Universidade Estadual de Londrina

HUGO QUEIROZ ABONIZIO

PRE-TRAINED DATA AUGMENTATION FOR TEXT CLASSIFICATION

Londrina
2021

HUGO QUEIROZ ABONIZIO

PRE-TRAINED DATA AUGMENTATION FOR TEXT CLASSIFICATION

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Sylvio Barbon Jr.

Londrina
2021

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UEL

Abonizio, Hugo Queiroz.
Pre-trained Data Augmentation for Text Classification / Hugo Queiroz Abonizio. - Londrina, 2021.
57 f. : il.

Orientador: Sylvio Barbon Junior.
Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Ciência da Computação, 2021.
Inclui bibliografia.

1. Data Augmentation - Tese. 2. Classificação de texto - Tese. 3. Redes Sociais - Tese. 4. Processamento de Linguagem Natural - Tese. I. Barbon Junior, Sylvio. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDU 519

HUGO QUEIROZ ABONIZIO

PRE-TRAINED DATA AUGMENTATION FOR TEXT CLASSIFICATION

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

BANCA EXAMINADORA

_____

Orientador: Prof. Dr. Sylvio Barbon Jr.
Universidade Estadual de Londrina - UEL

_____

Prof. Dr. Daniel dos Santos Kaster
Universidade Estadual de Londrina - UEL

_____

Prof. Dr. Bruno Bogaz Zarpelão
Universidade Estadual de Londrina - UEL

_____

Prof. Dr. Emerson Cabrera Paraiso
Pontifícia Universidade Católica do Paraná – PUCPR

Londrina, 26 de fevereiro de 2021.

ACKNOWLEDGEMENTS

Primeiramente gostaria de agradecer à minha família por todo o apoio e suporte que me deram ao longo de todos esses anos, e que sem os quais eu não seria nada.

Agradeço também à minha parceira Ana Luiza, que sempre me incentivou na busca pelos meus sonhos e é essencial em todas as partes da minha vida.

Agradeço ao meu orientador, professor e amigo, Sylvio, por todo apoio, orientação e inspiração ao longo do mestrado e da graduação.

Agradeço também aos meus amigos do laboratório REMID, que sempre contribuíram com minha pesquisa.

"Society does not consist of individuals, but expresses the sum of interrelations, the relations within which these individuals stand." - Karl Marx

ABONIZIO, H. Q.. ***Data Augmentation* Através de Modelos Pré-treinados para Classificação de Texto**. 2021. 56f. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2021.

RESUMO

Data augmentation é um método amplamente adotado para melhorar o desempenho de modelos em tarefas de classificação de imagens. Embora ainda não seja tão presente na comunidade de Processamento de Linguagem Natural (PLN), alguns métodos já foram propostos para aumentar a quantidade de dados de treinamento, como transformações simples no texto original ou a geração de novas amostras através de modelos de linguagem. No entanto, aplicações recentes de classificação de texto precisam lidar com domínios caracterizados por uma pequena quantidade de texto e escrita informal, como conteúdo de redes sociais virtuais, por exemplo, o que reduz a capacidade dos métodos atuais. Enfrentando esses desafios e tirando proveito dos modelos de linguagem pré-treinados e compressão de modelos, propusemos o método *PRE-training Data AugmenTOR* (PREDATOR). Nosso método de *augmentation* é composto por dois módulos: o Gerador, que sintetiza novas amostras baseadas em um modelo de linguagem de baixo custo computacional, e o Filtro, que seleciona apenas as amostras de alta qualidade. Os experimentos comparando *Bidirectional Encoder Representations from Transformer* (BERT), *Convolutional Neural Networks* (CNN), *Long Short-Term Memory* (LSTM) e *Multinomial Naïve Bayes* (NB) em seis conjuntos de dados demonstraram uma efetiva melhoria no desempenho. Foi obtida uma melhora de 28,5% de acurácia com LSTM no melhor cenário e uma melhoria média de 8% nos cenários de escassez de dados. Em conjuntos de dados com classes desbalanceadas o método melhorou em 6.04% o F_1-score. O PREDATOR conseguiu aumentar os conjuntos de dados de mídia social do mundo real e outros domínios, superando as técnicas recentes de *augmentation* de texto.

Palavras-chave: Data Augmentation. Classificação de Texto. Redes Sociais Virtuais.

ABONIZIO, H. Q.. **Pre-trained Data Augmentation for Text Classification**. 2021. 56p. Master's Thesis (Master in Science in Computer Science) – State University of Londrina, Londrina, 2021.

ABSTRACT

Data augmentation is a widely adopted method for improving model performance in image classification tasks. Despite not being as ubiquitous in the Natural Language Processing (NLP) community, some methods have already been proposed to increase 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 current methods' capabilities. Facing these challenges by taking advantage of pre-trained language models 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, which 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 Naïve Bayes (NB) in six datasets exposed an effective improvement in performance. It obtained 28.5% of accuracy improvement with LSTM on the best scenario and an average improvement of 8% on the low-data regime. On imbalanced datasets, it improved in 6.04% the F_1-score. PREDATOR was able to augment real-world datasets from social media, clinical reports, among other domains, overcoming recent text augmentation techniques.

Keywords: Data Augmentation. Text Classification. Online Social Networks.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

BERT Bidirectional Encoder Representations from Transformers.

BPE Bytepair Encoding.

BT Back-translation.

CNN Convolutional Neural Network.

EDA Easy Data Augmentation.

GPT Generative Pre-Training.

IDF Inverse Document Frequency.

LSTM Long Short-Term Memory.

ML Machine Learning.

NB Naïve Bayes.

NLP Natural Language Processing.

OSN Online Social Networks.

RNNs Recurrent Neural Networks.

TF Term Frequency.

TF-IDF Term Frequency-Inverse Document Frequency.

CONTENTS

1 INTRODUCTION

Data augmentation techniques have been successfully applied in Machine Learning (ML) models to improve their generalization capacity. It is a common strategy to avoid overfitting the training data, mainly on data scarcity scenarios and situations where labeled examples are expensive. Since the performance of ML 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 [4].

Those data scarcity scenarios can lead to either balanced and imbalanced datasets. When labeled samples, as a whole, are expensive to obtain, this leads to a balanced situation, where every class has roughly the same amount of samples. However, it is also common to have classes that are easier to obtain than others, leading to an imbalanced dataset, where a given class is under-represented compared to other classes [5]. The imbalanced problem poses another challenge to the data scarcity scenario. In addition to a small amount of data to learn from, imbalanced data is known to compromise the learning process [6]. Therefore, data augmentation techniques can play an essential role in those different scenarios to improve the model's generalization.

Several techniques have been proposed and evaluated for image data [7], 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 [8, 9], but this assumption does not hold for text data. Changing words order or removing some parts of a sentence might change its whole semantics, resulting in low-quality samples and negatively impacting the performance.

In recent years, different text transformation strategies have been proposed, varying from synonyms replacements [10, 11], paraphrasing through translation models [12] and text generation using language models [13]. A recent method, entitled Easy Data Augmentation (EDA) [14], has been proposed combining synonym replacement with other simple methods such as random deletion and random swap of words. Those methods reportedly increase the accuracy of classification on small datasets. However, a gold standard technique is yet to be settled.

Another often employed technique is the back-translation (BT), which works by making a round-trip translation using a secondary language. By using two models — one for translating from the original language to the secondary and other for the reverse — the intention is to create a paraphrase from the original sentence. This approach has been shown to yield better results on Neural Machine Translation and other tasks [12].

Most recent work has proposed using pre-trained language models [15, 16], lever-

aging 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 the simple sample transformations initially proposed. In addition to the resources requirements, those approaches can be prohibitively expensive and have raised a concern regarding their energy efficiency [17, 18].

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 [19], which leads to frequent out-of-vocabulary words in this scenario. On the other hand, language model-based approaches can learn to reproduce the dataset writing style and pre-trained models leverage a priori knowledge to extend their generation capabilities [20].

Tackling the challenges of text augmentation on 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. Our method is grounded on lightweight pre-trained models obtained by model compression [21]. 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 Naïve 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 all classifiers' accuracy, achieving an average of 8% improvement in accuracy and a maximum of 28.5% on the best scenario on the low-data regime. Statistical analysis demonstrated that its performance is similar to using real data to increase the dataset. On imbalanced datasets, our method achieved a 9.82% improvement on F_1-score. Our work is the first, to the best of our knowledge, to cover either balanced and imbalanced class distributions, binary and multiclass datasets, and a wide range of textual domains with a single method.

1.1 Objectives and contributions

The main objective of our work is to propose a method for text data augmentation suitable for diverse text classification tasks.

The main contributions of this work can be summarized as:

1. The introduction of a new method for data augmentation for text classification tasks;
2. The comparison with two widely applied augmentation methods;
3. Investigation of augmentation capabilities on different domains;
4. Improvement of text classification performance on data scarcity and imbalanced class scenarios.

1.2 Outline

This master thesis is structured as follows. Chapter 2 presents a review of works that proposed and applied data augmentation methods in the NLP area. Chapter 3 presents a theoretical foundation about the text classification task, the ML models evaluated and the concepts of transfer learning and model compression. The materials and methods are covered in Chapter 5. Experimental results are presented and analyzed in Chapter 6. Finally, limitations, conclusions and future work directions are discussed in Chapter 7.

2 RELATED WORK

Previous work on textual data augmentation has proposed different methods for sample transformations that can be categorized into three main groups: direct word transformations, back-translation, and language model-based. A common approach in the first category is the synonym replacement, explored with different methods. Zhang et al. [11] performed augmentation by replacing synonyms based on its similarity obtained from the WordNet thesaurus. After selecting all replaceable words, they choose r of them to be replaced according to the probability distribution with parameter p given by $P[r] \sim p^r$. The index s of the synonym chosen to be inserted replacing the original word is also given by a probability distribution with parameter q, which $P[r] \sim q^s$. Similar approaches of using synonyms replacement from a thesaurus were proposed by Kolomiyets et al. [10] and Wei and Zou [14].

Wang and Yang [22] proposed the usage of neighboring words in the continuous representation in the embedding space to create new instances for a tweet classification task. The proposed method searches for k-nearest neighbors for each word in the document using cosine similarity between the query term vector and the target vectors.

A recent method, EDA [14], 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. The method defines the augmentation based on two parameters: α as the percent of words in a sentence that should be changed, and n_{aug} for the number of generated sentences.

The second category of data augmentation approaches on NLP tasks is 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. It 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 [12, 23] and reading comprehension [24]. BT was also applied to low-resource text classification [25], yielding improved classification accuracy using different secondary languages.

More recently, Fadaee et al. [26] proposed the Translation Data Augmentation, which relies on a language model-based on a Bidirectional LSTM trained on large amounts of monolingual data. The method targets low-frequency words on the existing parallel sentence pairs by generating diverse contexts to improve low-resource languages' performance. Afterwards, the generated sentences are filtered to avoid semantically or syntac-

tically incorrect samples using the degree of confidence of the translation. Other works also applied BT on text classification tasks [27, 28].

The third category of approaches is the usage of a language model to generate new samples based on contextual information. Kobayashi [13] proposed a method of augmentation that relies on a Bidirectional LSTM Language Model pre-trained on a large amount of text to replace original words given their context. After selecting the target word in the original sample, the model predicts possible words for this position, given its surrounding. The method augments the samples by sampling words during the model's training using a temperature parameter τ, controlling the sampling probability distribution. For $\tau \to 0$, the augmenting words are always the highest probable ones, and when $\tau \to \infty$, the sampling becomes uniformly distributed. To show the effectiveness, the author evaluated CNN and LSTM as classifiers.

