Toward Text Data Augmentation for Sentiment Analysis

Hugo Queiroz Abonizio, Emerson Cabrera Paraiso, and Sylvio Barbon, Jr.

***Abstract*—A significant part of natural language processing (NLP) techniques for sentiment analysis is based on supervised methods, which are affected by the quality of data. Therefore, sentiment analysis needs to be prepared for data quality issues, such as imbalance and lack of labeled data. Data augmentation methods, widely adopted in image classification tasks, include data-space solutions to tackle the problem of limited data and enhance the size and quality of training datasets to provide better models. In this work, we study the advantages and drawbacks of text augmentation methods such as easy data augmentation, back-translation, BART, and pretrained data augmentor) with recent classification algorithms (long short-term memory, convolutional neural network, bidirectional encoder representations of transformers, support vector machine, gated recurrent units, random forests, and enhanced language representation with informative entities, that have attracted sentiment-analysis researchers and industry applications. We explored seven sentiment-analysis datasets to provide scenarios of imbalanced datasets and limited data to discuss the influence of a given classifier in overcoming these problems, and provide insights into promising combinations of transformation, paraphrasing, and generation methods of sentence augmentation. The results revealed improvements from the augmented dataset, mainly for reduced datasets. Furthermore, when balanced by augmenting the minority class, the datasets were found to have improved quality, leading to more robust classifiers. The contributions to this article include the taxonomy of NLP augmentation methods and their efficiency over several classifiers from recent research trends in sentiment analysis and related fields.**

***Impact Statement*—Data augmentation methods have substantially improved the data-driven predictive models. However, we considered text data augmentation methods that have been explored naively, particularly for sentiment-analysis problems. As a result, this field lacks discussion, analysis, and understanding of the entire phenomenon related to the augmented samples and their impact on the current classification methods. Here, we propose a new organization of categories and methods to shed light on this topic. Furthermore, we present advantages, drawbacks, and particularities when augmenting sentiment-analysis datasets by combining the most prominent augmented methods with several classification methods.**

***Index Terms*—Machine learning, natural language processing (NLP), sentiment analysis, text analysis, text mining.**

I. Introduction

WITH the rapid growth of textual data produced as a result of the Web and its interactions, sentiment analysis plays an important role in the effective application of AI models. Sentiments and emotions bring a degree of subjectivity, which is essential in human-to-human interactions. Therefore, sentiment analysis is a field of identifying and understanding these subjectivities and nuances, and is crucial for human-to-machine interactions. Sentiment-analysis applications range from commercial and academic tools to large and small companies, and have great potential as subcomponents for other technologies [1]. These techniques enable the automation of the analysis of a large amount of data [2] and the extraction of knowledge and insights from raw unstructured data [3], [4]. Although most of the research relies on deep learning methods [5], recent work has achieved advances in combining the bottom-up approach of learning language features from deep learning with a top-down approach of modeling commonsense knowledge [6]. However, sentiment-analysis models require a vast amount of training data to effectively learn these patterns. Low-quality datasets are often found when developing this type of system, with issues including data scarcity and the lack of labeled samples, which may degrade the performance of these models in real-world scenarios [7]. The scarcity of linguistic and textual resources has been a recurring issue in many NLP tasks [8]. Furthermore, the lack of data could compromise the sample quality and affect data distribution. As a result, the imbalanced data violates the assumption of a relative equilibrium distribution for most learning algorithms, which can significantly decrease the classification performance. Real-world datasets often suffer from data scarcity issues, which may lead to an overfitting scenario. When classification models are trained with few samples, they tend to memorize features from the training set instead of learning the underlying feature distribution, resulting in an inadequate generalization capacity [9]. In addition to data augmentation, different approaches have been proposed to handle data scarcity and imbalances in real-world scenarios. Neural network regularization [10], dropout regularization [11], batch normalization [12], and transfer learning [13], [14] are among the most widely adopted techniques, especially for deep learning methods. One-shot and zero-shot learning is a more recent paradigm for building models with minimal data that can deliver promising results [15]. Text data augmentation methods have been proposed to mitigate the data scarcity issue by performing class-preserving manipulation on the original data source [16]. These methods are common strategies to avoid overfitting the training data, mainly on small datasets and situations where labeled examples are expensive. However, unlike in the case of simple image transformations, such as rotation and translation, in these methods, preserving the original label after text perturbations may be more complex. Thus, different methods have been proposed in recent years to address this problem. Ranging from simple text transformations such as dictionary-based synonym replacement to more complex methods involving large language models and transfer learning, each technique has its own advantages and disadvantages. For example, more straightforward methods may fall short in more linguistically diverse scenarios such as social network media. By contrast, more complex methods may include significant overhead in the pipeline, which increases the training time. Thus, experiments evaluating the methods in diverse scenarios are required to recognize the benefits and drawbacks. To understand the effects of text data augmentation and how the classification method can handle data quality drawbacks, we systematically studied how sentiment analysis with different algorithms is affected by data augmentation methods. We performed our experiments using easy data augmentation (EDA), back-translation (BT), pretrained data augmentor (PREDATOR), and BART augmentation methods with recent classification algorithms. We augmented seven original datasets for sentiment-analysis tasks by using highly accurate classification methods: Long short-term memory (LSTM), convolutional neural network (CNN), bidirectional encoder representations of transformers (BERT), support vector machine (SVM), gated recurrent units (GRUs), random forests (RF), and enhanced language representation with informative entities (ERNIE). Three scenarios representing imbalanced datasets, small datasets, and different sample availabilities support our discussion. We discovered that, while these augmentation methods often contribute to a better performance than the original datasets, they can respond similarly to the original dataset according to the classification method in particular scenarios. We introduced a taxonomy for text data augmentation considering the most recent methods under both embedding and sentence categories. The main contributions of this article include the following.

1) A taxonomy devoted to text data augmentation, incorporating the last methods and their categories
2) Investigation of augmented methods under different scenarios (imbalanced data, small datasets, and different availability scales)
3) Examination of the advantages and disadvantages of modern classification methods and their relationship with augmented datasets

The rest of this article is organized as follows. Section II describes related work, detailing several recent and correlated papers. Section III presents the evaluated datasets, classification algorithms, and methods. In Section IV, we discuss the results of the proposed scenarios. Finally, Section V concludes this article.

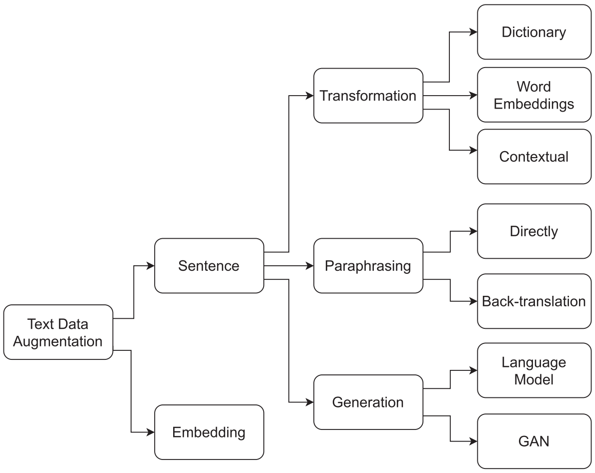

Fig. 1. Taxonomy of text data augmentation methods.

II. Related Work

Different methods of text data augmentation have been proposed over the years for different NLP tasks. However, in this work, we focus on applying data augmentation for text classification tasks, the field in which sentiment analysis traditionally lies. Although some of these methods can be used independently of the target class, some are designed explicitly for classification tasks, which we aim to discriminate samples among different categories. We can split text data augmentation methods into two primary approaches: Methods that rely on sentence manipulation and those that rely on model embedding manipulation. This work focuses on the former, which results in model-independent samples because they do not rely on a specific model's behavior, such as neural networks. Nevertheless, we briefly review some works proposed in the literature on the latter category. Fig. 1 presents the taxonomy of the different approaches proposed in the literature's most common methods. In sequence, we review the approaches of these different categories.

