applied sciences

MDPI

Article

Multi-Class Document Classification Using Lexical Ontology-Based Deep Learning †

Ilkay Yelmen [1,2,*], Ali Gunes [1] and Metin Zontul [3]

[1] Department of Computer Engineering, Faculty of Engineering, Istanbul Aydin University, Istanbul 34295, Turkey
[2] Turkcell Group Company Digital Educational Technologies Inc., Ankara 06800, Turkey
[3] Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Sivas Science and Technology University, Sivas 58100, Turkey
* Correspondence: ilkayyelmen@stu.aydin.edu.tr or ilkay.yelmen@turkcell.com.tr
† This study is based on a PhD dissertation titled "Deep Learning Approach with Ontology Based Dimension Reduction: An Application on Classification of Unstructured Documents", Istanbul Aydin University, Institute of Graduate Studies.

Abstract: With the recent growth of the Internet, the volume of data has also increased. In particular, the increase in the amount of unstructured data makes it difficult to manage data. Classification is also needed in order to be able to use the data for various purposes. Since it is difficult to manually classify the ever-increasing volume data for the purpose of various types of analysis and evaluation, automatic classification methods are needed. In addition, the performance of imbalanced and multi-class classification is a challenging task. As the number of classes increases, so does the number of decision boundaries a learning algorithm has to solve. Therefore, in this paper, an improvement model is proposed using WordNet lexical ontology and BERT to perform deeper learning on the features of text, thereby improving the classification effect of the model. It was observed that classification success increased when using WordNet 11 general lexicographer files based on synthesis sets, syntactic categories, and logical groupings. WordNet was used for feature dimension reduction. In experimental studies, word embedding methods were used without dimension reduction. Afterwards, Random Forest (RF), Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP) algorithms were employed to perform classification. These studies were then repeated with dimension reduction performed by WordNet. In addition to the machine learning model, experiments were also conducted with the pretrained BERT model with and without WordNet. The experimental results showed that, on an unstructured, seven-class, imbalanced dataset, the highest accuracy value of 93.77% was obtained when using our proposed model.

Keywords: document classification; multi-class classification; word embeddings; WordNet; BERT

Citation: Yelmen, I.; Gunes, A.; Zontul, M. Multi-Class Document Classification Using Lexical Ontology-Based Deep Learning. *Appl. Sci.* **2023**, *13*, 6139. https://doi.org/10.3390/app13106139

Academic Editors: Andrea Prati, Yu-Dong Zhang, Luis Javier García Villalba and Vincent A. Cicirello

Received: 8 February 2023
Revised: 6 May 2023
Accepted: 15 May 2023
Published: 17 May 2023

1. Introduction

Document and text classification is a fundamental task in Natural Language Processing (NLP). This is the task of assigning a class label to a new unclassified document. It has many applications, such as news content classification, spam filtering, opinion mining, etc.

Document classification is structurally different from sentence classification. Documents consist of multiple sentences. Sentences have ambiguous and complex semantic relationships, which makes it difficult to classify documents. In addition, as the number of document categories increases, their management becomes more difficult.

Automatic document classification is a supervised machine learning technique that involves determining whether a particular document belongs to a specific category by analyzing the words or terms used in the document and comparing them to those associated with the category [1]. Moreover, as the number of classes increases, the number of decision

boundaries will also increase. Therefore, it will be difficult for the algorithm to solve the problem. This is particularly difficult if the data is imbalanced.

In a study conducted with the aim of increasing classification success rate, it was emphasized that reducing the size of the feature vector is important for increasing the success of the model. In the study, size reduction algorithms were divided into feature selection and feature extraction algorithms [2]. Feature extraction methods use algebraic transformations to reduce high-dimensional feature vectors to lower-dimensional spaces. Feature extraction algorithms can be divided into two types: linear and nonlinear algorithms [3]. These algorithms perform data transformation using optimization techniques. The most important method is Principal Component Analysis (PCA) [4], which produces new features.

Developments in deep learning methods have resulted in developments in the field of text classification. Experimental studies have been carried out with the aim of increasing the success of classification models, especially with the emergence of the BERT model. In one study, the authors reported that methods using attention mechanisms such as BERT have the ability to capture contextual information present in the document. Within the scope of that study, the VGCNBERT model was proposed, which combines the capability of BERT with a Vocabulary Graph Convolution Network (VGCN). In their experiments, the best results were obtained when using the two-class SST-2 dataset, with an F1-Score of 91.93. The use of WordNet in future studies was also suggested by the authors, as the vocabulary graph can provide useful global information for BERT [5]. Therefore, it can be concluded that the use of WordNet could be beneficial.

In some recent studies, the WordNet lexical ontology and BERT language model were used together to perform document classification, where the role of WordNet was as a source of semantic knowledge, such as with respect to word embeddings, e.g., path2vec and wnet2vec, while that of BERT was to extract the local feature information of the documents and to classify them [5,6].

In contrast to the articles described above, in this study, we propose a new model of WordNet lexical ontology and the BERT language model in which WordNet is used for dimension reduction using lexicography instead of domain ontology with the aim of increasing the success of the classification model on an imbalanced, multi-class dataset containing seven classes. In the experiments carried out in this study, WordNet is applied before classification by machine learning algorithms and BERT. The advantage of our new hybrid model is that it reduces the feature vector size, catches the semantic similarities of the words in the sentence, and provides higher classification success on unstructured, imbalanced and high-dimensional multi-class documents. The contributions of this article are manifold, as detailed below.

1. Evaluation of the effect of using the WordNet ontology for feature dimension reduction on classification success on imbalanced and multi-class (seven class labels) data.
2. Comparison of the performance of machine learning and deep learning classification algorithms when used on a multi-class imbalanced dataset with different imbalance ratios.
3. The use of WordNet for dimension reduction using lexicography instead of domain ontology, together with some word embedding methods, before classical machine learning models increases the success of some of the classification models.
4. The highest success was achieved when using WordNet for dimension reduction with lexicography and the BERT algorithm as a hybrid model. It can be seen from our experimental studies that feature dimension reduction based on lexicography increased classification success.

