IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 December 2023, accepted 16 February 2024, date of publication 26 February 2024, date of current version 4 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3369918

RESEARCH ARTICLE

Text Data Augmentation Techniques for Word Embeddings in Fake News Classification

JOZEF KAPUSTA[1,2], DÁVID DRŽÍK[1], KIRSTEN ŠTEFLOVIČ[1], AND KITTI SZABÓ NAGY[1]
[1]Faculty of Natural Sciences and Informatics, Constantine the Philosopher University in Nitra, 949 01 Nitra, Slovakia
[2]Institute of Security and Computer Science, University of the National Education Commission, 30-084 Kraków, Poland

Corresponding author: Jozef Kapusta (jkapusta@ukf.sk)

This work was supported in part by the Scientific Grant Agency of the Ministry of Education of Slovak Republic and Slovak Academy of Sciences under Contract VEGA-1/0734/24, and in part by the European Commission ERASMUS+ Program 2021 under Grant 2021-1-SK01-KA220-HED-000032095.

ABSTRACT Contemporary language models heavily rely on large corpora for their training. The larger the corpus, the better a model can capture various semantic relationships. The issue at hand appears to be the limited scope of the corpora used. One potential solution to this problem is the application of data augmentation techniques to expand the existing corpus. Data augmentation encompasses several techniques for corpus augmentation. In this article, we delve deeper into the analysis of three techniques: Synonym Replacement, Back Translation, and Reduction of Function Words. Utilizing these three techniques, we prepared diverse versions of the corpus employed for training Word2Vec Skip-gram models. These techniques were validated through extrinsic evaluation, wherein Word2Vec Skip-gram models were used to generate word vectors for classifying fake news articles. Performance measures of the generated classifiers were analyzed. The study highlights significant statistical differences in classifier outcomes between augmented and original corpora. Specifically, Back Translation significantly enhances accuracy, notably with Support Vector and Bernoulli Naive Bayes models. Conversely, the Reduction of Function Words (FWD) improves Logistic Regression, while the original corpus excels in Random Forest classification. The article also includes an intrinsic evaluation involving lexical semantic relations between word pairs. The intrinsic evaluation highlights nuanced differences in semantic relations across augmented corpora. Notably, the Back Translation (BT) corpus better aligns with established lexical resources, showcasing promising improvements in understanding specific semantic relationships.

INDEX TERMS Back translation, function word deletion, synonym replacement, text data augmentation, Word2Vec, word embeddings.

I. INTRODUCTION

Data Augmentation (DA) can be defined as any technique for increasing the diversity of training examples without explicitly collecting new data [1]. The goal of data augmentation is to enhance the performance and robustness of machine learning models by exposing them to a wider range of variations and situations. Data augmentation has been successfully applied in various domains such as computer vision, natural language processing, and speech recognition. Moreover, data augmentation can also help mitigate overfitting by introducing more variation into the training data and preventing the model from memorizing the training examples. Despite unquestionable success in computer vision tasks, NLP research has not yet benefited as largely from DA systems [2]. There are a few popular techniques for text data augmentation: Back Translation, Synonym replacement, Paraphrasing, Random Insertion, Random Swap, Random Deletion, etc. Of course, there are a large number of data augmentation techniques, a clear summary of them can be found in papers [1], [2]. The use of text vectorization techniques is nowadays a necessity for many

The associate editor coordinating the review of this manuscript and approving it for publication was Varuna De Silva.

classification tasks in the field of natural language processing. Word embedding models like Word2Vec, Doc2Vec, and Glove, which continue to be widely used, rely on the semantic similarity among words.

From a pragmatic standpoint, word vectors find utility across various applications. This article, however, will center its attention on classification tasks. The objective of this article is to determine the extent to which data augmentation techniques enhance performance metrics in classification tasks. Within this article, we will scrutinize a subset of commonly employed data augmentation techniques to ascertain which among them yields the greatest assistance in classification tasks. The following techniques have been chosen from the pool of available options:

- **Synonym replacement (SR)** is a technique in which we replace a word by one of its synonyms. We use WordNet, a large linguistic database, to identify relevant synonyms.
- **Back translation (BT)** is a simple and effective data augmentation technique for textual data. It involves the translation of the original text into another language and subsequently translating it back into the source language. This process typically results in minor variations from the original text while retaining essential information.
- **Reduction of function words (FWD)** is similar to the Random Deletion technique. This technique randomly removes words from the sentence, following some probability parameter. In our scenario, the probability parameter was restricted to two values, one for function words and the other for content (non-function) words.
- **Original serves** as our reference, representing the unaltered corpus without the application of any data augmentation technique. We compared these techniques against the Original as our baseline for evaluation [1], [2].

The application of the mentioned techniques in addressing classification tasks is demanding in terms of both time and computational resources. Hence, their application is subject to debate. In our article, we aim to find an answer to the question of whether the application of these techniques impacts the enhancement of performance metrics in classification models. Apart from this inquiry, we also seek to identify which of these techniques has the most significant influence on the outcomes of classification models.

The contribution of our article lies in determining the significance of these techniques for classification tasks. Establishing their importance will assist us and other researchers in choosing which techniques to incorporate into the preprocessing phase for resolving future classification tasks.

We apply the mentioned techniques in the preparation of the corpus. Using the thus prepared corpus, the Word2Vec Skip-gram model will be trained, which will be used for word embeddings for the classification task. Fake news classification for the WELFake dataset was chosen as the classification task. Subsequently, we will evaluate the performance measures of the classification itself. I.e. we will not evaluate Data Augmentation techniques directly, but we will use them to solve the classification task and evaluate the success of the classification.

For the purpose of comparing the effectiveness of corpus preparation techniques, in the case of large language models, several approaches are available. Nazir et al. [3] assess the performance using the word similarity datasets WordSim-353 [4] and SimLex-999 [5], which contain computed similarities of selected word pairs. The computation of word similarities is often presented in educational examples focused on word vectors.

We will proceed according to the following methodology (figure 1):

1. Preprocessing of the corpus for training
2. Creation of a corpus (as a reference corpus) without the application of text data augmentation techniques (**Original**)
3. Preparation of augmentation corpuses using data augmentation techniques:
   - **SR corpus** – corpus prepared using the Synonym Replacement technique
   - **FWD corpus** - corpus prepared by Function Words Deletion
   - **BT corpus** – corpus prepared by Back Translation
4. From the four corpora created, 4 Word2Vec Skip-gram models were trained
5. Preprocessing of the dataset for the classification task
6. Creation of 4 input vectors to classifiers using 4 Word2Vec Skip-gram models
7. Creation of classification models using the following classification methods:
   - **Random Forest Classifier**
   - **Logistic Regression**
   - **BernoulliNB**
   - **SVC Classifier**
8. Obtaining model results according to k-fold cross validation
9. Identification and comparison of the performance of the created models

The article follows the following structure. In the second section (Related work), we provide a concise summary of existing research in the field of text data augmentation and other techniques related to word embedding models. In the third section (Materials and methods), we provide a detailed description of the data files used and their preprocessing, the application of synonym replacement techniques, back translation, and the deletion of functional words to create corpora. Furthermore, we describe the creation of vectors from these corpora and the classification of fake news based on these vectors. In the fourth section (Results), we present and analyze the achieved results through both internal and external evaluation. In the fifth section (Discussion),

