Pre-trained Data Augmentation for Text Classification

Hugo Queiroz Abonizio(✉) and Sylvio Barbon Junior

State University of Londrina (UEL), Londrina, Brazil
{hugo.abonizio,barbon}@uel.br

Abstract. Data augmentation is a widely adopted method for improving model performance in image classification tasks. Although it still not as ubiquitous in Natural Language Processing (NLP) community, some methods have already been proposed to increase the amount of training data using simple text transformations or text generation through language models. However, recent text classification tasks need to deal with domains characterized by a small amount of text and informal writing, e.g., Online Social Networks content, reducing the capabilities of current methods. Facing these challenges by taking advantage of the pre-trained language models, low computational resource consumption, and model compression, we proposed the *PRE-trained Data AugmenTOR* (PREDATOR) method. Our data augmentation method is composed of two modules: the Generator, which synthesizes new samples grounded on a lightweight model, and the Filter, that selects only the high-quality ones. The experiments comparing Bidirectional Encoder Representations from Transformer (BERT), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) and Multinomial Naive Bayes (NB) in three datasets exposed the effective improvement of accuracy. It was obtained 28.5% of accuracy improvement with LSTM on the best scenario and an average improvement of 8% across all scenarios. PREDATOR was able to augment real-world social media datasets and other domains, overcoming the recent text augmentation techniques.

Keywords: Data augmentation · Text classification · Online social networks

1 Introduction

Data augmentation techniques have been successfully applied in machine learning models to improve their generalization capacity. It is a common strategy to avoid overfitting the training data, mainly on scenarios of data scarcity and situations where labeled examples are expensive. Since the performance of machine

The authors would like to thank the financial support of the National Council for Scientific and Technological Development (CNPq) of Brazil - Grant of Project 420562/2018-4 - and Fundação Araucária.

learning models is highly correlated with the amount and the quality of the data used during its training, low-data scenarios become a challenge for practitioners [13].

Several techniques have been proposed and evaluated for image data [30], but the field of textual data augmentation is still incipient. Simple transformations, such as flipping, cropping, and other image manipulations, are often label-preserving on image classification tasks [3,18], but this assumption does not hold for text data. Changing words order or removing some parts of a sentence might change its whole semantic, resulting in low-quality samples and negatively impacting the performance.

In recent years, different text transformation strategies have been proposed. Varying from synonyms replacements [17,33], paraphrasing through translation models [23] and text generation using language models [16], however no gold standard technique has been developed yet. A recent method, entitled Easy Data Augmentation (EDA) [28], has been proposed combining synonym replacement with other simple methods such as random deletion and random swap of words. Those methods were found to increase the accuracy of classification on small datasets.

An often employed technique is the Back-Translation (BT), which works by making a round-trip translation using a secondary language. Given a sentence written in a language L_a and a translation system between L_a and a different language L_b, the BT approach firstly translates the sentence from L_a to L_b and then translates it back to the original L_a language, generating a slightly different sentence. Previous work demonstrated that this approach leads to better results on Neural Machine Translation [23] and reading comprehension [32]. BT was also applied to low-resource text classification [24], yielding an improvement in classification accuracy using different secondary languages.

Most recent work has proposed the use of pre-trained language models [1,19], leveraging transfer learning for improving text generation capabilities when synthesizing new samples. However, those pre-trained models increase the computational requirements of classification pipelines, in contrast to simple sample transformations initially proposed. Beyond its resources requirements, those approaches can be prohibitively expensive and have raised a concern regarding the energy efficiency of those models [22,26].

In contrast with dictionary-based approaches, such as EDA, those pre-trained language models can deal better with noisy text coming from Online Social Networks (OSN). OSN texts are characterized by an informal writing style and the presence of Internet slangs [14], which leads to frequent out-of-vocabulary words in this scenario. Language model-based approaches, on the other hand, can learn to reproduce the dataset writing style and pre-trained models leverage a priori knowledge to extend its generation capabilities [20].

Tackling the challenges of text augmentation in different and recent domains, we present the *Pre-trained Data Augmentor* (PREDATOR), a novel method for textual data augmentation that combines the high performance achieved by transfer learning of pre-trained models approaches with lower computational

resource consumption. The simplicity of our methods is grounded on simple pre-trained models obtained by model compression [9]. PREDATOR works by synthesizing new high-quality samples to improve classification performance, particularly on small datasets, proving its effectiveness even on noisy social media datasets. We evaluated our method in three different datasets (*SST-2*, *AG-NEWS*, and *CyberTrolls*) from different media sources, using four different classifiers (Bidirectional Encoder Representations from Transformer, Convolutional Neural Networks, Long Short-Term Memory, and Multinomial Naive Bayes) and comparing the results with two other techniques present in the literature (Easy Data Augmentation and Back-Translation). The results demonstrated that PREDATOR increased the accuracy of all classifiers, achieving an average of 8% improvement in accuracy and a maximum of 28.5% on the best scenario, and statistical analysis demonstrated that its performance is similar to the real data to increase the dataset. The code is publicly available at https://github.com/hugoabonizio/predator.

