applied sciences

MDPI

Article

Coreference Resolution for Improving Performance Measures of Classification Tasks

Kirsten Šteflovič [1,*] and Jozef Kapusta [1,2]

[1] Department of Informatics, Constantine the Philosopher University in Nitra, Trieda Andreja Hlinku 1, 949 74 Nitra, Slovakia; jkapusta@ukf.sk

[2] Institute of Computer Science, Pedagogical University of Cracow, ul. Podchorążych 2, 30-084 Kraków, Poland

* Correspondence: k.steflovic@gmail.com

Abstract: There are several possibilities to improve classification in natural language processing tasks. In this article, we focused on the issue of coreference resolution that was applied to a manually annotated dataset of true and fake news. This dataset was used for the classification task of fake news detection. The research aimed to determine whether performing coreference resolution on the input data before classification or classifying them without performing coreference resolution is more effective. We also wanted to verify whether it is possible to enhance classifier performance metrics by incorporating coreference resolution into the data preparation process. A methodology was proposed, in which we described the implementation methods in detail, starting from the identification of entity mentions in the text using the neuralcoref algorithm, then through word-embedding models (TF–IDF, Doc2Vec), and finally to several machine learning methods. The result was a comparison of the implemented classifiers based on the performance metrics described in the theoretical part. The best result for accuracy was observed for the dataset with coreference resolution applied, which had a median value of 0.8149, while for the F1 score, the best result had a median value of 0.8101. However, the more important finding is that the processed data with the application of coreference resolution led to an improvement in performance metrics in the classification tasks.

Keywords: fake news identification; text mining; natural language processing; dependency grammar; coreference resolution

check for updates

Citation: Šteflovič, K.; Kapusta, J. Coreference Resolution for Improving Performance Measures of Classification Tasks. *Appl. Sci.* **2023**, *13*, 9272. https://doi.org/10.3390/app13169272

Academic Editor: Chilukuri K. Mohan

Received: 11 May 2023
Revised: 5 August 2023
Accepted: 8 August 2023
Published: 15 August 2023

1. Introduction

As a part of computational linguistics, coreference resolution is still a research challenge as it is not enough to only find the first occurrence of an entity in the overall analysis of a text; the correct identification and assignment of all verbal references to these entities are also necessary. In this work, we decided to combine the issue of coreference resolution with the area of identifying fake news. We focused on entity determination methods and compared whether substituting an entity for the original reference can improve and specify the text representation for classifiers.

Natural language processing requires pre-processing the text into a form that is usable for the analysis and prediction of specific tasks. Coreference resolution (CR) involves searching and replacing the name of the given entity in the text. Every written text contains mentions or references to many entities from the real world, which may not always be labeled with the same words. Very often pronouns or synonyms of nominal phrases are used to avoid the repetition of words in the text. The mentioned method of referencing brings one problem: for each reference, its target entity must also be determined. CR deals with this problem. A coreference occurs when the same referent is referred to in the text under different terms.

As an example, we can mention the person “Marie Curie”, who can be mentioned later in a text as “Mrs. Curie”, by the pronoun “she”, or even by her initials “MC”. Mentions can occur in the form of a noun phrase, a pronoun, a proper name, etc. Each of these phrases refers to the same person, and the main task of coreferencing is to identify him or her and to find out the relationship between a given group of expressions. Another problem is determining the gender of the referent because it makes it much easier to determine the relationships between individual entities. We can see the illustration of CR in Figure 1, where mentions are identified in blue, which will be replaced by specific entities.

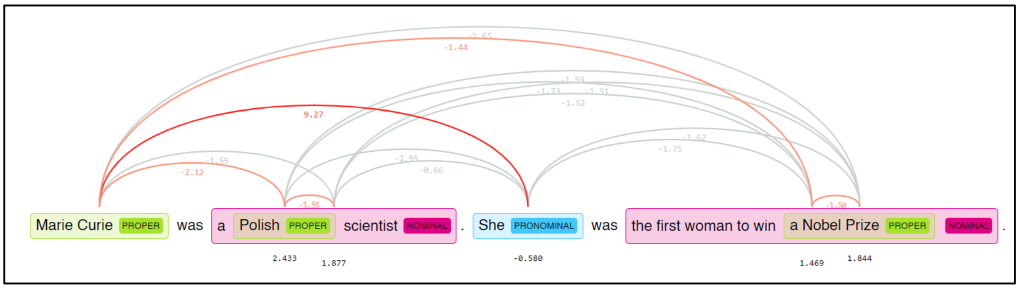


Figure 1. Example of identification of coreferences in text.

In our article, we focus on coreference resolution, which searches and identifies coreferential relationships between entities and their referring terms in a text. Using coreference resolution, we can add additional information to the text, e.g., replace pronouns with a weak declarative value by the objects they represent. This approach can be used to refine the representation of the text.

The aim of this article is to find out whether pre-processing a text using coreference resolution can improve performance measures in classification tasks. One of the classification tasks is the classification of fake news, which was chosen from our previous work as we have experience with this type of task and, at the same time, we have a validated existing dataset. It is obvious that coreference resolution can be applied in solving other text data classification tasks as well.

The basic goal of our work is to determine the significance of CR identification for identifying fake news messages. We believe that a text after CR identification will result in better outcomes. Our aim is not only to determine if the results will improve but also to quantify the improvement. Additionally, we aim to assess the impact of CR on the results based on two different embedding methods for creating feature vectors.

We used CR as a preliminary preparation method for texts that will be used for the vectorization of documents. Vectors created from documents are often used as input vectors to create training and test sets for classifiers in natural language processing classification tasks.

The following methodology (Figure 2) was used for the evaluation of the contribution of CR to improving classification tasks:

1. Data preparation;
2. The identification of the coreference resolution in the data file;
3. The creation of word vectors and document vectors using the Doc2Vec methods and the traditional TF–IDF method. The following datasets containing four-word vectors were created:
    (a) Doc2Vec_nocoref—a dataset in its original state without any identified coreference resolution, processed by the Doc2Vec method;
    (b) Doc2Vec_coref—a modified dataset with an identified coreference resolution, processed by the Doc2Vec method;
    (c) TfIdf_nocoref—a dataset in its original state without any identified coreference resolution, processed by the TF–IDF method,
    (d) TfIdf_coref—a modified dataset with an identified coreference resolution, processed by the TF–IDF method.
4. The creation of the text classification models (fake news classification) using pre-processed word vectors as the inputs:

   (a) Decision Tree;
   (b) Random Forest;
   (c) K-Nearest Neighbors;
   (d) MultinomialNB.

5. The evaluation and comparison of the performance measures of the created fake news classification models (accuracy, precision, recall, and F1 score).