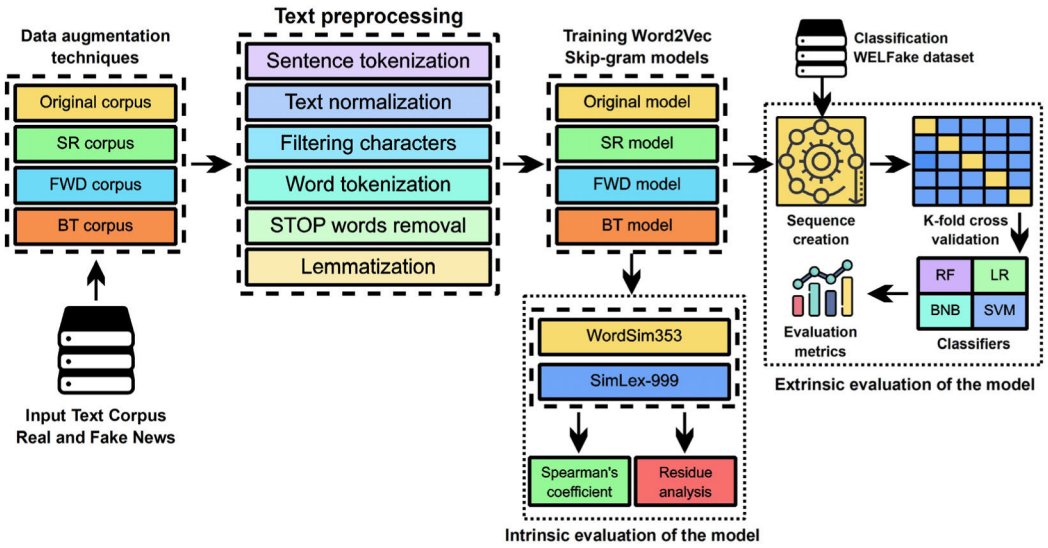


FIGURE 1. Methodology steps.

we evaluate our results and compare them with the findings of other authors.

This article provides a critical overview of the impact of text data augmentation techniques such as Synonym Replacement, Back Translation, and Reduction of Function Words on Word Embeddings outcomes in classifying fake news. The novelty of this research lies in the comprehensive examination of how these techniques notably enhance classification results. In addition to extrinsic evaluation, which analyzes these techniques' influence on classifier outcomes, the article delves into nuanced semantic differences among augmented corpora through intrinsic evaluation. This research contributes to identifying novel advancements in data augmentation techniques concerning the classification of fake news, emphasizing their significant influence on the performance of multiple classification models.

II. RELATED WORK

Text augmentation techniques have gained significant attention in recent years to improve the performance of text-based classification tasks. This section discusses relevant studies in the field of text augmentation, specifically focusing on the use of text augmentation techniques such as synonym augmentation, backtranslation and other text augmentation techniques in conjunction with the word embedding models.

Wei and Zou [6] introduced EDA, a set of text augmentation techniques including synonym replacement, random insertion, random swap, and random deletion. EDA improves text classification performance, especially on smaller datasets, demonstrating its effectiveness with convolutional and recurrent neural networks.

Synonym augmentation is a text data augmentation technique in natural language processing that replaces words in a text with their synonyms while maintaining the text's overall meaning. Commonly, it utilizes the WordNet thesaurus [7] to obtain synonyms and antonyms for words in the text, enhancing the diversity of data for machine learning models.

Marivate et al. [8] focused on evaluating the impact of various text augmentation techniques using diverse datasets, including social media and formal news articles to offer guidance to practitioners and researchers in classification tasks. The research highlights the effectiveness of Word2Vec-based augmentation as a practical choice, particularly when formal synonym models like WordNet are limited. Additionally, incorporating mix-up augmentation improves performance and mitigates overfitting in deep learning models. However, the study notes the cost challenges associated with round-trip translation, limiting its accessibility for both standard and resource-constrained scenarios.

Another research paper [9] introduces a novel data augmentation for labelled sentences called contextual augmentation. Their approach leverages the invariance of sentences to word replacements with paradigmatic relations, stochastically replacing words based on predictions from a bidirectional language model.

A novel approach, SynoExtractor [10], was presented for synonym extraction, crucial for natural language processing systems. Unlike traditional methods relying solely on word embeddings, their SynoExtractor pipeline addresses this challenge by filtering similar word embeddings based on linguistic rules to specifically identify synonyms. Experimentation with KSUCCA and Gigaword embeddings, along with training using CBOW and SG models, demonstrates significant precision enhancement over cosine similarity alone. Evaluation against Alma'any Arabic synonym thesauri shows a 0.605 mean average precision (MAP) for the King Saud University Corpus of Classical Arabic, marking a 21% improvement over the baseline, and a 0.748 MAP for the Gigaword corpus, indicating a 25% improvement. Outperforming the Sketch Engine thesaurus by 32% in terms

of MAP, SynoExtractor offers promising results for synonym extraction across languages.

As we delve into the related work, we will not only explore text augmentation techniques but also examine the intersection of these techniques with the critical task of fake news identification, where enhancing the quality and quantity of textual data plays a pivotal role in improving detection and classification accuracy.

Salah et al. [11] explored text augmentation for stance and fake news detection [12], assessing its impact on various classification algorithms. Their experimental analysis quantified the augmentation's actual contribution, pinpointing optimal (classifier, augmentation technique) pairs. They also introduced a novel ensemble approach combining bagging and stacking, utilizing text augmentation to enhance base classifiers' diversity and performance. Evaluation on real-world datasets demonstrated the approach's superior accuracy compared to state-of-the-art methods. Moreover, their investigation revealed that text augmentation effectively mitigates class imbalance issues, even under severe conditions, significantly improving predictive performance for minority classes.

Bucos et al. [13] investigated the utilization of a Romanian data source, different classifiers, and text data augmentation techniques for the implementation of a fake news detection system. The focus of the study centers on the application of text data augmentation, specifically Back-translation and Easy Data Augmentation (EDA), to enhance the efficiency of fake news detection tasks. The findings reveal that both BT and EDA techniques effectively improved classifier performance.

Keya et al. [14] introduced a novel approach that employs the exploration of fake news detection techniques, addressing challenges posed by biased datasets and the labor-intensive nature of manual labeling. In response, this study presents a text augmentation technique utilizing the Bidirectional Encoder Representation of Transformers (BERT) language model to create a synthetic fake data-enriched dataset. The resulting AugFake-BERT model outperforms twelve state-of-the-art models, achieving an accuracy of 92.45%. Performance metrics such as accuracy, precision, recall, and f1-score underscore the impact of a balanced dataset on classification efficacy, contributing valuable insights to the evolving field of fake news detection methodologies.

Haralabopoulos et al. [15] introduced a novel text augmentation techniques those improve classification while preserving crucial corpus statistics like term frequency and class distribution. Their approach includes negation-based augmentations, such as antonym replacement and negation insertion, which require mutually exclusive classes. Through comprehensive evaluations across diverse datasets, their permutation augmentation technique demonstrated a substantial 4.1 % increase in classification accuracy over the baseline and a 0.2 % improvement compared to the best-performing prior augmentation technique. Additionally, their antonym and negation augmentations consistently delivered enhancements of at least 0.35 % compared to permutation augmentation, highlighting the effectiveness of their novel techniques.

Dahou et al. [16] were dedicated to the detection of fake news posts in the Arabic language. They used the Multi-Task Learning (MTL) technique and a model based on the transformer architecture, augmented with a modified Nutcracker Optimization Algorithm, to extract contextual features from Arabic social media posts. Through extensive evaluation of various datasets of Arabic social media posts, they achieve an accuracy rate of 87% for binary and 69% for multiple classification. The authors claim that their methods also outperforms all compared algorithms and serve as a robust tool in the fight against the spread of misinformation.

In their work, Hua et al. [17] essentially developed a new framework that integrates textual and visual features for detecting fake news. Their approach involves using the BERT language model (based on transformer architecture) for text processing and employing a technique for text data augmentation through back-translation to acquire general topic characteristics. Additionally, they utilize contrastive learning to obtain improved multimodal representations of news by leveraging similar past news. The results of their research indicated that their methods outperforms current models in detecting fake news by 3.1% on the Mac F1 score.