This article is organized as follows. In Section 2, the existing literature is discussed under four different categories; then, the methodology is presented in Section 3. In Section 4, the preparations made before the experimental studies are explained. In Section 5, the experimental studies and results are evaluated. The last section of the study presents the conclusions and future work.

2. Literature Review

Studies related to text and document classification have been carried out on different subjects and with different purposes. In this section, we will examine the studies in which machine and deep learning methods were used together with semantic knowledge to increase the success of document classification.

2.1. Document Classification

Classifying massive numbers of textual documents is an important need in knowledge discovery. There are many studies available in the literature in the field of classification, and studies on this subject have been performed focusing on feature dimension reduction [7], word embedding [8], and the experimental evaluation of classification algorithms [9]. Document classification applications can be categorized into six groups, as follows: information retrieval, information filtering, sentiment analysis, recommender system, knowledge management, and document summarization [7].

In a study conducted on three different legal document sets using CNN, RNN, GRU and LSTM algorithms, the highest precision value of 92.11% was obtained using the CNN model [10].

In a study researching classification using word embedding methods, experiments were conducted using pretrained word2vec, GloVe, fastText methods and trained in-domain word2vec and Doc2Vec neural word embeddings. The study demonstrated that using word2vec and Doc2Vec neural word embeddings improved distance-based multi-class textual document classification [11].

Word embedding algorithms can be employed with the help of feedforward neural networks to extract semantic relationships from large numbers of text documents. However, they need huge amounts of textual data to perform well. Studies have been conducted using external semantic sources to improve the semantic relations among the words revealed by these models. In one of these studies, Wikipedia was used as an external semantic resource in order to better be able to work with distributed methods like word2vec in problems with small amount of labeled data. In another study, to resolve the requirement for vast training sets for the word2vec word embedding algorithm and to work well in practice, small labeled datasets of semantic features were used, which were reproduced by human experts [12].

2.2. Feature Dimension Reduction Using WordNet Ontology

Ontology-based research has recently become important in the search for linguistic patterns with the aim of increasing classification success. In a study on this subject aiming to classify clauses, entire lines, and sentences in legal text analytics work on contracts, smaller datasets were used, and fewer classes were focused on [13,14]. In another study, a method for extracting specific entities related to market analysis was presented utilizing domain ontology [15].

For the improvement of text classification performance, the information in knowledge bases such as Wikipedia and WordNet can be applied. Semantic correction was presented in [16] by including a priori information from WordNet in text classification. To incorporate semantic similarity between words, a smoothing matrix was employed, which was derived from WordNet. The smoothing was applied to TF-IDF feature vectors to improve semantic coherence. This causes the feature values of terms that are semantically related to increase. Another study that used WordNet to design a semantic smoothing kernel for text classification is [17]. The similarity between words was calculated based on their shared super concepts. Cristianini et al. [18] used LSI to incorporate semantic relations between terms calculated into a kernel.

In another study, sentiment classification was performed using 1,578,627 tweets. Classification success was measured using BoW and Semantic BoW. Here, the similarity calculation between the words was performed using WordNet, and the attribute size was reduced by reducing similar words to one. By using the AdaBoost Classifier and the

KNeighbors Classifier together with Semantic BoW, it was observed that the classification success increased by 1–4% compared to the classical BoW method. However, since the accuracy when using the Semantic BoW method only reached 69%, an improved success rate is needed [19].

Ontology-based dimension reduction methods have become important in classification studies. In a study in which feature dimension reduction was carried out by avoiding word repetition with WordNet, a structure was proposed in which classification was performed with CNN [20]. In another study on this subject, the authors compared Naive Bayes, Jrip, J48 and SVM classification methods with PCA and ontology-based feature reduction methods, and it was seen that ontology-based reduction gave better results than PCA. The highest success rate was obtained by SVM, at 85%. In the study, 15 imbalanced categories belonging to the Reuters-21578 dataset were used [21].

2.3. Binary and Multi-Class Classification

There are many studies in the literature related to binary text classification. Studies have been carried out in this area focusing on the detection of breast and skin cancer [6] and the classification of news-related tweets [22].

Two-class sentiment classification was performed using the SST-2, MR, CoLA and R8 datasets. The Text-GCN, Bi-LSTM, VGCN, BERT, STGCN + BERT + BiLSTM, VGCN-BERT and IMGCN models were used for classification. On other datasets, with the exception of R8, the IMGCN model achieved higher accuracy results than the other models. However, the highest accuracy value on the R8 dataset was obtained when using the STGCN + BERT + BiLSTM model. A two-layer BiLSTM model was utilized to integrate the local feature information obtained from the BERT model with the global information obtained from the GCN model [23].

However, studies using multi-class datasets are increasing day by day. As the number of classes increases, the decision boundaries that the algorithm must solve will also increase. This makes the process more difficult than a binary classification problem. In one of the studies, two different datasets were used. A set of 21,000 seven-class, balanced tweets was used for training and a set of 19,740 seven-class, balanced tweets was used for testing. In the study, three different experiments were conducted using the random forest model. In the first experiment, the seven class labels were consolidated into two classes: positive and negative. In this case, the accuracy value was 0.81. When consolidated into three classes—positive, neutral and negative—the accuracy value was 0.70. When seven class labels were used, the result was 0.60. The model success decreased with increasing numbers of class labels, making learning more difficult [24].

An experimental study was conducted on a total of 10,000 financial news reports from the Times of India, Money Control, Bloomberg and Financial Express news sources between 2017 and 2020. There were four class effects in total, and the unbalanced dataset was balanced using the SMOTE method. The Random Forest, Logistic Regression, Linear SVC, Multi-layer Perceptron and Decision Tree algorithms were used, together with TF-IDF. The highest accuracy value was obtained when using Random Forest, at 0.93. In the next experiment, DistilBERT was used instead of TF-IDF, and an accuracy value of 0.94 was obtained with Random Forest [25].