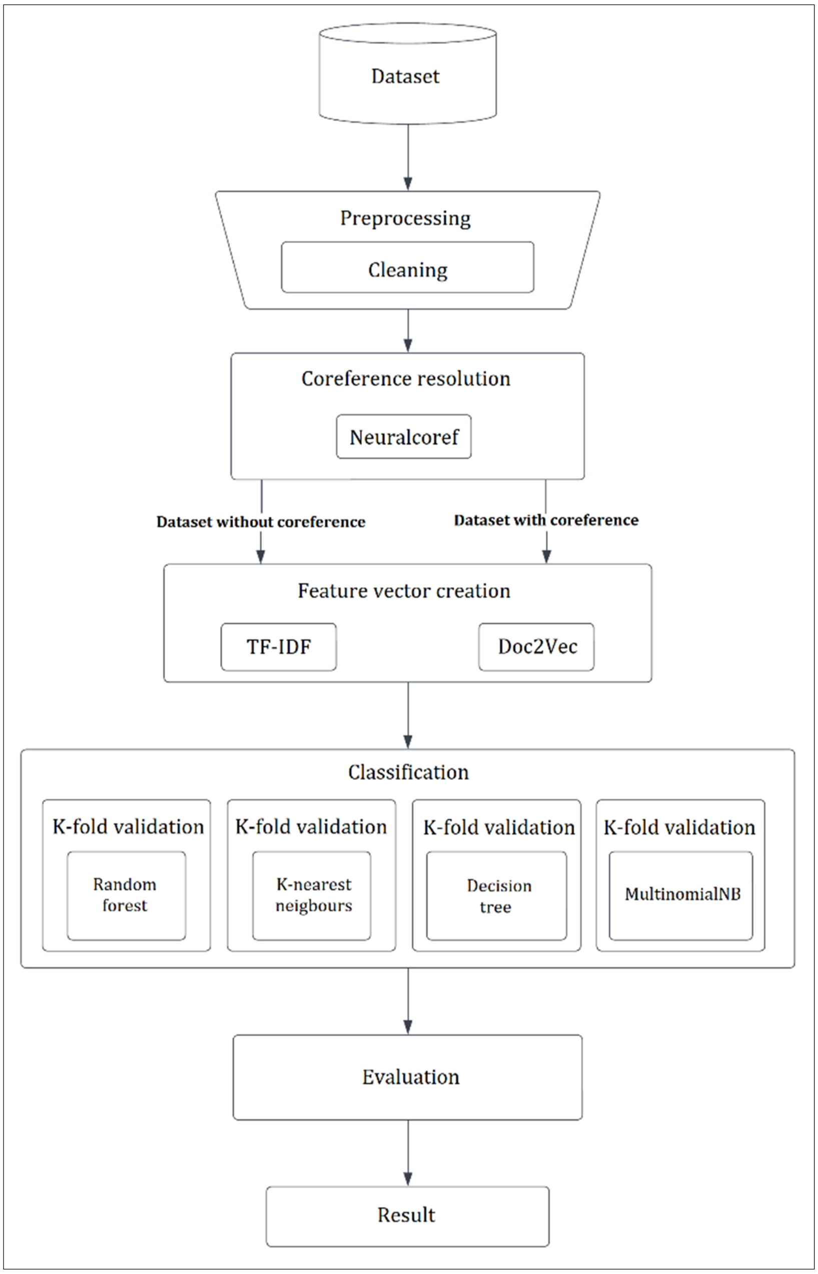
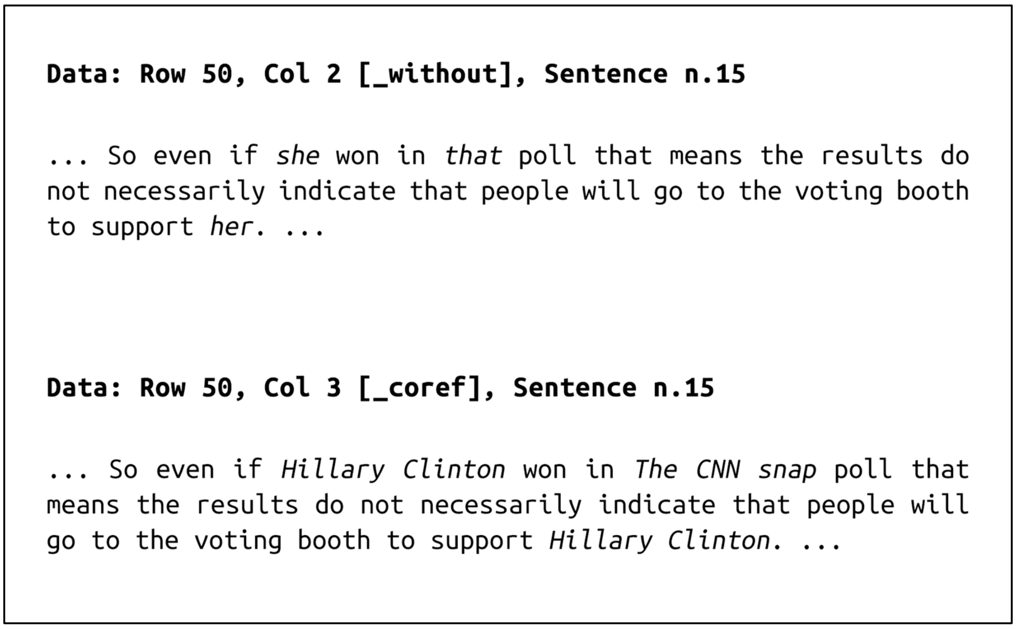


Figure 2. Workflow of our proposed method.

Steps 2 and 3 are the important parts of the mentioned methodology. In step 2, a new text data file named "_coref" is created, where mentions from the original text data file "_without" are identified and replaced (Figure 3). Then, word vectors are created from both the "_without" and "_coref" data files using both the TF–IDF and Doc2Vec methods.



Data: Row 50, Col 2 [_without], Sentence n.15

... So even if *she* won in *that* poll that means the results do not necessarily indicate that people will go to the voting booth to support *her*. ...

Data: Row 50, Col 3 [_coref], Sentence n.15

... So even if *Hillary Clinton* won in *The CNN snap* poll that means the results do not necessarily indicate that people will go to the voting booth to support *Hillary Clinton*. ...

Figure 3. An example of coreference annotation in used corpus.

The first phase of the methodology—data preparation—is a traditional stage of processing all NLP tasks. However, some of its parts (e.g., tokenization) are applied when creating word vectors only if the vectors are generated by available libraries.

The actual evaluation of CR involves assessing how well we can classify fake news using models created from pre-prepared datasets (with and without any coreference). In other words, the output evaluation will consist of evaluating the created models for classifying fake news, where news articles from the test set will be classified into two categories: fake and real news.

The paper is organized as follows. Section 2 provides a summary of the current state in the field of fake news identification studies. Section 3 describes the fake news dataset used in this research along with relevant pre-processing methods and the model creation. Section 4 provides a summary of the most important results. Section 6 of the article contains the discussion and conclusions.

2. Related Work

Coreference resolution is a method that usually supports other NLP techniques such as text summarization [1], a question answering system [2], or information extraction [3,4]. It can be used in solving several classification tasks such as identifying and classifying fake news or detecting phishing messages [5]. Nadeeem et al. [6] presented the FAKTA framework, which integrates various components of a fact checking process, i.e., document retrieval from media sources with various types of reliability, stance detection of documents with respect to given claims, evidence extraction, and linguistic analysis.

Bengtson et al. [7] described a rather simple pair-wise classification model for coreference resolution, which was developed with a well-designed set of features. Their work produced a state-of-the-art system that outperformed systems built with complex models.

Ming [8] investigated the improvement of Chinese CR. Their model proposes acquired word and character representations through pre-trained Skip-gram embeddings and pre-trained BERT. Then, it explicitly leverages span-level information by performing bidirectional LSTMs among the above representations. The proposed model achieved a 62.95% F1 score, outperforming their baseline methods. The only limitation of their article is the absence of an error analysis.

The BERT model (bidirectional representation from transformers), which is used by Google's search engines, was developed by Devlin et al. [9]. Unlike recent language representation models, they designed BERT to pre-train deep bidirectional representations from unlabeled text by joint conditioning on both the left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inferencing [1–3] without substantial task-specific architecture modifications. The results were spectacular; the GLUE score was 80.5%, and the MultiNLI accuracy was 86.7%. The Stanford question answering system dataset (SQuAD) v1.1 resulted in an F1 score of 93.2% and also a SQuAD v2.0 Test F1 score of 83.1%.

Denis et al. [10] investigated two strategies for improving coreference resolution. The first one involved training separate models that specialized in particular types of mentions (e.g., pronouns versus proper nouns), and the second one used a ranking loss function rather than a classification function. However accurate their models were at picking a correct antecedent for a true anaphor, the best they could achieve in terms of f-scores was 88.1% with MUC, 85.2% with B3, and 79.7% with CEAF.

Ferilli et al. [11] focused on anaphora resolution in English texts. Their approach improves on the traditional algorithm that is considered the standard baseline for comparison in the literature (by the Hobbs algorithm). Whilst the most significant contribution is provided by a gender agreement feature, the modification to the general rules alone already yields an improvement, for which they propose to use their algorithm as the new baseline in the literature.

Karthikeyan et al. [12] identified all types of anaphora with layered or step by step approaches so that everyone utilized the anaphora paradigm in their application. Their results presented an improved framework that performed all the necessary rules for distinguishing pronouns.

Veena et al. [13] introduced a conceptual graph model, which proposed a concept-based graph model that follows a triplet representation with coreference resolution, which extracts the concepts at both the sentence and document level. The extracted concepts are clustered using a modified DB Scan algorithm that then forms a belief network. Their model had an accuracy of 91%.

Veena et al. [14] proposed a machine learning approach using support vector machines (SVM) towards coreference resolution at the document level. In their research work, 17 well-defined syntactic and semantic features including the 13 baseline features with semantic role labeling (SRL) were used. The use of an SVM classifier led to a better outcome when compared to other machine learning models.

Novák [15] introduced the Treex CR system, which implemented a sequence of evaluation models for individual types of coreference expressions (personal, reflexive, indefinite, interrogative, and possessive pronouns). It used a rich set of features extracted from linguistically pre-processed data. The system seemed to outperform the baseline system in Czech. In English, although it could not outperform the best approaches in the Stanford system, its performance was high enough to be used in future experiments.

Mohan et al. [16] proposed a solution to the problem of ambiguous pronouns in the English language. In their work, they used a dataset called GAP (Webster et al., 2018), which contains 8908 labeled pairs of antecedent naming of a person and an ambiguous pronoun from the Wikipedia database. They trained the model using the BERT technique to obtain contextual embeddings from the text, which they applied to the SVM classifier. Their proposed model demonstrated promising performance; it had an accuracy of 78.35%, a precision of 75.82%, a recall of 69.30%, an F1 score of 71.50%, and a low loss of 0.53.