A. *Sentence*

Sentence-manipulation methods of augmentation directly transform, manipulate, and generate the sample text. Their input is the set of training samples, and the output is an the augmented set with the newly generated samples. The different methods proposed in this category can be further divided into three broad subcategories: Methods that transform the original sentences those that paraphrase the same way as the original sentence, and methods that generate entirely new samples based on the entire training set. Table I presents the related work classified by category, subcategory, and method strategy.

1) Transformation: The sentence-transformation subcategory often relies on simple lexical operations performed on a sentence's words to create variations in the original sentence. This approach involves methods that propose rule-based operations, such as lexical substitutions based on dictionaries and hand-crafted heuristics. Synonym replacement is one of the most traditional practices in this subcategory. Most strategies use a third-party thesaurus to find synonyms and related words for a

TABLE I
RELATED WORK CLASSIFIED BY CATEGORY, SUBCATEGORY AND METHOD'S APPROACH

Category	Subcategory	Strategy	References
Sentence	Transformation	Dictionary	[17], [18], [19] [20], [21], [22] [23], [24], [25] [26], [27]
		Word Embeddings	[28], [29]
		Contextual	[30], [31], [29]
	Paraphrasing	Directly	[20], [32], [33] [34], [35], [36]
		BT	[37], [20], [38] [39], [22], [25] [40], [41]
	Generation	Language model	[42], [7] [16], [43]
		GAN	[44]
Embedding			[45], [46], [47]

given target word [17]–[21], [43]. Zhang *et al.* [18], Mueller and Thyagarajan [19], and Wei and Zou [21] performed augmentation by replacing synonyms based on their similarity obtained from the WordNet thesaurus [48]. To decide which words to replace, they chose r of them to be replaced according to the probability distribution with parameter p given by $P[r] \sim p^r$. Then, from the synonym list, they selected the index s of the replacing synonym using a probability distribution with parameter q, where $P[r] \sim q^s$ [18]. Kolomiyets *et al.* [17] proposed a similar approach of replacing synonyms with WordNet to improve the model's performance, despite not explicitly naming this procedure augmentation. In addition to synonym replacement, Wei and Zou [21] proposed EDA, a technique composed of four operations: Synonym replacement, random insertion, random swap, and random deletion. The method relies on two parameters: α as the percentage of words in a sentence that should be changed, and n_{aug} for the number of generated sentences. This technique increased the accuracy of classification of benchmark datasets, including the SST-2 dataset. Coulombe [20] proposed a set of handcrafted transformations, such as verbal form contraction and expansion based on regular expressions and spelling error injection. That work also applied syntax-tree manipulation to create new sentences and textual noise injection among the proposed transformation techniques. In addition to noise injection, Kryscinski *et al.* [22] proposed rule-based methods of sentence negation, switching auxiliary verbs, named entities, pronouns, and number swapping. Rather than propose a fixed set of transformations, Ratner *et al.* [23] proposed a black-box approach for learning transformation functions. The evaluated transformation function candidates involve replacing specific word categories, swapping word order, and changing verbs around entities. These transform functions are optimized during training to produce augmented data that follow the original feature distribution. Similarly, Niu and Bansal [24] adapted AutoAugment [49] to discover effective perturbation policies on text automatically. Their search space involves random swapping, stop-word dropout, synonym replacement, grammar error injection, and a stammer. Xie *et al.* [25] proposed replacing less informative words in a sentence with the aim of not only maintaining its semantics, but also introducing a degree of variation. The authors used term frequency–inverse document frequency (TF-IDF) values to rank words and to select words with low values as being uninformative. This approach assigns a higher probability for replacing words with lower TF-IDF values. The choice of the substitute word in the vocabulary is made through sampling with the probabilities according to word frequency and IDF. This approach aims to retain keywords and replace uninformative words. The second strategy in this subcategory is to replace words in an embedding space with similar words, such as pretrained word vectors. Wang and Yang [28] proposed using neighboring words in the continuous representation in the embedding space. They selected the five most similar terms according to their cosine similarity by using a pretrained word2vec model [50], resulting in an augmented dataset five times larger than the original. This technique improved the performance of the topic classification of tweets. Instead of word2vec, Jiao *et al.* [29] used GloVe embeddings [51] to search for the nearest neighbors. They found the fifteen most similar terms and randomly sample them to replace the original words with a probability of 40%, repeating this process 20 times per sample. In addition to using word vectors to replace terms, they applied the third taxonomy strategy, which uses contextual models to predict the replacing terms. Using BERT, the authors proposed a masked language model, to fill a masked token in a sentence, thus repeating the same sampling process as in the word vector approach. Following this third strategy, Wu *et al.* [31] proposed a BERT-based method for labeled sentences called conditional BERT. The authors introduced a new pretraining task for BERT using a conditional masked language model. This novel task randomly masks tokens from the labeled sentence, and the objective is to predict the original tokens based on their contexts, such as the original task and its label. The proposed technique achieved better results than the compared methods on six text classification datasets, including three for sentiment analysis. Kobayashi [30] proposed using a bidirectional LSTM language model to make word predictions based on context. First, the LSTM model is pretrained WikiText-103 corpus [52], an English subset of Wikipedia articles. Second, the pretrained model is fine-tuned on the target-labeled dataset, introducing the label conditional constraint. To achieve this conditioning, the study altered the traditional language model objective to a label-conditional language model objective to assign word probabilities considering the sample's label. Similar to other proposed methods, the results that use augmentation outperformed the baselines on six datasets, including three sentiment-analysis datasets.

2) Paraphrasing: Another subcategory for manipulating sentences concerns paraphrasing techniques to obtain rephrased variations from the original samples. Two approaches for generating paraphrases explored in literature consist of using different techniques to directly paraphrase sentences and performing BT. Traditional paraphrase generation approaches usually require handcrafted rules [53]–[55]. Unlike sentence transformation, which concerns rule-based methods to perform word-level operations, this category relies mainly on sentence-level operations, often performed by neural models. The different strategies can be split into those that directly generate paraphrases and those that generate paraphrases through BT. Following the first approach, Coulombe [20] as well as Şahin and Steedman [32] proposed techniques for manipulating the dependency tree of sentences to create paraphrases. After parsing sentences and obtaining the dependency tree, they proposed a set of rule-based transformations such as the transition from passive to active verb form and replacing a noun or a nominal group by a pronoun. Şahin and Steedman [32] also proposed the rotation of sentence fragments around their roots on dependency trees to generate paraphrased sentences. Cho *et al.* [33] proposed a semisupervised learning pipeline containing a paraphrase generator based on a transformer model [56]. They based their paraphrase generation model on translation models but adapted it to use the same language as the source and target. Sokolov and Filimonov [34] used a similar approach of using a neural machine translation model to generate paraphrases and applied it to intent classification and named entity recognition. Jolly *et al.* [35] also applied paraphrase generation to the intent classification task but used an approach of learning to generate the original utterance given its expected output. Then, they obtain various paraphrases by sampling token by token. Huang and Chang [36] proposed a syntactically controlled paraphrase generator (SynPG), an encoder-decoder model that learns both syntactic and semantic embeddings. SynPG can be further used to generate syntactically controlled paraphrases to augment datasets for text classification. The authors reported an improvement in adversarial attack robustness using their method, although it did not improve the accuracy before attacking compared with the baseline. The evaluated datasets included SST-2, a sentiment-analysis dataset. The second approach is the generation of paraphrases through BT, which is one of the most widespread methods for augmenting datasets for NLP tasks [20], [22], [25], [37], [39]–[41]. With the development of translation systems, implementing and including NLP pipelines has become easier. Previous work demonstrated that those paraphrased samples improve the model's results on different NLP tasks, including text classification [39]. BT works by generating a new sample for a given input sentence x by translating it into an intermediate language and making the round-trip back to the source language to obtain the augmented sample $\hat{x}$. Different methods have been proposed to increase the diversity of BT because the traditional translating system output is obtained deterministically through beam search. Kryscinski *et al.* [22] used different intermediate languages to obtain diverse augmented samples. Specifically, they used French, German, Chinese, Spanish, and Russian as intermediate languages, resulting in variations when translating back to their source language: English. Other studies included a variation of the decoding step to produce diverse outputs. Xie *et al.* [25] used random sampling decoding with a temperature hyperparameter instead of a beam search. They used only English/French models and achieved better results than the baseline on different text classification datasets, including Yelp and Amazon Reviews for sentiment analysis. Edunov *et al.* [38] evaluated different decoding strategies and demonstrated that they outperformed the baselines, beams, and greedy search. Another approach for performing BT, adopted by Coulombe [20] and Kryscinski *et al.* [22], used cloud vendor APIs to perform the translation. This approach has the advantage of being scalable and being easy to implement because all infrastructures for running the translation models is delegated to the vendor. However, this approach provides less control over the translation generation to the practitioner and can be more expensive, depending on the scenario.