Marwat et al. [18] addressed the automated analysis of end-user reviews on social media concerning products. They proposed the SentiDecpective approach for categorizing reviews into negative, positive, and neutral sentiments and for detecting deceptive information in ratings. Gathering over 11,000 comments from online stores, they employed content analysis and machine learning algorithms (Multinomial Naive Bayes, Logistic Regression, Gradient Boosting Classifier, Linear SVC, and Random Forest Classifier) for classification. They developed a tool displaying collective customer opinions via a graph, aiding decision-making. Their approach achieved an average precision of 94.01% in identifying positive sentiments from online user feedback.

Khan et al. [19] laid the groundwork by highlighting the significance of user forums in providing valuable insights for software evolution based on end-user reviews. Building on this, our work introduces the Crowd-based Requirements Engineering by Valuation Argumentation (CrowdRE-VArg) approach. This approach addresses the challenge of fragmented user-generated information on Reddit forums by systematically analyzing discussions, identifying conflict-free new features, design alternatives, or issues, and reaching rationale-based requirements decisions. Utilizing a bipolar gradual valuation argumentation framework, extended from abstract argumentation and valuation frameworks, our automated CrowdRE-VArg approach negotiates conflicts among crowd-users in real-time. We demonstrate the proof-of-concept through a sample conversation topic from the Reddit forum on the Google Map mobile application, employing natural language processing (TF-IDF method)

and machine learning algorithms to automatically prioritize requirements-related information for software engineers.

III. MATERIALS AND METHODS

A. DATASETS AND THEIR PREPROCESSING

In our article, we used two freely available data files. The first text dataset or corpus [20] contains 143,000 English articles from 15 American journals. The selection of this dataset was deliberate, as we aimed to address the classification task of detecting fake news in everyday contexts where misinformation proliferates. We chose this dataset because we believe it comprehensively covers a wide range of topics encountered in daily life. Before using this text, it is necessary to preprocess it as precisely as possible. We started by tokenizing the text into sentences using the NLTK library [21]. This dataset contains a total of up to 704,357 sentences. Subsequently, we converted the text in each sentence to lowercase letters, cleaned the text from hyperlinks, from white characters, and from special characters that are not numbers or letters. We split the sentences into words. We then removed words that contained numbers as well as numbers themselves, ensuring that only words with linguistic meaning appeared in the dictionary. We have also removed stop words that appear too often in the text but have no meaning in themselves. The last step of text preprocessing was lemmatization using WordNetLemmatizer, which reduced the number of words in the dictionary since they can appear in the dictionary in different forms but carry the same meaning. For further work, we did not use all this text, but only part of it, namely the first 10,000 sentences.

The second dataset, WELFake [22], contains a total of 72,134 articles correctly labeled as fake news (35,028) and real news (37,106). Despite its limitations, we've opted to use this dataset because it's commonly applied in smaller classification tasks thanks to its accurate labeling, making it a practical choice despite its drawbacks. We also preprocessed this file in the same way as the first one. We will use this classification dataset to verify the quality of the vector model.

B. CREATING A LIST OF SYNONYMS FROM WORDNET SYNSETS

In this article, we want to experiment with word vectors and see if replacing words with their synonyms improves their quality. Therefore, we need to create a list of synonyms for the words in our examined text.

To create a list of synonyms, we used the sentences from the first data set, because in this text we will replace words with their synonyms, so it does not make sense to look for synonyms for words that we will not need later. The process of creating synonyms was as follows. We gradually went through the individual words of all preprocessed sentences. Then, for each word, we searched for synsets (a list of synonyms) using the function synsets through the WordNet library [7] from the NLTK library [20]. If no synonyms were found for the specified word in the WordNet dictionary, we tried to lemmatize the word again, this time using the WordNet morphy function, which tries to find its lemmatized form, which is listed in the WordNet dictionary. If the word is found, a list of synonyms for the modified word is searched.

After finding a synset for our word, we went through each synonym in turn and filtered this list based on semantic similarity (by using path similarity, the result value is between 0 and 1, where a value of 1 indicates maximum similarity) to our word. If this similarity was higher than 0.5, we kept this synonym, otherwise, we removed this synonym from the synset. In the end, we lemmatized all valid synonyms and stored a list with a maximum length of 5 synonyms per word. We saved the list as a binary file using the Pickle library.

C. REPLACING WORDS IN THE ORIGINAL SENTENCES WITH THEIR SYNONYMS

After creating a list of synonyms (a word and a list of synonyms for this word), of which there were 9,704, we decided to gradually replace each word for which we registered a list of synonyms in all sentences. We used a technique known as Data Augmentation - replacement by synonyms [6]. In our experiment, we go through all words in all sentences and check for each word whether it contains a list of synonyms. If so, then we will duplicate the whole sentence and replace the one specific word with each synonym for this word. As a result, we will have several times more sentences than the original ones. Just to give an idea, from the original 10,000 sentences, we created 175,354 sentences with this technique, which is more than 17 times.

D. BACKTRANSLATION AS AN AUGMENTATION METHOD

Back translation is a powerful data augmentation technique used in Natural Language Processing (NLP) to enhance the diversity and size of a given dataset. The technique involves translating a sentence or text from its original language into another language using a machine translation model. Subsequently, the translated text is translated back into the original language. The purpose of this process is to introduce variations in the phrasing and wording while preserving the underlying meaning of the text [1].

This technique leverages the strengths of machine translation models to generate new and semantically equivalent versions of the original text. By translating the sentence into a different language and then back, the model may encounter linguistic nuances and different word choices. This forces the model to learn to reconstruct the initial input, promoting robustness and improving the generalization capability of NLP models [1].

In this study, the Backtranslation data augmentation technique using the EasyNMT framework was employed to enhance the diversity and size of our dataset. The Opus-MT pre-trained model [23] was used to translate the monolingual NLP dataset from the source language to a target language. The translated sentences in the target language were then re-translated back to the source language using the same model. The resulting backtranslated sentences, together

with the original sentences, were combined to augment our dataset. By integrating these new variations, an aim was made to increase the diversity of our data, reduce overfitting, and enhance the generalization capability of our NLP model.

E. CREATION OF WORD VECTORS

Word vectors are one way to represent words using numbers. They are n-dimensional vectors of real numbers, which are placed in the vector space in such a way that they capture the meaning or relationships between individual words. In our article, we used the word model Word2Vec to create word vectors. Using the Gensim library [24], we implemented this vector model in Python by creating corresponding models for all the augmented corpora.

1) WORD2VEC

The Word2Vec model operates on the fundamental principle that the contextual usage of a word encapsulates its inherent meaning. In essence, this model generates a multidimensional vector representation for each word, preserving both syntactic and semantic relationships among words. The vector distances between individual words reflect the human perception of word associations [6], [25].

The creators of the Word2Vec model have introduced two primary architectures: Continuous Bag-of-Words (CBOW) and Skip-gram. CBOW focuses on predicting a target word based on the surrounding context words within a specific window, whereas Skip-gram focuses on predicting context words given a target word within a given context window. These architectures can be understood as simple neural networks comprising input, hidden, and output layers. However, the training outcome of these networks is not the output layer itself but rather the weights of the hidden layer, which serve as the word vectors. While a Softmax activation function is employed for the output layer, Word2Vec also incorporates algorithmic enhancements like hierarchical Softmax and negative sampling to reduce computational complexity [24], [25].