In this article, we focus on improving the classification of textual data using coreference resolution. This approach is very similar to data augmentation methods. Wei et al. [17] utilized four techniques for expanding the text corpus, synonym replacement, random insertion, swapping, and deletion, which are collectively known as easy data augmentation (EDA). Through experiments on five classification tasks, they found an improved performance for convolutional and recurrent neural networks. Models trained using EDA surpassed this number by achieving an average accuracy of 88.6% while only using 50%

of the available training data. A limitation of this paper is that the performance gains achieved with the proposed data augmentation method (EDA) may be marginal when the dataset is already sufficient. While EDA shows promise for small datasets, it might not yield significant improvements when using pre-trained models like ULMFit, ELMo, or BERT. Additionally, comparing EDA with other data augmentation methods used in NLP becomes challenging due to variations in the models and datasets used in different studies.

Haralabopoulos et al. [18] used a text data augmentation technique—sentence permutations—to create synthetic data based on an existing labeled dataset. As a conclusion from the mentioned research works, the solution to the CR issue entails the design of new methods and a search for ways in which we can identify references in the text in the most efficient way. Their permutation augmentation improved the baseline classification accuracy by 4% on average and outperformed all the other augmentations proposed in their work by an average of 0.2%.

All the research works are summarized in Table 1.

Table 1. Summarization of research papers.

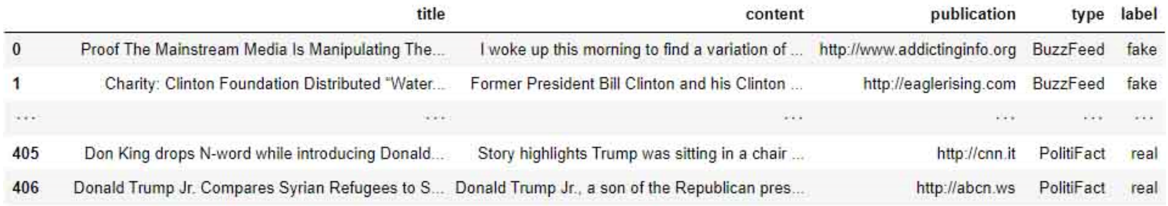
Author	Approach	Models and Methods	Dataset	Keywords
Nadeem et al. [6]	Created the FAKTA framework for verifying fake news.	Model of obtaining and reviewing documents.	Fact extraction and verification dataset	Fake news
Bengtson et al. [7]	Developed a simple pairwise classification model for CR.	Document-level decision model.	ACE training dataset	Coreference resolution
Ming [8]	Presented a model that retrieves word and character representations from Chinese texts.	Skip-gram and BERT models, bidirectional LSTM.	CoNLL-2012	Coreference resolution, BERT, and bidirectional LSTMs
Devlin et al. [9]	Developed the BERT technique.	The BERT model.	GLUE, SQuAD, and SST	BERT, NLP, and nested words
Denis et al. [10]	Presented two strategies for improving CR.	A standard one-model approach based on classification.	ACE training dataset	Coreference resolution and cross-linguistic learning
Ferilli [11]	Improved the traditional algorithm to handle different kinds of anaphora using gender recognition.	Extension of the Hobbs algorithm.	Brown corpus and NLTK corpus	Anaphora resolution
Karthikeyan et al. [12]	Presented an improved framework for distinguishing pronouns.	Anaphora resolution model.	-	Anaphora resolution, reference
Veena et al. [13]	Introduced a conceptual graph model.	Vector space model called the bag-of-words model, modified DB Scan algorithm.	Two hundred articles collected from the ACM digital library	Coreference resolution, information extraction, and RDF
Veena et al. [14]	Proposed a machine learning approach using the support vector method.	SVM classifier, comparison with Decision Tree.	Two hundred articles per training set, one hundred articles per test set from a digital library	Coreference resolution and machine learning
Novák [15]	Introduced the Treex system for setting coreferences for the Czech and English languages.	Treex CR model implemented from the Vowpal Wabbit Toolkit.	Czech and English datasets	Coreference resolution, parallel corpus, and bilingual coreferencing
Mohan et al. [16]	Proposed a solution to ambiguous pronouns in the English language.	The BERT model with a subsequent application of the SVM classifier.	GAP dataset	Coreference resolution and ambiguous pronouns
Wei et al. [17]	Presented the easy data augmentation techniques.	Four EDA operations and convolutional and recurrent neural networks.	Various datasets	EDA, CNN, and RNN models
Haralabopoulos et al. [18]	Developed a sentence permutation method to augment an initial dataset.	Deep learning model LSTM.	Eight comprehensive datasets	Augmentation, multilabel, and LSTM

3. Methodology

To verify the suitability of the pre-preparation of text data using coreference resolution for classification tasks, we proceeded according to the methodology we mentioned at the beginning of this article. We applied this method to the selected dataset. In this chapter, we describe the dataset and the algorithm for coreference resolution in more detail. It was necessary to create several versions of the classifier and verify their success using performance measures. We also describe the methods used to create the word vectors that represent the inputs to the classifiers.

3.1. Dataset

We used the freely available dataset KaiDMML, which was originally created for the needs of the FakeNewsTracker system [19,20]. The dataset contains records collected from the PolitiFact and GossipCop projects. On Figure 4, a sample of the used dataset is presented.



	title	content	publication	type	label
0	Proof The Mainstream Media Is Manipulating The...	I woke up this morning to find a variation of ...	http://www.addictinginfo.org	BuzzFeed	fake
1	Charity: Clinton Foundation Distributed "Water...	Former President Bill Clinton and his Clinton ...	http://eaglerising.com	BuzzFeed	fake
...	...	...	...	...	...
405	Don King drops N-word while introducing Donald...	Story highlights Trump was sitting in a chair ...	http://cnn.it	PolitiFact	real
406	Donald Trump Jr. Compares Syrian Refugees to S...	Donald Trump Jr., a son of the Republican pres...	http://abcn.ws	PolitiFact	real

Figure 4. KaiDMML dataset sample [21].

Both projects tried to verify the facts to determine their truth and correctness, mainly from news portals or social networks. The first project focused on the field of politics, and the second one focused on the field of verifying information about famous personalities. The dataset was manually annotated, where each message was marked as real or fake. Table 2 provides an overview of the basic analysis of the number of words in the dataset.

Table 2. Number of records in the KAIDMML dataset.

Dataset Name	All Logs	All Words	Unique Words	Average Number of Words in a Log	Shortest Log	Longest Log
KaiDMML (real)	197	124,390	6920	631.42	31	5459
KaiDMML (fake)	208	87,909	5728	422.64	5	4155
KaiDMML fake + real	405	212,299	9224	524.20	5	5459

3.2. Coreference Resolution Algorithm—Neuralcoref

Before we introduce the characteristics of the neuralcoref algorithm, we should consider one of the three different CR approaches [19]:

- Deterministic—coreference identification based on natural language rules [20,22];
- Statistical—coreference identification based on machine learning [19];
- Neural—coreference identification based on the neural network [20,22].

We decided to use the open-source library neuralcoref in our work, which is an implementation of the mention-ranking coreference model from [23] and provides very good results in terms of coreference resolution. The current version of the neuralcoref library (as of November 2022) is 4.0. It was necessary to have Spacy version 2.1.0 installed.

The product of the model is the rating (score) for the pair s(c,m), where c is designated as a candidate for antecedent and m is a mention. The given score represents the compatibility of the coreference with the feedforward neural network [24].

At the same time, the model extracts different words, e.g., the base form of a mention with a group of words. Each word is represented by a vector $w_i \in R^{d_w}$, and each group of words is represented by the average of the vectors of all the words in the group. It also considers the distance and match of strings. Heuristic loss functions were also used when training the model [25].

In the context of gender association, the trained word embeddings refer to vectors that capture the relationship between words and their gender connotations. These vectors are generated using algorithms that analyze large datasets, such as the OntoNotes corpus. By utilizing the OntoNotes corpus, the algorithm is able to train the word embeddings to accurately determine the gender of given entities. This includes the ability to assign a coreference even for unknown entities (such as nicknames, names of celebrities, etc.). Overall, the trained word embeddings enable the algorithm to effectively analyze and determine gender associations, providing valuable insights into the feminine and masculine gender connotations of the given entities [26].

3.3. The Techniques Used to Create the Input Vectors

We worked with two versions of the dataset: one with and one without coreference resolution. These two versions were used separately for the next steps of creating the word vector models. We considered using the TF–IDF and Doc2Vec techniques to create the input vector for classification. The vectors were subsequently used as an input for the following classification algorithms: Decision Tree, Random Forest, MultinomialNB, K-Nearest Neighbors, and Logistic Regression. The individual steps of our methodology are depicted in Figure 2.

3.3.1. Term Frequency–Inverse Document Frequency

Nowadays, the classification of text documents is used in various areas of computer science. Algorithms dealing with this task usually need to represent the input text as a vector with a fixed size [27]. The simplest method for creating word vectors from documents is TF (term frequency) [28], which is calculated according to (1):

$$tf_{i,j} = \frac{n_{i,j}}{dl_j} \tag{1}$$

where:

- $n_{i,j}$ is the number of terms I in the document j;
- dl_j is the number of all terms in the document.