The combination of WordNet and deep learning-based models such as BERT is currently a hot research topic. In a study using WordNet and BERT, Text-GCN, Bi-LSTM, VGCN, BERT, VGCN-BERT and the proposed Mutual Graph Convolution Networks (MGCN) model were used on two-class SST-2, MR and CoLA datasets. The Mutual Graph Convolution Networks (MGCN) model introduced the semantic dictionary, which is dependent on WordNet and the BERT model. MGCN uses dependency to address context dependence issues and utilizes WordNet to gather additional semantic information. Experimental studies were carried out, and the highest success rate was obtained when using MGCN, the recommended model, which achieved a Macro F1-score of 92.31% [26]. WordNet can also be used for embedding. WN embeddings have demonstrated a reason-

able color density towards [CLS]. In this study, the WN2V-BERT model was proposed, and the highest accuracy value of 93.23% was obtained on the two-class dataset [6]. In addition to binary classification, studies have been carried out on multi-class datasets using BERT and BERT-like methods. In a related study, experiments were conducted using the LSTM, Bidirectional LSTM, BERT, BERT uncased, DistilBERT, RoBERTa, XLM-RoBERTa, GPT-2, RoBERTa CustomNet and RoBERTa ConvNet methods using four-class news data. The highest results were obtained with the use of RoBERTa, at 91.91% [27]. Studies have shown that the success rate of classification is affected by the increase in the number of classes [24]. In addition, since WordNet provides more semantic information, it has been used in research on text/document classification.

2.4. Statistical Analysis of Document Classification

There are a few statistical analysis methods available to evaluate performance. The Friedman test is one of the more popular multiple comparison methods [28]. The Friedman test was applied for statistical analysis in a study in which the Multivariate Relative Discrimination Criterion (MRDC) method was applied to perform feature selection in text classification. This test was applied to the Precision and Recall values, where the number of methods was $M = 5$ and the number of datasets was $N = 3$, and the critical value of Fisher distribution with M—1 degrees of freedom was $F(4, 8) = 3.838$ for $\alpha = 0.05$. As a result, it was seen that the FF value was greater than the critical value of 3.838 [29].

In a different study on text classification, a two-stage feature selection method was proposed involving univariate feature selection and feature clustering. The objective was to first narrow down the search space and then to choose feature sets that were relatively independent. In the study, FS-CHICLUST, SVM, KNN, Decision Tree and Naive Bayes algorithms were used, and when Friedman test was performed, the following results were obtained: Friedman chi-square = 39.9447, df = 4, p value = 4.444×10^8. According to these results, since the p value is very low, the null hypothesis that the difference in ranks is not significant is rejected, indicating that the FSCHICLUST algorithm demonstrates better performance than the other classifiers [30].

3. Methodology

In this study, experimental activities are carried out using WordNet Ontology, four different word embedding methods and four different classification algorithms. The rest of this section describes the machine learning, deep learning, and ontology framework that was structured specifically for this study.

3.1. WordNet Ontology

WordNet 3.0 is a large lexical ontology that connects over 117,000 English synonyms (synsets) through semantic relationships. It is widely used in NLP works [31], and it was created and is maintained under the direction of psychology professor George A. Miller [32].

WordNet is used in many different areas, such as automatic text classification, machine translation, word-sense disambiguation, information retrieval, automatic text summarization [33].

WordNet includes many components. Lexicographer files are the most important components of WordNet. These files are categorized as domain-based, and each of them contains the synsets for a specific syntactic category. WordNet contains 45 PoS (noun, adj, verb or adv)-based lexicographer files [34].

Figure 1 shows the structure of the WordNet lexical ontology. The figure excludes some classes and features that are not necessary for comprehending the dataset and experiments [35].

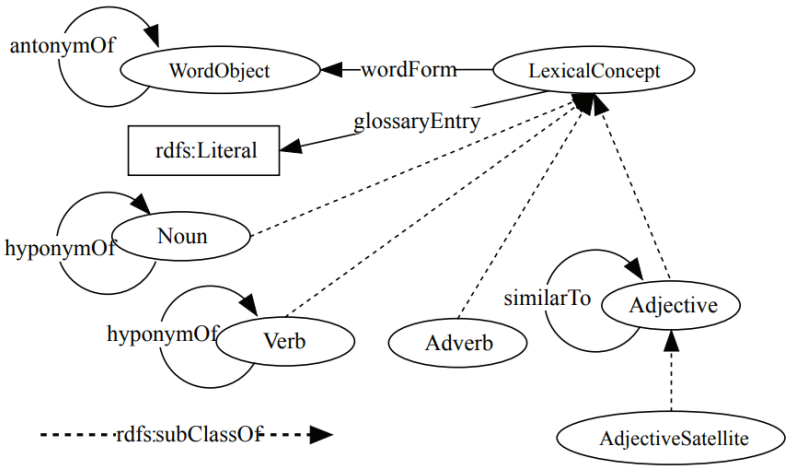

Figure 1. WordNet ontology structure [35].

3.2. Word Embedding Methods

3.2.1. Bag of Words (BoW)

BoW is a widely used technique for extracting textual attributes, where the document is represented as a vector consisting of word frequencies [36]. In this model, given a premade dictionary, a text is represented by a set of words. This notation can be binary. If a word is in the text, it scores 1, otherwise it scores 0 [37].

3.2.2. TF-IDF

The word weight calculation method is the most widely used feature in vector space models, and was first proposed in [38] in the context of the TF-IDF algorithm. It basically consists of two parts: the frequency of the words and the frequency of the reversed texts.

When creating TF-IDF vectors, first, term frequencies are calculated [39]. Different methods can be used to carry out this process. In general, the basic method is to calculate the term frequency by dividing the number of occurrences of a word in the document by the total number of words in that document. That is, the term frequency measures how often a word appears in a document [40]. After calculating the TF (term frequency), IDF (inverse document frequency) is calculated [41].

3.2.3. Word2Vec

Word2vec is a word embedding approach and was suggested by Mikolov as a method for use on Google in 2013 [42]. This model avoids nonlinear transformations and therefore makes training extremely efficient. It allows embedded word vectors to be learned from the millions of words in this dictionary, as well as from very large datasets with billions of words [43].

The Word2vec method has two different learning models. These are CBOW (continuous bag-of-words model) and skip-gram (continuous skip-gram model) [44]. The architectures of these models are illustrated in Figure 2.