The remainder of this paper of structured as follows: Sect. 2 describes the proposed method, detailing the pipeline and required parameters. Section 3 presents the evaluated datasets, the classification algorithms, and the methods compared with ours. In Sect. 4 we discuss the results regarding the amount of augmentation with all combinations of classifiers and datasets, and compare our method with two other widely adopted techniques on the same augmentation amount setups. Finally, on Sect. 5 we conclude and discuss future work.

2 Proposed Method

The main idea of our proposal regards the joint contribution of a text generator module and a filter module leading a two-step sample synthesizing approach. Thus, boosted by semi-supervised classification model, our method delivers a robust text augmentation method.

The PREDATOR architecture is composed of two modules: the Generator and the Filter, as shown in Fig. 1. These modules are responsible for synthesizing new samples and filtering high-quality ones, respectively. The first one, the Generator, is based on a language model [2], i.e., a model trained to predict the probability distribution for next tokens for a given context until it reaches a stop condition. This module is responsible for learning to generate new data corresponding to original classes while increasing its variability. The Filter module uses a text classifier trained on the original dataset towards selecting the high-quality new samples, i.e., accept as augmented samples only the synthetic samples in which the classifier has a high confidence of belonging to one of the given classes.

Figure 1 illustrates the main steps of our method pipeline. The first step is to initialize the Filter module by training its classifier to predict new sample labels based on the original dataset. Then, the Generator is trained to learn to synthesize new samples based on the original sentences by fine-tuning its language model. With both modules initialized, for each iteration of the method,

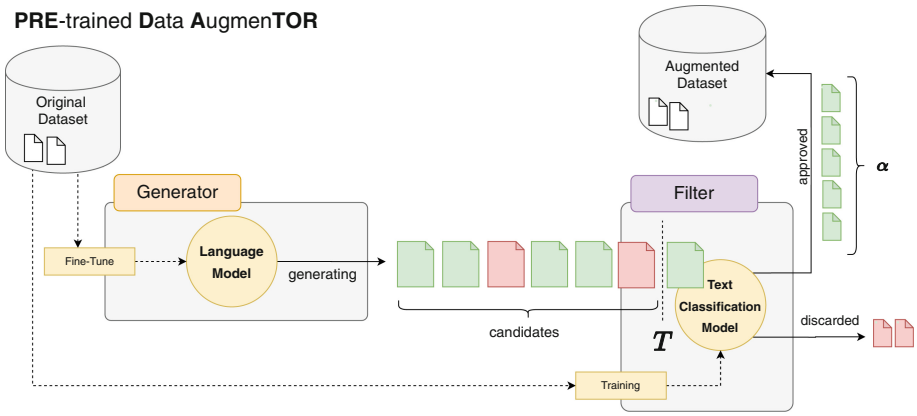

Fig. 1. Overview of proposed approach: PRE-trained Data AugmentOR

the generated samples are filtered, discarding low-quality sentences. The selected synthetic samples are accumulated until it reaches a previously defined number of augmented samples.

Among the several recently developed language models, we propose the usage of DistilGPT2 [21] on the Generator module. DistilGPT2 is a compressed version of GPT-2 obtained through knowledge distillation [9], becoming two times faster and having 33% fewer parameters than the smallest version of original GPT-2 with a minimal reduction in performance. This reduction makes the process of fine-tuning and posterior text generation much faster and reproducible with lower resources when compared to prior works.

The fine-tuning step may vary depending on the dataset, especially when its content is very different from the original corpus DistilGPT2 was trained. However, since DistilGPT2 was trained using OpenWebTextCorpus [8], a very diverse corpus, experimental results indicate that fine-tuning for only one epoch was enough to generate high-quality texts for augmenting the target dataset, even with a noisy dataset collected from social media interactions.