3) Generation: Sentence-generation strategies have gained popularity recently because of advances in neural text generation. Different from the approaches explored until now, which modified the original sentences on word- or sentence-level, sentence-generation approaches are intended to create entirely new samples based on the entire original set. With the advances in autoregressive language models such as GPT [57], GPT-2 [58], and XLNet [59] pretrained on an enormous amount of data, these models could leverage transfer learning to provide breakthrough results. Methods for generating completely new samples can be split into two strategies: Those based on language models and those based on generative adversarial networks (GANs) [60]. By following the first approach, Anaby-Tavor *et al.* [42] proposed language-model-based data augmentation (LAMBADA), a technique that leverages a pretrained GPT-2 model to generate high-quality and diverse text on the target's dataset domain. Their technique follows a semisupervised approach, in which they use a generative model and a classification model to generate pseudolabels. Furthermore, they fine-tune the pretrained language model and the classifier on the target dataset. Then, the technique keeps synthesizing new samples by concatenating the label and the associated sentences with bootstrap to initialize the language model generation. The study evaluated the technique on small subsets of three datasets, with sizes varying from 5 to 100 samples per class, demonstrating a performance gain obtained by using it. Similarly, Abonizio and Barbon [7] proposed PREDATOR, a two-module technique composed of a generator module with an autoregressive language model in its kernel and a filter module with a classification model. The language model fine-tuning and text generation procedures vary from LAMBADA, where the language model fine-tuning does not involve concatenating the label but only the sentence texts. The text generation procedure is also performed by concatenating examples from the target class separated by a special token and allowing the model infer the continuation. With the fine-tuned generator, the technique generates new samples and pseudolabels with the filter module, which is also responsible for accepting only samples with classification confidence above a threshold. Another difference from previous techniques is the use of compressed models [61], [62], which makes the approach computationally cheaper. The experiments were conducted on a low-data regime with different subsamples from three datasets, improving every scenario's performance, including that of the SST-2 dataset. Ollagnier and Williams [43] followed a similar method to handle the class imbalance problem with data augmentation but used a CNN-LSTM architecture instead of a transformer for its language model. Another difference in their approach is the lack of a filtering step to control the quality of the synthesized examples. Additionally, because the proposed language model is trained on the target dataset, it cannot introduce out-of-vocabulary words, which might improve the model's generalization. Transfer learning with pretrained models on large amounts of data has the advantage of including novel terms that the target dataset might not include. However, this method is computationally cheaper than using large transformer models. Moreover, by tackling the challenge of class imbalance, Ibrahim *et al.* [26] applied augmentation methods to solve the problem of toxic comment identification. In addition to synonym replacement, they removed duplicated words and randomly masked 20% of the original words in the sentences. These perturbations lead to an increase in the F1-score, improving the performance of the minority class. In addition to autoregressive language models, Kumar *et al.* [16] evaluated the use of the autoencoder [63] and sequence-to-sequence [64] models. The authors evaluated different text generation approaches by including and omitting the label in the generation process and predicting only words or a continuous chunk of words in the sequence-to-sequence experiment. Similar to other studies, the study conducted experiments by following a low-data regime, with subsamples from the original datasets, including the SST-2 benchmark. The results showed that the sequence-to-sequence approach–specifically BART–was superior on average in terms of classification accuracy and superior to the BT and autoencoder for maintaining the original semantic fidelity. Gupta [44] proposed using GANs to augment data for sentiment analysis. The author first pretrained the generative model on a larger external dataset from a related task, and then conducted a fine-tuning step on the target small dataset. In addition to improving the accuracy of the models, the author demonstrated a visualization technique that the synthetic examples generated after a real data distribution.

B. Embedding

The embedding manipulation category varies from the sentence-manipulation category in that it does not operate on the text itself but on the representation vectors of sentences in the model's embedding space. The methods of this category depend more on the model's architecture because they assume how sentences are represented internally and are usually based on neural networks. Embedding-based methods are less explored than sentence-based methods, but initial work has made advances in the field. Chen *et al.* [47] proposed MixText, an augmentation technique based on the interpolation of the intern representation of sentences and imputation of labels for unlabeled data using a semisupervised approach. The authors obtained sentence embeddings from a combination of different layers of BERT. Kumar *et al.* [46] also proposed a featured space data augmentation technique based on evaluating different perturbations to representation learning using different feature extractors, such as bidirectional LSTM and BERT. Another method of combining embeddings was proposed by Guo *et al.* [45]. Based on Mixup [65], a technique initially intended for image data by interpolating pixels from images in a mini-batch, the authors proposed an adaptation for applying to text data. They had two approaches: Interpolation between word and sentence embeddings, with each method leading to better results for the different scenarios. They evaluated the method on five datasets, three of which were used for sentiment analysis. From the current state of the art, we identified a lack of studies aimed at understanding the effects of text data augmentation and how the classification method can handle data quality drawbacks, especially for sentiment-analysis tasks.

III. Methodology

This article evaluated different strategies of text data augmentation methods on seven datasets for sentiment analysis and related tasks using four different classifiers. To represent each different augmentation strategy, we selected a representative from each of the three subcategories of sentence-manipulation methods. The following sections explain the methodology procedures for the experiments.

A. Datasets

Because this article focused on sentiment analysis, we selected benchmark datasets following sentiment analysis and related tasks to evaluate the different scenarios. The datasets have different class cardinality and distribution and originate from diverse domains. Table II presents the summary statistics of the datasets grouped by the different evaluated scenarios.

B. Classifiers

We conducted the experiments using seven classification algorithms to evaluate different families of classifiers: Deep learning approaches, current state-of-the-art transformer-based models, and traditional approaches. We selected CNNs [74], LSTM [75], and GRUs to represent common deep learning approaches. They are initialized using fixed pretrained GloVe embeddings [51] of dimension 200. We based CNN model architectures on Kim [74] using filters of sizes 3, 4, and 5 that were max-pooled, concatenated, and followed by a fully connected layer of size 150 and used a dropout regularization with a probability of 0.3. Single-layer LSTM models have a hidden size of 150 and dropout probability of 0.3, and they are bidirectional. We implemented GRU models with the same hyperparameters as LSTM. We trained CNN, LSTM, and GRU for 15 epochs with early stopping, with an initial learning rate of $1\mathrm{e}{-3}$ using Adam as an optimizer [76], [77] and gradient clipping [78]. We developed all implementations using PyTorch [79]. Representing the current state-of-the-art transformer-based models that achieve the most recent progress on numerous NLP tasks, including sentiment analysis, we selected BERT [63] and ERNIE [80], [81]. BERT and ERNIE represent a breakthrough in NLP tasks, outperforming previous methods by a large margin, with the latter being the current state-of-the-art method on the GLUE benchmark [82] at the time of this writing. However, this high accuracy is computationally expensive because these large methods require a higher amount of resources to be trained. We developed the implementations using the transformers library [83] and PyTorch [79]. We fine-tuned the models on the target classification dataset over three epochs with a learning rate of 5e–5, as recommended in the original article [63]. To represent traditional methods, we selected a variation of SVM [84] proposed by Çöltekin and

TABLE II
EXPLORED SCENARIOS WITH DETAILED INFORMATION ABOUT DATASETS, CLASSES, SIZE, BALANCE, AVERAGE SENTENCE LENGTH, AND AUTHORS

Scenario	Dataset name	Number of classes	Dataset size	Majority / minority class ratio	Average sentence length	
Imbalanced	Financial PhraseBank	3	2,264	4.4x	23	[66]
	IEMOCAP	6	9,292	2.6x	14	[67]
	SEMAINE	3	5,627	2.7x	14	[68], [69]
Small	Ethos	2	753	1.7x	24	[70]
	Amazon	2	1,000	1.0x	12	[71], [72]
	Yelp	2	1,000	1.0x	12	[71], [72]
Data availability	SST-2	2	9,613	1.1	19	[73]

Rama [85], which achieved results competitive with deep neural models. They proposed a linear SVM with TF-IDF features considering both character n-grams and word n-grams, with a maximum n-gram size of 6 for character n-grams and 4 for word n-grams. We also selected traditional RF, a robust ensemble method widely adopted for various classification tasks [86]. We developed SVM and RF models by using Scikit-learn [87] and its default set of hyperparameters.