In CBOW, the model is trained to predict the target word based on the context provided by the surrounding words. The target word wi is predicted by taking the input words wi − 2, wi − 1, wi + 1, wi + 2.

In the field of NLP, skip-gram is commonly used to generate word representations that can be used to predict the surrounding words in a sentence or document. According to Mikolov, although skip-gram is slower for infrequent words, it works well with small amounts of training data [44].

In the CBOW method, words outside the center of the window size parameter are taken as input. In the skip-gram method, the word in the middle of the window size parameter is taken as input [42].

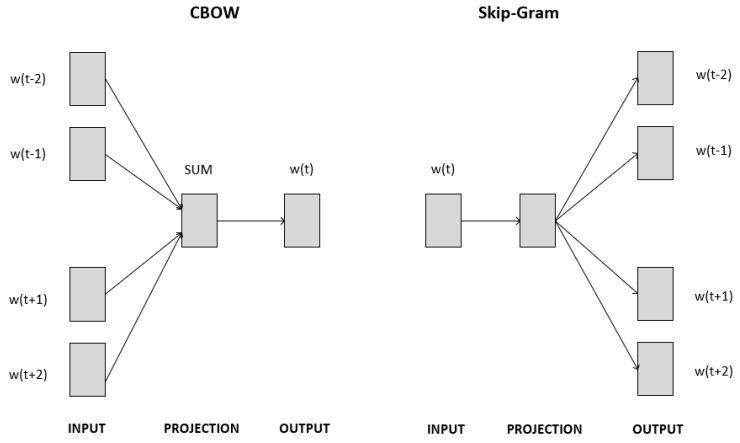

Figure 2. Architecture of Word2Vec models: CBOW and skip-gram.

3.2.4. Doc2Vec

Doc2Vec generates a vector representation of a document to predict the target word [37]. In doing so, document length is not considered. Doc2Vec can be divided into the Distributed Memory Model of Paragraph Vectors (PV-DM), and the Distributed Bag of Words version of Paragraph Vector (PV-DBOW). The architectures of these models are illustrated in Figure 3.

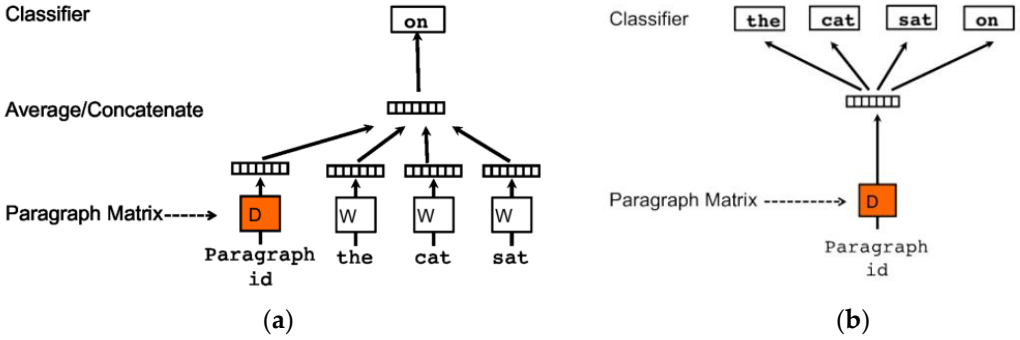

Figure 3. Architecture of Doc2Vec [44]. (**a**) PV-DM architecture; (**b**) PV-DBOW architecture.

In the PV-DM method, a single word is considered from each paragraph, and each paragraph has a unique identity. PV-DBOW uses a paragraph vector to classify the words in the document.

While both the PV-DM and PV-DBOW methods are used for generating distributed representations of words, there are some differences between them. PV-DM, as illustrated in Figure 3a,b, predicts four words from a single input. Another key difference is that PV-DBOW tends to require less storage space, as it only stores SoftMax weights, whereas PV-DM requires more data storage. The third difference is that the PV-DM target pulls the words around it, while PV-DBOW extracts one word from the entire paragraph [44].

These word embedding methods are summarized in terms of their advantages and disadvantages in Table 1, below.

Table 1. Advantages and disadvantages of word embedding methods.

Word Embedding Method	Advantages	Disadvantages
BoW	Characterized by ease of implementation. It does not require extensive training data. It can be used to create an initial draft model before proceeding to more sophisticated word embeddings.	It does have a few limitations and drawbacks. As the size of the vocabulary increases, the size of the BoW vector representation grows accordingly.

Table 1. *Cont.*

Word Embedding Method	Advantages	Disadvantages
TF-IDF	The algorithm is simple to use, as it is computationally efficient, cost effective to run, and provides a clear basis for similarity calculations.	TF-IDF cannot assist in carrying semantic meaning. TF-IDF disregards word order.
Word2Vec	It is a computationally efficient method for generating word embeddings. It produces embeddings that capture semantic relationships between words. It employs dimensionality reduction techniques to compress the high-dimensional vector space of words into a lower-dimensional space. It is flexible and can be trained on different datasets. Word2vec can be used for language modeling, which involves predicting the likelihood of a sequence of words occurring in a text corpus.	Word2vec does not handle polysemy very well, which is the phenomenon of a single word having multiple meanings. Word2vec requires a large corpus of text to train on, and it may not perform well on words that are not present in the training data. While word2vec produces vector representations of words, it can be difficult to interpret what these vectors actually represent. While word2vec is generally computationally efficient, it can still require significant computing resources, especially when training on large datasets or with complex models.
Doc2Vec	Doc2vec can generate vector representations that capture the semantic relationships between entire documents, making it useful for tasks such as document classification, clustering, and similarity searches. Unlike traditional bag-of-words models, doc2vec can handle variable-length documents, which makes it useful for processing long and complex documents. Doc2vec can incorporate the context of a document, including the surrounding documents and other relevant information, into the vector representation of the document. Doc2vec is computationally efficient and can be used on large datasets.	Doc2vec is a more complex model than traditional bag-of-words models, and can be more difficult to implement and interpret. Doc2vec can require significant computing resources, especially when trained on large datasets or with complex models. The quality of the document embeddings produced by doc2vec depends on the quality and size of the training data. While doc2vec produces vector representations of documents, it can be difficult to interpret what these vectors actually represent. This can make it challenging to understand why certain documents are more similar to each other than others.

