on average, Single Masking with ten augmented sentences has the best accuracy on RNN, this wasn't the case for CNN, giving the best results for three augmented sentences.

Cuadro 1. Results for DA with Single Masking for one, three, five and ten sentences augmented

		Plain	Plain + Single Masking				State of the Art
			1	3	5	10	
Spam	**RNN**	0.79	0.86	0.84	0.86	0.92	**ELCADP**
	CNN	0.59	0.68	0.67	0.65	0.63	**0.95**
IMDB	**RNN**	0.71	0.72	0.77	0.75	0.71	**XLNET**
	CNN	0.79	0.76	0.77	0.72	0.74	**0.96**
Sentence Type	**RNN**	0.35	0.42	0.46	0.53	0.56	**BERT**
	CNN	0.40	0.43	0.44	0.40	0.40	**0.65**
SST2	**RNN**	0.59	0.63	0.68	0.66	0.67	**ALBERT**
	CNN	0.68	0.63	0.66	0.65	0.71	**0.97**
Average	**RNN**	0.61	0.65	0.68	0.70	0.71	**0.88**
	CNN	0.61	0.62	0.63	0.60	0.62	

Double Masking Augmentation Based on 5.2, we can conclude that, on average, Double Masking has better accuracy results for RNN and CNN. We can also conclude, that, on average, augmenting ten sentences gave better results for RNN and CNN. Thus our initial hypothesis were correct, the more amount of sentences, the better the accuracy.

Cuadro 2. Results for DA with Double Masking for one, three, five and ten sentences augmented

		Plain	Plain + Double Masking				State of the Art
			1	3	5	10	
Spam	**RNN**	0.79	0.82	0.77	0.78	0.86	**ELCADP**
	CNN	0.59	0.65	0.57	0.66	0.74	**0.95**
IMDB	**RNN**	0.71	0.71	0.69	0.76	0.70	**XLNET**
	CNN	0.79	0.76	0.76	0.74	0.70	**0.96**
Sentence Type	**RNN**	0.35	0.37	0.44	0.35	0.50	**BERT**
	CNN	0.40	0.43	0.40	0.35	0.50	**0.65**
SST2	**RNN**	0.59	0.64	0.66	0.69	0.66	**ALBERT**
	CNN	0.68	0.68	0.65	0.70	0.70	**0.97**
Average	**RNN**	0.61	0.63	0.64	0.64	0.68	**0.88**
	CNN	0.61	0.63	0.59	0.61	0.66	

Triple Masking Augmentation Based on 3, we can conclude that, on average, DA with Triple Masking obtains better results on accuracy on all classification tasks for RNN and CNN. We can also conclude, that, on average, ten sentences was the best classification for RNN and one sentence for CNN.

Cuadro 3. Results for DA with Triple Masking for one, three, five and ten sentences

		Plain	Plain + Triple Masking				State of the Art
			1	3	5	10	
Spam	**RNN**	0.79	0.74	0.77	0.67	0.85	**ELCADP**
	CNN	0.59	0.69	0.65	0.62	0.57	**0.95**
IMDB	**RNN**	0.71	0.72	0.70	0.70	0.77	**XLNET**
	CNN	0.79	0.75	0.76	0.70	0.76	**0.96**
Sentence Type	**RNN**	0.35	0.43	0.38	0.47	0.51	**BERT**
	CNN	0.40	0.42	0.35	0.36	0.50	**0.65**
SST2	**RNN**	0.59	0.64	0.69	0.70	0.63	**ALBERT**
	CNN	0.68	0.70	0.71	0.70	0.63	**0.97**
Average	**RNN**	0.61	0.63	0.63	0.63	0.69	**0.88**
	CNN	0.61	0.64	0.62	0.59	0.61	

Augmented Sentence Based on 5.2, we can conclude that, on average, DA method Augmented Sentence, has better accuracy results for RNN.