Recently, approaches using pre-trained language models leveraging the capacity of the Transformer [29] architecture trained on large corpus started being proposed. Anaby-Tavor et al. [15], Wu et al. [30] and Kumar et al. [16] are examples of this recent trend. Anaby-Tavor et al. [15] proposed the usage of GPT-2 [31] for generating synthetic samples conditioned by using the class label as prompt to the language model. This class label prompt combined with a previously trained classifier output results on a double voting mechanism for selecting generated samples. Kumar et al. [16] go further and explore different types of pre-trained Transformers, namely BERT [32] and BART [33].

Some works tackled the imbalanced class problem, such as Ibrahim et al. [34], where duplicated words removal, random deletion and synonyms replacement were conducted to augment an imbalanced multi-label dataset. Zhang et al. [35] also explored an imbalanced multi-label dataset, using GPT-2 to generate new samples.

Table 1 shows a summary of the related studies with the corresponding year of publication. Notably, more sophisticated methods gained popularity in recent years. The development and popularization of those more complex tools in recent years, e.g., pre-trained language models, explain the exploration of them for data augmentation purposes.

Our approach is situated on the language model-based methods, taking advantage of a pre-trained model through transfer learning. However, most of those recent Transformer models are computationally expensive. They sometimes require specific hardware, which creates a barrier to its usage on supporting tasks such as data augmentation since the objective is to improve the main task's performance. The usage of large models might be prohibitive in those scenarios of low computational resources. Therefore, our method is the first, to the best of our knowledge, to propose and evaluate the usage of compressed models to achieve a similar result with much smaller, faster and computationally cheaper models than their original versions.

Table 1 – Summary of related studies applying and proposing data augmentation for NLP tasks and its year of publication for timeline comparison.

Category	Year of Publication	Reference
Word transformation	2011	Kolomiyets et al. [10]
	2015	Zhang et al. [11]
	2015	Wang and Yang [22]
	2018	Ibrahim et al. [34]
	2019	Wei and Zou [14]
Back-translation	2016	Sennrich et al. [12]
	2018	Yu et al. [24]
	2018	Nishimura et al. [23]
	2018	Aroyehun and Gelbukh [27]
	2019	Shleifer [25]
	2020	Marivate and Sefara [28]
Language model	2018	Kobayashi [13]
	2019	Anaby-Tavor et al. [15]
	2019	Wu et al. [30]
	2020	Kumar et al. [16]
	2020	Zhang et al. [35]

3 THEORETICAL FOUNDATION

The method proposed in this thesis is built upon a theoretical foundation of algorithms and techniques previously proposed that were adapted to our proposal. In this chapter, we present the main concepts explored in this thesis. In Section 3.1, an overview of the text classification task is provided. In Section 3.2, the different ML algorithms employed in this work are introduced. Moreover, since the proposed method relies on transfer learning, Section 3.3 explains the concept and its different approaches. Finally, the concept of neural model compression is explained and motivated in Section 3.4.

3.1 Text Classification

Due to the increasing amount of textual data being generated on several media such as Online Social Networks (OSN), news outlets and patient records, Natural Language Processing (NLP) field has gained traction in recent years. The advent of Web 2.0 [36], where users actively create the content, has amplified the content generation to a new level. Written documents are strongly tied to human activities, and useful insights can be brought by the analysis of such in practically all domains [37].

Text data is unstructured information, which raises a range of technical challenges for machines to understand. However, the information contained on the large amount of text produced on a daily basis is valuable. Thus, the development of algorithms capable of extracting this knowledge is a must [38]. The discipline responsible for extracting knowledge from text data is known as text mining. Feldman and Sanger [39] define text mining as a knowledge-intensive process in which a user explores documents through analytic tools to extract useful patterns. Žižka [37] differentiates text mining from data mining primarily due to the data structure, where data mining handles mainly tabular data records. While data mining can be defined as the automatic process of finding implicit knowledge in collections of electronically stored data [40], text mining deals with unstructured documents with variable lengths.

Under this broad umbrella of text mining, many approaches have been applied to extract valuable information from documents. Document clustering, information retrieval and trend detection are examples of those major topic areas in text mining [41]. The categorization of documents is one of the major areas, and algorithms for automatic categorization are at the core of many systems that process text data at scale. Email filtering, language identification and topic detection are a few procedures where the categorization is employed.

In this context, the task of automatically categorizing text documents into pre-

defined classes is called text classification [42]. The usual pipeline for text classification systems involves transforming the raw text into feature vectors to feed a supervised learning model [43]. This model is trained to predict the category for a given document based on its feature vector. The main objective is to extract knowledge from training examples and generalize to infer the category of unseen examples.

Aggarwal and Zhai [44] formally define the text classification problem as follows. Given a set of training samples $\mathcal{D} = \{x_1, ..., x_n\}$ labeled as a discrete k class index, the classification model is then trained on those samples to relate the features to its corresponding label. Thus, for a given sample in which class is unknown, the classification model might be able to predict its class based on its features and the relations learned on the training set. This classification model training can be defined as learning a function f that maps documents in $\mathcal{D}$ into the corresponding label y_i from the label set $\mathcal{L} = \{y_1, ..., y_n\}$, as defined in Equation 3.1.

$$\begin{aligned} f &: \mathcal{D} \rightarrow \mathcal{L} \\ f(x_i) &= y_i \end{aligned} \tag{3.1}$$

Text classification applications vary, being widely applied in several domains such as healthcare, social sciences, and marketing. A typical application of text classification is sentiment analysis, where, given an opinionated document about an entity, its opinion orientation is automatically determined as being positive, negative, or neutral [45]. Another common application is document categorization, where documents are categorized into predefined concepts or domains, e.g., news topics or message subjects [46]. In addition to these examples, text classification techniques has been applied to predict suicide risk , automatizing diagnoses categories [47] and fake news detection [48].

3.2 Machine Learning Algorithms

Machine Learning (ML) algorithms can be split into two main approaches: supervised and unsupervised learning [43]. Briefly, supervised algorithms have the main goal of finding a function f which maps $f : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{X}$ is the feature set and $\mathcal{Y}$ is the target set. On the other hand, unsupervised algorithms do not have access to the target set, they extract patterns solely from the feature vectors. The classification tasks, which are the focus of this work, lie on the supervised learning category, where the feature set is composed of features extracted from the documents and the target set assumes discrete values representing the labels.

This work also employs two other hybrid approaches: semi-supervised and self-supervised learning. Semi-supervised learning in classification tasks consists of learning

from both labeled and unlabeled examples. Since classification models require many labeled data and the labeling process is costly, a semi-supervised approach aims to improve the model performance using unlabeled data [49]. Blum and Mitchell [50] proposed the usage of a few labeled samples to train an initial model to automatically label more samples to augment the training set using a graph-based approach. A similar approach of using model's prediction on unlabeled data to increase the data during training was used by Ruder and Plank [51]. This process is also known as self-training [52, 53].

The self-supervised approach is employed when the labels are natural to the task, i.e., no manual labeling process is required to define the label for an instance since it can be implied. An example of a self-supervised task is the prediction of a word given its context [54], since both the context and the target word can be implied from the dataset. Modern pre-training objectives for NLP models use self-supervised learning tasks to take advantage of the large amount of raw text available on the web [55].

3.2.1 Naïve Bayes

The Naïve Bayes (NB) classifier has been widely used for document classification for decades [56]. It is based on the Bayes theorem [57], which assigns the most likely class to a given example based on its feature vector, assuming each feature as independent. Despite this unrealistic assumption, especially on textual domain where words on the same sentence tend to be strongly dependent, NB classifier has proven to be effective in many applications [58, 59].

Historically, the most basic version of NB classifier used Term Frequency (TF), also known as bag-of-words, as feature [60]. The TF technique counts the frequency of each word of a document to represent it as a fixed length vector. Later, Jones [61] proposed the usage of Inverse Document Frequency (IDF) combined with TF as a method to soften the effect of common words, such as articles, present in the corpus. Thus, for a given document d and a term t, being N the number of documents, $df\,(t)$ the number of documents that contains the term t, Equation 3.2 defines the Term Frequency-Inverse Document Frequency (TF-IDF) calculation.

$$W\,(d,t) = TF\,(d,t) * log\left(\frac{N}{df\,(t)}\right) \tag{3.2}$$

Wang and Manning [62] demonstrated that simple NB models — which are computationally inexpensive and have little memory consumption — can perform on par or even better than more complex models. Therefore they are an excellent baseline method to compare. The variation recommended by them is the Multinomial NB, which assumes that each probability for a feature comes from a multinomial distribution. Equation 3.3

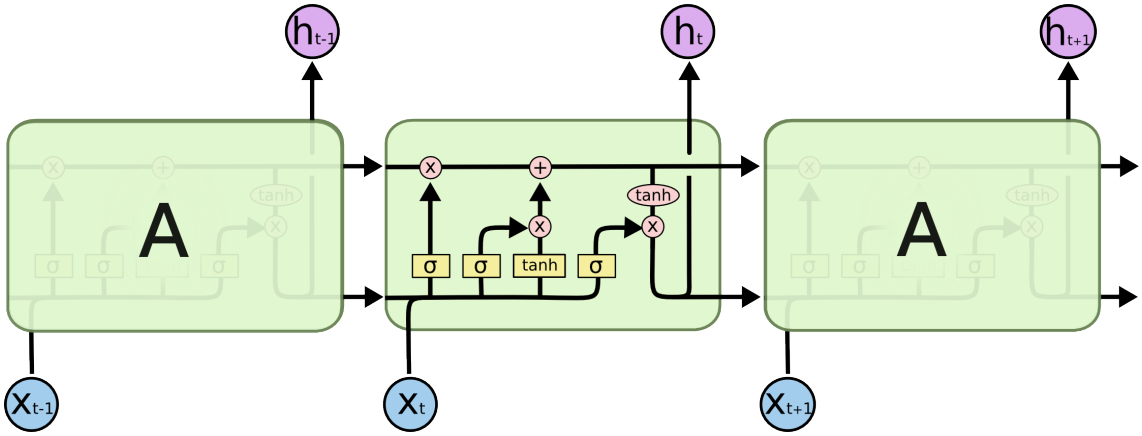

Figure 1 – LSTM architecture. Source: Olah [1].

shows the probability of document d belonging to class c.

$$P(c|d) = \frac{P(c) \prod_{w \in d} P(d|c)^{n_{wd}}}{P(d)} \tag{3.3}$$

3.2.2 Long Short-Term Memory

Long Short-Term Memory (LSTM) is a variation of Recurrent Neural Networks architecture designed to better capture long term dependencies [63]. Recurrent architectures are intended to make use of sequential information by representing the text as a sequence of words, i.e., the output y_t for the time step t is determined not only by current input x_t but also its precursors $x_0, x_1, ..., x_{t-1}$. This is done by sharing a hidden state h_t, which contains information from all previous steps. This gives RNNs the ability to incorporate sequential information in their predictions and makes them a great fit for modeling text data. However, the vanilla version of RNNs suffers from problems of exploding and vanishing gradient, making the model biased towards recent inputs and prone to forget longer-term data [64].