3.3. Classification Methods

3.3.1. Random Forest (RF)

The RF Algorithm, which was first presented in the literature in [45], performs classification by training each decision tree on a different observation sample and producing various models.

These are the steps involved in the Random Forest (RF) algorithm:

1. Random samples are selected from the input dataset.
2. The algorithm constructs a decision tree that will yield the prediction result for each selected sample.
3. This model is used in the classification problem for each of the predicted outcomes.
4. The prediction that receives the most votes is the final result.

The definition of the k trained decision trees in the Random Forest model is given in Equation (1), below.

$$H(X, \theta_j) = \sum_{i=0}^{k} h_i(x, \theta_j), \qquad (j = 1, 2, 3, \ldots, m) \tag{1}$$

The general architecture of the RF, depicting this situation, is illustrated in Figure 4.

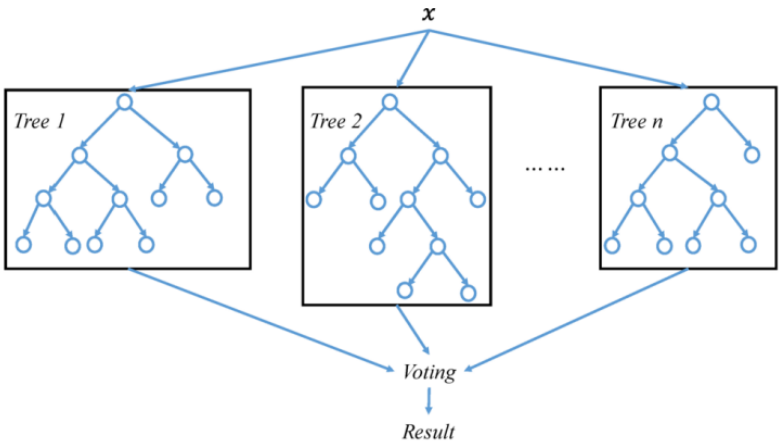

Figure 4. General architecture of RF [46].

3.3.2. Support Vector Machine (SVM)

The SVM was invented in 1960 [47]. Following its invention, it was presented in 1998 by Vapnik [48]. Initally, SVM was presented as a form of initial binary classification [49]. Then, two types appeared: linear and multi-class SVM [50].

SVM is used to classify data on the basis of two distinct categories and to find the best hyperplane [51]. The primary objective is to find a hyperplane that maximizes the distance with the nearest data points [52,53]. To maximize the minimum distance between the hyperplane and the training data, an algorithmically selected hyperplane is used. The minimum distance is usually referred to as the margin [54]. The structure of the SVM is shown in Figure 5.

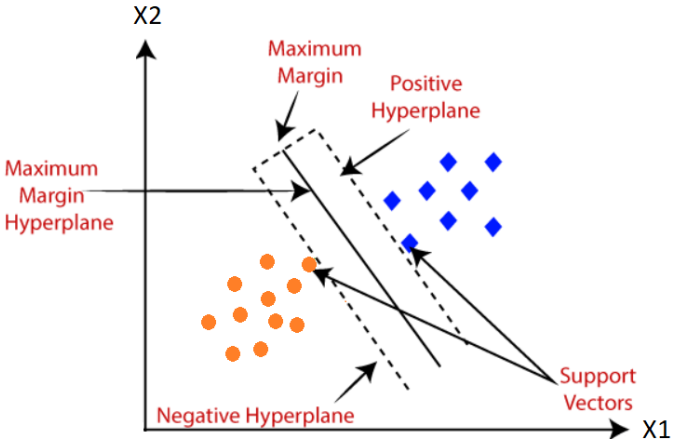

Figure 5. SVM structure.

The hyperplane formed by SVM can be formulated according to Equation (2) [12]:

$$f(x) = w^T \cdot x + b \tag{2}$$

The optimization problem of SVM can be summarized as shown in Equation (3):

$$\min \frac{1}{2} ||x||^2 \tag{3}$$

subject to Equation (4):

$$y_i(wTxi + b) \geq 1, \ \forall i_1 = 1, \ 2, \ldots, \ N \tag{4}$$

3.3.3. Multi-Layer Perceptron (MLP)

MLP is one of the most significant classes of Artificial Neural Network. It is a powerful modeling method that performs supervised learning using labeled data samples. This method creates a nonlinear function model that allows the estimation of outputs from given inputs [55].

In MLP, each of the components of the network determines a bias-weighted sum of its inputs and passes the calculated values through a transfer function to produce their output, and the units are designed in a layered feed-forward topology. By modeling functions with a number of hidden layers and neurons in each layer, MLP networks can determine the complexity of a function. The number of hidden layers to use depends on the problem and the data type used for the models [56].

The layers of the MLP, the architectural structure of which is shown in Figure 6, are as follows [57]:

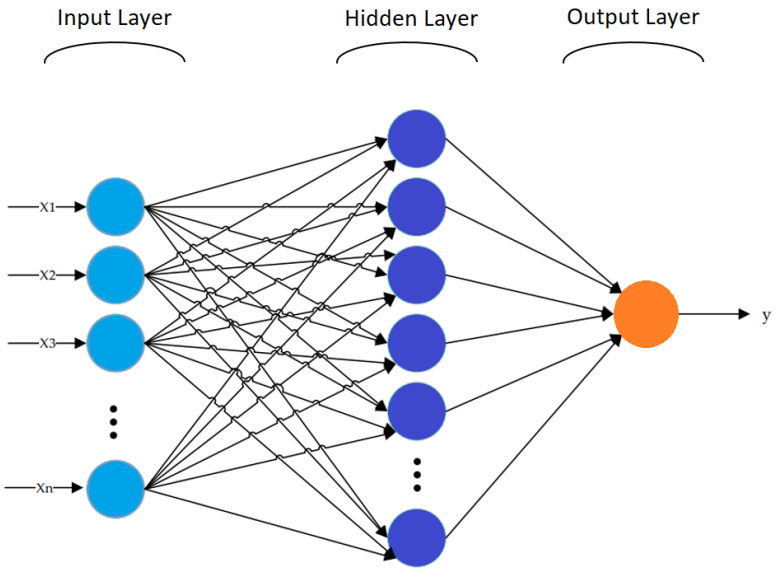

Figure 6. Architecture of MLP.