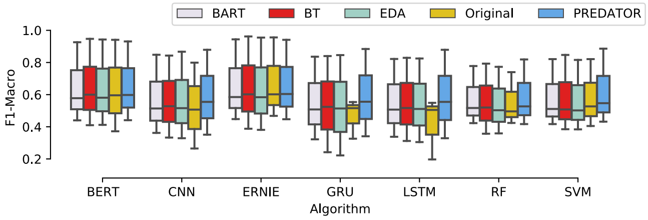

Fig. 2. Comparison of F1-Macro among LSTM, GRU, CNN, RF, BERT, ERNIE, and SVM classification methods induced from original and augmented datasets using EDA, BT, BART, and PREDATOR (sorted by F1-Macro) with three different imbalanced datasets (Financial PhraseBank, IEMOCAP, SEMAINE).

C. Scenarios

To understand the impact of data augmentation techniques on a wide variety of scenarios, we conducted three different experiments to cover a large set of real-world problems. We selected augmentation methods for comparing different subcategories and strategies: EDA for representing sentence-transformation methods, BT for sentence paraphrasing methods, and PREDATOR and BART for representing the most recent sentence-generation methods. For all augmentation methods, we increased the original dataset three times to ensure a fair comparison. Specifically for EDA, we used the original publicly available implementation[1] and the recommended hyperparameters. For BT, we used the translation models proposed by Edunov *et al.* [38], a transformer model made publicly available[2] and, rather than use traditional beam search for decoding, we adapted the decoding step to use top-k sampling with $k = 10$, as proposed by Edunov *et al.* [38]. We used the publicly available PREDATOR implementation,[3] with the augmentation ratio hyperparameters as 3, to generate new samples until attaining thrice the original dataset size and retaining the other hyperparameters by default. The BART augmentation uses the pretrained sequence-to-sequence model to predict a span of 40% of the words in the sentence, as described by Kumar *et al.* [16], using the transformers library [83] implementation of BART with beam-search decoding with a beam size of 5. We will formulate the code for reproducing our experiments publicly available.[4] Our first evaluated scenario was the imbalanced class distribution setup, which poses a challenge for classification algorithms. In this setup, all classifiers applied augmentation methods when training the original datasets. We applied the augmentation methods to balance the class distributions, thus making every class have the same number of examples and augmenting the dataset by three times, that is, in the final dataset, every class contains three times the size of the original majority class. For the second setup, we evaluated small datasets, each containing at most 500 samples per class. These small datasets posed a new challenge for classifiers because we have few examples to learn generalizable patterns. Finally, we conducted experiments on the SST-2 dataset, a standard sentiment-analysis benchmark, but with different available samples. We trained classifiers by using 5%, 10%, 50%, and 100% of the samples and augmented each breakpoint three times to study the impact of augmentation compared to the addition of real data.

IV. RESULTS

The following sections present and discuss each scenario and its results.

A. Imbalanced Datasets

In this scenario, we applied seven different classification algorithms (LSTM, GRU, CNN, RF, BERT, ERNIE, and SVM) over 12 different combinations based on three original imbalanced multiclass sentiment-analysis datasets (Financial PhraseBank, IEMOCAP, and SEMAINE). We used F1-Score macroaveraged (F1-Macro) instead of the traditional accuracy measures to make a fair comparison of methods considering the class distribution because the accuracy may mask poor performance on minority classes. EDA, BT, BART, and PREDATOR created augmented versions of the original versions, thus balancing the dataset to discuss the improvements provided by these methods in this constraint. BERT and ERNIE were the classification methods with higher average performance, whereas the other competitors (LSTM, GRU, CNN, RF, and SVM) achieved similar F1-Macro values. As Fig. 2 shows, the boosting performance provided by

[1] [Online]. Available: https://github.com/jasonwei20/eda_nlp
[2] [Online]. Available: https://pytorch.org/hub/pytorch_fairseq_translation/
[3] [Online]. Available: https://github.com/hugoabonizio/predator
[4] [Online]. Available: https://github.com/hugoabonizio/tai-sentiment

TABLE III
Relative Performance of Seven Classifiers With Augmented Datasets From Four Different Methods Over Imbalanced Datasets

Method	Classifier	Dataset		
		Financial	IEMOCAP	SEMAINE
EDA	BERT	1.00	1.11	0.98
	CNN	1.08	1.24	**1.02**
	ERNIE	1.00	0.82	0.97
	GRU	**1.53**	0.68	1.00
	LSTM	1.50	**1.55**	1.01
	RF	1.04	0.85	1.01
	SVM	0.99	0.95	0.95
BT	BERT	1.01	1.10	1.01
	CNN	1.05	1.26	1.0
	ERNIE	1.01	0.83	1.00
	GRU	**1.52**	0.74	1.01
	LSTM	1.51	**1.59**	1.02
	RF	1.07	0.84	**1.05**
	SVM	1.03	0.95	0.96
PREDATOR	BERT	0.99	1.18	1.00
	CNN	1.10	1.32	1.09
	ERNIE	0.98	0.96	1.00
	GRU	**1.60**	1.05	1.08
	LSTM	**1.60**	**1.67**	**1.10**
	RF	1.11	0.98	1.06
	SVM	1.08	1.07	1.04
BART	BERT	0.98	1.18	0.97
	CNN	1.06	1.37	1.01
	ERNIE	0.99	0.96	0.97
	GRU	**1.51**	0.99	0.98
	LSTM	1.50	**1.72**	1.00
	RF	1.05	1.00	**1.04**
	SVM	1.00	1.03	0.97

We have highlighted best improvements and worst results per dataset in bold and underline, respectively.

the augmentation methods over a significant number of classifiers is clear. In addition to the average improvement, most augmented models resulted in higher values for the lower bounds. PREDATOR stood out from the other techniques regardless of the classifier applied.

BT was the second-best augmentation method, boosting LSTM, GRU, CNN, RF, ERNIE, and BERT. However, the dataset augmented by BT reduced the induction capacity of the SVM classifier. The EDA method obtained similar results, in which SVM and BERT classifiers achieved better performance (F1-Macro) when created over the original dataset instead of using an augmented dataset. By observing the relative performance of the original dataset presented in Table III, LSTM trained on PREDATOR augmented dataset provided the most significant improvements for all imbalanced datasets. Classification improved by 60% in Financial PhraseBank, 67% with IEMOCAP, and 10% with SEMAINE. Note that ERNIE and BERT over the Financial PhraseBank did not improve by using the PREDATOR method.

The models based on SVM demonstrated the least increase, especially considering models augmented with EDA, which degraded their performance in all evaluated scenarios. This poor performance might indicate that data augmentation through sentence manipulation is not the best option for the traditional methods. Instead, a technique such as SMOTE with TF-IDF representation might be more appropriate [88]. However, more experiments need to be conducted for this comparison. The original datasets provided a better training set for SVM, considering the augmented version created using EDA. For example, when using the IEMOCAP dataset, the reduction was approximately 5% of the original dataset capacity when using the EDA augmentation method. Also, the classification performance of the GRU reduced by 32% when using EDA as an augmentation method on the IEMOCAP dataset. BT did not achieve the topmost improvements, but, remarkably, this method delivered improvements on most classifiers and scenarios. The IEMOCAP dataset demonstrated the greatest advantage of augmentation, reaching a 67% increase in the F1-Macro. This dataset has a higher class cardinality evaluated in our experiments, which might imply that imbalanced multiclass datasets with higher cardinality are an appropriate scenario for applying data augmentation techniques. We evaluated the results based on statistical analysis grounded on the nonparametric Friedman test to determine any significant differences between the augmented datasets with different methods and the original data. We used the *post hoc* Nemenyi test to infer statistically significant differences. Fig. 3 shows significant differences between populations. In particular, we assumed no significant differences between PREDATOR and BT or between BT, BART, original, and EDA. All other differences were statistically significant. In other words, PREDATOR and BT achieved superior results that were statistically different from the original data. By contrast, EDA, BART, and the original dataset obtained statistically similar results.