In our study, we employed the Word2Vec model implementation provided by the Gensim library [24] in the Python programming language. Our model received pre-processed sentences as input, and we configured the dimension size to 150 while setting the context window size to 7. These parameter values were selected based on our prior research, which aimed to identify appropriate settings for optimal results. Utilizing both the Skip-gram architecture, we developed one model for the original sentences and three additional models for sentences subjected to specific modifications, namely synonym replacement, function word deletion, and back translation. By training the model according to this defined approach, we obtained word vectors for all available words. We stored the individual word vectors and their corresponding words in a binary file format using the Gensim library [24] and the KeyedVectors class, creating a comprehensive dictionary-like structure.

F. VERIFICATION OF THE QUALITY OF THE VECTOR MODEL ON THE CLASSIFICATION TASK

It is commonly acknowledged that language models are evaluated through intrinsic and extrinsic evaluations to assess their performance. The key distinction lies in the fact that intrinsic evaluations directly assess the system's performance on the specific task it was designed for. For instance, if a language model aims to generate coherent sentences, the evaluation would involve measuring the coherence of the generated sentences. On the other hand, extrinsic evaluation entails assessing the system's performance on a subsequent task that it was not originally designed for. For example, if a language model is intended to generate coherent sentences, an extrinsic evaluation would measure its ability to enhance the performance of a downstream task such as machine translation or sentiment analysis [26], [27].

In our article, we aim to determine whether word vectors created from modified sentences or from original sentences exhibit better quality. For this purpose, we will employ extrinsic evaluation. Specifically, we will address the classification of fake news using the word vectors we have created. We will compare the effectiveness of classifying fake news using vectors generated from the original sentences versus those generated from the modified sentences.

G. CREATING SEQUENCES OF WORD VECTORS

Before generating the word vector sequences, it is necessary to modify the dataset for fake news classification. We have completed the initial preprocessing, but the dataset contains words that do not have corresponding word vectors. Such words do not contribute to the classification of fake news or the accuracy of our word vectors, so we will eliminate them. As a result, we observed that some records became empty or had a reduced number of words. To enhance the quality of our outcomes, we removed the top and bottom 5 % of entries with extreme word counts.

Another challenge we encountered was the varying word counts in each record. Some entries had too few words, while others had an excessive amount. As we aim to create sequences of word vectors for each word in the record and consider that each vector has a dimension of 150, dealing with a large number of words would result in sequences of enormous dimensions. Consequently, we decided to only consider the first ten words from each record for fake news classification.

Creating the sequences became straightforward as we iterated through the individual records, replacing the words with their corresponding word vectors. The outcome is a list of sequences, where each sequence comprises ten vectors with a dimension of 150.

H. CLASSIFICATION OF FAKE NEWS

Due to the large size of our classification dataset, a decision was made to focus solely on the initial 15,000 records. To ensure a balanced dataset with an equal representation of

false and true message classes, we employed the undersampling method. This involved randomly reducing the number of entries from the larger class to match the number of entries from the smaller class. Consequently, we were left with 7,265 records for each class. As a result, our classification model's input will consist of a list of word vector sequences, accompanied by an indication of whether they represent false messages or not.

For the classification of fake news, we will utilize various classifiers available in the Scikit-learn library [27], which will be introduced in the following section. To perform the classification, we employed the Stratified k-fold cross-validation method. This method involves dividing the entire dataset into k-subsets, where one subset is used for testing and the remaining subsets are utilized for training the classification model. This process is repeated k-times. It is worth noting that all subsets maintain a consistent representation of each class.

In our evaluation, we set k to be 10, resulting in the dataset being divided into 10 subsamples. To assess the model's performance within each cross-validation fold, we employed fundamental performance metrics such as accuracy, precision, recall, and F1 score. Once we obtained the results from all the folds, we computed the average metrics to derive an overall assessment of the classification accuracy for fake news.

IV. RESULTS

After constructing the Word2Vec model, we leveraged these models to generate word vectors for the purpose of classifying fake news. The Word2Vec model was trained on diverse English article corpora, employing a dimensionality of 150 and a window size of 7. In accordance with the approach outlined in the Introduction section, we proceeded to establish classification models designed to identify fabricated news based on the word vectors. These classification models were trained using the WELFake dataset introduced in Chapter 3.1. The subsequent classification algorithms were employed in the creation of these models:

- Random Forest
- Logistic Regression
- BernoulliNB
- SVC.

All algorithms were used with basic settings, without hyperparameter optimization, for comparability of results. Stratified k-fold validation was employed for evaluating classification models. The evaluation metric used was classification accuracy (1), calculated as the ratio of correct predictions (TP + TN) to the total number of samples (TP + TN + FP + FN). Precision (2), recall (3), and F1 score (4) were also computed for evaluating, analysing, and describing the model results. Precision is the ratio of true positive results (TP) to the sum of true positives and false positives (TP + FP). Recall is the ratio of TP to the sum of TP and false negatives (TP + FN). F1 score is the weighted average of precision and recall, considering both false positives and false negatives.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FP} \tag{1}$$

$$precision = \frac{TP}{TP + FP} \tag{2}$$

$$recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1\ score = 2 * \frac{precision * recall}{precision + recall} \tag{4}$$

To begin, it is necessary to perform a comprehensive data survey to assess the accuracy and F1 score metrics. Table 1 presents basic statistics for the accuracy metric. In the case of the mean, the best results were achieved for the SVC classifier and the BT corpus (0.8596). When comparing accuracy values across different classifiers, the highest values were observed for the BT corpus in SVC and BernoulliNB, and for the FWD corpus in Logistic Regression and the Original corpus in Random Forest. For the BT corpus, the most homogeneous results in terms of Quartile Range (Quartile) were observed in the Random Forest and SVC classifiers. In the remaining two classifiers, the lowest Quartile Range values were observed for the FWD corpus.

In the description of our results, we will focus on the F1 score. This metric is usually more useful than accuracy because it also considers precision and recall. Visual representations in the form of boxplots (refer to figures 1 and 2) provide a clear depiction of key statistical measures, including maximum, minimum, median, upper quartile, and lower quartile, for the F1 score metric across the four classifiers utilized in the analysis.

The graphs show the differences between the measured values. Based on the results of descriptive statistics, we therefore need to verify the validity of the following hypothesis:

H0: The dependent variable F1 does not depend on the corpus preparation factor.

We want to reject this hypothesis because by rejecting this hypothesis, the alternative hypothesis becomes valid. First, we check the normality (Table 2) for the dependent variable (F1 score) in individual groups.

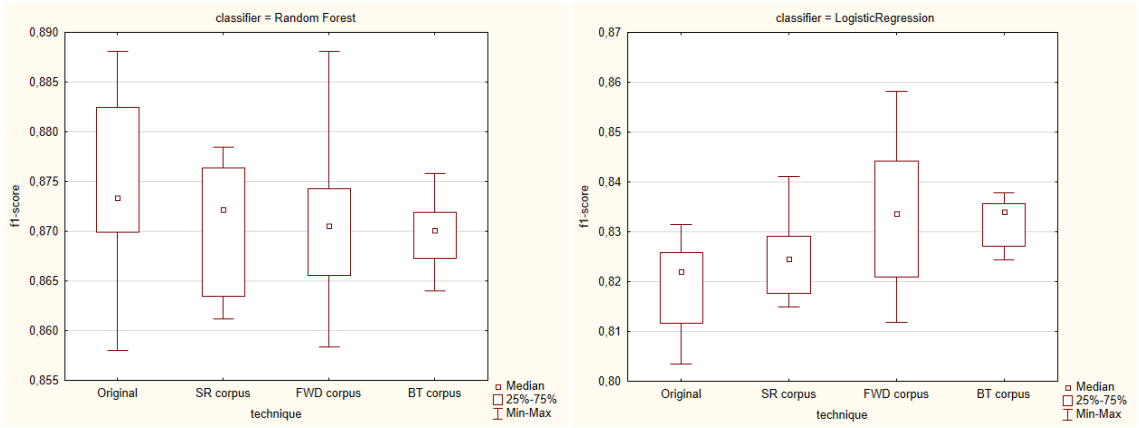
To test the equality of variances, we opted for a non-parametric test due to the identified deviations from normality and the limited dataset size. The Wilcoxon matched pairs test serves as a non-parametric alternative to the t-test for dependent samples. It assesses whether the scores of two variables are derived from the same distribution.