3.3.4. Bidirectional Encoder Representations from Transformers (BERT)

BERT, which has recently entered the literature, is a Transformer (deep learning model)-based machine learning model for NLP pretraining that was propounded in October 2018 by Google [58].

BERT uses the Transformer [59] architecture to learn word representations. Transformer is a new architecture for array modeling that can be incorporated into deep networks. Using only attention mechanisms, the Transformer learns about global dependencies between input and output [60].

BERT is used in two ways, one of which is the Masked Language Prediction approach. In this method, a few words of the input text are masked and then input into the BERT model, which is then used to predict the masked words. To predict a masked word, the BERT model considers the context of the unmasked words that precede and follow it.

BERT can be applied in two phases: pretraining and fine-tuning. During pretraining, the model learns to recognize the input text data and its contextual relationships. In the fine-tuning phase, the model adapts to a specific task and refines its understanding to provide the best solution. By adding a new layer, a pretrained BERT model can be fine-tuned to achieve exemplary results [61]. BERT uses the same architecture in the pretraining and fine-tuning stages, and both stages are shown in Figure 7.

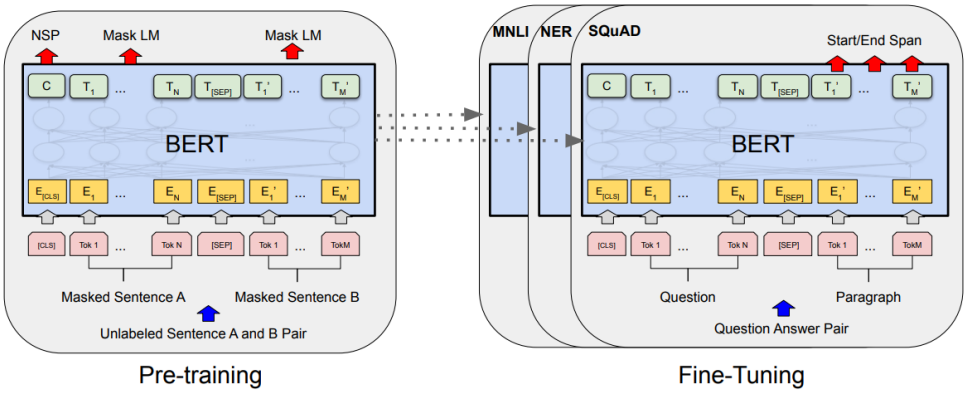

Figure 7. BERT architecture of pretraining and fine-tuning tasks [55].

The BERT model consists of three parts: token placement, partition placement, and position placement [62]. In the model, sentences always start with [CLS] and end with [SEP]. The overall input text sequence is classified according to the output embedding associated with the [CLS] token.

BERT has two different variants: Base and Large. BERT-Base consists of 12 layers, 768 hidden dimensions, and 12 attention heads, with 110M total parameters. On the other hand, BERT-Large consists of 24 layers, 16 attention heads, 1024 hidden dimensions, and 340M total parameters [55]. The Base and Large variants also have two different versions each: cased and uncased. In the uncased version, text is converted to lowercase before the word tokenization process. Conversely, the cased version is case sensitive [63].

These classification methods are summarized in terms of their advantages and disadvantages in Table 2 below.

Table 2. Advantages and disadvantages of classification methods.

Classification Method	Advantage	Disadvantage
RF	It helps to enhance accuracy by mitigating overfitting in decision trees. It is versatile and can be used for both classification and regression problems. It is effective for working with both categorical and continuous values. It can handle missing values in the data without the need for imputation or pre-processing.	It requires a substantial amount of computational power and resources. It can take a long time to train, as it combines a large number of decision trees to determine the class. Due to the use of an ensemble of decision trees, RF also faces challenges with interpretability and struggles to determine the significance of each variable.
SVM	SVM is particularly effective when there is a clear separation between classes. In high-dimensional spaces, SVM tends to be more effective. SVM is effective in scenarios where the number of dimensions exceeds the number of samples. It is known for its relatively efficient memory usage.	The SVM algorithm is not suitable for processing large datasets due to its higher computational complexity and memory requirements. SVM does not perform well when the dataset contains more noise. SVM does not perform well if the number of features per data point exceeds the number of training samples.
MLP	It is capable of addressing complex nonlinear problems. It can effectively handle large amounts of input data. After the training process, MLP can make quick predictions. MLP can achieve comparable levels of accuracy even with smaller sample sizes.	MLP includes too many parameters because it is fully connected.
BERT	BERT is a technology for generating "contextualized" word embeddings/vectors, which is its biggest advantage.	It is very computationally intensive at inference time, meaning that if you want to use it in production at scale, it can become costly.

4. Experimental Design

This section provides an overview of the datasets used, outlines the data preprocessing steps, and details the dimension reduction approach employed in our study.

4.1. Application Design

In the context of this study, all experiments were carried out on a computer with a 2.60 GHz six-core Intel core i7 processor and 16 GB memory running the Windows 10 Enterprise operating system. Python 3.8.2 was used as the programming language and Visual Studio Code was used as the IDE. As Python libraries, Sklearn for TF-IDF, BoW, MLP, SVM, RF; Gensim for Word2Vec and Doc2vec; and Transformers for BERT were used.

A console application was developed in order to be able to perform the experiments quickly and easily. Depending on the criteria selected on the application menu, data preprocessing, WordNet, word embedding and classification methods can be run.

4.2. Dataset

Studies in document classification have been carried out using many different datasets. For example, a minimum of two class labels are generally used in sentiment analysis studies. Apart from this, studies have been performed using datasets such as Twitter, complaints, comments, spam and news. In the context of this study, a dataset with a small amount of imbalanced with seven class labels was selected, and the success of machine learning and deep learning models was evaluated. In the experiments, 4817 lines of news content data from the English news website Inshorts, which we downloaded from Kaggle, were used. In the dataset, there was news belonging to seven different categories (class labels): "Automobile", "Entertainment", "Politics", "Science", "Sports", "Technology", and "World" [64]. Detailed information about the dataset used in the experimental studies is shown in Table 3.