Given the language model fine-tuned in the given dataset domain, different methods can be employed to generate new texts. The previous work attached the class labels to condition the generation of text. Our proposed approach differs from previous [1] by simply concatenating three random samples from the target class using the language model input, i.e., given random samples s from from a target class, and a separator token already included on model vocabulary `SEP`, the input is given by $s_1\texttt{SEP}s_2\texttt{SEP}s_3\texttt{SEP}$. Thus, the following generation maintains the characteristics of the target class. The generation is done by sampling the probability of the language model, which, in contrast with beam-search and greedy decoding, generates higher-quality and more diverse texts [11]. The decoding strategy used in PREDATOR is top-k sampling [7], with $k = 40$.

After Generator, the next step is the selection of those new synthesized samples and the imputation of its class by a classifier trained on the original dataset. This process is performed by the Filter module for avoiding low-quality samples in the final augmented dataset, essential to leverage high accurate outcomes. Current state-of-the-art classification models are often large Transformer-based classifiers [29], which makes them too resource-hungry. Therefore they are not well suited to be directly applied to a pipeline of data augmentation. Thus, Sanh et al. [21] developed DistilBERT, a compressed model obtained through knowledge distillation which results in a 40% smaller model than original BERT, while being 60% faster and achieving 97% of its original performance. Therefore, DistilBERT is used as the classifier for the Filter module, maintaining a competitive performance, and meeting the requirements of computational resources.

PREDATOR requires two main hyperparameters: α as the augmentation rate, determining how many new samples need to be synthesized by the Generator module, and T as the threshold confidence for the Filter module. The α parameter controls the increase of the augmented dataset regarding the original sample size, i.e., given a dataset with n samples per class and an α of 0.25, the resulting augmented dataset will have $1.25n$ samples for each class. In this work, we conduct a more in-depth experiment with different values for α, showing its behavior on different datasets.

The T parameter controls the flexibility of the Filter, determining whether it is stricter or more flexible on the sample quality selection. Quality refers to the predictive power regarding classification task considering the given samples. With a more strict T value, the module only selects the samples that its classifier predicts the class with the most confidence. In contrast, a lower value of T might approve samples that the classifier has more uncertainty. This value represents a trade-off since a higher value makes it difficult for the Generator to synthesize enough samples, making it necessary to do more iterations to satisfy α requirement. On the other hand, lower values can lead to low-quality augmented dataset due to noisy samples.

Another aspect that needs to be assessed is the variability sought by data augmentation methods. With a high T, only samples with high certainty will be selected, i.e., only samples very similar to the original dataset will be included in the augmented dataset, which is a suboptimal result. Since the objective of data augmentation is to enrich the dataset with different samples without losing its class nature, a certain amount of uncertainty is required. Preliminary experiments showed that a value of 0.7 (default value) for T is a reasonable default since it is a balance on prediction confidence. However, different values of T for different scenarios might be explored in future works, especially on datasets with a higher class carnality.

2.1 PREDATOR Augmentation Example

Table 1 shows examples of original samples and synthesized to illustrate the learning of writing style from the original dataset and the class preserving of the

generated text. Besides writing style, the text length also tends to be similar, even though it was not directly enforced.

Table 1. Examples of original and synthesized samples for each dataset showing samples from the same class.

Dataset	Real samples	Augmented samples
AG-NEWS	*Apple cuts prices and improves products. Apple introduced a range of new machines on Tuesday, as it gears up for the annual christmas shopping season. As part of the launch, it cut the price of its entry-level iBook G4 notebook computer and boosted chip speed across the line*	*Apple Shares. A European market leader says heś ready to share it. Apple stock was high Tuesday on the eve of its annual earnings*
CyberTrolls	*i hate that! That sucks!!!*	*oh man that sucks!! LOL*
SST-2	*The jokes are flat, and the action looks fake*	*... even some jokes are off the charts and probably just too out of place*

3 Materials and Methods

3.1 Datasets

We evaluated the method on three different text classification datasets. *SST-2* (Stanford Sentiment Treebank)[1] [25], a classic dataset for sentiment classification on movie reviews widely applied as benchmark [16,19,28], with two classes: positive and negative. *AG-NEWS*[2] [33], another common benchmark dataset, but for topic classification task [12,31] composed of news belonging to four classes: world, sports, business and science/technology. *CyberTrolls*[3] is a more recent dataset for the task of aggressiveness detection in social networks, with examples of two classes: cyber-aggressive and non cyber-aggressive. *CyberTrolls* presents a challenge for text mining given the noisy characteristics of this media. With those three datasets we test our method with representations of different written styles to compare its behavior on more formal and more informal data sources.