The Wilcoxon matched pairs test was employed to evaluate null hypotheses suggesting the independence of the F1 score from corpus preparation. These hypotheses were tested using results from individual classifiers. If the P value in the Wilcoxon matched pairs test is small, we can reject the idea that the difference is due to chance. Marked tests are significant at $p < 0.05$ (it is marked red color in the tables).

Based on the significance level of the Wilcoxon matched pairs test ($p < 0.05$), we can conclude that we have

TABLE 1. Descriptive statistics for accuracy metric.

Classifier	Technique	Valid N	Mean	Median	Min.	Max.	Lower	Upper	Range	Quartile	Std. Dev.
Random Forest	Original	10	0.8588	0.8548	0.8466	0.8719	0.8534	0.8685	0.0253	0.0151	0.0088
	SR corpus	10	0.8528	0.8524	0.8397	0.8692	0.8431	0.8609	0.0294	0.0177	0.0103
	FWD corpus	10	0.8519	0.8493	0.8383	0.8700	0.8466	0.8603	0.0316	0.0137	0.0094
	BT corpus	10	0.8526	0.8534	0.8437	0.8602	0.8492	0.8547	0.0165	0.0055	0.0044
Logistic Regression	Original	10	0.8167	0.8190	0.7950	0.8306	0.8094	0.8245	0.0356	0.0150	0.0104
	SR corpus	10	0.8215	0.8194	0.8141	0.8382	0.8155	0.8258	0.0240	0.0103	0.0075
	FWD corpus	10	0.8307	0.8301	0.8107	0.8528	0.8204	0.8390	0.0420	0.0185	0.0145
	BT corpus	10	0.8294	0.8306	0.8211	0.8348	0.8238	0.8335	0.0136	0.0097	0.0051
Bernoulli NB	Original	10	0.8056	0.8039	0.7894	0.8211	0.7990	0.8162	0.0317	0.0172	0.0101
	SR corpus	10	0.7984	0.7973	0.7854	0.8128	0.7900	0.8067	0.0273	0.0166	0.0088
	FWD corpus	10	0.8137	0.8125	0.8038	0.8314	0.8101	0.8163	0.0276	0.0061	0.0074
	BT corpus	10	0.8170	0.8155	0.8052	0.8293	0.8107	0.8239	0.0240	0.0131	0.0079
SVC	Original	10	0.8290	0.8320	0.8184	0.8383	0.8232	0.8327	0.0199	0.0095	0.0068
	SR corpus	10	0.8575	0.8548	0.8506	0.8699	0.8528	0.8623	0.0192	0.0095	0.0071
	FWD corpus	10	0.8593	0.8569	0.8492	0.8789	0.8555	0.8610	0.0296	0.0055	0.0078
	BT corpus	10	0.8596	0.8613	0.8485	0.8671	0.8568	0.8637	0.0185	0.0068	0.0063



(a) (b)

FIGURE 2. Boxplot for F1 score a) Random forest b) Logistic regression.

established a statistically significant distinction in individual measurements (Table 3 - 6). Therefore, we reject the null hypothesis.

For the Random Forest classifier, we observed statistically significant differences between the SR and Original groups (Table 3), favoring the Original group (Figure 1a). No statistically significant differences were detected in relation to other groups.

Statistically significant differences were demonstrated between Original and FWD corpus, as well as Original and BT corpus (Table 4), similarly between SR corpus and BT corpus for the Logistic Regression classifier. According to (Figure 1b), the best technique is the BT corpus, the results of which are statistically significant compared to the Original and SR corpus.

For the BernoulliNB classifier, the worst technique is the Original (Table 5), the results of which are statistically significantly worse than in the case of other techniques. The second worst is the SR corpus technique. There were statistically insignificant differences between the BT corpus and the FWD corpus, which were evaluated as the best techniques (Figure 2a).

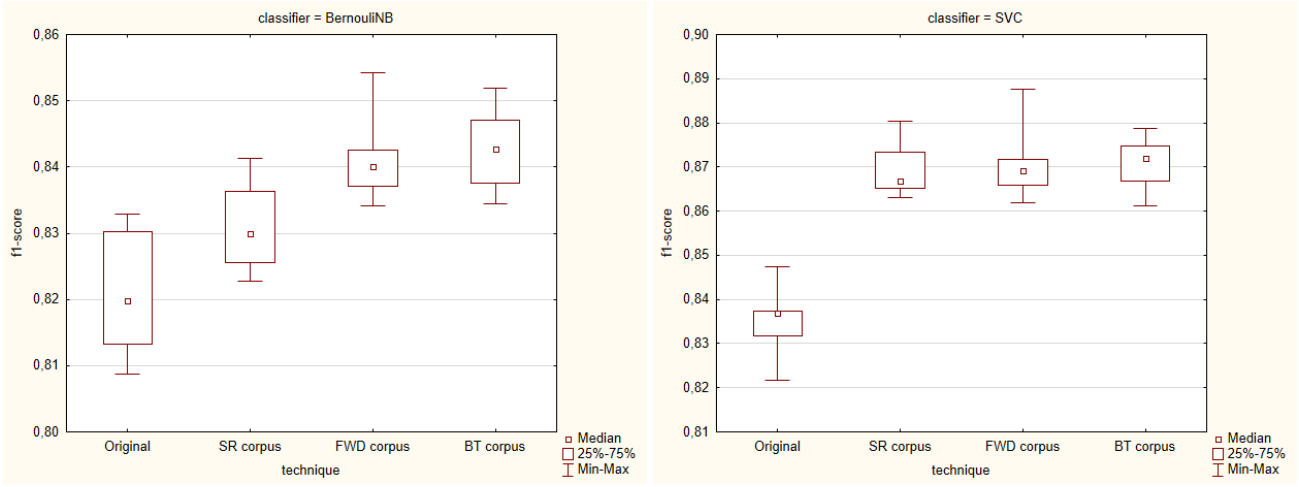


FIGURE 3. Boxplot for F1 score a) BernoulliNB b) SVC classifier.

TABLE 2. Shapiro-wilk test (normality).

Classifier	Level of Factor	N	SW-W	p-value
Random Forest	Original	10	0.9547	0.7247
	SR corpus	10	0.8778	0.1232
	FWD corpus	10	0.9589	0.7738
	BT corpus	10	0.9495	0.6624
Logistic Regression	Original	10	0.9531	0.7055
	SR corpus	10	0.8571	0.0705
	FWD corpus	10	0.8207	0.0259
	BT corpus	10	0.9499	0.6670
BernoulliNB	Original	10	0.9376	0.5265
	SR corpus	10	0.9623	0.8117
	FWD corpus	10	0.8721	0.1058
	BT corpus	10	0.9741	0.9261
SVC	Original	10	0.9288	0.4358
	SR corpus	10	0.9461	0.6226
	FWD corpus	10	0.9369	0.5189
	BT corpus	10	0.8687	0.0966

TABLE 3. Wilcoxon matched pairs test (Random Forest Classifier) marked tests are significant at $p < 0.05$.