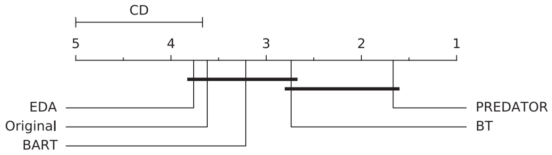

Fig. 3. Nemenyi *post hoc* test (significance of $\alpha = 0.05$ and critical distance of 1.331) considering F1-Macro obtained from original, EDA, BART, BT, and PREDATOR (sorted by F1-Macro) methods over imbalanced datasets (5 populations with 12 paired samples).

Earlier work found that EDA might not yield substantial improvements when using pretrained models [21] (such as BERT and ERNIE) and may even harm its performance [27]. However, our findings show that EDA had no statistically significant difference from BT and BART in these datasets, although it did not achieve the best results.

B. Small Datasets

Small datasets are extremely delicate when supporting sentiment classification. To evaluate the impact of augmentation methods in this scenario, we evaluated three different binary classification algorithms with a reduced number of samples. The ERNIE and BERT classifiers obtained the highest performance, followed by the SVM models. By confirming the theoretical foundations, the deep learning methods (LSTM, GRU, and CNN) achieved similar low performance when trained with reduced datasets. This corroborates with the common sense of the data-hungriness nature of deep learning models [89], and Anaby-Tavor *et al.* [42] found the superiority of SVM over LSTM on small datasets. Meanwhile, the BERT and ERNIE

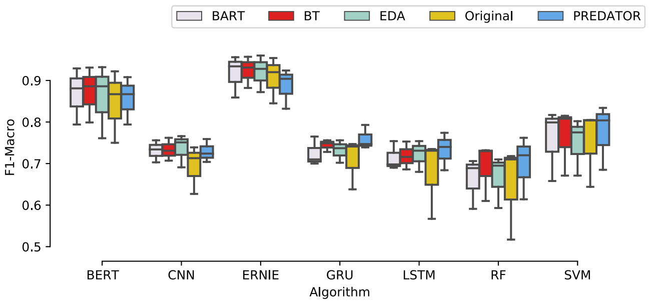

Fig. 4. Comparison of F1-Macro among LSTM, CNN, BERT, and SVM classification methods induced from original and augmented datasets using EDA, BT, and PREDATOR with three different small datasets (Ethos, Amazon, and Yelp).

TABLE IV
RELATIVE PERFORMANCE OF SEVEN CLASSIFIERS WITH AUGMENTED DATASETS FROM THREE DIFFERENT METHODS OVER SMALL DATASETS

Method	Classifier	Dataset		
		Ethos	Amazon	Yelp
EDA	BERT	1.01	1.01	1.02
	CNN	**1.04**	1.10	**1.05**
	ERNIE	1.01	1.03	1.01
	GRU	1.01	1.10	0.99
	LSTM	1.03	**1.20**	0.99
	RF	0.97	1.15	1.00
	SVM	1.00	1.04	0.96
BT	BERT	1.01	1.07	1.02
	CNN	**1.03**	1.13	**1.03**
	ERNIE	1.00	1.04	1.01
	GRU	1.01	1.17	0.98
	LSTM	**1.03**	**1.21**	0.97
	RF	1.02	1.18	**1.03**
	SVM	1.00	1.04	1.01
PREDATOR	BERT	0.98	1.06	1.00
	CNN	1.03	1.12	1.02
	ERNIE	0.97	0.99	0.98
	GRU	**1.06**	1.16	1.01
	LSTM	**1.06**	**1.21**	1.01
	RF	1.00	1.19	**1.07**
	SVM	1.04	1.06	1.00
BART	BERT	1.01	1.06	1.02
	CNN	1.02	1.12	**1.03**
	ERNIE	1.00	1.02	1.02
	GRU	1.02	1.10	0.96
	LSTM	**1.03**	**1.22**	0.95
	RF	0.96	1.14	0.99
	SVM	1.01	1.02	0.99

Best improvements and worst results per dataset were marked in bold and underline, respectively.

models are deeper and have more parameters, but their transfer learning pipeline fits well on small datasets. Fig. 4 shows the F1-Macro for all evaluated scenarios.

Table IV presents the relative performances for all scenarios, with the better improvements per dataset highlighted in bold and the deterioration of performance caused by the augmentation methods underlined. In a significant number of the combinations, the augmentation methods improved the classifier performance. BT and PREDATOR improved almost all datasets and classifiers similar to the experiment with imbalanced datasets. PREDATOR achieved the best improvement with LSTM over the Amazon dataset: 21%. However, in the same dataset, the BERT classifier obtained a more predictive model using the original dataset.

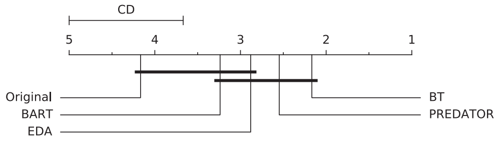

Fig. 5. Nemenyi *post hoc* test (significance of $\alpha = 0.05$ and critical distance of 1.331) considering F1-Macro obtained from original, EDA, BT, BART, and PREDATOR methods over small datasets (5 populations with 21 paired samples).

EDA delivered the best improvement for the Yelp dataset when using the CNN classifier: 5%. However, as previously seen in the imbalanced scenario, the combination of SVM with EDA-augmented datasets obtained results worse than the original datasets with 99%, 95%, and 95% relative performances for Amazon, Ethos, and Yelp, respectively. Another important conclusion is that even the best-performing models based on BERT and ERNIE benefited from augmentation in most scenarios. For small datasets, we carried out the same statistical procedure for the imbalanced scenario. We applied the nonparametric Friedman test to determine any significant differences between the augmented datasets with different methods and the original data. Furthermore, we used the *post hoc* Nemenyi test for statistically significant differences. As Fig. 5 shows, the differences between populations are significant. We noticed no significant differences within PREDATOR, BT, BART, and EDA, nor with BART, EDA, and original. All other differences were statistically significant. In this case, only BT and PREDATOR overcame the results from the original small data with a statistical difference.

C. *Impact of Augmentation on Different Sample Availability*

We evaluated the impact of augmentation with different data availabilities on a single dataset using the best classifiers of each class in the previous scenario (SVM, LSTM, and ERNIE). We evaluated the tradeoff between the less availability and improvement provided by the augmentation methods. SVM, LSTM, and ERNIE supported this evaluation to provide insights from two different text augmentation categories. The aim was to obtain insights from the improvements provided by augmenting a dataset when dealing with different data amounts. We created different splits from the SST-2 dataset to accomplish this task, ranging from 5%, 10%, 50%, and 100%. Each augmentation lead to thrice the original value, reaching sizes equivalent to 15%, 30%, 150%, and 300% considering the original dataset size. Classification results achieved by SVM, LSTM, and ERNIE fit the hypothesis that more data leads to a better induction of predictive models. Fig. 6 shows improvement provided by increasing the number of samples from the original and augmented datasets. However, by comparing both classifiers dealing with the same problem, observing particularities related to the augmented methods and the obtained predictive performance was possible.