The experiments were conducted to reproduce a small data scenario, where the classifier is trained on a restricted number of samples, and data augmentation performance is compared with the subsampled set. Previous work has simulated low-data regime setting by subsampling original datasets [13,19,28],

[1] https://nlp.stanford.edu/sentiment/.
[2] http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html.
[3] https://www.kaggle.com/datathurks/dataset-for-detection-of-cybertrolls.

becoming a common practice when evaluating augmentation techniques. Thus, for each dataset, 100 samples per class were subsampled and, since it is a non-deterministic process, this procedure was repeated ten times to average the final results. Those subsample of the dataset are assigned as the original performance on experiments, and all augmentation was made based on them. It is important to emphasize that only train sets were subsampled, validation and test sets were kept the same as originals.

3.2 Text Classification Algorithms

We conducted the experiments using four different classification models: Bidirectional Encoder Representations from Transformers (BERT) [5], Convolutional Neural Networks (CNN) [15], Long Short-Term Memory (LSTM) [10], and Multinomial Naive Bayes (NB) [27]. Those classifiers were selected to represent different categories, having NB as a classic text classifier often compared as a baseline method [27], LSTM, and CNN as common deep learning classifiers [33], and BERT representing the most recent progress achieving state-of-the-art on numerous NLP tasks including text classification.

3.3 Augmentation Methods

Since the text data augmentation is still an emergent topic, several methods have been proposed, but there is still no de facto standard technique. The two most applied techniques found in the literature are synonyms replacement and BT. EDA is an extension of synonyms replacing, introducing simple text transformations that were successfully applied to other works. Therefore, we compared PREDATOR with those two widely applied augmentation techniques (EDA and BT), observing the boosting on performance on different domains.

The first compared technique was BT, where each sentence of the original dataset is translated into a different language and then translated back to the source. This method requires two models: a model to translate from source language to a target one, and the inverse model. Among the alternatives of models, we conducted the experiments using the models proposed by Edunov et al. [6], a Transformer model made publicly available[4].

The second compared technique was EDA, which the code is publicly available[5]. The hyperparameters used to generate new samples were the recommended default, generating 9 new samples for each sample in the original datasets.

4 Results and Discussion

In this section, we expose two perspectives to evaluate the proposed method. First, it is discussed the augmentation ratio and classification performance from

[4] https://pytorch.org/hub/pytorch_fairseq_translation/.
[5] https://github.com/jasonwei20/eda_nlp.

the original size to nine augmented outcomes using three different datasets (*AG-NEWS*, *CyberTrolls*, and *SST-2*) and four classification algorithms (BERT, CNN, LSTM, and NB). The second perspective supports the comparison between PREDATOR and current text augmentation methods (EDA and BT) using the original textual resource (Truth) with the same amount of samples for each method.

4.1 Augmentation Capabilities

The main goal when using augmentation methods, independent of problem, is to improve the predictive performance. We performed 9 augmentation rates (0.1, 0.25, 0.50, 1.0, 1.5, 2.0, 3.0, 6.0 and 9.0) on *AG-NEWS*, *CyberTrolls* and *SST-2* datasets. Figure 2 shows the accuracy of different augmentation rates across all four different classification algorithms (BERT, CNN, LSTM and NB).

A prominent boosting in performance was obtained by LSTM on *AG-NEWS* (first column and third row in Fig. 2) since the original accuracy of 71% reach 84% when augmented. In the same combination of algorithm and dataset, we can observe an improvement of the model quality grounded in the reduction on the accuracy standard deviation, highlighted by the performance shadowed mark.

The overall accuracy improvement between the original size and the maximum augmentation (9x) across all scenarios is exposed in Table 2. The results were grouped by dataset with the biggest improvement highlighted in bold and the smallest underlined. As the previous case presented, LSTM obtained the biggest improvement in all scenarios. BERT provided small improvement on *AG-NEWS* and *CyberTrolls* and CNN on *SST-2*.

Table 2. Augmentation results grouped by datasets for each classifier, highlighting the biggest improvements in bold and the smallest improvements underlined.