Table 3. Dataset details.

Class Number	Category (Class Label) Name	Total Number of Data	Number of Data in Training Set	Number of Data in Test Set
1	Automobile	256	190	66
2	Entertainment	998	697	301
3	Politics	546	380	166
4	Science	389	277	112
5	Sports	856	594	262
6	Technology	751	514	237
7	World	1021	719	302

4.3. Data Preprocessing

In the data preprocessing stage, the following steps were carried out, in turn:

- Punctuation marks were removed.
- HTML tags were removed.
- Numeric expressions were removed.
- All words were converted to lowercase.
- Stop words were removed.
- Word spellings were corrected.
- Lemmatization was applied.
- Stop words formed after lemmatization were removed.

In the preprocessing stage, Python SymSpell was used to perform word correction, and the NLTK library was used for the removal of stop words and lemmatization.

At the end of this step, we decomposed the dataset into two, comprising 70% training and 30% testing. Details of the training set and testing set data after decomposition are shown in Table 3.

4.4. Ontology-Based Feature Dimension Reduction

One of the most important and distinctive stages of this study is the feature dimension reduction stage. In this step, the size of the vector space is narrowed using the NLTK WordNet library.

WordNet has forty-five lexicographer files based on synthesis sets, syntactic categories, and logical groupings. Each lexicographer file can be described using a file number. Additionally, to indicate a lexicographer filename in an effective way, file numbers are encoded in multiple parts of the WordNet ontology. File word names, which provide the connection between file names and numbers, are used by end users or programs to interact with them.

There are 45 lexicographer files in total, and within the scope of this study, 11 general lexicographer files suitable for the dataset were selected and used. Lexicographer files which were not associated with the dataset were ignored. The names of the lexicographer files used in the study and their corresponding file numbers, along with a brief description of the contents of each file, are shown in Table 4.

Table 4. Lexicographer file names and their file numbers.

File Number	Name	Contents
13	noun.food	nouns denoting foods and drinks
15	noun.location	nouns denoting spatial position
18	noun.person	nouns denoting people
21	noun.possession	nouns denoting possession and transfer of possession
23	noun.quantity	nouns denoting quantities and units of measure
25	noun.shape	nouns denoting two- and three-dimensional shapes
28	noun.time	nouns denoting time and temporal relations
37	verb.emotion	verbs of feeling
40	verb.possession	verbs of buying, selling, owning
41	verb.social	verbs of political and social activities and events
43	verb.weather	verbs of raining, snowing, thawing, thundering

Table 5 shows the original state of the three sample rows in the dataset, after the data preprocessing step, and after applying the WordNet lexical ontology. Words marked in bold indicate changes with respect to the WordNet lexicographer file.

Table 5. Change in data after applying data preprocessing and WordNet.

Original Data	After Preprocessing	After WordNet
Iranian authorities on Saturday executed journalist Ruhollah Zam over his online work that helped inspire nationwide economic protests in 2017. A court had sentenced Zam to death in June after he was found guilty of “corruption on earth”, one of the country\’s most serious offences. Zam had been living in exile in France but was arrested in October last year.	iranian authority saturday executed journalist roll online work helped inspire nationwide economic protest court sentenced death june found guilty corruption earth one country serious offence living exile france arrested october last year	**person** authority **time social person** roll online work **social emotion** nationwide economic protest court sentenced death **time possession** guilty corruption earth quantity country serious offence living **person location** arrested **time time time**
Tokyo Stock Exchange (TSE) President and CEO Koichiro Miyahara will step down to accept responsibility over a system failure last month that resulted in the first all-day stoppage of trading since the exchange switched to all-electronic trading in 1999. Akira Kiyota, the Group CEO of Japan Exchange Group that runs the TSE, will temporarily take over Miyahara’s role.	tokyo stock exchange president co cairo micah ara step accept responsibility system failure last month resulted first day stoppage trading since exchange switched electronic trading akita toyota group co japan exchange group run temporarily take micah ara role	**location possession** exchange **person** co **location person** ara step accept responsibility system failure **time time** resulted first **time** stoppage trading since exchange switched electronic trading akita toyota group co **location** exchange group run temporarily **possession person** ara role

Table 5. *Cont.*

Original Data	After Preprocessing	After WordNet
Mick Schumacher, son of seven-time world champion Michael Schumacher, will be racing for Haas in the next Formula One season. The 21-year-old German signed a multi-year agreement and will partner Russian Nikita Mazepin. "The prospect of being on the Formula One grid next year makes me incredibly happy . . . I\'m simply speechless", said Mick. He is currently leading the Formula Two championship.	mick schumacher son seven time world champion michael schumacher racing haas next formula one season year old german signed multi year agreement partner russian nikita maze prospect formula one grid next year make incredibly happy simply speechless said mick currently leading formula two championship	**person** schumacher **person** **quantity time** world champion **person** schumacher racing haas next formula **quantity time time** **time person** signed multi **time** agreement **person person** nikita maze prospect formula **quantity** grid next **time** make incredibly happy simply speechless said person currently leading formula **quantity** championship

5. Experiment and Results

5.1. Proposed Model

After the data preprocessing phase, the subsequent experiments are divided into two parts: with the use of WordNet and without the use of WordNet. Afterwards, word embedding methods were used both with WordNet and directly with the RF, SVM and MLP classification algorithms. Since there is no need to use word embedding methods in the BERT algorithm, separate experiments were conducted both with and without WordNet. The purpose of using WordNet in the study was to investigate its contribution to classification success by reducing the feature vector space. The system architecture for the studies carried out is illustrated in Figure 8. This proposed architecture is composed of five main parts: dataset, data preprocessing, feature reduction with WordNet, word embedding, and classification, respectively.