Among the different variations of vanilla RNNs, the LSTM is the most widely applied in NLP tasks [65]. It was proposed to overcome the problems faced by vanilla RNNs by introducing a memory cell to remember values through time. Figure 1 illustrates the structure of LSTM with its gates and the recurrent mechanism. The LSTM cell works by deciding which information to keep through a forget gate, defined in Equation 3.4, followed by the input gate and the candidate values, Equation 3.5 and Equation 3.6, which decide what information to store in the cell state. Then, the current state is updated by Equation 3.7 based on the forget gate and candidate values. The output gate for the step t is given by Equation 3.8, and the hidden cell state is passed to the next step according to Equation 3.9.

$$f_t = \sigma(\mathbf{W}_f[h_{t-1}, x_t] + b_f), \tag{3.4}$$

$$i_t = \sigma(\mathbf{W}_i \cdot [h_{t-1}, x_t] + b_i), \tag{3.5}$$

$$\widetilde{C}_t = tanh(\mathbf{W}_C \cdot [h_{t-1}, x_t] + b_C), \tag{3.6}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C}_t, \tag{3.7}$$

$$o_t = \sigma(\mathbf{W}_o[h_{t-1}, x_t] + b_o), \tag{3.8}$$

$$h_t = o_t \cdot tanh(C_t), \tag{3.9}$$

Due to the suitability of LSTM structure to variable-length data, it has also been applied to various non-NLP tasks. Some examples are time-series forecasting [66] and malware classification [67]. On NLP tasks, some examples are classification of legal documents [68], cross-lingual sentiment classification [69], language translation [70], and credibility analysis on social media [71].

3.2.3 Convolutional Neural Network

A Convolutional Neural Network (CNN) is a neural architecture commonly used for image processing, but is also one of the most popular model architectures for text classification [65]. While RNNs aims to recognize patterns through time, CNNs learn to recognize patterns across space [72], i.e., CNNs work by extracting local patterns such as key phrases in order to discriminate the categories. Thus, CNNs have been effectively applied to classification tasks such as sentiment analysis [73], news categorization [74] and other NLP tasks [75].

The main mechanism of CNN is the convolution operation [76], which, in the context of text data, applies a k-word sliding window over the sentences. This sliding window is done by a $d \times d$ filter, also known as kernel, and converts each window into a d-dimensional vector that can be combined and called feature map [60]. However, different from convolution over a 2-dimensional matrix on images, convolution over sentences are carried over one dimension. Given a sequence of words $x_1, ..., x_n$ and its corresponding d-dimensional word embeddings w_i, a 1-dimensional convolution with width k results on m vectors $p_1, ..., p_n$ such that:

$$p_i = g(w_i W + b)$$

where g is an element-wise non-linear activation function, W and b are learned parameters, and p_i encodes the information present in w_i [77]. Those vectors are often combined using a max-pooling layer resulting in a single vector picking the most important features extracted to be used on prediction. In practice, multiple window sizes are used on different convolutional layers and then concatenated to be fed to the output layer [78].

3.2.4 Transformer

The Transformer architecture was introduced by Vaswani et al. [2] and, instead of recurrence or convolution, is based on the attention mechanism. This architecture allows much more parallelization when compared to RNNs, making it possible to train bigger models on a larger dataset efficiently. The self-attention mechanism employed by Transformer computes in parallel an attention score for each word on a sentence to determine its influence on the remaining.

The attention mechanism was proposed by Bahdanau et al. [70] and since then became ubiquitous in modern state-of-the-art NLP models. The main idea is to focus on relevant parts of the input sequence. As the model processes each word, the attention allows it to look at other parts of the input in order to produce better encoding. This mechanism creates three vectors for each input vector, query, key and value vectors, which are created by multiplying the input by three correspondent matrices that have their values learned during the training. Then, the attention score is calculated by the dot product of the query vector and the key vector for the given word and pass it to a softmax normalization. With the normalized scores, each value vector is multiplied by its score to keep only the relevant ones. The sum of the weighted value vectors is the output of the self-attention layer that is passed to a feed-forward layer, in the case of Transformer. Equation 3.10 provides a formal definition, called scaled dot-product attention, where Q is the query vector, K is the key vector, V is the value vector, and d_k is the dimension of the key vector.

$$Attention\,(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.10}$$

However, as the attention mechanism does not provide sequential information because each item is computed independently with no notion of word order, the Transformer models rely on positional sinusoidal to represent the sequence position. Other Transformer based architectures also introduce the positional embeddings that are learned through the training phase. For a given vector index i, position pos, and d_{model} representing the dimension of the input embeddings, this positional sinusoidal encoding is given by:

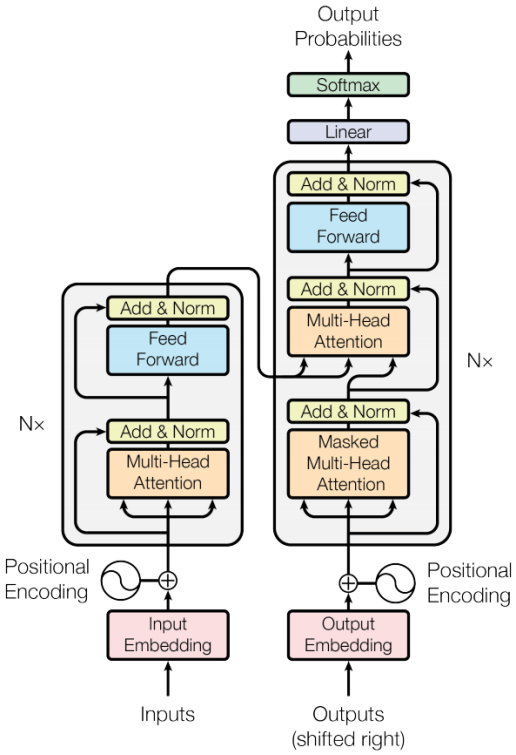

Figure 2 – Encoder-decoder Transformer model architecture composed of N_x Transformer blocks with multi-head performing self-attention mechanism multiple times. Source: Vaswani et al. [2].

$$PE_{(pos,2i)} = sin\left(pos/10000^{2i/d_{model}}\right)$$

$$PE_{(pos,2i+1)} = cos\left(pos/10000^{2i/d_{model}}\right)$$

Figure 2 illustrates an encoder-decoder model composed of Transformer blocks with its multi-head attention module to perform attention mechanism multiple times, the feed-forward layer, the layer normalization and the other components, whose in-depth explanation goes beyond this work. This architecture makes Transformer-based models robust to be applied to a variety of NLP tasks, achieving better results than previous models on text classification, question answering, natural language inference, language modeling, among others [32, 79, 80, 81].

3.2.5 GPT-2

GPT-2 is the successor of Generative Pre-Training (GPT) [82], an autoregressive Transformer language model trained on a large text corpus on a semi-supervised manner. The corpus is composed of 8 million webtexts obtained by scraping about 45 million links from Reddit, resulting in approximately 40 GB of text. The pre-training task provides GPT the ability to generate text by iteratively selecting words from the model output

until it reaches a stop condition. The language modeling task has as input a sequence of words on a sentence and outputs the probability distribution for the next word given the input context. Given a sequence of tokens $\mathcal{U} = \{u_1, ..., u_n\}$, the objective is to maximize the following conditional probability:

$$p(\mathcal{U}) = \prod_{i=1}^{n} p(u_i \mid u_1, \ldots, u_{i-1}) \tag{3.11}$$

GPT-2 handles tokens as subword units, instead of complete words, using Bytepair Encoding (BPE) [83], resulting on a vocabulary of 50,257 tokens and avoiding out-of-vocabulary with rare words. The pre-trained model was released in different model sizes, varying from 117 million parameters to 1.5 billion parameters.

The text generation capacity of GPT-2 raised concerns due to its potential of misusing on large scale disinformation and propaganda. Previous research was already conducted to adapt the model to different domains to generate neural fake news [84], realistic propaganda for extremist groups [85], and automatic patent claims [86]. The community also noted that the synthetic text can be difficult to detect and supporting tools need to be developed [87, 88]. Different methods for text decoding were also proposed, culminating on the current standard sampling methods [89].

3.2.6 Bidirectional Encoder Representations from Transformers

Bidirectional Encoder Representations from Transformers (BERT) [32] is an autoencoder Transformer language model pre-trained on masked language model, i.e., during the training phase, the model learns to fill missing words within a sentence. Unlike autoregressive language models that predict the next word for a given sequence, the input words are randomly replaced with a `[MASK]` token on a probability of 15%, representing the word that should be predicted. BERT also uses a next sentence prediction task during the pre-training, where the model receives a pair of sentences and should predict if the second sentence is subsequent to the first in the original document.

Different from GPT-2, which used a left-to-right architecture, BERT incorporates information from both directions in order to apply contextual data to sentence-level and token-level tasks such as classification and question answering. BERT's ability to take bidirectional information in consideration to make a decision, makes it very attractive for text classification tasks, which rely not only on syntactic but also semantic understanding to improve the performance.

BERT achieved state-of-the-art performance on a wide range of tasks by the time of its releasing, including natural language inference, question answering and text classification [32]. After that, many BERT-based models were proposed to improve the original

architecture by making it more robust, such as RoBERTa [90], which is trained on much more data, and ALBERT [55] which lowers the memory consumption.

3.3 Transfer Learning

Transfer learning in ML models is inspired by the human ability to transfer acquired knowledge across tasks. Instead of learning from scratch, humans are capable of apply abstract knowledge to improve another task [3]. The motivations from transferring knowledge between domains include the data scarcity problem, which is faced by many applications where the data labeling process can be costly. Model robustness is another motivation, since the assumption that both the training and test data are drawn from the same distribution does not always hold as expected. Learning from a larger amount of data and adapt to smaller problems tends to generate more robust models. Privacy and data security are also relevant issues that can be addressed by transfer learning methods, where edge devices can build new models with local data on top of a more general model, ensuring privacy of the data [91].

According to Yang et al. [91], we define a domain $\mathbb{D}$ composed of a feature space such that $x \in \mathbb{X}$ and a marginal probability distribution $\mathbb{P}^X$. Given a specific domain, a task $\mathbb{T}$ is composed of a label space Y and a predictive function $f(x) = P(y|x)$ which makes predictions based on feature values. The values of Y can be binary, discrete or continuous, depending on the task. Thus, given a source domain $\mathbb{D}_s$ and a source task $\mathbb{T}_s$, and a target domain $\mathbb{D}_t$ with a target task $\mathbb{T}_t$, transfer learning refers to using the knowledge in $\mathbb{D}_s$ and $\mathbb{T}_s$ to improve $f_t(.)$.

The transfer learning paradigm refers to extracting the knowledge from one or more scenarios to help boosting the learning performance in the target scenario. This methodology is shown to be effective to leverage the performance of machine translation of low-resource languages [92] and sample-efficient text classification [93]

In recent years, NLP models make heavy use of transfer learning to leverage the performance on many tasks [94]. The traditional NLP learning task was decomposed into two stages: pre-training and fine-tuning. During the pre-training phase, the model learns from large amounts of unlabeled text data, intending to capture the general grammar and semantic information of a language. The fine-tuning phase learns a specific task from a particular domain, e.g., sentiment analysis or question answering. However, pre-training is a time-consuming procedure that requires massive computational resources. Meanwhile, the pre-training is a one-time step and the model can be fine-tuned starting from the saved model as many times as needed.