Dataset	Algorithm	Avg. Improvement	Original Acc
AG-NEWS	BERT	+0.7%	87.2%
AG-NEWS	CNN	+2.2%	83.1%
AG-NEWS	LSTM	**+28.5%**	65.4%
AG-NEWS	NB	+2.8%	80.0%
CyberTrolls	BERT	+6.8%	63.5%
CyberTrolls	CNN	+0.7%	65.2%
CyberTrolls	LSTM	**+10.0%**	58.7%
CyberTrolls	NB	+3.5%	62.8%
SST-2	BERT	+2.0%	81.5%
SST-2	CNN	+12.9%	63.8%
SST-2	LSTM	**+17.9%**	59.8%
SST-2	NB	+8.6%	63.4%

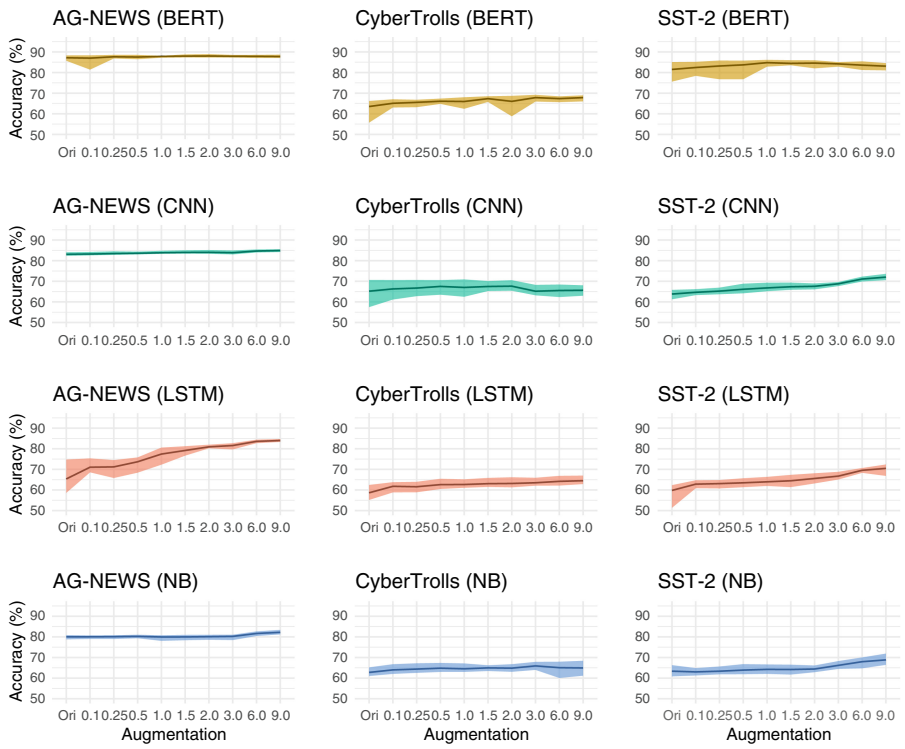

Fig. 2. Performance obtained from different algorithms and datasets using the original size and different augmented sources with PREDATOR.

It is important to note that improvement depends on the original performance, since LSTM was the algorithm that took the most advantage of PREDATOR, but obtained the lower average performance in comparison to the other algorithms. The high improvement on LSTM results indicates it tends to overfit the training data when using a small number of samples, thus the variances introduced by augmentation act as a regularizer, highly improving its performance. Conversely, BERT presented the smallest difference when using our augmentation, but high classification performance.

The higher average performance increases were achieved in *SST-2* and *AG-NEWS* datasets, traditional benchmarks with cleaned texts. *SST-2* is composed of movie reviews and *AG-NEWS* is composed of news articles, which tend to have a more formal writing style. On the other hand, the lowest average increase happened in *CyberTrolls* dataset, which is composed of highly noisy texts, containing emojis and specific social media expressions. Even though this writing style might be more difficult for the language model to reproduce, the fine-tuning step and the proposed input seed strategy was proved effective in writing style conditioning and robust on noisy datasets.

4.2 Methods Comparison

EDA and BT were proposed discussing its advantage over smaller datasets and intensively experiment with a specific augmentation rate. Thus, we created particular scenarios to provide a fair comparison between the methods and PREDATOR.

For the first scenario, we created the Truth baseline from the original data with the same amount of text augmented by the compared methods. Particularly, it was used the triple of size from the original training size for Truth and PREDATOR was configured to augment twice when comparing BT. Figure 3 presents boxplots of accuracy with all classification algorithms using all datasets grouped by the augmentation method. PREDATOR overcome BT with all classification algorithms, with a greater accuracy difference with CNN (2.3%). The smallest difference was obtained with BERT about (0.2%), the most predictive algorithm. The truth was superior to all methods.