However, the above-mentioned method is not reliable, as the most important words will usually be words with frequent occurrences, e.g., pronouns, conjunctions, prepositions, particles, or articles, unless they have been removed as stop words. Simultaneously, a document also can contain words that describe the essence of the text much better, and those words may not be marked with the TF method.

Therefore, the IDF (inverse document frequency) method is a method (2) where $idf_{i,j}$ is determined by a logarithmic calculation. The method involves dividing the number of all documents N and the number of documents in which the word i occurs. Priority is given to words that are specific to a small subset of documents.

$$idf_{i,j} = log\frac{N}{|(j : i \in j)|}\ , \tag{2}$$

If the word does not exist in the corpus, the denominator is adjusted to the form $|(j : i \in j)| + 1$, so the division by zero does not occur. Subsequently, the TF–IDF method, which is the improved product of the TF statistic and the IDF statistic, has the following form (3):

$$TFIDF_{i,j} = tf_{i,j} \times idf_{i,j}, \tag{3}$$

There are many formulas for calculating the TF–IDF, as there are also other parameters that can affect the vector itself [29]. More advanced keyword extraction methods include the KEA (keyphrase extraction algorithm) [30] or TextRank [31], in which the similarities of individual word subsequences are heuristically evaluated. Another popular algorithm is RAKE (rapid automatic keyword extraction) [32], the main advantage of which is that it can work with an article regardless of the corpus.

3.3.2. Word2Vec

A weak point of the previous approaches is the neglecting of the semantics of a text [33]. Word2Vec was introduced to overcome this. The Word2Vec model uses a two-layer neural network and was created in 2013 from the need to capture the semantic similarity of words [34]. A corpus—a set of text—is needed for the training process itself (learning neural networks).

The output of the trained Word2Vec models are word vectors containing a large cluster of data, where individual elements (words) are represented by their position in a multidimensional space. Logical and mathematical operations can be performed with the mentioned vectors, and, thus, they can simulate semantic or lexical relations between words. Formula (4) gives the most common example of a trained Word2Vec vector and determines that the difference between the words "father" and "man" is equal to the difference between the words "mother" and "woman". The mentioned relations are language-independent.

$$v_{\text{father}} - v_{\text{man}} = v_{\text{mother}} - v_{\text{woman}}, \tag{4}$$

Word2vec can utilize one of two possible model architectures to produce these distributed representations of words:

- CBOW (continuous bag of words)—the algorithm predicts the missing word depending on the other words in the document;
- Skip-gram—opposite of CBOW. It receives one word for the input to predict the surrounding window of context words.

Both algorithms work as a neural network, which is trained by a standard procedure. The architecture is shown in Figure 5. The implementation of both algorithms is also used by the Deeplearning4j [35] and Gensim [36] tools.

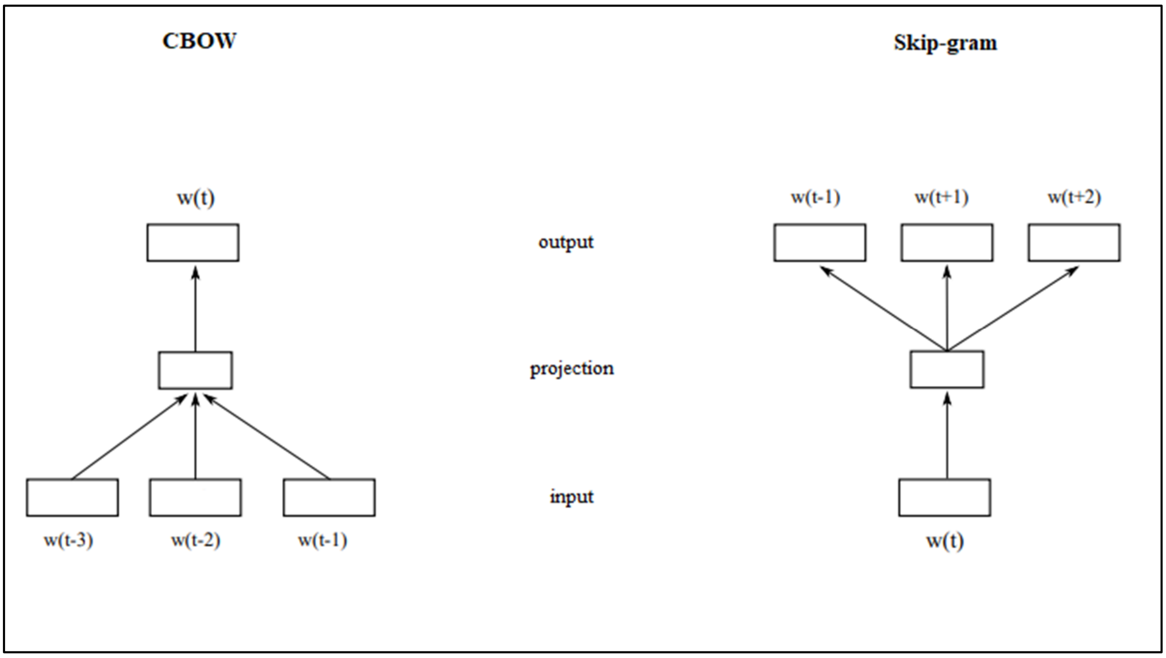


Figure 5. Model architecture of CBOW and Skip-gram [22].

3.3.3. Doc2Vec

The Doc2Vec model works by describing whole sentences using vectors, while corpus words are learned separately. This model is strongly based on the principle of the Word2Vec model. They differ in the case of Doc2Vec, the goal of which is to create a vector representation of the entire document instead of the words [37].

The training of sentence vectors in the Doc2Vec method is based on the word vector methods from Word2Vec. The first of the methods is PV–DM (distributed memory model of paragraph vectors), and it works on the CBOW principle. A document ID is added to the model, which uniquely identifies said document. Each sentence is mapped to a vector of the same size as the individual word vectors. Vectors are also created for the document itself in the same way [38].

Another Doc2Vec method is PV–DBOW (distributed bag of words of paragraph vector), which ignores the context of words in the input. It is less memory-demanding than the PV–DM technique, and it is very similar to the Skip-gram algorithm from Word2Vec [39]. The architecture of both algorithms is shown in Figure 6.

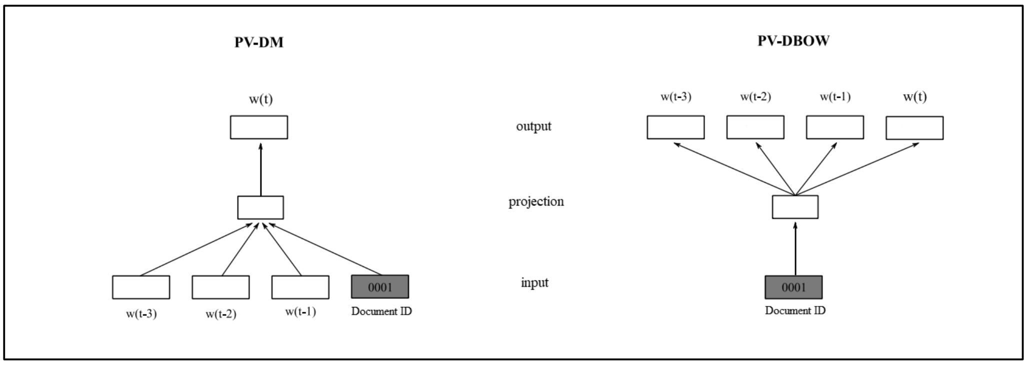


Figure 6. Doc2Vec architecture [22].

We decided to use the last-mentioned model in our work, as we needed to determine the vectors of the entire document with and without the use of CR.

3.3.4. The Techniques Used to Create Classification Methods

We prepared multiple fake news classifiers to verify the effectiveness of the coreference resolution. These were created using the following classification algorithms:

- Decision Tree algorithm;
- Random Forest classifier [40];
- K-Nearest Neighbors [41];
- Multinomial Naive Bayes model [42];
- Logistic Regression.

We used K-fold validation to evaluate the created models. The K-fold technique is a popular and easy-to-understand technique, which generally results in a less biased model. The reason for this is that it ensures that every observation from the original dataset has the chance of appearing in the training and test set.

Firstly, we shuffled our dataset so that the order of the inputs and outputs was completely random. We conducted this step to make sure that our inputs were not biased in any way. In the K-fold cross-validation step, the original sample was randomly partitioned into k equally sized subsamples. Of the k subsamples, a single subsample was retained as the validation data for testing the model, and the remaining $k-1$ subsamples were used as the training data. The cross-validation process was then repeated k times with each of the k subsamples used exactly once as the validation data.

The advantage of this method is that all observations were used for both training and validation, and each observation was used for validation exactly once. When evaluating all the models we created, $k = 10$, i.e., the input dataset was randomly divided into 10 parts.