Pair of Variables	N	T	Z	p-value
Original & Original				
Original & SR corpus	10	7.00000	2.089553	0.036659
Original & FWD corpus	10	15.0000	1.274118	0.202623
Original & BT corpus	10	9.00000	1.885695	0.059337
SR corpus & Original	10	7.00000	2.089553	0.036659
SR corpus & SR corpus				
SR corpus & FWD corpus	10	27.00000	0.050965	0.959354
SR corpus & BT corpus	10	21.00000	0.662541	0.507625
FWD corpus & Original	10	15.00000	1.274118	0.202623
FWD corpus & SR corpus	10	27.00000	0.050965	0.959354
FWD corpus & FWD corpus				
FWD corpus & BT corpus	10	20.00000	0.764471	0.444587
BT corpus & Original	10	9.00000	1.885695	0.059337
BT corpus & SR corpus	10	21.00000	0.662541	0.507625
BT corpus & FWD corpus	10	20.00000	0.764471	0.444587
BT corpus & BT corpus				

In the case of the SVC classifier, statistically significant differences were recorded (Table 6) between the techniques SR corpus, FWD corpus, BT corpus and Original in favor of Original (Figure 2b).

A common evaluation practice in the vector models literature is to measure the models' ability to predict human judgments regarding lexical semantic relations between word pairs. Most existing evaluation sets consist of scores collected for word pairs. This type of evaluation is referred to as intrinsic evaluation. In our experiment, we used it only as supplementary analysis for the evaluation of Data Augmentation Techniques.

SimLex-999 [5] and WordSim353 [4] are widely recognized lexical resources commonly referred to as a gold standard resource for evaluating distributional semantic models. Using these two lexical resources, we conducted

TABLE 4. Wilcoxon matched pairs test (Logistic Regression) marked tests are significant at $p < 0.05$.

Pair of Variables	N	T	Z	p-value
Original & Original				
Original & SR corpus	10	13.00000	1.477977	0.139415
Original & FWD corpus	10	6.00000	2.191483	0.028418
Original & BT corpus	10	0.00000	2.803060	0.005062
SR corpus & Original	10	13.00000	1.477977	0.139415
SR corpus & SR corpus				
SR corpus & FWD corpus	10	16.00000	1.172189	0.241122
SR corpus & BT corpus	10	6.00000	2.191483	0.028418
FWD corpus & Original	10	6.00000	2.191483	0.028418
FWD corpus & SR corpus	10	16.00000	1.172189	0.241122
FWD corpus & FWD corpus				
FWD corpus & BT corpus	10	25.00000	0.254824	0.798860
BT corpus & Original	10	0.00000	2.803060	0.005062
BT corpus & SR corpus	10	6.00000	2.191483	0.028418
BT corpus & FWD corpus	10	25.00000	0.254824	0.798860
BT corpus & BT corpus				

TABLE 5. Wilcoxon matched pairs test (Bernoullinb Classifier) marked tests are significant at $p < 0.05$.

Pair of Variables	N	T	Z	p-value
Original & Original				
Original & SR corpus	10	1.00000	2.701130	0.006911
Original & FWD corpus	10	0.00000	2.803060	0.005062
Original & BT corpus	10	0.00000	2.803060	0.005062
SR corpus & Original	10	1.00000	2.701130	0.006911
SR corpus & SR corpus				
SR corpus & FWD corpus	10	0.00000	2.803060	0.005062
SR corpus & BT corpus	10	1.00000	2.701130	0.006911
FWD corpus & Original	10	0.00000	2.803060	0.005062
FWD corpus & SR corpus	10	0.00000	2.803060	0.005062
FWD corpus & FWD corpus				
FWD corpus & BT corpus	10	17.00000	1.070259	0.284504
BT corpus & Original	10	0.00000	2.803060	0.005062
BT corpus & SR corpus	10	1.00000	2.701130	0.006911
BT corpus & FWD corpus	10	17.00000	1.070259	0.284504
BT corpus & BT corpus				

TABLE 6. Wilcoxon matched pairs test (SVC) marked tests are significant at $p < 0.05$.

Pair of Variables	N	T	Z	p-value
Original & Original				
Original & SR corpus	10	0.00000	2.803060	0.005062
Original & FWD corpus	10	0.00000	2.803060	0.005062
Original & BT corpus	10	0.00000	2.803060	0.005062
SR corpus & Original	10	0.00000	2.803060	0.005062
SR corpus & SR corpus				
SR corpus & FWD corpus	10	27.00000	0.050965	0.959354
SR corpus & BT corpus	10	26.00000	0.152894	0.878482
FWD corpus & Original	10	0.00000	2.803060	0.005062
FWD corpus & SR corpus	10	27.00000	0.050965	0.959354
FWD corpus & FWD corpus				
FWD corpus & BT corpus	10	22.00000	0.560612	0.575063
BT corpus & Original	10	0.00000	2.803060	0.005062
BT corpus & SR corpus	10	26.00000	0.152894	0.878482
BT corpus & FWD corpus	10	22.00000	0.560612	0.575063
BT corpus & BT corpus				

supplementary internal evaluations of the techniques we investigated. From the four corpora created (Original, SR corpus, FWD corpus, and BT corpus), we trained four Word2Vec Skip-gram models. Subsequently, we calculated the semantic distances between individual words in SimLex-999 and WordSim353.

To assess the corpus on the SimLex-999 and WordSim353 datasets, we computed Spearman's rank correlation coefficient for each model, comparing the relationship between model similarity scores and human similarity judgments. The results of Spearman's rank correlation coefficient, along with information about coverage, are presented in Table 7. It is essential to note that SimLex-999 comprises 999 word pairs, while WordSim353 is divided into three parts: WordSim overall (350 word pairs), WordSim Similarity (201 word pairs), and WordSim Relatedness (252 word pairs). The coverage value indicates how many word pairs were successfully used to calculate semantic distances. Given the size of the corpora we worked with, it is evident that they may not contain all the words present in the SimLex-999 and WordSim353 resources.

The results show small differences in the correlation coefficient. In all three resources, WordSim achieved the best match with BT corpus, while in the case of SimLex, the best match was observed in Original, although it should be noted that the differences are indeed small.

For our intrinsic evaluation, we decided to take a different perspective on the semantic distances between individual words. In this view of the results, we experimented

TABLE 7. Spearman's rank correlation coefficient and coverage for models.

Model	SimLex-999		WordSim overall		WordSim Similarity		WordSim Relatedness	
	coeff.	coverage	coeff.	coverage	coeff.	coverage	coeff.	coverage
Original	0.160	971	0.423	317	0.474	179	0.359	231
SR corpus	0.132	988	0.371	325	0.430	184	0.373	236
FWD corpus	0.134	971	0.398	320	0.424	179	0.363	235
BT corpus	0.118	953	0.465	313	0.507	172	0.459	231

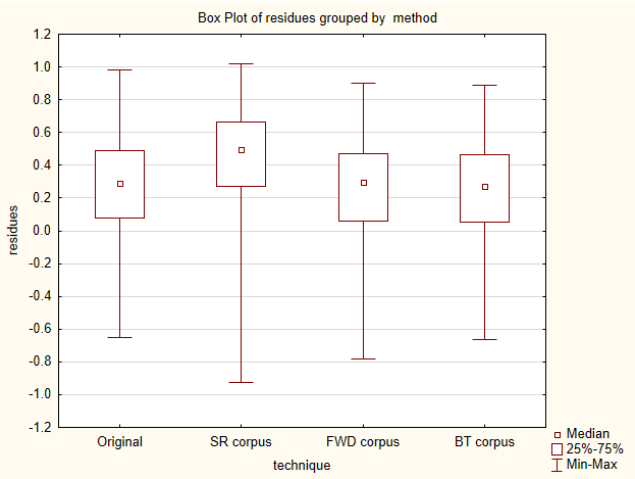


FIGURE 4. Boxplot for residues.

only with the lexical resource SimLex-999. In contrast to WordSim-353, this resource explicitly quantifies similarity rather than association or relatedness, resulting in lower ratings for pairs of entities that are associated but not actually similar.