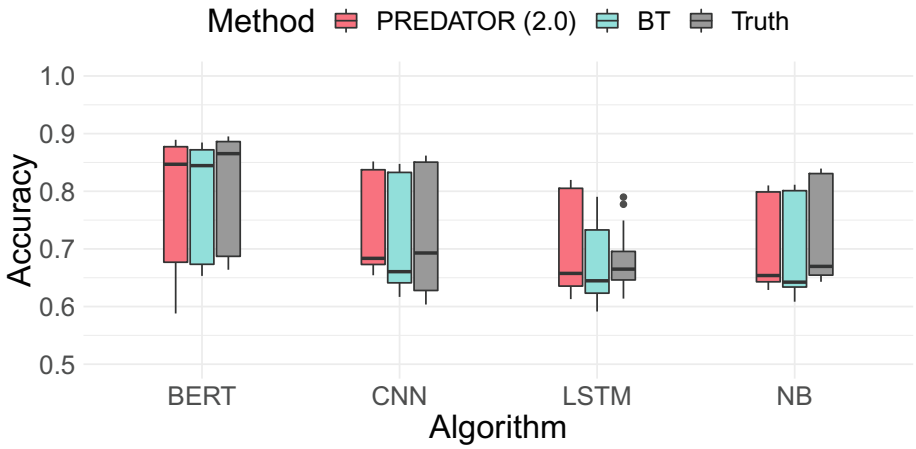

Fig. 3. Accuracy comparison among augmentation methods (PREDATOR and BT) and truth dataset. The boxplots were computed with the three datasets (*AG-NEWS*, *CyberTrolls* and *SST-2*) grouped by classification algorithms.

In order to observe a performance superiority of a method, we evaluated the statistical significance using the Friedman test, with a significance level of $\alpha = 0.05$. The null hypothesis is that the augmentation methods are similar. Anytime the null hypothesis is rejected, the Nemenyi post hoc test is applied, stating that the performance of a pair of methods are significantly different if their corresponding average ranks differ by at least a critical distance value. When multiple methods are compared in this way, a graphic representation can be used to represent the results with the Critical Difference Diagram, as proposed by Demšar [4]. This analysis is shown in Fig. 4, where it is possible to conclude

that Truth and PREDATOR are similar, PREDATOR and BT are similar, and Truth and BT are statistically different. Thus, we can claim that our proposal is statistically similar to the usage of the original data.

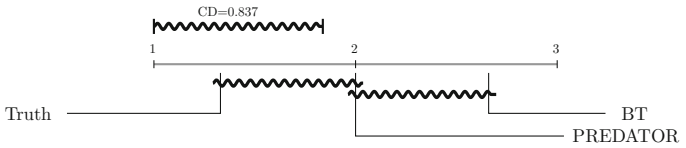

Fig. 4. Comparison of the accuracy values obtained by augmentation methods (PREDATOR and BT) and truth with the nemenyi test. Groups that are not significantly different ($\alpha = 0.05$ and $CD = 0.83$)

In the second scenario, PREDATOR was compared to EDA, one of the most recent proposals. A situation to the first scenario was found. Truth, the real data, obtained the best performance followed by PREDATOR and EDA. In this scenario, PREDATOR augmented the original dataset by nine to match the same amount of Truth and EDA. The more significant difference between the proposed method and EDA was using NB, about 6.3% accuracy, as Fig. 5 shows. Again, BERT achieved the smallest accuracy difference, an average of 0.1%.

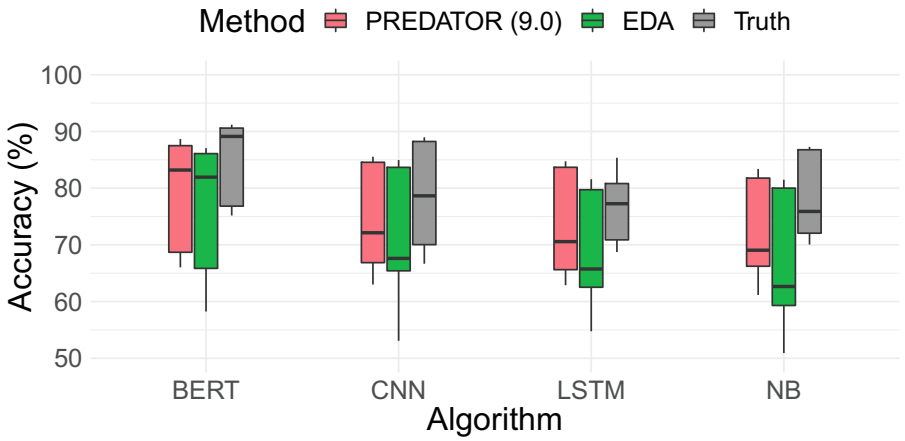

Fig. 5. Accuracy comparison among augmentation methods (PREDATOR and EDA) and truth dataset. The boxplots were computed with the three datasets (*AG-NEWS*, *CyberTrolls* and *SST-2*) grouped by classification algorithms