We also focused on the following four widely used metrics, accuracy, precision, F1 score, and recall, to assess the performance of our models.

Accuracy is a fundamental performance measure used in evaluating machine learning models. It represents the ratio of correctly predicted instances to the total number of instances in the dataset. Mathematically, it is defined as per Formula (5):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{5}$$

In the binary classification, we used four key metrics: true positives (*TP*), true negatives (*TN*), false positives (*FP*), and false negatives (*FN*). TP represents the number of correctly predicted positive instances, *TN* represents the number of correctly predicted negative instances, *FP* refers to the number of incorrectly predicted positive instances, and *FN* indicates the number of incorrectly predicted negative instances. These metrics helped us to assess the model's performance and make informed decisions as to how to improve it.

Precision assesses the accuracy of positive predictions made by the model. It represents the proportion of true-positive predictions (correctly predicted positive instances) among all the positive predictions (Formula (6))

$$Precision = \frac{TP}{TP + FP}, \tag{6}$$

Precision is particularly useful when the cost of false positives is high, as it helps in minimizing the number of false alarms. However, a high precision score may be accompanied by a low recall score, leading to an increased number of false negatives.

Recall calculates the proportion of true-positive predictions among all the actual positive instances in the dataset (Formula (7)).

$$Recall = \frac{TP}{TP + FN}, \tag{7}$$

The final measure is the F1 score, which is a metric that combines precision and recall to provide a balanced evaluation of a model's performance. It is the harmonic mean of precision and recall and is mathematically represented in Formula (8):

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall},$$

The F1 score reaches its highest value at 1 (perfect precision and recall) and its lowest at 0. It is particularly useful when dealing with imbalanced datasets, where accuracy can be misleading. The F1 score considers both false positives and false negatives, making it a suitable measure when the overall model performance needs to be balanced.

4. Results

We applied the existing coreference resolution method to the selected dataset using the neuralcoref library. We cleaned the data and removed null values in the form of empty rows in the pre-processing stage of our work. We did not remove the stop words or lemmatization since we could have significantly disrupted the identification of pronouns, as the library needs to work with the original text.

We used the Scikit-learn library to create word vectors using TF–IDF. The gensim library was used for the Doc2Vec method. The investigated classification models were created using the methods of the Scikit-learn library. We used K-fold validation for all the models, and we set the number of splits as $k = 10$.

Five classification methods (Decision Tree, Random Forest, K-Nearest Neighbors, MultinomialNB, and Logistic Regression) and two word-embedding methods (TF–IDF, Doc2Vec) were selected. The used methods were examined using K-fold cross-validation, where k represented 10 measurements. In this way, 160 classification models were created (data without coreferencing and data with coreferencing (2) × word embedding methods (2) × classification methods (4) × k-fold validation (10)) for the investigated fake news dataset. The quality of the created models was evaluated using evaluation metrics (*accuracy, precision, recall, f1 score, precision_fake, recall_fake, precision_real,* and *recall_real*). The individual stages of our proposed method are described in Figure 2.

The descriptive statistics for the accuracy results are presented in Table 3. The Valid N value represents the four classification methods used for the 10-fold cross-validation. The mean and median proved that better results were observed for the TF–IDF method compared to the Doc2Vec method. After closer inspection, better outcomes were achieved (mean and median) for the models created from the input data after applying coreference resolution. The range of the variation in both word-embedding methods (*TfIdf_coref* and *Doc2Vec_coref*) also decreased.

Table 3. Descriptive statistics for accuracy results.

	Tfidf _Without	**Tfidf _Coref**	**Doc2Vec _Without**	**Doc2Vec _Coref**
Valid N	50	50	50	50
Mean	0.725991	0.747058	0.668720	0.680290
Median	0.775000	0.780488	0.682927	0.682927
Min	0.475000	0.487805	0.414634	0.487805
Max	0.900000	0.878049	0.950000	0.900000
Lower quartile	0.646341	0.703659	0.600000	0.617378
Upper quartile	0.804878	0.825000	0.740854	0.756098
Std. dev.	0.120152	0.099022	0.111137	0.103146
Skewness	−0.834287	−1.020730	0.044494	−0.037169

Similarly, better results were also observed in the case of coreference resolution for the other performance measures. The results for the F1 score, which is a combination of precision and recall, are presented in Table 4.

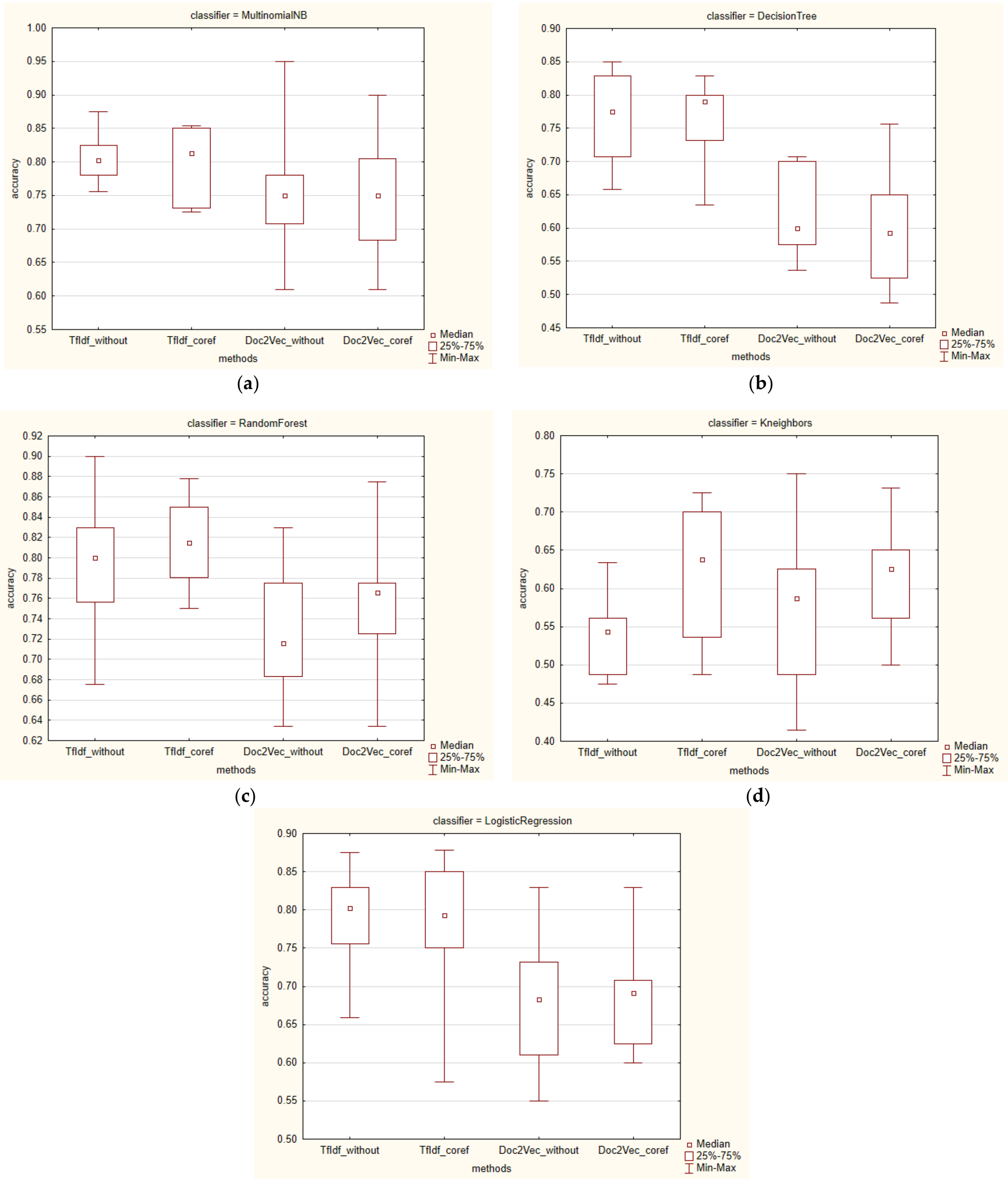
Table 4. Descriptive statistics for F1 score results.

	Tfidf _Without	**Tfidf _Coref**	**Doc2Vec _Without**	**Doc2Vec _Coref**
Valid N	50	50	50	50
Mean	0.701709	0.737156	0.648874	0.659382
Median	0.766992	0.777031	0.653872	0.654276
Min	0.365512	0.482883	0.349206	0.404762
Max	0.895833	0.878049	0.949875	0.899749
Lower quartile	0.601746	0.663276	0.567196	0.576818
Upper quartile	0.802880	0.816453	0.720862	0.754184
Std. dev.	0.151805	0.107234	0.121631	0.114638
Skewness	−0.957966	−1.02115	−0.029760	−0.073310

The results of the accuracy and F1 score (Tables 3 and 4) showed that the difference between the models created from the datasets without and with the application of coreference resolution were not so striking. Similarly, insufficient results of the models were observed with the other evaluation measurements (*precision, recall, precision_fake, recall_fake, precision_real,* and *recall_real*).