Cuadro 4. Results for DA with Triple Masking for one, three, five and ten sentences augmented

		Plain	Plain + Augmented Sentence	State of the Art
Spam	RNN	**0.79**	**0.75**	**ELCADP**
	CNN	**0.59**	**0.61**	**0.95**
IMDB	RNN	**0.71**	**0.75**	**XLNET**
	CNN	**0.79**	**0.76**	**0.96**
Sentence Type	RNN	**0.35**	**0.46**	**BERT**
	CNN	**0.40**	**0.41**	**0.65**
SST2	RNN	**0.59**	**0.62**	**ALBERT**
	CNN	**0.68**	**0.67**	**0.97**
Average	RNN	**0.61**	**0.64**	**0.88**
	CNN	**0.61**	**0.61**	

Graphs for Accuracy Classification with and without Augmentation Based on 5.2, we don't see the logarithmic-curve form we would've expected, we blame this on the fact that the first percentage of the values don't have enough amount of data and the classifier returns noise. However in the graph related to SST-2, we can see a good behaviour of the data and, in general, we can see the yellow curve above the blue curve, a expected behavior. In the four graphs, in general, we can see the orange curve above the blue one, this is in accordance to the results in our table, however, it is clear that we need to work with bigger datasets.

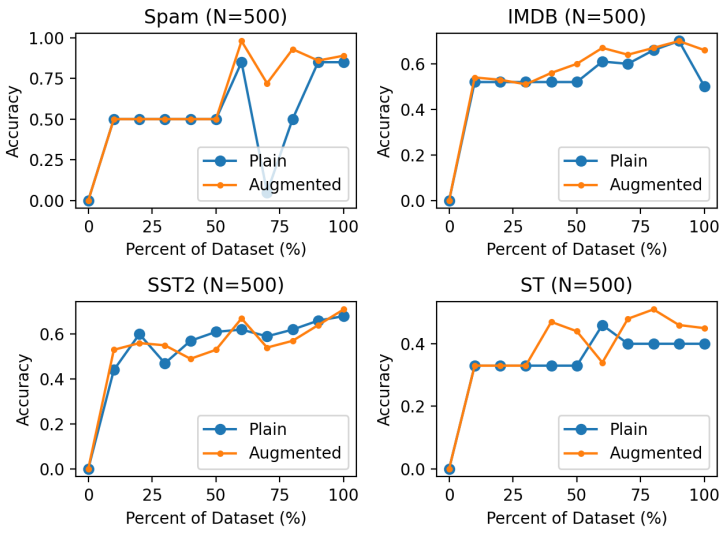


Figura 5. Performance on text classification tasks with and without Augmented Data for Single Masking in three sentences augmented

6. Conclusions

We proposed four DA methods for text classification based on Transformers (BERT and GPT-2): Single Masking, Double Masking, Triple Masking, and Augmented Sentence. The four methods, on average, improve accuracy results on RNN and CNN implementations. Related to the DA methods for BERT, on average, Single Masking has the best results, but Double Masking and Triple Masking also improve accuracy, thus indicating that multi-masking is a viable text transformation rule for DA. It is also important to note, that in each of these methods, the more sentences added, the better the accuracy results, thus multi-masking in combination with at least ten sentences is an efficient DA technique for text classification.

The Graphs are a clear indicator of two things: first of all, we need more data in order to perform more experiments and to validate through graphs the results from the tables, and second, the curves related to the augmented data classification are, in general, on top of the curves related to the plain data, thus indicating that the DA methods do improve accuracy.

As future work, we would first like to work with more data, with it we could produce better graphs and a second round of average results for the methods. Second, we would like to implement a class-label-related as part of our model, this in order to conserve the label within the text augmentation process. And finally, we would like to run experiments on combinations of the methods developed in this work. This would surely bring light over which could be the best tool in order to enhance our training data.

Referencias

1. Kobayashi, S. (2018). Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*.
2. Wei, J., & Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.
3. Kumar, V., Choudhary, A., & Cho, E. (2020). Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.
4. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems (pp. 5998-6008)*.
5. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
6. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 9.
7. Loper, E., & Bird, S. (2002). NLTK: the natural language toolkit. *arXiv preprint cs/0205028*.