We observed an F1-Macro improvement of 22% from 0.67 (5% of the total dataset, 480 samples) to 0.82 when using the total available (9613 samples), with the SVM model using the original dataset. The augmentation methods yielded different

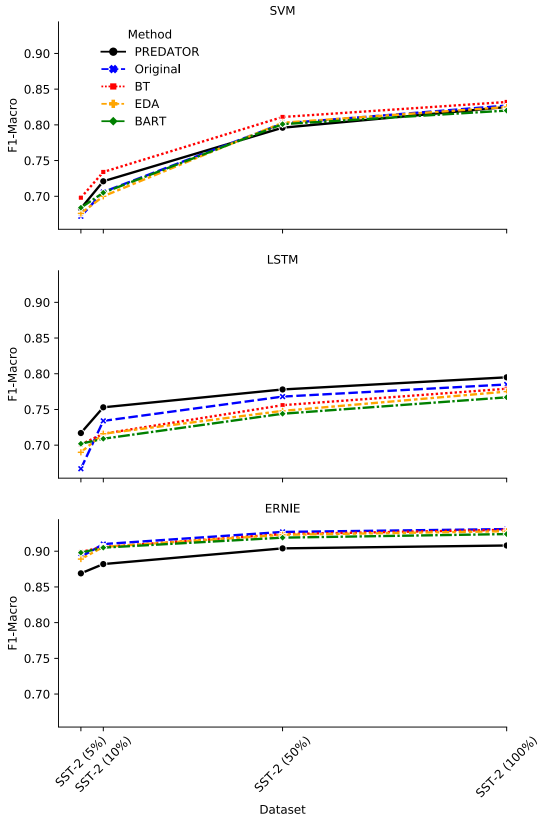

Fig. 6. Comparison of F1-Macro among SVM, LSTM, and ERNIE classification methods induced from original and augmented datasets using EDA, BT, BART, and PREDATOR with four different numbers of available samples (5%, 10%, 50%, and 100%) from SST dataset (a total of 9613 samples).

improvements. A notable result was obtained using the BT. This method improved by 18%, starting from the F1-Macro of 0.70 (5%) to 0.82 (100%). The EDA-augmented dataset presented results similar to those of the original dataset. PREDATOR obtained better results than EDA and the original but smaller than BT. Considering the efficiency of augmentation, the positive impact of addressing a small number of available samples is clear. By taking an SVM model trained with an augmented dataset from BT as an example, we obtained these improvement rates of approximately 4.4% (5% of the total), 4.2% (10% of the total), 1.2% (50% of the total), and 0.6% (100% of the total). Thus, the contribution of the augmented datasets was reduced as the number of available samples increased. PREDATOR was the best augmentation method for LSTM classifier for all dataset sizes (5%, 10%, 50%, and 100%). The most significant F1-Macro improvements occurred when augmenting the smallest datasets, from 0.66 (original) to 0.71 and from 0.73 (original) to 75% for 5% and 10% of SST-2, respectively. BT and EDA could deliver a better augmented dataset than the original only for 5% of SST-2. The ERNIE classifier presented a similar tendency of improvement with respect to the increase in available samples. However, the improvements provided by the augmented datasets were reduced compared with those of the SVM classifier. This flat classification improvement occurred with the original dataset, which obtained 0.89, 0.91, 0.92, and 0.93 for the F1-Macro with 5%, 10%, 50%, and 100%, respectively. As the most efficient augmentation method, BT obtained the F1-macro values of 0.90 (5%), 0.91 (10%), 0.92 (50%), and 0.93 (100%). However, using augmentation methods with ERNIE did not result in any significant improvement. When training ERNIE with 10%, 50%, and 100% of the available samples, the augmentation methods provided no improvements. Instead, PREDATOR delivered datasets that lead to building a less accurate classifier. EDA and BART did not obtain improvements in the shorter dataset (5%). Again, BT was the best augmentation method.

D. *Recommendations for Practitioners*

After all experiments with different data scenarios and classifier combinations with texts from diverse source domains, we can draw conclusions and recommendations for applying the augmentation methods for text classification problems. First, the statistical results indicated that using augmentation techniques improves the classification performance on limited data, especially when using the most advanced methods. Thus, facing a real-world problem where labeled data are scarce, including a data augmentation step in the training pipeline, tends to contribute to models with a better generalization capacity. Another recommendation is to use data augmentation to ease the class imbalance problem because all evaluated methods obtained better results than oversampling in most scenarios. Even simpler methods such as EDA, which is cheaper than neural methods, might improve imbalanced scenarios, although not expressively. BT is a good default strategy, considering its widespread adoption in related studies. This strategy steadily leads to better-augmented datasets, especially when using a sampling decoding strategy such as top-k sampling. Furthermore, the ease of implementation using vendor translation APIs eases the entry barrier of this strategy. However, when using vendor-translation APIs, we do not control the decoding strategy, so a common strategy is to use different intermediate languages to achieve variations. Nonetheless, sentence-generation strategies, such as PREDATOR and BART, may lead to better results than BT. Both PREDATOR and BART led to higher performing models, but no method was found suitable for all scenarios. However, because they have a training phase before the generation, they tend to be more expensive than unsupervised methods that transform or paraphrase sentences. Although it was not the purpose of this work to thoroughly evaluate the computational cost of different methods, each strategy has different resource requirements, which need to be considered when choosing the method. When using classifiers based on large pretrained models, the data augmentation method can slightly improve the generalization of the model. However, pretrained models deal well with data scarcity and imbalanced scenarios out of the box, and other training strategies may lead to better results. Meanwhile, with traditional classifiers, such as SVM and RF, sentence-manipulation augmentation may not be the choice for improving the model's performance, and sentence-generation strategies may add the variability needed to improve generalization, but the efficiency of augmentation methods depends on the dataset. Considering the application of augmentation in languages other than English and low-resource

TABLE V

Overview of Text-Augmented Methods Over Different Scenarios With Respective Advantages and Drawbacks

Scenario	Classifier	EDA	BT	PREDATOR	BART
Imbalanced	Traditional	-	-	✓	✓
	Deep learning	✓	✓	✓	✓
	Transformers	-	-	-	-
Small	Traditional	-	✓	✓	✓
	Deep learning	✓	✓	✓	✓
	Transformers	✓	✓	✓	✓
Advantages		Computationally cheap method	Stable improvements in data quality and easy pipeline setup	May achieve higher improvements	May achieve higher improvements
Drawbacks		May introduce harmful variations leading to inferior performance than the original set	Expensive to translate several documents when using clound vendor APIs	Mildly expensive methods and may require tuning	Very expensive method

Recommended methods are marked with ✓.

languages, sentence-transformation methods—such as those based on dictionaries—need this kind of resource present in the target language, such as a Portuguese WordNet [90]. Furthermore, sentence-generation methods based on pretrained language models require using these models as inferred. This type of tool is harder to access in low-resource languages, whereas translation models or vendor APIs might be easier to find [91]. Thus, BT strategies have more potential for multilanguage and low-resource language setups. We summarized the main advantages and drawbacks of the combined algorithms in Table V. Furthermore, we recommended (marked as ✓) the augmentation methods for particular sentiment-analysis scenarios.

Because data augmentation is performed during training, the inference cost is not impacted, so its addition to text classification pipelines may only improve the model's generalization without harming its computational performance.

V. Conclusion

With the increasing popularity of classification models in sentiment analysis, some data limitations require effective data augmentation methods. In this article, we provide a comprehensive discussion of sentiment-analysis data augmentation methods under various scenarios. First, we introduced a taxonomy that serves as a classification framework for text data augmentation approaches. Next, the experiments evaluated different augmentation methods, based on their capacities, to improve model performance on target tasks under different scenarios. The study discussion comprises imbalanced scenarios, reduced datasets, and the relation of some available samples and improvements led by an augmented dataset. Furthermore, we evaluated the efficiency of the augmented datasets using different classification algorithms. LSTM boosted their result (F1-Macro) by 67% after balancing the IEMOCAP dataset using the PREDATOR method. BERT and ERNIE worked the best with small datasets, primarily with BT augmentation boosting the classifier performance by 21% for BERT. An extremely competitive SVM resulted in more significant improvements from the original dataset within different levels of available data. The contribution of the BT augmentation method for all classifiers and PREDATOR for imbalanced scenarios is remarkable. GRU obtained a similar contribution to LSTM, but with less effective augmentation. RF took advantage of augmented datasets in small and imbalanced scenarios, mainly when using BART for the latter scenario. Meanwhile, EDA and BART resulted in slight improvements for particular cases, but their augmented datasets provided less induction potential than the original dataset in several experiments. The future of text data augmentation is promising. Presently, data augmentation methods cannot overcome all issues in a text mining dataset. With particular sentiment-analysis cases, selecting a classification method can interfere more than the augmentation itself. Future work can also evaluate the impact of augmentation on mitigating the model's societal biases and the interpretation capacity of augmented models.