Therefore, we analyzed the results for the used classification algorithms individually in more detail. The results of the 10-fold cross-validation for each classification technique (Decision Tree, Random Forest, K-Nearest Neighbors, MultinomialNB, and Logistic Regression) depended on the methods of preparing the input vectors for the classification methods. The input vectors were prepared without coreference resolution (parameter *_without*) and with coreference resolution (parameter *_coref*). At the same time, two word-embedding methods (TF–IDF and Doc2Vec) were applied for the preparation of the vectors.

The results for accuracy are visualized in the following individual graphs (Figure 7a–d). We compared the combinations of the text preparation methods (*_without* and *_coref*) and word-embedding methods (TF–IDF and Doc2Vec).



(**a**) (**b**)

(**c**) (**d**)

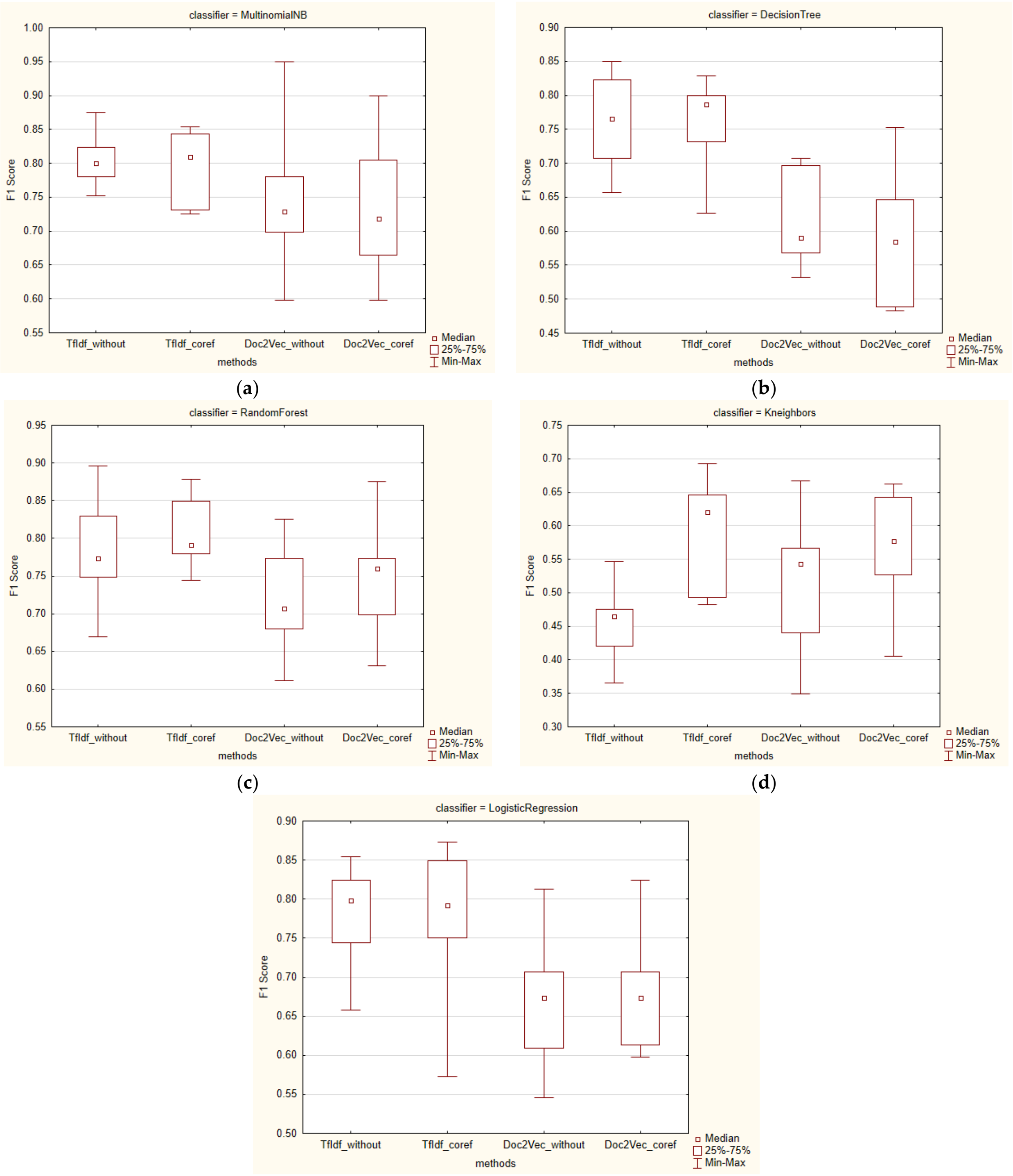
(**e**)

Figure 7. Box plot for the model performance in terms of measurement accuracy for different classifiers: (**a**) MultinomialNB, (**b**) Decision Tree, (**c**) Random Forest, (**d**) K –Nearest Neighbors, and (**e**) Logistic Regression.

The graphs show that the median was generally higher for the classification methods with the application of coreference resolution. This was observed in all five classifiers, except for TF–IDF for Logistic Regression. Similarly, a higher value of Q3 was observed for most methods, except for Decision Tree and Doc2Vec for Logistic Regression, in favor of the methods with the application of coreference resolution. On the other hand, in the K-Nearest

Neighbors and MultinomialNB classifiers, there was a greater degree of variability in the coreference resolution methods.

Simultaneously, we also analyzed the results from the F1 score point of view (Figure 8a–d), which combines precision and recall. Even in this case, higher median values were observed for all F1 score metrics except for word embedding with Doc2Vec and the MultinomialNB classifier.



(**a**)

(**b**)

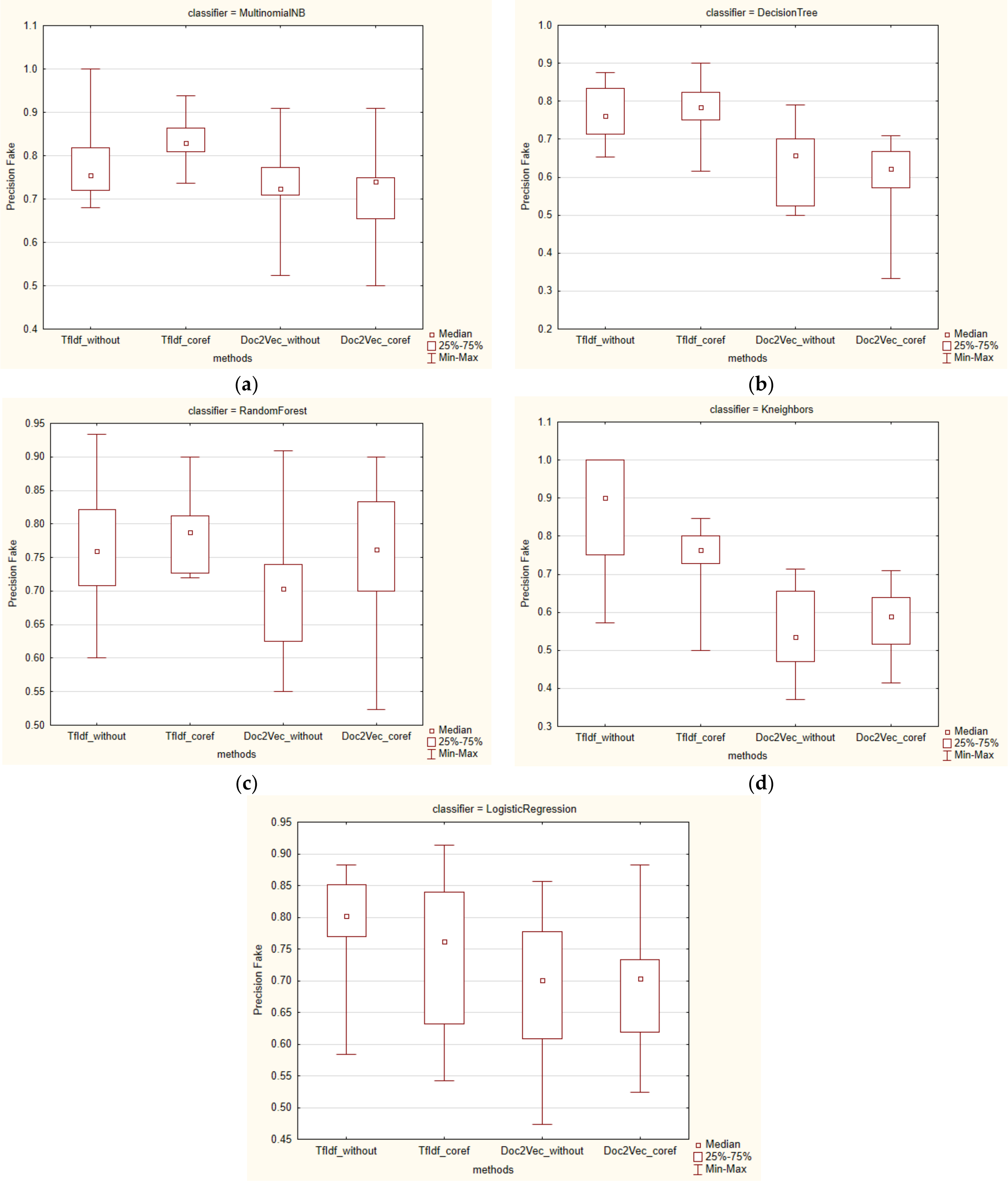
(**c**)

(**d**)

(**e**)

Figure 8. Box plots for the model performance in terms of F1 scores for different classifiers: (**a**) MultinomialNB, (**b**) Decision Tree, (**c**) Random Forest, (**d**) K-Nearest Neighbors, and (**e**) Logistic Regression.