Ruder [3] proposed the taxonomy for NLP transfer learning illustrated in Figure 3. The category of transfer learning methods applied in this works is situated inside Inductive

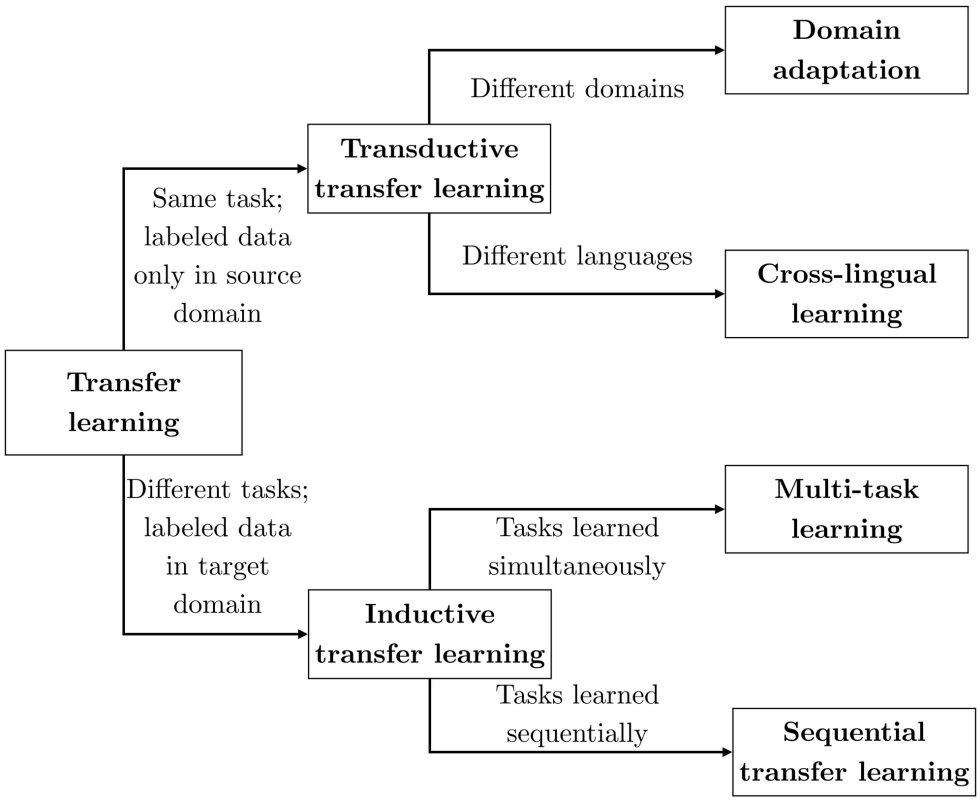

Figure 3 – Transfer learning taxonomy for NLP. Source: Ruder [3].

transfer learning, more specifically on Sequential transfer learning. It is defined as the setting where the source and target tasks are different and performed in sequence, i.e., each task is performed separately, one after the other.

3.4 Model Compression

The current trend towards bigger NLP models motivated a reaction in the research community and industry, seeking ways to compress those models into faster and lightweight models [95]. Current state-of-the-art models demand a large memory footprint and heavy computing power to be trained. While traditional transfer learning softens this impact with task-specific fine-tuning, the model size remains the same. Different solutions for compressing the original models were proposed, such as pruning and quantization [96], but a widely researched technique in recent years is the knowledge distillation [21]. This technique proposes the usage of a smaller model – known as the student – and the original model – the teacher. The student is trained to reproduce the behavior of the teacher. Standard classification models are trained to maximize the estimated probability of the original labels, usually by minimizing the cross-entropy between predictions and the true label distribution. However, the loss function used in the distillation process is different. The student's loss depends on the teacher's output.

This work proposed the usage of DistilBERT, a model proposed by Sanh et al.

[97] which uses a distillation loss defined by $\mathcal{L}_{ce} = \sum t_i * log(s_i)$, being t_i the probability estimated by the teacher and s_i by the student. The authors also used a masked language modeling loss $\mathcal{L}_{mlm}$ and a cosine embedding loss $\mathcal{L}_{cos}$ to align the hidden state vectors of both models. Those losses are combined composing a tripe loss for distilling the knowledge from BERT-base. DistilBERT achieves 97% of the teacher performance on the GLUE benchmark [98], using 40% fewer parameters and being 60% faster at inference time. This work also employs the DistilGPT2 model, which was obtained through an analogous process to that of DistilBERT.

This makes room for researchers from peripheral countries and small companies to employ state-of-the-art models in an energy-efficient and minimum-cost production environment. While the trend of building larger models does not hit a plateau [99], model compression techniques keep evolving and democratizing ML.

4 PROPOSED APPROACH

The main idea of our proposal regards the joint contribution of a text generator module and a filter module making up a two-step sample synthesizing approach. Thus, boosted by a semi-supervised classification model, our method delivers a robust text augmentation method. The algorithm behind our method is designed to cover both balanced and imbalanced classes datasets.

The PREDATOR architecture is composed of two modules: the Generator and the Filter, as shown in Figure 4. 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 [100], 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 high-quality new samples, i.e., it accepts as augmented samples only the synthetic samples in which the classifier has a high confidence of belonging to one of the given classes.

Figure 4 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, the generated samples are filtered, discarding low-quality sentences. The selected synthetic samples are accumulated until reaching a previously defined number of augmented samples.

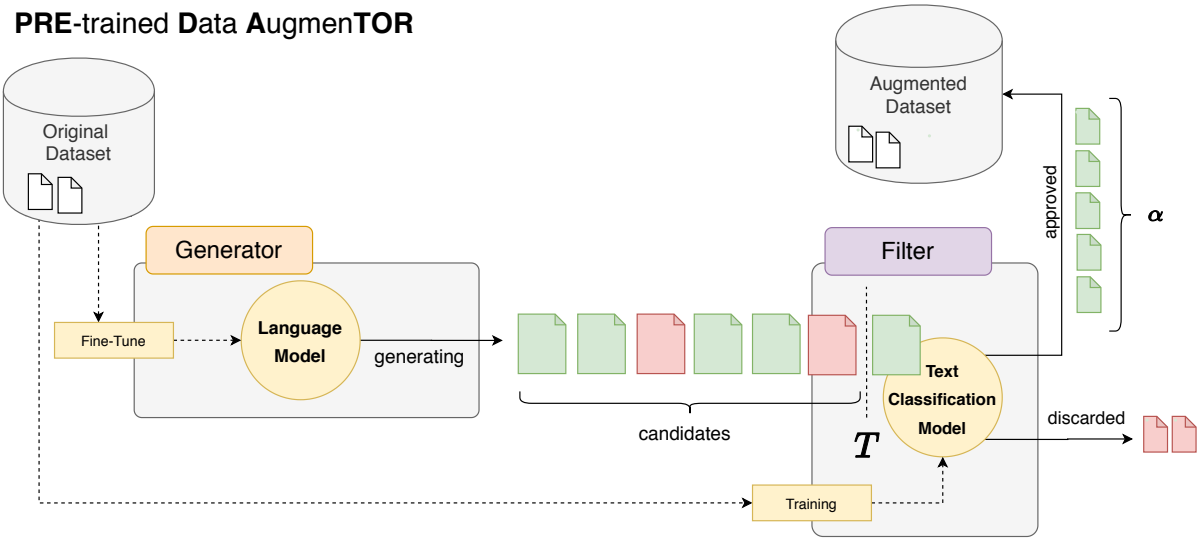

Figure 4 – Overview of the proposed approach: PRE-trained Data AugmentOR

Algorithm 1: Augmentation procedure of PREDATOR

Input: Training dataset $\mathcal{D}_{aug}$
Majority classes to ignore $\mathcal{Y}_{ignore}$ (default $\emptyset$)
Augmentation ratio α
Confidence threshold T
Number of input prompts n

$\mathcal{M}_{filter} \leftarrow \text{train}(\mathcal{M}_{filter}, \mathcal{D}_{train})$;
$\hat{\mathcal{D}}_{train} = \{d_{label} \in \mathcal{D}_{train} \mid label \notin \mathcal{Y}_{ignore}\}$;
$\mathcal{M}_{generator} \leftarrow \text{train}(\mathcal{M}_{generator}, \hat{\mathcal{D}}_{train})$;
$targetsize \leftarrow \max\{|d_{label}| \in \mathcal{D}_{train}\} * \alpha$
$\mathcal{D}_{aug} \leftarrow \mathcal{D}_{train}$
while $targetsize > \min\{|d_{label}| \in \mathcal{D}_{train}\} * \alpha$ **do**
  $l \leftarrow \arg\min_{label}\{|d_{label}| \in \mathcal{D}_{train}\}$;
  $prompt \leftarrow Uniform(\{|d_{label}| \in \mathcal{D}_{train} \mid label = l\}, n)$;
  $generated \leftarrow generate(\mathcal{M}_{generator}, prompt)$;
  $selected \leftarrow filter(\mathcal{M}_{filter}, generated, T)$;
  $\mathcal{D}_{aug} \leftarrow \mathcal{D}_{aug} \cup selected$
end
Result: Augmented dataset $\mathcal{D}_{aug}$

The augmentation procedure is summarized in Algorithm 1. The inputs for the algorithm are: the training dataset $\mathcal{D}_{train}$, which is the target of augmenting, a set containing classes to ignore on generation $\mathcal{Y}_{ignore}$, the augmentation ratio α, the confidence threshold T, and the number of input prompts n that are used to feed the language model. We start by training the Filter classifier $\mathcal{M}_{filter}$ on the original dataset $\mathcal{D}_{train}$.

Then, we create a $\hat{\mathcal{D}}_{train}$ with only samples from classes that are not ignored, where $d_{label} \subset \mathcal{D}_{train}$ with d_{label} containing only samples from a specific label. This step is important on imbalanced scenarios where we have over-represented classes that may harm the learning of the Generator. On balanced scenarios all labels are considered, therefore $\hat{\mathcal{D}}_{train} = \mathcal{D}_{train}$. Next, the $\mathcal{M}_{generator}$ is trained on $\hat{\mathcal{D}}_{train}$.

The stop criteria for our augmentation loop is the reaching of a target size, which is defined by the number of samples of the majority class multiplied by our augmentation ratio. The loop stops when the minority class reaches the size of the majority one. That way, we can cover both balanced and imbalanced setups because, with the imbalance, the number of samples is naturally different, and, when balanced, the α controls this target size. Therefore, for balanced datasets, the algorithm requires $\alpha > 1$. However, for imbalanced scenarios, lower values for α can also be used when the goal is to reduce the imbalance among the classes.

At each iteration, the algorithm randomly select n (where $n \in \mathbb{N}$) samples from the minority class as the Generator prompt. Those concatenated samples are fed to the language model in order to start the conditioned text generation. With the synthesized

samples, we select only those above the threshold T and include them into the resulting dataset $\mathcal{D}_{aug}$, until it reaches the stop criteria.

Among the several recently developed language models, we propose the usage of DistilGPT2 [97] on the Generator module. DistilGPT2 is a compressed version of GPT-2 obtained through knowledge distillation [21], becoming two times faster and having 33% fewer parameters than the smallest version of the 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 that DistilGPT2 was trained on. However, since DistilGPT2 was trained using OpenWebTextCorpus [101], 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. Previous work attached the class labels to condition the generation of text [15]. Our proposed approach differs from previous by simply concatenating three random samples from the target class using the language model input. That is, given random samples s_i from a target class. A separator token already included in the model vocabulary `SEP`, the input is given by s_1`SEP`s_2`SEP`s_3`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 [89].

To develop the decoding strategy used in PREDATOR, we evaluated different combinations of currently applied methods. We evaluated top-k sampling [102] with different values for k, combined with nucleus sampling [89] with different values for p. We also evaluated a *temperature* parameter to control the shape of the probability distribution [103, 102]. However, the best results were obtained using the value 1, i.e., not reshaping the distribution. The resulting decoding strategy is a combination of top-k and nucleus sampling, with $k = 50$ and $p = 0.9$. Those values are the proposed default because they demonstrated the best performance on evaluated scenarios. Nevertheless, they can be treated as a hyperparameter using different values in subsequent researches and applications.