Using the same statistical assumptions of the first scenario, we employed Friedman and Nemenyi test to compare PREDATOR, EDA, and Truth. As Fig. 6

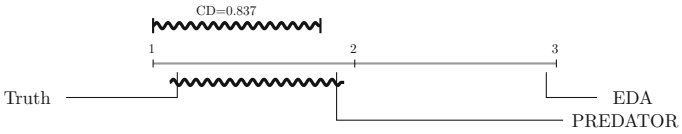

Fig. 6. Comparison of the accuracy values obtained by augmentation methods (PREDATOR and EDA) and truth with the nemenyi test. Groups that are not significantly different ($\alpha = 0.05$ and $CD = 0.83$)

exposes, it is possible to conclude that Truth and PREDATOR are similar and statistically different from EDA. Thus, we can affirm that our proposal produces synthetic data able to support the training of a text classification model capable of obtaining results statistically similar to the usage of real data. Therefore, PREDATOR generates samples that lead to superior performance than EDA.

The results reveal that the PREDATOR approach is an effective method for improving the performance of the classifiers, resulting in similar or greater performance than its alternatives. It also proves to be robust to noise on the text and informal writing, improving the accuracy of the model on a real-world social media dataset. Other methods, such as EDA, depend on a fixed dictionary to work, causing out-of-vocabulary issues on rare words, neologisms, and internet slang. This explains the poor performance of EDA on *CyberTrolls* dataset and shows better handling of these issues by our method.

Another aspect is the amount of added samples to the training set. While EDA increases the training set in ten times its size by default, our method achieves the same or higher performance with less than one-fifth of it on average. On the other hand, the BT approach depends on the number of available secondary languages to be translated, i.e., with only one language to translate, the training set is doubled. Although PREDATOR and BT did not show a significant difference for the same amount of augmented samples, PREDATOR can increase this amount considerably using the same model due to its sampling generation method. At the same time, BT depends on new translation models to increase the samples.

4.3 Open Issues and Limitations

The main limitation of PREDATOR is its Anglophone-centric nature since the pre-trained models are trained in the English language. However, this issue can be easily overcome with the usage of models pre-trained on other languages or even multilingual models on its modules. The PREDATOR architecture enables the change of its kernel models, making it possible to be applied to different languages or taking advantage of newer classifiers and language models in the future.

The minimum amount of samples required to apply the method successfully is not yet clear. Too few amounts of samples can not be enough to train an effective Filter module since it depends on its classifier quality. More experiments need

to be taken in order to define the optimal hyperparameters for each different scenario.

5 Conclusion

We presented PREDATOR, a novel data augmentation technique for text classification leveraged by transfer learning using a language model to synthesize new high-quality samples. Our proposed method was experimentally compared with two widely adopted methods across three datasets using four text classifiers. The results show that PREDATOR is effective with either cleaner benchmark datasets and noisy real-world datasets. Besides achieving a better average performance than the compared methods, it is statistically similar to using the original dataset with the same amount of data.

A natural progression of this work is to analyze the behavior of different kernels, using different language models and classifiers to improve its applicability to lower-resourced languages. Another possible area of future research would be to expand the experiments with newer language model-based approaches, assessing its resources consumption and complexity.

References

1. Anaby-Tavor, A., et al.: Not enough data deep learning to the rescue. arXiv preprint arXiv:1911.03118 (2019)
2. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**(2), 1137–1155 (2003)
3. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: learning augmentation policies from data (2019).https://arxiv.org/pdf/1805.09501.pdf
4. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota, June 2019. https://doi.org/10.18653/v1/N19-1423, https://www.aclweb.org/anthology/N19-1423
6. Edunov, S., Ott, M., Auli, M., Grangier, D.: Understanding back-translation at scale. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 489–500. Association for Computational Linguistics, Brussels, Belgium, Oct-Nov 2018. https://doi.org/10.18653/v1/D18-1045, https://www.aclweb.org/anthology/D18-1045
7. Fan, A., Lewis, M., Dauphin, Y.: Hierarchical neural story generation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 889–898. Association for Computational Linguistics, Melbourne, Australia, July 2018. https://doi.org/10.18653/v1/P18-1082, https://www.aclweb.org/anthology/P18-1082