References

[1] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 102–107, Mar./Apr. 2016.

[2] D. M. E.-D. M. Hussein, "A survey on sentiment analysis challenges," *J. King Saud Univ.- Eng. Sci.*, vol. 30, no. 4, pp. 330–338, 2018.

[3] S. Shayaa *et al.*, "Sentiment analysis of big data: Methods, applications, and open challenges," *IEEE Access*, vol. 6, pp. 37807–37827, 2018.

[4] A. M. Almeida, R. Cerri, E. C. Paraiso, R. G. Mantovani, and S. B. Junior, "Applying multi-label techniques in emotion identification of short texts," *Neurocomputing*, vol. 320, pp. 35–46, 2018.

[5] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdiscip. Rev., Data Mining Knowl. Discov.*, vol. 8, no. 4, 2018, Art no. e1253.

[6] E. Cambria, Y. Li, F. Z. Xing, S. Poria, and K. Kwok, *SenticNet 6: Ensemble Application of Symbolic and Subsymbolic AI for Sentiment Analysis*. New York, NY, USA: Assoc. Comput. Machinery, 2020, pp. 105–114.

[7] H. Q. Abonizio and S. Barbon Junior, "Pre-trained data augmentation for text classification," in *Intelligent Systems*, R. Cerri and R. C. Prati, Eds. Cham, Switzerland: Springer, 2020, pp. 551–565.

[8] K. M. Yoo, Y. Shin, and S.-G. Lee, "Data augmentation for spoken language understanding via joint variational generation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, no. 1, pp. 7402–7409.

[9] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: A methodology review," *J. Biomed. Informat.*, vol. 35, no. 5, pp. 352–359, 2002.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer, 2006.

[11] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014.

[12] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[13] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[14] S. Niu, Y. Liu, J. Wang, and H. Song, "A decade survey of transfer learning (2010–2020)," *IEEE Trans. Artif. Intell.*, vol. 1, no. 2, pp. 151–166, Oct. 2020.

[15] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, Jul. 2019, Art. no. 60.

[16] V. Kumar, A. Choudhary, and E. Cho, “Data augmentation using pre-trained transformer models,” in *Proc. 2nd Workshop Life-long Learn. Spoken Lang. Syst.*, Dec. 2020, pp. 18–26.

[17] O. Kolomiyets, S. Bethard, and M.-F. Moens, “Model-portability experiments for textual temporal analysis,” in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics., Human Lang. Technol., Short Papers*, 2011, pp. 271–276.

[18] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 649–657.

[19] J. Mueller and A. Thyagarajan, “Siamese recurrent architectures for learning sentence similarity,” in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 2786–2792.

[20] C. Coulombe, “Text data augmentation made simple by leveraging NLP cloud APIS,” 2018. [Online]. Available: https://dblp.uni-trier.de/rec/journals/corr/abs-1812-04718.html?view=BibTeX

[21] J. Wei and K. Zou, “EDA: Easy data augmentation techniques for boosting performance on text classification tasks,” in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process*, Nov. 2019, pp. 6382–6388.

[22] W. Kryscinski, B. McCann, C. Xiong, and R. Socher, “Evaluating the factual consistency of abstractive text summarization,” in *Proc. Conf. Empirical Methods Natural Lang. Process*, Nov. 2020, pp. 9332–9346. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-main.750

[23] A. J. Ratner, H. R. Ehrenberg, Z. Hussain, J. Dunnmon, and C. Ré, “Learning to compose domain-specific transformations for data augmentation,” in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 3239–3249.

[24] T. Niu and M. Bansal, “Automatically learning data augmentation policies for dialogue tasks,” in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process*, Nov. 2019, pp. 1317–1323.

[25] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, “Unsupervised data augmentation for consistency training,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., 2020, pp. 6256–6268.

[26] M. Ibrahim, M. Torki, and N. El-Makky, “Imbalanced toxic comments classification using data augmentation and deep learning,” in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl.*, 2018, pp. 875–878.

[27] H. Tayyar Madabushi, E. Kochkina, and M. Castelle, “Cost-sensitive BERT for generalisable sentence classification on imbalanced data,” in *Proc. 2nd Workshop Natural Lang. Process. Internet Freedom., Censorship, Disinf. Propaganda*, Nov. 2019, pp. 125–134.

[28] 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 *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2015, pp. 2557–2563.

[29] X. Jiao *et al.*, “TinyBERT: Distilling BERT for natural language understanding,” in *Proc. Findings Assoc. Comput. Linguistics., Empirical Methods Natural Lang. Process*, Nov. 2020, pp. 4163–4174.

[30] S. Kobayashi, “Contextual augmentation: Data augmentation by words with paradigmatic relations,” in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics., Hum. Lang. Technol., (Short Papers)*, Jun. 2018, pp. 452–457.

[31] X. Wu, S. Lv, L. Zang, J. Han, and S. Hu, “Conditional BERT contextual augmentation,” in *Computational Science - ICCS 2019*, J. M. F. Rodrigues, J. J. Dongarra, and P. M. Sloot *et al.*, Eds. Cham, Switzerland: Springer, 2019, pp. 84–95.

[32] G. G. Şahin and M. Steedman, “Data augmentation via dependency tree morphing for low-resource languages,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Oct./Nov. 2018, pp. 5004–5009.

[33] E. Cho, H. Xie, and W. M. Campbell, “Paraphrase generation for semi-supervised learning in NLU,” in *Proc. Workshop Methods Optimizing Evaluating Neural Lang. Gener.*, Jun. 2019, pp. 45–54.

[34] A. Sokolov and D. Filimonov, “Neural machine translation for paraphrase generation,” 2020, *arXiv:2006.14223*.

[35] S. Jolly, T. Falke, C. Tirkaz, and D. Sorokin, “Data-efficient paraphrase generation to bootstrap intent classification and slot labeling for new features in task-oriented dialog systems,” in *Proc. 28th Int. Conf. Comput. Linguistics., Ind. Track. Online., Int. Committee Comput. Linguistics*, Dec. 2020, pp. 10–20.

[36] K.-H. Huang and K.-W. Chang, “Generating syntactically controlled paraphrases without using annotated parallel pairs,” in *Proc. Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2021, pp. 1022–1033.

[37] R. Sennrich, B. Haddow, and A. Birch, “Improving neural machine translation models with monolingual data,” in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers)*, Aug. 2016, pp. 86–96.

[38] S. Edunov, M. Ott, M. Auli, and D. Grangier, “Understanding back-translation at scale,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Oct./Nov. 2018, pp. 489–500.

[39] A. W. Yu, D. Dohan, Q. Le, T. Luong, R. Zhao, and K. Chen, “Fast and accurate reading comprehension by combining self-attention and convolution,” in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–16.

[40] J.-P. Corbeil and H. Abdi Ghavidel, “Bet: A backtranslation approach for easy data augmentation in transformer-based paraphrase identification context,” 2020, *arXiv:2009.12452*.

[41] S. Gao, Y. Zhang, Z. Ou, and Z. Yu, “Paraphrase augmented task-oriented dialog generation,” in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds., 2020, pp. 639–649.

[42] A. Anaby-Tavor *et al.*, “Do not have enough data? Deep learning to the rescue!” in *Proc. 34th AAAI Conf. Artif. Intell., AAAI 2020, 32nd Innov. Appl. Artif. Intell. Conf., 10th AAAI Symp. Educ. Adv. Artif. Intell.*, 2020, pp. 7383–7390.

[43] A. Ollagnier and H. Williams, “Text augmentation techniques for clinical case classification,” in *Proc. Work. Notes Conf. Labs Eval. Forum. CEUR Workshop Proc.*, 2020, pp. 22–25.

[44] R. Gupta, “Data augmentation for low resource sentiment analysis using generative adversarial networks,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 7380–7384.

[45] H. Guo, Y. Mao, and R. Zhang, “Augmenting data with mixup for sentence classification: An empirical study,” 2019, *arXiv:1905.08941*.

[46] V. Kumar, H. Glaude, C. de Lichy, and W. Campbell, “A closer look at feature space data augmentation for few-shot intent classification,” 2019, *arXiv:1910.04176*.