After generating, the next step is selecting new synthesized samples and the imputation of its class by a classifier trained on the original dataset. The Filter module performs this process for avoiding low-quality samples in the final augmented dataset, essential to leverage highly accurate outcomes. Current state-of-the-art classification models are often

large Transformer-based classifiers [80], which makes them too resource-hungry. Therefore they may not be well suited to be applied to a pipeline of data augmentation due to their requirements of expensive resources such as GPUs with large amounts of memory. Therefore, DistilBERT is used as the Filter module classifier, maintaining a competitive performance and meeting computational resources' requirements.

The α hyperparameter 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 the classification task considering the given samples. With a high T value, the module only selects the samples that its classifier predicts the class with the highest confidence.

In contrast, a low value of T might approve samples to which the classifier is less certain about the predicted class. 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 the α requirement. On the other hand, lower values can lead to a 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 data augmentation aims to enrich the dataset with different samples without losing its class characteristics, a certain amount of uncertainty is required. Our experiments showed that a value of 0.7 (default value) for T is the best choice on evaluated scenarios.

5 MATERIAL AND METHODS

This section describes the experiments carried out in this work. Section 5.1 describes the three explored datasets and their characteristics. Section 5.2 describes the classification setup and each model hyperparameters. Finally, Section 5.3 describes the experimental comparison between the proposed method and literature techniques.

5.1 Datasets

We evaluated the method on six different datasets to cover both balanced and imbalanced scenarios in different domains within text classification tasks. Firstly, we selected three datasets for the low-data regime balanced scenario. *SST-2* (Stanford Sentiment Treebank)[1] [104], a classic dataset for sentiment classification on movie reviews widely applied as benchmark [13, 16, 14], with two classes: positive and negative. *AG-NEWS*[2] [11], another common benchmark dataset, but for topic classification task [93, 105] 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 this media's noisy characteristics. 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 on balanced datasets 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 maintaining the equal class distribution. Previous work has simulated low-data regime settings by subsampling original datasets [4, 16, 14], becominga 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 subsamples of the dataset are treated 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.

Then, we selected other three datasets with the class imbalance problem to be evaluated as their original sizes. Instead of a data scarcity scenario as a whole, those datasets contains under-represented classes, imposing a challenge for the learning process.

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

Ethos[4] [106] is a recently proposed dataset for hate speech detection on social media platforms. *TREC-6*[5] [107], a question classification dataset that separates questions into six categories. *ADE*[6] [108], a medical case reports dataset annotated as having drug-related adverse effect mentions. Table 2 summarizes the statistics of the datasets.

Table 2 – Statistics for the datasets evaluated in this work.

Dataset	Original training size	Test size	Number of classes	Category
AG-NEWS	120,000	7,600	4	Balanced
CyberTrolls	12,671	1,563	2	Balanced
SST-2	6,920	1,821	2	Balanced
Ethos	602	151	2	Imbalanced
TREC-6	5,452	500	6	Imbalanced
ADE	18,812	4,704	2	Imbalanced

5.2 Text classification algorithms

We carried out the experiments using four different classification models: Bidirectional Encoder Representations from Transformers (BERT) [32], Convolutional Neural Networks (CNN) [73], Long Short-Term Memory (LSTM) [63], and Multinomial Naive Bayes (NB) [62]. Those classifiers were selected to represent different categories, having NB as a classic text classifier often used as a baseline method [62], LSTM, and CNN as common deep learning classifiers [11], and BERT representing the most recent progress achieving state-of-the-art on numerous NLP tasks including text classification.

Before running the experiments, a preliminary process of hyperparameter tuning was conducted, defining the parameters for all classifiers based on the performance on the corresponding validation set. For NB, we used the default parameters of Scikit-learn distribution [109]. LSTM and CNN implementations were developed with PyTorch [110], initialized with GloVe embeddings [111], trained for a maximum of 20 epochs with early stopping, and Adam [112] optimizer with a learning rate of 1e−3. The BERT classifier was developed with Transformers library [80], fine-tuning the pre-trained model for sequence classification for 2 epochs and learning rate of 5e−5, as recommended [32].

5.3 Augmentation methods

Since 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

[4] <https://github.com/intelligence-csd-auth-gr/Ethos-Hate-Speech-Dataset>
[5] <https://cogcomp.seas.upenn.edu/Data/QA/QC/>
[6] <https://sites.google.com/site/adecorpus/>

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 they provide 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. [113], a Transformer model made publicly available[7]. For the balanced dataset scenario, we generated the translations using beam search, from which the results are deterministic. For the imbalanced scenario, where we need more sentences, we adapted the decoding step to use top-k sampling with $k = 10$ as proposed in Edunov et al. [113], performing similarly to the PREDATOR decoding step.

The second compared technique was EDA, whose code is publicly available[8]. The hyperparameters used to generate new samples were the recommended default, generating 9 new samples for each sample in the original datasets. We generated as many new sentences as needed to make the class distribution balanced on the imbalanced scenario.

[7] <https://pytorch.org/hub/pytorch_fairseq_translation/>
[8] <https://github.com/jasonwei20/eda_nlp>

6 RESULTS

In this section, we explore three perspectives to evaluate the proposed method. First, we discuss the augmentation ratio and classification performance from 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. Lastly, we evaluate our method on a class imbalance scenario with original size datasets.

6.1 Synthesized examples

Table 3 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. The samples were manually selected to show semantically similar sentences. The writing style tends to be similar in synthetic examples while creating variations.

6.2 Augmentation capabilities

The main goal when using augmentation methods, despite the problem, is to improve the predictive performance. We performed 9 augmentation rates ($\alpha \in \{0.1, 0.25, 0.50, 1.0, 1.5, 2.0, 3.0, 6.0, 9.0\}$) on *AG-NEWS*, *CyberTrolls* and *SST-2* datasets. Fig. 5 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.5) since the original accuracy of 65% reaches 84% when augmented. In the same combination of algorithm and dataset, we can observe an improvement of the model quality based on 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 4. 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*. The average improvement is the ratio between the accuracy after augmenting and the original accuracy.

Table 3 – 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 he's 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.*
Ethos	*I really don't like your accent. I don't want to call support and call to rats like you. I want a cute lady with British accent to answer my call. You are third world pieces with no job to this planet*	*The "Militia of the West is a popular ideology, often seen as a racist ideology or racial chauvinism. It does not*
TREC-6	*What North American city sprouts the most parking meters?*	*Where is the U.S. Postal Service headquartered?*
ADE	*Severe diffuse interstitial pneumonitis induced by carmustine (BCNU).*	*erythemaemia following the ingestion of PDE5 inhibitor tadalafil after treatment with lafiparam.*

The results show that our method improved the average accuracy of the classifiers on all combinations of datasets and algorithms. 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 big 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, improving significantly 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

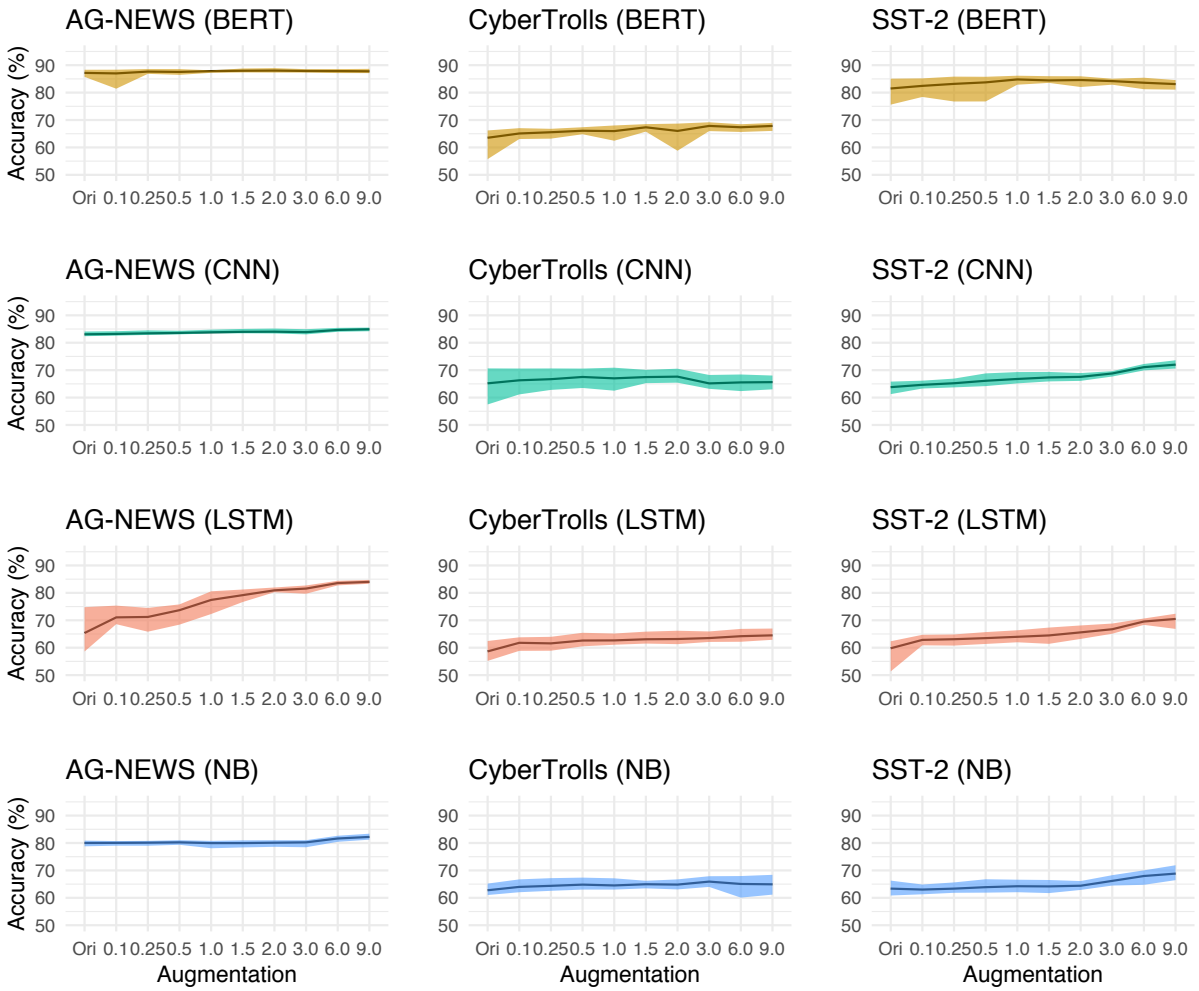

Figure 5 – Performance obtained from different algorithms and datasets using the original size and different augmented sources with PREDATOR.

Table 4 – Augmentation results grouped by datasets for each classifier, highlighting the biggest improvements in bold and the smallest improvements underlined.

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

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 showed to be effective in writing style conditioning and robust on noisy datasets.

6.3 Methods Comparison

We compared PREDATOR with two other augmentation methods: BT and EDA. Each of them provide a specific amount of generated samples by default. BT, using one secondary language and deterministic translations, doubles the size of the original dataset. EDA results on an augmented dataset that is 10 times larger than the original using its default hyperparameters. Thus, for a fair comparison with these methods, the amount of augmentation for PREDATOR was adjusted according to the compared method.