8. Gokaslan, A., Cohen, V.: Openwebtext corpus (2019). http://Skylion007.github.io/OpenWebTextCorpus
9. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: NIPS Deep Learning and Representation Learning Workshop (2015). http://arxiv.org/abs/1503.02531
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
11. Holtzman, A., Buys, J., Du, L., Forbes, M., Choi, Y.: The curious case of neural text degeneration. arXiv preprint arXiv:1904.09751 (2019)
12. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 328–339. Association for Computational Linguistics, Melbourne, Australia, July 2018. https://doi.org/10.18653/v1/P18-1031, https://www.aclweb.org/anthology/P18-1031
13. Hu, Z., Tan, B., Salakhutdinov, R.R., Mitchell, T.M., Xing, E.P.: Learning data manipulation for augmentation and weighting. In: Wallach, H., Larochelle, H., Beygelzimer, A., dÁlché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32, pp. 15764–15775. Curran Associates, Inc. (2019). http://papers.nips.cc/paper/9706-learning-data-manipulation-for-augmentation-and-weighting.pdf
14. Igawa, R.A., et al.: Account classification in online social networks with lbca and wavelets. Inform. Sci. **332**, 72–83 (2016)
15. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751. Association for Computational Linguistics, Doha, Qatar, October 2014. https://doi.org/10.3115/v1/D14-1181, https://www.aclweb.org/anthology/D14-1181
16. Kobayashi, S.: Contextual augmentation: data augmentation by words with paradigmatic relations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 2 (Short Papers), pp. 452–457. Association for Computational Linguistics, New Orleans, Louisiana, June 2018. https://doi.org/10.18653/v1/N18-2072, https://www.aclweb.org/anthology/N18-2072
17. Kolomiyets, O., Bethard, S., Moens, M.F.: Model-portability experiments for textual temporal analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers, Vol. 2, pp. 271–276. HLT '11, Association for Computational Linguistics, USA (2011)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
19. Kumar, V., Choudhary, A., Cho, E.: Data augmentation using pre-trained transformer models. arXiv preprint arXiv:2003.02245 (2020)
20. Petroni, F., et al.: Language models as knowledge bases. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 2463–2473. Association for Computational Linguistics, Hong Kong, China, November 2019. https://doi.org/10.18653/v1/D19-1250, https://www.aclweb.org/anthology/D19-1250
21. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)

22. Schwartz, R., Dodge, J., Smith, N.A., Etzioni, O.: Green ai. ArXiv abs/1907.10597 (2019)
23. Sennrich, R., Haddow, B., Birch, A.: Improving neural machine translation models with monolingual data. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 86–96. Association for Computational Linguistics, Berlin, Germany, Auguest 2016. https://doi.org/10.18653/v1/P16-1009, https://www.aclweb.org/anthology/P16-1009
24. Shleifer, S.: Low resource text classification with ulmfit and backtranslation. arXiv preprint arXiv:1903.09244 (2019)
25. Socher, R., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1631–1642 (2013)
26. Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for deep learning in NLP. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3645–3650. Association for Computational Linguistics, Florence, Italy, July 2019. https://doi.org/10.18653/v1/P19-1355, https://www.aclweb.org/anthology/P19-1355
27. Wang, S., Manning, C.D.: Baselines and bigrams: simple, good sentiment and topic classification. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers, vol. 2, pp. 90–94. Association for Computational Linguistics (2012)
28. Wei, J., Zou, K.: EDA: easy data augmentation techniques for boosting performance on text classification tasks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP), pp. 6382–6388. Association for Computational Linguistics, Hong Kong, China, November 2019. https://doi.org/10.18653/v1/D19-1670, https://www.aclweb.org/anthology/D19-1670
29. Wolf, T., et al.: Huggingface's transformers: state-of-the-art natural language processing. ArXiv abs/1910.03771 (2019)
30. Wong, S.C., Gatt, A., Stamatescu, V., McDonnell, M.D.: Understanding data augmentation for classification: when to warp. In: 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp. 1–6. IEEE (2016)
31. Yogatama, D., Dyer, C., Ling, W., Blunsom, P.: Generative and discriminative text classification with recurrent neural networks. arXiv preprint arXiv:1703.01898 (2017)
32. Yu, A.W., Dohan, D., Luong, T., Zhao, R., Chen, K., Le, Q.: Qanet: combining local convolution with global self-attention for reading comprehension (2018). https://openreview.net/pdf?id=B14TlG-RW
33. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, Vol. 1, pp. 649–657. NIPS'15, MIT Press, Cambridge, MA, USA (2015)