All the methods were applied with the aim of identifying fake news. For this reason, we present the results (Figure 9a–d) for the performance measure precision_fake.



(**a**) (**b**)

(**c**) (**d**)

(**e**)

Figure 9. Box plots for the model performance in terms of precision_fake metric for different classifiers: (**a**) MultinomialNB, (**b**) Decision Tree, (**c**) Random Forest, (**d**) K-Nearest Neighbors, and (**e**) Logistic Regression.

In addition, the perspective of the median of the variability in these results was also interesting. The median, except for TF–IDF for the K-Nearest Neighbors and Logistic Regression methods and Doc2Vec for the Decision Tree method, was higher in all the other methods after applying coreference resolution.

Also, the quartile range for the degree of variability was analyzed. An interesting finding was that, except for Doc2Vec for the Random Forest and MultinomialNB methods and TF–IDF for the Logistic Regression method, smaller quartile ranges were found for the methods with coreference resolution.

5. Discussion

We aimed at improving text classification using coreference resolution in the presented work. We used the freely available KaiDMML dataset, which contains 405 news articles, half of which were annotated as fake news and the other half as real news. The advantage of this dataset is its manual annotation. However, it is natural that the observed performance metrics of our models were lower using this dataset than in the case of the datasets that are not manually annotated. The second disadvantage of this dataset is its small size.

We applied CR analysis to the selected dataset and compared the results of the classification of fake news without applying the CR method. The TF–IDF and Doc2Vec methods were used to create the input vectors. We chose Decision Tree, Random Forest, K-Nearest Neighbors, MultinomialNB, and Logistic Regression from the classification methods.

Despite our efforts, no significant differences in the results were identified. Therefore, we only analyzed the results with descriptive statistics, and we did not verify the statistical significance of the differences between the investigated approaches.

This finding is important. The results were improved by applying the CR method. Unfortunately, its impact was not statistically significant. The only difference was observed for the K-Nearest Neighbors classifier. However, this classifier achieved very poor results compared to the other classifiers used. The difference in the results between the methods with CR and those without applying CR was therefore rather due to the poor results of the K-Nearest Neighbors classifier.

The finding that CR led to better classification results but not to statistically significant differences was surprising. In the case of the TF–IDF method in particular, we expected significantly better results for CR. This expectation was given by the TF–IDF method itself, where the values of the vector element for specific objects were practically increased, e.g., nouns and decreased values and pronouns or abbreviations (mentions). We assumed that by increasing the values of the elements expressing specific objects at the expense of mentions, we would achieve significantly better results.

However, it was clear from the results of the descriptive statistics that applying CR contributed to better results, especially when examining the median value for accuracy, the F1 score, and the precision_fake metric. For this reason, we can conclude that the application of CR improved the classification tasks in general.

An interesting and surprising finding was that the TF–IDF method achieved better results compared to the Doc2Vec method. In our opinion, this was caused by every creation of a vector in the Doc2Vec method, which needs a robust corpus for its effective deployment. In our case, it was only 405 records.

Although the results were not statistically significant, it can be concluded from the descriptive statistics that the application of CR can contribute to improving the classification of fake news.

The application of coreference resolution (CR) to a dataset can be classified as a data augmentation technique. There are two related research studies in this area. Haralabopoulos et al. [18] used a text data augmentation technique—sentence permutations—to create synthetic data based on an existing labeled dataset. Their method achieved a significant improvement in terms of classification accuracy, averaging around 4.1% across eight different datasets. Additionally, they proposed two other methods for text data expansion: antonym replacement and negation. These methods were tested on three suitable datasets and achieved accuracy improvements of 0.35% (antonyms) and 0.4% (negation) compared to the permutation method they proposed.

Wei et al. [17] utilized four techniques for expanding the text corpus called easy data augmentation (EDA) which included synonym replacement, random insertion, swapping, and deletion. Through experiments on five classification tasks, they found an improved performance for convolutional and recurrent neural networks.

To improve the accuracy and F1 score results, the application of CR increased the median value for the accuracy by 0.0942 (K-NN/TF–IDF) and for the F1 score by 0.1558 (K-NN/TF–IDF). In comparison to the mentioned studies, our improvement may seem relatively small, but it is important to note that our article focused solely on one technique. In the mentioned experiments, the authors worked with a combination of 3–4 techniques, making the results appear comparable. Nevertheless, we believe that the more crucial finding is the fact that the application of CR led to a classification improvement across practically all the classifiers. It is also worth noting that in augmenting the data of natural language processing, the primary concern is often corpus augmentation, whereas in our article, we directly applied coreference resolution to the classified data.

6. Conclusions

We focused our work on the design and verification of procedures for the field of natural language processing. Within this area, we chose the issue of coreference identification and a dataset containing fake news. We used the coreference algorithm for the input data, and, thus, we created two sets of texts—a text with the application of coreference resolution and a text without the application of coreference resolution. Subsequently, we evaluated these two texts using vector models, namely TF–IDF and Doc2Vec. Finally, we implemented various machine-learning-based classifiers in our research, from which we selected the Decision Tree, Random Forest, K-Nearest Neighbors, MultinomialNB, and Logistic Regression methods as the most suitable. The classifiers categorized news articles as either fake or real news based on their classification result.

According to the achieved results, the data processing with the application of coreference resolution resulted in an increase in the accuracy of the classification of fake news.

In the case of the accuracy performance measure, the best result was observed for the Random Forest classifier using the *TfIdf_coref* method (*median* = 0.8149). For the F1 score, the best result was observed for the MultinomialNB classifier using the *TfIdf_coref* method (*median* = 0.8101). When looking at the results in terms of increased accuracy, an improvement was observed in all methods except for the Decision Tree classifier with Doc2Vec, where a decrease in accuracy was observed when comparing the metrics *_coref* and *_without*. The most significant increase in accuracy was observed for the K-Nearest Neighbors classifier with the TF–IDF embedding method (0.0942 increase for *_coref*).

Regarding the results from the perspective of an increased F1 score, an improvement was observed in all the methods except for the Decision Tree and MultinomialNB classifiers with Doc2Vec, where a decrease in accuracy was observed when comparing the metrics *_coref* and *_without*. The most notable increase in the F1 score was observed for the K-Nearest Neighbors classifier and the TF–IDF embedding method (0.1557 increase for *_coref*).

Coreference resolution represents a single method for dataset preparation in the classification tasks of fake news identification. For this reason, even a slight improvement in the method could cause an important shift in the researched area, because the method would be used as part of other methods for dataset preparation.

We want to find and use other tools for the application of coreference resolution in future work. The neuralcoref library was used as a deep learning neural network model in this study. This library has not been updated for a long time, and its implementation is only possible with the support of the Spacy 2.1.0 software, which has long been replaced by newer versions.

It is clear from the results that the method needs to be verified with several other datasets. We plan to validate this method on other datasets with a different thematic focus.

An important part of further work would also be improving the accuracy of the method in various other classification tasks of natural language processing, such as classifications of text type, content, or tone. It would also be a research challenge to validate this method in collaboration with other word vector preparation methods.