For the first comparison, with BT, 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 double of the size of the original training size for Truth and PREDATOR was configured to augment twice. Fig. 6 presents boxplots of accuracy with all classification algorithms using all datasets grouped by the augmentation method. PREDATOR overcomes BT for all classification algorithms, with a greater accuracy difference for CNN (2.3%). The smallest difference was obtained with BERT (0.2%), the best performing algorithm. The truth was superior to other methods.

To check whether a method outperformed the other ones, 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 [114]. This analysis is presented in Fig. 7, 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.

In the second scenario, PREDATOR was compared to EDA, one of the most recent proposals. A situation similar 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 matching the number of Truth samples and EDA aug-

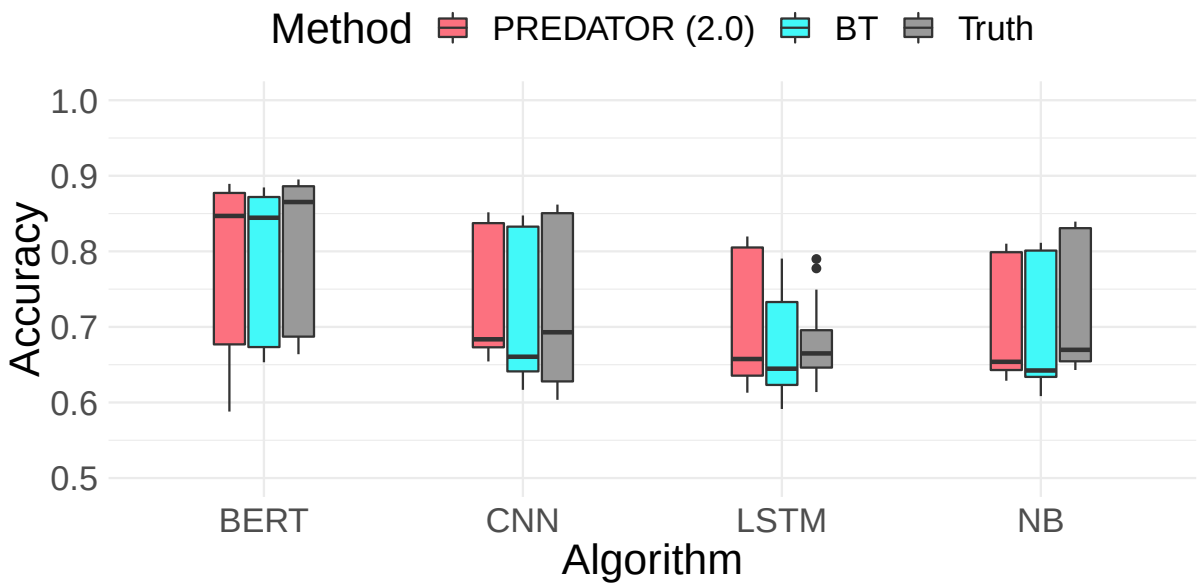

Figure 6 – Accuracy comparison among augmentation methods (PREDATOR and BT) and Truth dataset. The boxplots were computed with the three balanced datasets (*AG-NEWS*, *CyberTrolls* and *SST-2*) grouped by classification algorithms.

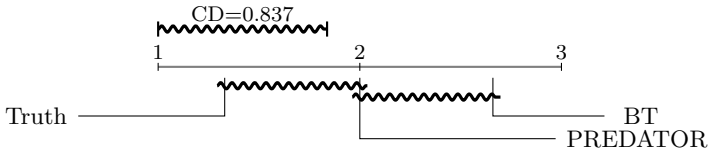

Figure 7 – Comparison of the accuracy values obtained by augmentation methods (PREDATOR and BT) and Truth with the Nemenyi test. Groups within critical distance are not significantly different ($\alpha = 0.05$ and $CD = 0.83$)

mented samples. The most significant difference between the proposed method and EDA was using NB, about 6.3% accuracy, as Fig. 8 shows. Again, BERT achieved the smallest accuracy difference, an average of 0.1%.

Using the same statistical assumptions of the first scenario, we employed Friedman and Nemenyi test to compare PREDATOR, EDA, and Truth. As Fig. 9 shows, it is possible to conclude that Truth and PREDATOR are similar and statistically different from EDA. Thus, we can claim 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,

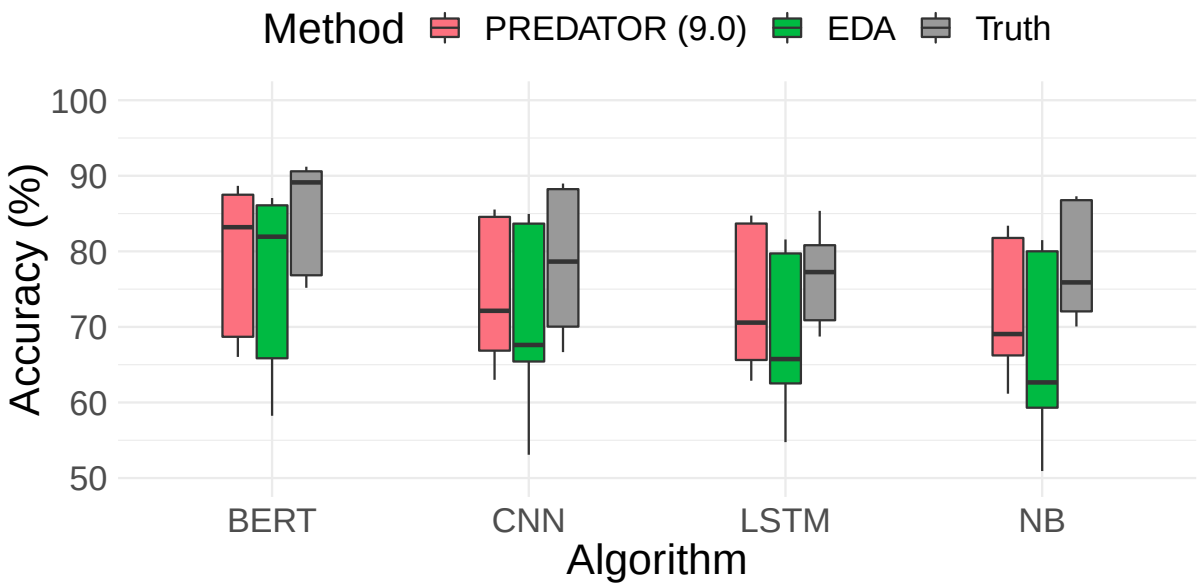

Figure 8 – Accuracy comparison among augmentation methods (PREDATOR and EDA) and Truth dataset. The boxplots were computed with the three balanced datasets (*AG-NEWS*, *CyberTrolls* and *SST-2*) grouped by classification algorithms

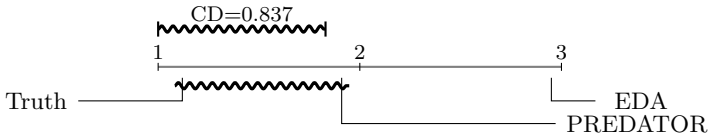

Figure 9 – Comparison of the accuracy values obtained by augmentation methods (PREDATOR and EDA) and Truth with the Nemenyi test. Groups within critical distance are not significantly different ($\alpha = 0.05$ and $CD = 0.83$)

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, where it degraded the accuracy on -0.9% on average, while PREDATOR improved the accuracy by 5.1%, showing a 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.

6.4 Imbalanced class distribution

In addition to the balanced low-data regime scenario, we evaluated our method on imbalanced tasks. The imbalance in the class distribution itself is a challenge for learning algorithms. Instead of having data scarcity and learning from a small number of samples, we have one or more under-represented classes with varying dataset sizes in this scenario. In this scenario, instead of accuracy, the results are better evaluated using F_1-score [115]. Accuracy, in this case, may mask a poor performance because the right predictions for the majority class result in high accuracy, while the performance on minority classes is not relevant. With F_1-score, we aim to balance precision and recall to quantify the classifier's performance better.

The selected datasets have different imbalance ratio, i.e., the ratio between majority and minority class. The dataset with the higher imbalance ratio is *TREC-6*, where the ratio between the most frequent class and the less frequent is 14.6x, followed by *ADE* with 2.4x and *Ethos* with 1.7x. The baseline performance to compare the augmentation techniques is the oversampling strategy, where minority classes are sampled with replacement at random to reach the majority size. The training with all strategies was repeated ten times to get an average result.

The experiments on imbalanced datasets aimed to make the original datasets balanced by generating samples using the evaluated techniques until all classes reach the majority class. For the BT technique, this means that we could not rely on just deterministic translation, or we would need a large number of translation models for different languages. Thus, we employed a sampling strategy when decoding the resulting translation. For both forward and backward translation, we applied top-k sampling to result in different sentences. So we managed to augment minority classes with synthesized examples avoiding repeated samples. For EDA, we took a slightly different approach than the previous scenario. We sampled sentences from minority classes and generated four variations using EDA to include in the augmented dataset. We repeated this process iteratively until all classes were balanced. For PREDATOR experiments, we simply set the α hyperparameter to 1 and $\mathcal{Y}_{ignore}$ to the majority class. Since we are not interested in generating examples for the majority class, we can ignore it in the Generator learning. The $\alpha = 1$ hyperparameter makes the augmentation procedure stop when minority classes reach the majority one.

Table 5 shows the average improvement in F_1-score using each evaluated augmentation technique in comparison to the oversampling performance. The average improvement was obtained considering performance boosting on all four classifiers (BERT, CNN, LSTM and NB). On each dataset, one of the evaluated techniques leads to higher improvement, highlighted in bold. However, it is noteworthy that PREDATOR did not fall into the smallest value in any case. The global average improvement using PREDATOR was

+6.04%, followed by BT with +5.92% and EDA with +4.21%. Even though PREDATOR was not the best case on all datasets, on average, it was more stable leading to higher performance classifiers, with a small advantage over BT.

Table 5 – Augmentation results on imbalanced scenario grouped by datasets for each augmentation technique, highlighting the biggest improvements in bold and the smallest improvements underlined.

Dataset	Augmentation technique	Average improvement
Ethos	Oversampling baseline	0.649
	BT	+9.72%
	EDA	<u>+6.29%</u>
	PREDATOR	**+9.82%**
TREC-6	Oversampling baseline	0.793
	BT	**+6.37%**
	EDA	<u>+4.09%</u>
	PREDATOR	+6.32%
ADE	Oversampling baseline	0.833
	BT	<u>+1.66%</u>
	EDA	**+2.23%**
	PREDATOR	+1.99%

The dataset that took the most advantage of augmentation, in general, was *Ethos*, which was also the most challenging considering baseline performance. *TREC-6* dataset got a smaller improvement, but the baseline performance was already high — especially considering it has six classes. The *ADE* dataset was the one that took the less advantage of augmentations and the only which EDA was the best method, considering the six datasets evaluated in this work.

In this scenario, we also employed Friedman and Nemenyi test to compare PREDATOR, EDA and BT with classifiers trained with oversampling as a baseline. In addition to the oversampling baseline, we included random and majority classifiers, i.e., a classifier that predicts classes randomly independent of the input features and a majority classifier that assigns the majority class to all predictions. Figure 10 presents the result of this statistical analysis, where we can conclude that all augmentation methods are statistically superior to the oversampling baseline. We also can verify that a majority classifier, in this scenario using F_1-score, is the worst-case — even worse than a random classifier. Another conclusion is that PREDATOR and BT are statistically similar, despite the slightly higher average performance improvement of PREDATOR. BT and EDA also are statistically similar. The same is not true for PREDATOR and EDA, agreeing with previous results.