We drew inspiration from residual analysis. The main idea of this method assumes that:

$$\textbf{Data} = \textbf{prediction using the model (function)} + \textbf{residual value}.$$

By subtracting the values obtained from the model (expected values) from the observed values, we obtained errors (residual values). We could analyze the residual values for the purpose of model assessment. In our case, the expected values are semantic distance values of words obtained from our models, and the observed values are values from the lexical resource SimLex-999. The selection residues e_i are defined as:

$$e_i = y_i - \hat{y}_i \tag{5}$$

where $\hat{y}_i$ are expected values predicted by the model and y_i are observed values.

After calculating the residues, we visualized them using a box plot (Figure 3). From the graph, it is evident that, based on the positive medians for all models, our models tended to overestimate the semantic word distance values. Interestingly, the words were most overestimated by the SR model. However, a more critical perspective lies in the examination of the variance and quartile range. For good models, this range should be small, indicating that residue values are close to 0 or the median. The smallest variance range was observed for FWD (1.512). It is also clear from the graph that this variance range, along with BT (1.554), is much lower than the variance range for SR (1.925) and Original (1.947). In terms of quartile range, the values were quite similar for Original (0.412), SR (0.400), FWD (0.401), and BT (0.409).

From this perspective on the results, it is evident that the FWD and BT models were better at estimating the semantic distances between words.

V. DISCUSSION

Not every Text Data Augmentation technique is equally straightforward to implement. While the FWD technique essentially involved applying an algorithm to remove stop words and using libraries to identify POS tags, the BT technique required making API calls for translation, which was relatively slow, with necessary adjustments and limitations, and, of course, it was not freely available. The most complex in terms of implementation was the SR technique. Here, it was necessary to work with lists of synsets and available libraries.

There are similarly focused research studies, as we have mentioned in the Related work chapter. Haralabopoulos et al. [15] employed the technique of text data augmentation, specifically sentence permutations, to generate synthetic data based on an existing labeled dataset. Their technique achieved a significant improvement in classification accuracy by an average of 4.1 % across eight different datasets. Furthermore, they proposed two additional text data augmentation techniques: synonym replacement and negation. These techniques were tested on three appropriate datasets and yielded an accuracy improvement of 0.35 % (synonyms) and 0.4 % (negation) when compared to the permutation method they proposed. In our research, we also achieved similar increases in accuracy (up to 3.06 % for BT, FWD techniques and SVC classifiers), with the SR technique yielding an accuracy improvement of up to 2.85 % (SR technique and SVC classifier).

Wei et al. [6] utilized four techniques for text corpus augmentation (synonym replacement, random insertion,

swapping, and deletion) known as Easy Data Augmentation (EDA). Through experiments conducted on five classification tasks, they observed performance improvements for convolutional and recurrent neural networks. The research was evaluated in a different manner compared to our case; however, in terms of accuracy, their results were comparable.

It's noteworthy to acknowledge the findings of Salah et al. [11], where their stacking approach surpassed our results. Specifically, their approach achieved an accuracy of 82.77 % for synonym augmentation and 90.60 % for BT augmentation when employing the Random Forest classifier.

On the other hand, Bucos et al. [13] state that their BT technique led to better performance across all four classification models (Extra Trees Classifier, Random Forest, Logistic Regression, and SVM). In their content-based approach, BT achieved an average accuracy of 77.82 %, which was worse than our accuracy.

In comparison to the referenced studies, our enhancement may appear to be comparable. However, it is important to note that our objective was not to achieve the highest possible performance measures with the investigated techniques. We chose the f1-score metric because it also accounts for precision and recall. Our goal was to ascertain whether data augmentation techniques statistically contribute to improving performance measures in NLP classification tasks. The results indicate that statistically significant differences in results were primarily observed for the FWD and BT techniques. Additionally, besides extrinsic evaluation (using a classification task), intrinsic evaluation was also employed (through the analysis of lexical semantic relations between word pairs).

The potential issues in implementing data augmentation techniques depend on the technique used. In the case of the SR technique, different languages other than English pose problems. For English, there are sets of synsets available in libraries. However, these are not available for every language. The FWD technique relies on additional classifiers that must identify part-of-speech for individual words. The use of the BT technique raises a multitude of questions. More attention should be given to individual languages, determining which languages are suitable for BT depending on the source language, how to optimize the process for BT, and so on.

Our research was conducted in the context of the English language. All the analyzed techniques are likely applicable in other languages as well. The technique SR relies on the existence of lists of semantically similar words for the chosen language. In our case, for English, sets of synsets were available in libraries. Such lists are not readily available for minority languages. The lack of these lists can be addressed by employing multiple language models that extract semantically similar words from accessible corpora. However, this solution is time and computation-intensive, not to mention the requirement for existing accessible corpora in the chosen language. Similarly, the FWD technique depends on technologies that identify the part-of-speech for individual words. Unlike lists of semantically similar words, there are numerous POS taggers available for most languages. The BT technique appears to be the most accessible for majority languages. Currently, there are numerous successful machine translators for the majority of languages. In our experiment, this technique even yielded the best results.

Another limitation of the study may be the datasets used, especially the dataset utilized as a corpus for training Word2Vec. Large language models are trained on much larger corpora. However, implementing larger corpora was not feasible due to computational and memory constraints. It is evident, though, that if statistically significant differences in results were observed using a relatively small corpus (143,000 English articles), those differences would also be detected with a larger corpus.

VI. CONCLUSION

In the article, we focused on optimizing a text corpus through various text data augmentation techniques. Subsequently, we conducted training in word vectors on these expanded corpora. The result of our experiment in classification tasks is that the new word vectors obtained through this augmentation led to a significant improvement in classification performance compared to the original texts.

In addition to this result, during the implementation of our experiment, we encountered several issues and questions suitable for our future research. In the case of the SR (synonym replacement) technique, it appears interesting to seek solutions for minority languages, where semantically similar words are not part of synsets. Often, these languages only have synonym dictionaries (often not digitized). The question also remains as to which part-of-speech categories are suitable for replacement and replacing which does not make sense to enhance the results. We assume that nouns, adjectives, and verbs will certainly be more important for replacement than pronouns, prepositions, conjunctions, and the like. For the FWD (function words deletion) corpus preparation technique, it would be interesting to determine which languages benefit the most from this technique. At the same time, it would be fascinating to investigate the ideal number of words to delete, possibly in combination with a detailed analysis of part-of-speech and their impact on the results. With the BT (back translation) technique, there is a natural opportunity for experimenting with multiple languages. In addition to a closer look at individual languages, it is also possible to experiment with various machine translation systems.

Our article focuses on the currently most progressive area of large language models. It is evident that with their current proliferation, any improvement is desired. Enhancements can also be achieved directly in the methods for creating LLMs. However, these are already completed libraries in programming languages. Improving them is challenging. Data augmentation techniques, on the other hand, appear to be a simple enhancement where we don't need to reprogram models from scratch. We just need to modify and enrich the input corpus. In this, we see the greatest advantage of these techniques.

REFERENCES

[1] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy, "A survey of data augmentation approaches for NLP," 2021, *arXiv:2105.03075*.

[2] L. F. A. O. Pellicer, T. M. Ferreira, and A. H. R. Costa, "Data augmentation techniques in natural language processing," *Appl. Soft Comput.*, vol. 132, Jan. 2023, Art. no. 109803, doi: 10.1016/j.asoc.2022.109803.

[3] S. Nazir, M. Asif, S. A. Sahi, S. Ahmad, Y. Y. Ghadi, and M. H. Aziz, "Toward the development of large-scale word embedding for low-resourced language," *IEEE Access*, vol. 10, pp. 54091–54097, 2022, doi: 10.1109/ACCESS.2022.3173259.