Author Contributions: Conceptualization, K.Š. and J.K.; methodology, K.Š. and J.K.; validation, K.Š. and J.K.; formal analysis, K.Š. and J.K.; resources, K.Š. and J.K.; data curation, K.Š. and J.K.; writing—original draft preparation, K.Š. and J.K.; writing—review and editing, K.Š. and J.K.; visualization, K.Š.; supervision, J.K.; project administration, J.K.; funding acquisition, J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Scientific Grant Agency of the Ministry of Education of the Slovak Republic and Slovak Academy of Sciences under Contract VEGA-1/0490/22, also by the Slovak Research and Development Agency under contract no. APVV-18-0473, and also by the European Commission ERASMUS+ Program 2021, under Grant 2021-1-SK01-KA220-HED-000032095.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study are freely available online at: https://github.com/KaiDMML/FakeNewsNet.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Afsharizadeh, M.; Ebrahimpour-Komleh, H.; Bagheri, A. Automatic Text Summarization of COVID-19 Research Articles Using Recurrent Neural Networks and Coreference Resolution. *Front. Biomed. Technol.* **2020**, *7*, 236–248. [CrossRef]
2. Bhattacharjee, S.; Haque, R.; de Buy Wenniger, G.M.; Way, A. Investigating Query Expansion and Coreference Resolution in Question Answering on BERT. In *Natural Language Processing and Information Systems*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 47–59. [CrossRef]
3. Beheshti, S.-M.-R.; Benatallah, B.; Venugopal, S.; Ryu, S.H.; Motahari-Nezhad, H.R.; Wang, W. A systematic review and comparative analysis of cross-document coreference resolution methods and tools. *Computing* **2017**, *99*, 313–349. [CrossRef]
4. Seljan, S.; Tolj, N.; Dunđer, I. Information Extraction from Security-Related Datasets. In Proceedings of the 2023 46th MIPRO ICT and Electronics Convention (MIPRO), Opatija, Croatia, 22–26 May 2023; pp. 539–544. [CrossRef]
5. Kovač, A.; Dunđer, I.; Seljan, S. An overview of machine learning algorithms for detecting phishing attacks on electronic messaging services. In Proceedings of the 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 23–27 May 2022; pp. 954–961. [CrossRef]
6. Nadeem, M.; Fang, W.; Xu, B.; Mohtarami, M.; Glass, J. FAKTA: An Automatic End-to-End Fact Checking System. In Proceedings of the NAACL HLT 2019—2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies—Proceedings of the Demonstrations Session, Minneapolis, MN, USA, 2–7 June 2019; pp. 78–83. [CrossRef]
7. Bengtson, E.; Roth, D. Understanding the Value of Features for Coreference Resolution. 2008, pp. 294–303. Available online: http://L2R.cs.uiuc.edu/ (accessed on 24 August 2022).
8. Ming, K. Chinese Coreference Resolution via Bidirectional LSTMs using Word and Token Level Representations. In Proceedings of the 2020 16th International Conference on Computational Intelligence and Security, CIS 2020, Nanning, China, 27–30 November 2020; pp. 73–76. [CrossRef]
9. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL HLT 2019—2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies—Proceedings of the Conference, Minneapolis, MN, USA, 11 October 2018; pp. 4171–4186. [CrossRef]
10. Denis, P.; Baldridge, J. Specialized Models and Ranking for Coreference Resolution. In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, HI, USA, 25–27 October 2008; pp. 660–669.
11. Ferilli, S.; Redavid, D. Experiences on the Improvement of Logic-Based Anaphora Resolution in English Texts. *Electronics* **2022**, *11*, 372. [CrossRef]
12. Karthikeyan, K.; Karthikeyani, V. Understanding text using Anaphora Resolution. In Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, PRIME 2013, Salem, India, 21–22 February 2013; pp. 346–350. [CrossRef]
13. Veena, G.; Krishnan, S. A concept based graph model for document representation using coreference resolution. *Adv. Intell. Syst. Comput.* **2016**, *384*, 367–379.

14. Veena, G.; Gupta, D.; Daniel, A.N.; Roshny, S. A learning method for coreference resolution using semantic role labeling features. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017, Udupi, India, 13–16 September 2017; pp. 67–72. [CrossRef]
15. Novák, M. Coreference Resolution System Not Only for Czech. Available online: https://github.com/ufal/treex/ (accessed on 24 August 2022).
16. Mohan, M.; Nair, J.J. Coreference Resolution in Ambiguous Pronouns Using BERT and SVM. In Proceedings of the 2019 International Symposium on Embedded Computing and System Design, ISED 2019, Kollam, India, 13–14 December 2019; pp. 68–72. [CrossRef]
17. 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, 3–7 November 2019; pp. 6382–6388.
18. Haralabopoulos, G.; Torres, M.T.; Anagnostopoulos, I.; McAuley, D. Text data augmentations: Permutation, antonyms and negation. *Expert Syst. Appl.* **2021**, *177*, 114769. [CrossRef]
19. Shu, K.; Mahudeswaran, D.; Liu, H. FakeNewsTracker: A tool for fake news collection, detection, and visualization. In *Comput Math Organ Theory*; Carley, K.M., Frantz, T.L., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 25, pp. 60–71.
20. Shu, K.; Mahudeswaran, D.; Wang, S.; Lee, D.; Liu, H. FakeNewsNet: A Data Repository with News Content, Social Context and Spatialtemporal Information for Studying Fake News on Social Media. Big Data. *arXiv* **2018**, arXiv:1809.01286. [CrossRef] [PubMed]
21. Lee, H.; Chang, A.; Peirsman, Y.; Chambers, N.; Surdeanu, M.; Jurafsky, D. Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules. *Comput. Linguist.* **2013**, *39*, 885–916. [CrossRef]
22. Manning, C.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.; McClosky, D. The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of the ACL Anthology, Baltimore, MD, USA, 22 June 2014; pp. 55–60. [CrossRef]
23. Clark, K.; Manning, C.D. Deep Reinforcement Learning for Mention-Ranking Coreference Models. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–4 November 2016; pp. 2256–2262. [CrossRef]
24. Clark, K.; Manning, C.D. Entity-Centric Coreference Resolution with Model Stacking. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 26–31 July 2015; pp. 1405–1415. [CrossRef]
25. Clark, K.; Manning, C.D. Improving Coreference Resolution by Learning Entity-Level Distributed Representations. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 643–653. [CrossRef]
26. Weischedel, R.M.; Hovy, E.H.; Marcus, M.P.; Palmer, M. OntoNotes: A Large Training Corpus for Enhanced Processing. In *Handbook of Natural LanguageProcessing and Machine Translation: DARPA Global Autonomous Language Exploitation*; Olive, J., Christianson, C., McCary, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2011.
27. Le, Q.; Mikolov, T. Distributed Representations of Sentences and Documents. In Proceedings of the 31st International Conference on Machine Learning, ICML 2014, Beijing, China, 22–24 June 2014; pp. 1188–1196. [CrossRef]
28. Wang, R.; Shi, Y. Research on application of article recommendation algorithm based on Word2Vec and Tfidf. In Proceedings of the 2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms, EEBDA 2022, Changchun, China, 25–27 February 2022; pp. 454–457. [CrossRef]
29. Soucy, P.; Mineau, G. *Beyond TFIDF Weighting for Text Categorization in the Vector Space Model*; ACM Digital Library: New York, NY, USA, 2005; pp. 1130–1135.
30. Li, T.; Hu, L.; Li, H.; Sun, C.; Li, S.; Chi, L. TripleRank: An unsupervised keyphrase extraction algorithm. *Knowl. -Based Syst.* **2021**, *219*, 106846. [CrossRef]
31. Mihalcea, R.; Tarau, P. TextRank: Bringing Order into Text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004; pp. 404–411.
32. Rose, S.; Engel, D.; Cramer, N.; Cowley, W. Automatic Keyword Extraction from Individual Documents. In *Text Mining: Applications and Theory*; Berry, M.W., Kogan, J., Eds.; Wiley: Hoboken, NJ, USA, 2010; pp. 1–20.
33. Zhang, Y.; Jin, R.; Zhou, Z.-H. Understanding bag-of-words model: A statistical framework. *Int. J. Mach. Learn. Cybern.* **2010**, *1*, 43–52. [CrossRef]
34. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, 3111–3119. [CrossRef]
35. Papadimitriou, S. Scientific Scripting in Java with JShellLab and application to Deep Learning using DeepLearning4j. *Int. J. Model. Simul. Sci. Comput.* **2020**, *11*, 2050031. [CrossRef]
36. Beysolow, T., II. Applied Natural Language Processing with Python. In *Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing*, 1st ed.; Apress: Berkeley, CA, USA, 2018.
37. Recasens, M.; de Marneffe, M.-C.; Potts, C. The Life and Death of Discourse Entities: Identifying Singleton Mentions. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, GA, USA, 9–14 June 2013; pp. 627–633.
38. Dai, A.M.; Olah, C.; Le, Q.V. Document Embedding with Paragraph Vectors. *arXiv* **2015**, arXiv:1507.07998.

39. Mani, K.; Verma, I.; Meisheri, H.; Dey, L. Multi-Document Summarization Using Distributed Bag-of-Words Model. In Proceedings of the 2018 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2018, Santiago, Chile, 3–6 December 2018; pp. 672–675. [CrossRef]
40. Ho, T.K. Random decision forests. In Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, Montreal, QC, Canada, 14–16 August 1995; pp. 278–282. [CrossRef]
41. Sun, S.; Huang, R. An adaptive k-nearest neighbor algorithm. In Proceedings of the 2010 7th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2010, Yantai, China, 10–12 August 2010; pp. 91–94. [CrossRef]
42. Kibriya, A.M.; Frank, E.; Pfahringer, B.; Holmes, G. Multinomial naive bayes for text categorization revisited. In *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3339, pp. 488–499.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.