PREDATOR and BT achieved similar results throughout the experiments, but they have different advantages and disadvantages. While PREDATOR is already designed

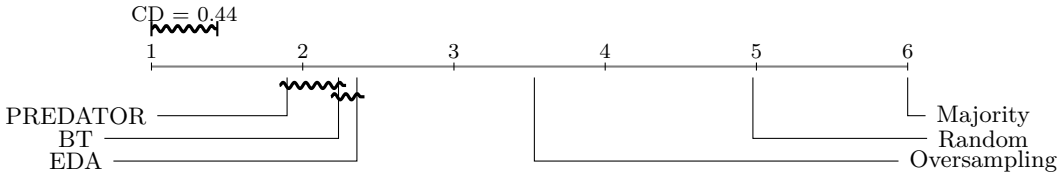

Figure 10 – Comparison of the F_1-score obtained by augmentation methods (PREDATOR, BT and EDA), the oversampling baseline and random and majority classifiers with the Nemenyi test. Groups within critical distance are not significantly different ($\alpha = 0.05$ and $CD = 0.44$)

to generate as many samples as needed, traditional BT needs to be adapted using sampling for decoding strategies to generate more diverse results. This can be a limitation for BT when using vendor APIs (Application Programming Interface), where the results are deterministic. However, this is often surpassed using different secondary languages. BT's advantage is the lack of a training step, which is time-consuming. Depending on the model used on PREDATOR's kernels, this may require too much computational resources, which is the primary motivation to propose the usage of compressed models. On the other hand, this training step may lead to a more appropriate domain adaptation that possibly explains the highest performance of PREDATOR on noisy social media datasets.

7 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 six 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.

Therefore, the results show that PREDATOR is a competitive augmentation method with a carefully designed pipeline to cover either balanced and imbalanced scenarios, including binary and multiclass datasets. We also provide default values to be used on future research and applications obtained through an extensive set of experiments on a variety of textual domains, number of samples and imbalance ratios.

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 by using models pre-trained on other languages or even multilingual models on its modules' kernel. 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.

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

BIBLIOGRAPHY

[1] OLAH, C. *Understanding LSTM Networks*. 2015. Disponível em: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.

[2] VASWANI, A. et al. Attention is all you need. In: GUYON, I. et al. (Ed.). *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017. p. 5998–6008. Disponível em: <http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>.

[3] RUDER, S. *Neural Transfer Learning for Natural Language Processing*. Tese (Doutorado) — National University of Ireland, Galway, 2019.

[4] HU, Z. et al. Learning data manipulation for augmentation and weighting. In: WALLACH, H. et al. (Ed.). *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019. p. 15764–15775. Disponível em: <http://papers.nips.cc/paper/9706-learning-data-manipulation-for-augmentation-and-weighting.pdf>.

[5] CHAWLA, N. V. Data mining for imbalanced datasets: An overview. In: *Data Mining and Knowledge Discovery Handbook*. Springer US, 2009. p. 875–886. Disponível em: <https://doi.org/10.1007/978-0-387-09823-4_45>.

[6] LEMAÎTRE, G.; NOGUEIRA, F.; ARIDAS, C. K. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. JMLR.org, v. 18, n. 1, p. 559–563, jan. 2017. ISSN 1532-4435.

[7] Wong, S. C. et al. Understanding data augmentation for classification: When to warp? In: *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. [S.l.: s.n.], 2016. p. 1–6.

[8] KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2012. p. 1097–1105.

[9] CUBUK, E. D. et al. Autoaugment: Learning augmentation strategies from data. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. [S.l.]: Computer Vision Foundation / IEEE, 2019. p. 113–123.

[10] 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 - Volume 2*. USA: Association for Computational Linguistics, 2011. (HLT '11), p. 271–276. ISBN 9781932432886.

[11] 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 - Volume 1*. Cambridge, MA, USA: MIT Press, 2015. (NIPS'15), p. 649–657.

[12] 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)*. Berlin, Germany: Association for Computational Linguistics, 2016. p. 86–96. Disponível em: <https://www.aclweb.org/anthology/P16-1009>.

[13] 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, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, 2018. p. 452–457. Disponível em: <https://www.aclweb.org/anthology/N18-2072>.

[14] 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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019. p. 6382–6388. Disponível em: <https://www.aclweb.org/anthology/D19-1670>.

[15] ANABY-TAVOR, A. et al. Not enough data? deep learning to the rescue! *arXiv preprint arXiv:1911.03118*, 2019.

[16] KUMAR, V.; CHOUDHARY, A.; CHO, E. Data augmentation using pre-trained transformer models. Association for Computational Linguistics, Suzhou, China, p. 18–26, dez. 2020. Disponível em: <https://www.aclweb.org/anthology/2020.lifelongnlp-1.3>.

[17] 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*. Florence, Italy: Association for Computational Linguistics, 2019. p. 3645–3650. Disponível em: <https://www.aclweb.org/anthology/P19-1355>.

[18] SCHWARTZ, R. et al. Green ai. *ArXiv*, abs/1907.10597, 2019.

[19] IGAWA, R. A. et al. Account classification in online social networks with lbca and wavelets. *Information Sciences*, Elsevier, v. 332, p. 72–83, 2016.

[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)*. Hong Kong, China: Association for Computational Linguistics, 2019. p. 2463–2473. Disponível em: <https://www.aclweb.org/anthology/D19-1250>.

[21] HINTON, G.; VINYALS, O.; DEAN, J. Distilling the knowledge in a neural network. In: *NIPS Deep Learning and Representation Learning Workshop*. [s.n.], 2015. Disponível em: <http://arxiv.org/abs/1503.02531>.

[22] WANG, W. Y.; YANG, D. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. In: *Proceedings of the 2015*

Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, 2015. p. 2557–2563. Disponível em: <https://www.aclweb.org/anthology/D15-1306>.

[23] NISHIMURA, Y. et al. Multi-source neural machine translation with missing data. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, v. 28, p. 569–580, 2020.

[24] YU, A. W. et al. Qanet: Combining local convolution with global self-attention for reading comprehension. In: . [s.n.], 2018. Disponível em: <https://openreview.net/pdf?id=B14TlG-RW>.

[25] SHLEIFER, S. Low resource text classification with ulmfit and backtranslation. *arXiv preprint arXiv:1903.09244*, 2019.

[26] XIA, M. et al. Generalized data augmentation for low-resource translation. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019. p. 5786–5796. Disponível em: <https://www.aclweb.org/anthology/P19-1579>.

[27] AROYEHUN, S. T.; GELBUKH, A. Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In: *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018. p. 90–97. Disponível em: <https://www.aclweb.org/anthology/W18-4411>.

[28] MARIVATE, V.; SEFARA, T. Improving short text classification through global augmentation methods. In: HOLZINGER, A. et al. (Ed.). *Machine Learning and Knowledge Extraction*. Cham: Springer International Publishing, 2020. p. 385–399.

[29] VASWANI, A. et al. Attention is all you need. In: GUYON, I. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. v. 30, p. 5998–6008. Disponível em: <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>.

[30] WU, X. et al. Conditional bert contextual augmentation. In: SPRINGER. *International Conference on Computational Science*. [S.l.], 2019. p. 84–95.

[31] RADFORD, A. et al. Language models are unsupervised multitask learners. v. 1, n. 8, p. 9, 2019.

[32] DEVLIN, J. et al. 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, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019. p. 4171–4186. Disponível em: <https://www.aclweb.org/anthology/N19-1423>.

[33] LEWIS, M. et al. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. Association for Computational Linguistics, Online, p. 7871–7880, jul. 2020. Disponível em: <https://www.aclweb.org/anthology/2020.acl-main.703>.

[34] IBRAHIM, M.; TORKI, M.; EL-MAKKY, N. Imbalanced toxic comments classification using data augmentation and deep learning. In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. [S.l.: s.n.], 2018. p. 875–878.

[35] ZHANG, D. et al. On data augmentation for extreme multi-label classification. *arXiv preprint arXiv:2009.10778*, 2020.

[36] BLANK, G.; REISDORF, B. C. The participatory web: A user perspective on web 2.0. *Information, Communication & Society*, Taylor & Francis, v. 15, n. 4, p. 537–554, 2012.

[37] ŽIŽKA, J.; DAŘENA, F.; SVOBODA, A. *Text Mining with Machine Learning: Principles and Techniques*. [S.l.]: CRC Press, 2019.

[38] ALLAHYARI, M. et al. A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*, 2017.

[39] FELDMAN, R.; SANGER, J. et al. *The text mining handbook: advanced approaches in analyzing unstructured data*. [S.l.]: Cambridge university press, 2007.

[40] WITTEN, I. H.; FRANK, E. Data mining: practical machine learning tools and techniques with java implementations. *Acm Sigmod Record*, ACM New York, NY, USA, v. 31, n. 1, p. 76–77, 2002.

[41] BERRY, M. W.; CASTELLANOS, M. Survey of text mining. *Computing Reviews*, Springer, v. 45, n. 9, p. 548, 2004.

[42] KO, Y. A study of term weighting schemes using class information for text classification. In: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: Association for Computing Machinery, 2012. (SIGIR '12), p. 1029–1030. ISBN 9781450314725. Disponível em: <https://doi.org/10.1145/2348283.2348453>.

[43] BISHOP, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 0387310738.

[44] AGGARWAL, C. C.; ZHAI, C. A survey of text classification algorithms. In: *Mining text data*. [S.l.]: Springer, 2012. p. 163–222.

[45] LIU, B.; ZHANG, L. A survey of opinion mining and sentiment analysis. In: _____. *Mining Text Data*. Boston, MA: Springer US, 2012. p. 415–463. ISBN 978-1-4614-3223-4. Disponível em: <https://doi.org/10.1007/978-1-4614-3223-4_13>.

[46] SUMATHY, K. L.; CHIDAMBARAM, M. Article: Text mining: Concepts, applications, tools and issues - an overview. *International Journal of Computer Applications*, v. 80, n. 4, p. 29–32, October 2013. Full text available.

[47] LAURÍA, E. J. M.; MARCH, A. D. Combining bayesian text classification and shrinkage to automate healthcare coding: A data quality analysis. *J. Data and Information Quality*, Association for Computing Machinery, New York, NY, USA, v. 2, n. 3, dez. 2011. ISSN 1936-1955. Disponível em: <https://doi.org/10.1145/2063504.2063506>.

[48] ABONIZIO, H. Q. et al. Language-independent fake news detection: English, portuguese, and spanish mutual features. *Future Internet*, v. 12, n. 5, 2020. ISSN 1999-5903. Disponível em: <https://www.mdpi.com/1999-5903/12/5/87>.

[49] SHAMS, R. Semi-supervised classification for natural language processing. *arXiv preprint arXiv:1409.7612*, 2014.

[50] BLUM, A.; MITCHELL, T. Combining labeled and unlabeled data with co-training. In: *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*. New York, NY, USA: Association for Computing Machinery, 1998. (COLT' 98), p. 92–100. ISBN 1581130570. Disponível em: <https://doi.org/10.1145/279943.279962>.

[51] RUDER, S.; PLANK, B. Strong baselines for neural semi-supervised learning under domain shift. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, 2018. p. 1044–1054. Disponível em: <https://www.aclweb.org/anthology/P18-1096>.

[52] YAROWSKY, D. Unsupervised word sense disambiguation rivaling supervised methods. In: *33rd annual meeting of the association for computational linguistics*. [S.l.: s.n.], 1995. p. 189–196.