[4] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pasca, and A. Soroa, "A study on similarity and relatedness using distributional and WordNet-based approaches," in *Proc. Human Lang. Technologies, Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2009, pp. 19–27.

[5] F. Hill, R. Reichart, and A. Korhonen, "SimLex-999: Evaluating semantic models with (Genuine) similarity estimation," *Comput. Linguistics*, vol. 41, no. 4, pp. 665–695, Dec. 2015, doi: 10.1162/coli_a_00237.

[6] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 6381–6387, doi: 10.18653/v1/d19-1670.

[7] G. A. Miller, "WordNet," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995, doi: 10.1145/219717.219748.

[8] V. Marivate and T. Sefara, "Improving short text classification through global augmentation methods," in *Proc. Int. Cross-Domain Conf. Mach. Learn. Knowl. Extraction*, 2020, pp. 385–399, doi: 10.1007/978-3-030-57321-8_21.

[9] S. Kobayashi, "Contextual augmentation: Data augmentation by words with paradigmatic relations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2018, pp. 452–457, doi: 10.18653/v1/n18-2072.

[10] R. N. Al-Matham and H. S. Al-Khalifa, "SynoExtractor: A novel pipeline for Arabic synonym extraction using Word2 Vec word embeddings," *Complexity*, vol. 2021, pp. 1–13, Feb. 2021, doi: 10.1155/2021/6627434.

[11] I. Salah, K. Jouini, and O. Korbaa, "On the use of text augmentation for stance and fake news detection," *J. Inf. Telecommun.*, vol. 7, no. 3, pp. 359–375, Jul. 2023, doi: 10.1080/24751839.2023.2198820.

[12] I. Salah, K. Jouini, and O. Korbaa, "Augmentation-based ensemble learning for stance and fake news detection," in *Proc. Int. Conf. Comput. Collective Intell.*, 2022, pp. 29–41, doi: 10.1007/978-3-031-16210-7_3.

[13] M. Bucos and G. Ţucudean, "Text data augmentation techniques for fake news detection in the Romanian language," *Appl. Sci.*, vol. 13, no. 13, p. 7389, Jun. 2023, doi: 10.3390/app13137389.

[14] A. J. Keya, M. A. H. Wadud, M. F. Mridha, M. Alatiyyah, and M. A. Hamid, "AugFake-BERT: Handling imbalance through augmentation of fake news using BERT to enhance the performance of fake news classification," *Appl. Sci.*, vol. 12, no. 17, p. 8398, Aug. 2022, doi: 10.3390/app12178398.

[15] G. Haralabopoulos, M. T. Torres, I. Anagnostopoulos, and D. McAuley, "Text data augmentations: Permutation, antonyms and negation," *Expert Syst. Appl.*, vol. 177, Sep. 2021, Art. no. 114769, doi: 10.1016/j.eswa.2021.114769.

[16] A. Dahou, A. A. Ewees, F. A. Hashim, M. A. A. Al-Qaness, D. A. Orabi, E. M. Soliman, E. M. Tag-Eldin, A. O. Aseeri, and M. A. Elaziz, "Optimizing fake news detection for Arabic context: A multitask learning approach with transformers and an enhanced nutcracker optimization algorithm," *Knowl.-Based Syst.*, vol. 280, Nov. 2023, Art. no. 111023, doi: 10.1016/j.knosys.2023.111023.

[17] J. Hua, X. Cui, X. Li, K. Tang, and P. Zhu, "Multimodal fake news detection through data augmentation-based contrastive learning," *Appl. Soft Comput.*, vol. 136, Mar. 2023, Art. no. 110125, doi: 10.1016/j.asoc.2023.110125.

[18] M. I. Marwat, J. A. Khan, M. D. Alshehri, "Sentiment analysis of product reviews to identify deceptive rating information in social media: A SentiDeceptive approach," *KSII Trans. Internet Inf. Syst.*, vol. 16, no. 3, pp. 830–860, Dec. 2022, doi: 10.3837/tiis.2022.03.005.

[19] J. A. Khan, A. Yasin, R. Fatima, D. Vasan, A. A. Khan, and A. W. Khan, "Valuating requirements arguments in the online user's forum for requirements decision-making: The CrowdRE-VArg framework," *Softw., Pract. Exper.*, vol. 52, no. 12, pp. 2537–2573, Dec. 2022, doi: 10.1002/spe.3137.

[20] M. Risdal. (2016). *Getting Real About Fake News*. Kaggle. Accessed: Dec. 28, 2023. [Online]. Available: https://www.kaggle.com/code/anthonyc1/gathering-real-news-for-oct-dec-2016/output

[21] S. Bird, E. Klein, and E. Loper, *Natural Language Processing With Python: Analyzing Text With the Natural Language Toolkit*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2009.

[22] P. K. Verma, P. Agrawal, I. Amorim, and R. Prodan, "WELFake: Word embedding over linguistic features for fake news detection," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 4, pp. 881–893, Aug. 2021. [Online]. Available: https://www.kaggle.com/datasets/saurabhshahane/fake-news-classification/data

[23] J. Tiedemann and S. Thottingal, "OPUS-MT—Building open translation services for the world," in *Proc. 22nd Annu. Conf. Eur. Assoc. Mach. Transl.*, A. Martins, H. Moniz, S. Fumega, B. Martins, F. Batista, L. Coheur, C. Parra, I. Trancoso, M. Turchi, A. Bisazza, J. Moorkens, A. Guerberof, M. Nurminen, L. Marg, M. L. Forcada, Eds., 2020, pp. 479–480.

[24] R. Řehuřek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proc. LREC Workshop New Challenges for NLP Frameworks, ELRA*, 2010, pp. 45–50.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[26] M. Zhai, J. Tan, and J. Choi, "Intrinsic and extrinsic evaluations of word embeddings," in *Proc. AAAI Conf. Artif. Intell.*, Nov. 2016, vol. 30, no. 1, pp. 4282–4283, doi: 10.1609/aaai.v30i1.9959.

[27] Y. Shi, Y. Zheng, K. Guo, L. Zhu, and Y. Qu, "Intrinsic or extrinsic evaluation: An overview of word embedding evaluation," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2018, pp. 1255–1262, doi: 10.1109/ICDMW.2018.00179.

JOZEF KAPUSTA is currently an Associate Professor with the Department of Informatics, Faculty of Natural Sciences, Constantine the Philosopher University in Nitra. His research interests include natural language processing and artificial intelligence.

DÁVID DRŽÍK received the master's degree in applied informatics from the Faculty of Natural Sciences and Informatics, Constantine the Philosopher University in Nitra, Slovakia, in 2022, where he is currently pursuing the Ph.D. degree, with his doctoral work centered on the exploration of Slovak language morphology within the field of artificial intelligence. In the past, his research has encompassed the investigation of various behavioral characteristics of users during interactions with computers and smartphones, with a focus on classifying user's emotional states.

KIRSTEN ŠTEFLOVIČ was born in Lucenec, Slovakia, in 1997. She received the M.S. degree in applied informatics from the Faculty of Natural sciences and Informatics, Constantine the Philosopher University in Nitra, Slovakia, in 2021, where she is currently pursuing the Ph.D. degree in applied informatics. Her research interests include natural language processing, machine learning, and web development.

KITTI SZABÓ NAGY received the master's degree in intelligent software systems from the Faculty of Informatics and Information Technology, Bratislava, in 2020, and the Ph.D. degree from Constantine the Philosopher University in Nitra, in 2023. She is actively involved in research with a focus on natural language processing (NLP). The core of her doctoral research is centered on the development of innovative techniques for knowledge extraction from unstructured texts, primarily aimed at the classification of fake news.