[47] J. Chen, Z. Yang, and D. Yang, “MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification,” in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2020, pp. 2147–2157.

[48] G. A. Miller, “Wordnet: A lexical database for English,” *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995.

[49] E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, and Q. V. Le, “Autoaugment: Learning augmentation strategies from data,” in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 113–123.

[50] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., 2013, pp. 3111–3119.

[51] J. Pennington, R. Socher, and C. Manning, “GloVe: Global vectors for word representation,” in *Proc. Conf. Empirical Methods Natural Lang. Process*, Oct. 2014, pp. 1532–1543.

[52] S. Merity, C. Xiong, J. Bradbury, and R. Socher, “Pointer sentinel mixture models,” in *Proc. 5th Int. Conf. Learn. Representations*, 2017, pp. 1–15.

[53] K. R. McKeown, “Paraphrasing questions using given and new information,” *Comput. Linguistics*, vol. 9, no. 1, pp. 1–10, Jan. 1983.

[54] I. A. Bolshakov and A. Gelbukh, “Synonymous paraphrasing using wordnet and internet,” in *Proc. Natural Lang. Process. Inf. Syst.*, F. Meziane and E. Métais, Eds., 2004, pp. 312–323.

[55] D. Kauchak and R. Barzilay, “Paraphrasing for automatic evaluation,” in *Proc. Main Conf. Human Lang. Technol. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2006, pp. 455–462.

[56] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush, “OpenNMT: Open-source toolkit for neural machine translation,” in *Proc. ACL, Syst. Demonstrations*, Jul. 2017, pp. 67–72.

[57] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding by generative pre-training,” 2018.

[58] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[59] Z. Yang *et al.*, “Xlnet: Generalized autoregressive pretraining for language understanding,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, Eds., 2019, pp. 1–11.

[60] I. J. Goodfellow *et al.*, “Generative adversarial nets,” in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[61] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” in *Proc. NIPS Deep Learn. Representation Learn. Workshop*, 2015, pp. 1–9.

[62] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “Distilbert, A distilled version of bert: smaller, faster, cheaper and lighter,” 2019, *arXiv:1910.01108*.

[63] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics., Human Lang. Technol., (Long and Short Papers)*, Jun. 2019, pp. 4171–4186.

[64] M. Lewis *et al.*, “BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” Jul. 2020, pp. 7871–7880.

[65] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–13.

[66] P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala, “Good debt or bad debt: Detecting semantic orientations in economic texts,” *J. Assoc. Inf. Sci. Technol.*, vol. 65, no. 4, pp. 782–796, Apr. 2014.

[67] C. Busso *et al.*, “Iemocap: Interactive emotional dyadic motion capture database,” *Lang. Resour. Eval.*, vol. 42, pp. 335–359, 2008.

[68] G. McKeown, M. Valstar, R. Cowie, M. Pantic, and M. Schroder, “The semaine database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent,” *IEEE Trans. Affect. Comput.*, vol. 3, no. 1, pp. 5–17, Jan./Mar. 2012.

[69] V. Barriere, C. Clavel, and S. Essid, “Attitude classification in adjacency pairs of a human-agent interaction with hidden conditional random fields,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 4949–4953.

[70] I. Mollas, Z. Chrysopoulou, S. Karlos, and G. Tsoumakas, “Ethos: An online hate speech detection dataset,” 2020, *arXiv:2006.08328*.

[71] J. McAuley and J. Leskovec, “Hidden factors and hidden topics: Understanding rating dimensions with review text,” in *Proc. 7th ACM Conf. Recommender Syst.*, 2013, pp. 165–172.

[72] D. Kotzias, M. Denil, N. De Freitas, and P. Smyth, “From group to individual labels using deep features,” in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 597–606.

[73] R. Socher *et al.*, “Recursive deep models for semantic compositionality over a sentiment treebank,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1631–1642.

[74] Y. Kim, “Convolutional neural networks for sentence classification,” in *Proc. Conf. Empirical Methods Natural Lang. Process*, Oct. 2014, pp. 1746–1751.

[75] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[76] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. 3rd Int. Conf. Learn. Representations*, Y. Bengio and Y. LeCun, Eds., 2015.

[77] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–18.

[78] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *Proc. 30th Int. Conf. Int. Conf. Mach. Learn.*, 2013, pp. III-1310–III-1318.

[79] A. Paszke *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” in *Proc. Adv. Neural Inf. Process. Syst. 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dÁlché-Buc, E. Fox, and R. Garnett, Eds., 2019, pp. 8024–8035.

[80] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, “ERNIE: Enhanced language representation with informative entities,” in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2019, pp. 1441–1451.

[81] Y. Sun *et al.*, “Ernie 2.0: A continual pre-training framework for language understanding,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 5, pp. 8968–8975, Apr. 2020.

[82] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, “GLUE: A multi-task benchmark and analysis platform for natural language understanding,” in *Proc. EMNLP Workshop BlackboxNLP., Analyzing Interpreting Neural Netw. NLP*, Nov. 2018, pp. 353–355.

[83] T. Wolf *et al.*, “Huggingface’s transformers: State-of-the-art natural language processing,” 2019, *arXiv:1910.03771*.

[84] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[85] Ç. Çöltekin and T. Rama, “Tübingen-Oslo at SemEval-2018 task 2: SVMs perform better than RNNs in emoji prediction,” in *Proc. 12th Int. Workshop Semantic Eval.*, Jun. 2018, pp. 34–38.

[86] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[87] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[88] W. Satriaji and R. Kusumaningrum, “Effect of synthetic minority oversampling technique (SMOTE), feature representation, and classification algorithm on imbalanced sentiment analysis,” in *Proc. 2nd Int. Conf. Informat. Comput. Sci.*, 2018, pp. 1–5.

[89] C. C. Aggarwal *et al.*, *Neural Networks and Deep Learning*, vol. 10. Cham, Switzerland: Springer, 2018.

[90] P. Marrafa, “Portuguese wordnet: General architecture and internal semantic relations,” *DELTA: Documentação de Estudos em Lingüística Teórica e Aplicada*, vol. 18, no. SPE, pp. 131–146, 2002.

[91] J. Gu, H. Hassan, J. Devlin, and V. O. Li, “Universal neural machine translation for extremely low resource languages,” in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics., Human Lang. Technol., (Long Papers)*, Jun. 2018, pp. 344–354.

Hugo Queiroz Abonizio received the B.Sc. and M.Sc. degrees in computer science from the State University of Londrina (UEL), Londrina, Brazil, in 2016 and 2021, respectively.

He is currently an auxiliary Professor with the Pontifícia Universidade Católica do Paraná, Curitiba, Brazil. His research interests include natural language processing, deep learning, and text mining.

Emerson Cabrera Paraiso received the Ph.D. degree in systems and information technology from the Université de Technologie de Compiègne, Compiègne, France, in 2005.

He is an Associate Professor and the head of the Graduate Program in Informatics (PPGIa) with the Pontifícia Universidade Católica do Paraná, Curitiba, Brazil. His research interests include text mining, natural language processing, information retrieval, and collaborative systems.

Dr. Paraiso is a member of the Technical and Scientific Council on Systems and Computing with the Paraná Institute of Technology, a member of the Computer Science Advisory Committee with the Araucaria Foundation, and a member of ACM, and of the Brazilian Computer Society.

Sylvio Barbon, Jr. received the B.Sc. degree in computer science and the M.Sc. degree in computational physics from the University of São Paulo, São Paulo, Brazil, in 2005 and 2007, respectively, and the M.Sc. degree in computational engineering and the Ph.D. degree from IFSC/USP, São Carlos, Brazil, in 2008 and 2011, respectively.

He is a Professor and Leader of the research group that studies machine learning with Computer Science Department, the State University of Londrina (UEL), Londrina, Brazil. In 2017, he was a Visiting Researcher with the University of Modena and Reggio Emilia, Modena, Italy, working on multispectral analysis and machine learning. Since 2021, he has been a Visiting Professor with Universita Degli Studi Di Milano, Milano, Italy, he focused on data stream and process mining. He is currently a Professor of postgraduate and graduate programs. His research interests include digital signal processing, pattern recognition, and machine learning.