[53] MCCLOSKY, D.; CHARNIAK, E.; JOHNSON, M. Reranking and self-training for parser adaptation. In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*. USA: Association for Computational Linguistics, 2006. (ACL-44), p. 337–344. Disponível em: <https://doi.org/10.3115/1220175.1220218>.

[54] MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2013. p. 3111–3119.

[55] LAN, Z. et al. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[56] ANDERSON, J. Diagnosis by logistic discriminant function: further practical problems and results. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, Wiley Online Library, v. 23, n. 3, p. 397–404, 1974.

[57] PEARSON, E. S. Bayes' theorem, examined in the light of experimental sampling. *Biometrika*, JSTOR, p. 388–442, 1925.

[58] DOMINGOS, P.; PAZZANI, M. On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning*, Springer, v. 29, n. 2-3, p. 103–130, 1997.

[59] HELLERSTEIN, J. L. et al. *Recognizing end-user transactions in performance management*. [S.l.]: IBM Thomas J. Watson Research Division Hawthorne, NY, 2000.

[60] KOWSARI, K. et al. Text classification algorithms: A survey. *Information*, v. 10, n. 4, 2019. ISSN 2078-2489. Disponível em: <https://www.mdpi.com/2078-2489/10/4/150>.

[61] JONES, K. S. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, MCB UP Ltd, 1972.

[62] WANG, S.; MANNING, C. D. Baselines and bigrams: Simple, good sentiment and topic classification. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2*. [S.l.], 2012. p. 90–94.

[63] HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Comput.*, MIT Press, Cambridge, MA, USA, v. 9, n. 8, p. 1735–1780, nov. 1997. ISSN 0899-7667. Disponível em: <https://doi.org/10.1162/neco.1997.9.8.1735>.

[64] BENGIO, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, v. 5, n. 2, p. 157–166, 1994.

[65] MINAEE, S. et al. Deep learning based text classification: A comprehensive review. *arXiv preprint arXiv:2004.03705*, 2020.

[66] ABDEL-NASSER, M.; MAHMOUD, K. Accurate photovoltaic power forecasting models using deep lstm-rnn. *Neural Computing and Applications*, Springer, v. 31, n. 7, p. 2727–2740, 2019.

[67] Athiwaratkun, B.; Stokes, J. W. Malware classification with lstm and gru language models and a character-level cnn. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2017. p. 2482–2486.

[68] BRAZ, F. A. et al. Document classification using a bi-lstm to unclog brazil's supreme court. *arXiv preprint arXiv:1811.11569*, 2018.

[69] ZHOU, X.; WAN, X.; XIAO, J. Attention-based LSTM network for cross-lingual sentiment classification. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, 2016. p. 247–256. Disponível em: <https://www.aclweb.org/anthology/D16-1024>.

[70] BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[71] GIACHANOU, A.; ROSSO, P.; CRESTANI, F. Leveraging emotional signals for credibility detection. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. [S.l.: s.n.], 2019. p. 877–880.

[72] LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998.

[73] KIM, Y. Convolutional neural networks for sentence classification. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014. p. 1746–1751. Disponível em: <https://www.aclweb.org/anthology/D14-1181>.

[74] ZHANG, X.; ZHAO, J.; LECUN, Y. Character-level convolutional networks for text classification. In: CORTES, C. et al. (Ed.). *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015. p. 649–657. Disponível em: <http://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classification.pdf>.

[75] COLLOBERT, R. et al. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, JMLR.org, v. 12, n. null, p. 2493–2537, nov. 2011. ISSN 1532-4435.

[76] LECUN, Y. et al. Object recognition with gradient-based learning. In: _____. *Shape, Contour and Grouping in Computer Vision*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999. p. 319–345. ISBN 978-3-540-46805-9. Disponível em: <https://doi.org/10.1007/3-540-46805-6_19>.

[77] GOLDBERG, Y. A primer on neural network models for natural language processing. *J. Artif. Int. Res.*, AI Access Foundation, El Segundo, CA, USA, v. 57, n. 1, p. 345–420, set. 2016. ISSN 1076-9757.

[78] JACOVI, A.; SHALOM, O. S.; GOLDBERG, Y. Understanding convolutional neural networks for text classification. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, 2018. p. 56–65. Disponível em: <https://www.aclweb.org/anthology/W18-5408>.

[79] HOULSBY, N. et al. Parameter-efficient transfer learning for NLP. In: CHAUDHURI, K.; SALAKHUTDINOV, R. (Ed.). *Proceedings of the 36th International Conference on Machine Learning*. Long Beach, California, USA: PMLR, 2019. (Proceedings of Machine Learning Research, v. 97), p. 2790–2799. Disponível em: <http://proceedings.mlr.press/v97/houlsby19a.html>.

[80] WOLF, T. et al. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[81] CLARK, K. et al. ELECTRA: Pre-training text encoders as discriminators rather than generators. In: *ICLR*. [s.n.], 2020. Disponível em: <https://openreview.net/pdf?id=r1xMH1BtvB>.

[82] RADFORD, A. et al. *Improving language understanding by generative pre-training*. 2018.

[83] SENNRICH, R.; HADDOW, B.; BIRCH, A. Neural machine translation of rare words with subword units. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, 2016. p. 1715–1725. Disponível em: <https://www.aclweb.org/anthology/P16-1162>.

[84] ZELLERS, R. et al. Defending against neural fake news. In: WALLACH, H. et al. (Ed.). *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019. p. 9054–9065. Disponível em: <http://papers.nips.cc/paper/9106-defending-against-neural-fake-news.pdf>.

[85] SOLAIMAN, I. et al. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*, 2019.

[86] LEE, J.-S.; HSIANG, J. Patent claim generation by fine-tuning openai gpt-2. *arXiv preprint arXiv:1907.02052*, 2019.

[87] ADELANI, D. I. et al. Generating sentiment-preserving fake online reviews using neural language models and their human-and machine-based detection. In: SPRINGER. *International Conference on Advanced Information Networking and Applications*. [S.l.], 2020. p. 1341–1354.

[88] GEHRMANN, S.; STROBELT, H.; RUSH, A. GLTR: Statistical detection and visualization of generated text. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Florence, Italy: Association for Computational Linguistics, 2019. p. 111–116. Disponível em: <https://www.aclweb.org/anthology/P19-3019>.

[89] HOLTZMAN, A. et al. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

[90] LIU, Y. et al. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[91] YANG, Q. et al. *Transfer Learning*. Cambridge: Cambridge University Press, 2020.

[92] ZOPH, B. et al. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*, 2016.

[93] 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)*. Melbourne, Australia: Association for Computational Linguistics, 2018. p. 328–339. Disponível em: <https://www.aclweb.org/anthology/P18-1031>.

[94] MALTE, A.; RATADIYA, P. Evolution of transfer learning in natural language processing. *arXiv preprint arXiv:1910.07370*, 2019.

[95] ZAFRIR, O. et al. Q8bert: Quantized 8bit bert. *arXiv preprint arXiv:1910.06188*, 2019.

[96] HAN, S.; MAO, H.; DALLY, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[97] SANH, V. et al. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[98] WANG, A. et al. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, 2018. p. 353–355. Disponível em: <https://www.aclweb.org/anthology/W18-5446>.

[99] BROWN, T. B. et al. Language models are few-shot learners. 2020.

[100]BENGIO, Y. et al. A neural probabilistic language model. *Journal of machine learning research*, v. 3, n. Feb, p. 1137–1155, 2003.

[101]GOKASLAN, A.; COHEN, V. *OpenWebText Corpus*. 2019. <http://Skylion007.github.io/OpenWebTextCorpus>.

[102]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)*. Melbourne, Australia: Association for Computational Linguistics, 2018. p. 889–898. Disponível em: <https://www.aclweb.org/anthology/P18-1082>.

[103]ACKLEY, D. H.; HINTON, G. E.; SEJNOWSKI, T. J. A learning algorithm for boltzmann machines. *Cognitive Science*, v. 9, n. 1, p. 147 – 169, 1985. ISSN 0364-0213. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0364021385800124>.

[104]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*. [S.l.: s.n.], 2013. p. 1631–1642.

[105]YOGATAMA, D. et al. Generative and discriminative text classification with recurrent neural networks. *arXiv preprint arXiv:1703.01898*, 2017.

[106]MOLLAS, I. et al. Ethos: an online hate speech detection dataset. *ArXiv*, abs/2006.08328, 2020.

[107]LI, X.; ROTH, D. Learning question classifiers. In: *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*. USA: Association for Computational Linguistics, 2002. (COLING '02), p. 1–7. Disponível em: <https://doi.org/10.3115/1072228.1072378>.

[108]GURULINGAPPA, H. et al. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics*, v. 45, n. 5, p. 885 – 892, 2012. ISSN 1532-0464. Text Mining and Natural Language Processing in Pharmacogenomics. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1532046412000615>.

[109]PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

[110]PASZKE, A. et al. Pytorch: An imperative style, high-performance deep learning library. In: WALLACH, H. et al. (Ed.). *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019. p. 8024–8035. Disponível em: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.

[111]PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. [S.l.: s.n.], 2014. p. 1532–1543.

[112]KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[113]EDUNOV, S. et al. Understanding back-translation at scale. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018. p. 489–500. Disponível em: <https://www.aclweb.org/anthology/D18-1045>.

[114]DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, v. 7, n. Jan, p. 1–30, 2006.

[115]JENI, L. A.; COHN, J. F.; TORRE, F. D. L. Facing imbalanced data–recommendations for the use of performance metrics. In: *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*. [S.l.: s.n.], 2013. p. 245–251.

WORKS PUBLISHED BY THE AUTHOR

1. Abonizio, H. Q.; Júnior, S. B. **Pre-trained Data Augmentation for Text Classification**. Brazilian Conference on Intelligent Systems 2020 (BRACIS). Lecture Notes in Computer Science, vol 12319. Springer, Cham. (Qualis CC 2016, B2)

2. Abonizio, H. Q.; de Morais, J. I.; Tavares, G. M.; Barbon Junior, S.. **Language-Independent Fake News Detection: English, Portuguese, and Spanish Mutual Features.** Future Internet, v. 12 n. 5 p. 87, 2020. (Qualis 2019, B1)

3. de Morais, J. I.; Abonizio, H. Q.; Tavares, G. M.; da Fonseca, A. A.; Barbon Jr, S. **Deciding among Fake, Satirical, Objective and Legitimate news: A multi-label classification system.** Proceedings of the XV Brazilian Symposium on Information Systems. 2019. (Qualis CC 2016, B2)

4. de Morais, J. I.; Abonizio, H. Q.; Tavares, G. M.; da Fonseca, A. A.; Barbon Jr, S. **A Multi-label Classification System to Distinguish among Fake, Satirical, Objective and Legitimate News in Brazilian Portuguese.** iSys-Revista Brasileira de Sistemas de Informação, v. 13, n. 4, 2020. (Qualis CC 2016, B3)

5. Leão, A. L. F.; Abonizio, H. Q.; Júnior, S. B.; Kanashiro, M. **Identificação de composições da paisagem urbana: uma abordagem de deep learning.** Revista de Morfologia Urbana, v. 8, n. 1, p. e00140, 2020. (Qualis 2019, B1)

6. Leão, A. L. F.; Abonizio, H. Q.; Reis, R. S.; Kanashiro, M. **Walkability variables: an empirical study in Rolândia-PR, Brazil.** Ambiente Construído, v. 20, n. 2 p. 475-488, 2020. (Qualis 2019, A3)