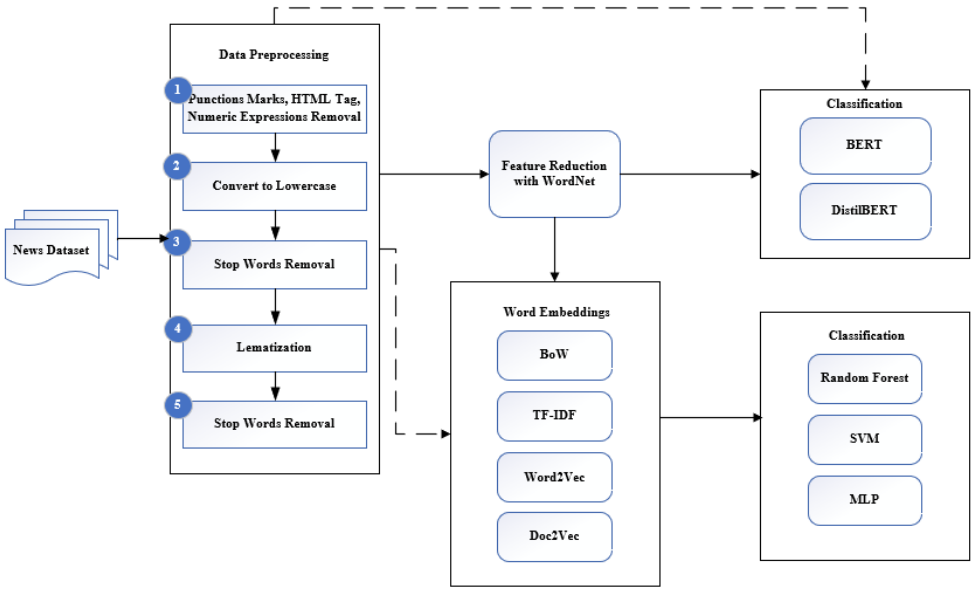

Figure 8. Architecture of the proposed system.

5.2. Machine Learning Classifiers

In the experiments, different parameter values were used for each of the word embedding methods. The optimal parameters of the BoW, TF-IDF, Word2Vec, Doc2Vec, SVM, and MLP algorithms, which were determined experimentally, are shown in Table 6.

Table 6. Optimal parameters adapted for BoW, TF-IDF, Word2Vec, Doc2Vec, SVM, MLP.

(a) Parameters for BoW	
Parameter	**Parameter Value**
Max Features	500
Min df	5
Max df	0.7
(b) Parameters for TF-IDF	
Parameter	**Parameter Value**
Max Features	1000
Min df	5
Max df	0.7
(c) Parameters for Word2Vec	
Parameter	**Parameter Value**
Training Algorithm	skip-gram
Window	5
Min Count	5
Size	200
Workers	100
Epoch	100
(d) Parameters for Doc2Vec	
Parameter	**Parameter Value**
Training Algorithm	PV-DM
Vector Size	200
Window	8
Workers	100
Epoch	25
(e) Parameters for RF	
Parameter	**Parameter Value**
Random State	0
(f) Parameters for SVM	
Parameter	**Parameter Value**
Max Iter	15,000
Kernel	Linear
Gamma	Auto
(g) Parameters for MLP	
Parameter	**Parameter Value**
Solver	lbfgs
Max Iter	50
Hidden Layer Sizes	50, 50, 50
Activation	Relu

In each of the machine learning-based classification algorithms, the BoW method was used first, and experiments were carried out with the optimal parameters, as shown in Table 6a, in order to compare the success of the models under the same conditions. The experiments were then continued using the optimal parameters in Tables 6b, 6c and 6d, respectively. We conducted experiments with different parameter settings, depending on the model. In RF, we used the default parameters. In SVM, we tested kernel functions such as linear, polynomial, RBF, and sigmoid, a maximum number of iterations ranging from 1000 to 15,000, and gamma set to 'auto'. Finally, in MLP, we tested activation functions such as relu and sigmoid, a maximum number of iterations ranging from 50 to 100, and hidden

layer sizes ranging from 30 to 50. After analyzing the experimental results, we determined the optimal parameters for each model, and these are presented in Table 6e–g.

The results of the testing performance of different ML classifiers on our dataset are shown in Tables 7–9.

Table 7. Macro averaged scores for RF classification.

Method	Precision	Recall	F1-Score	Accuracy
BoW + RF	91.29%	90.53%	90.87%	91.35%
TF-IDF + RF	90.66%	90.33%	90.43%	91.35%
Word2Vec + RF	90.57%	88.34%	89.32%	91.14%
Doc2Vec + RF	92.67%	91.41%	91.96%	92.39%
WordNet + BoW + RF	90.40%	89.78%	90.04%	90.73%
WordNet + TF-IDF + RF	92.19%	91.70%	91.94%	91.90%
WordNet + Word2Vec + RF	92.60%	90.27%	91.33%	91.77%
WordNet + Doc2Vec + RF	92.55%	92.79%	92.62%	93.01%

Table 8. Macro averaged scores for SVM classification.

Method	Precision	Recall	F1-Score	Accuracy
BoW + SVM	89.36%	90.42%	89.70%	90.45%
TF-IDF + SVM	89.95%	90.09%	89.91%	91.00%
Word2Vec + SVM	86.96%	87.36%	87.07%	88.03%
Doc2Vec + SVM	91.46%	91.52%	91.41%	92.18%
WordNet + BoW + SVM	91.76%	91.10%	91.38%	91.90%
WordNet + TF-IDF + SVM	90.04%	90.94%	90.37%	90.66%
WordNet + Word2Vec + SVM	87.20%	86.39%	86.68%	87.89%
WordNet + Doc2Vec + SVM	91.70%	92.05%	91.85%	92.25%

Table 9. Macro averaged scores for MLP classification.

Method	Precision	Recall	F1-Score	Accuracy
BoW + MLP	89.54%	89.37%	89.43%	90.38%
TF-IDF + MLP	83.72%	82.64%	82.87%	84.99%
Word2Vec + MLP	75.91%	73.89%	74.63%	81.32%
Doc2Vec + MLP	90.69%	89.48%	90.00%	91.07%
WordNet + BoW + MLP	90.35%	90.78%	90.51%	91.07%
WordNet + TF-IDF + MLP	87.60%	87.30%	8.41%	89.14%
WordNet + Word2Vec + MLP	79.41%	75.75%	77.03%	81.88%
WordNet + Doc2Vec + MLP	91.84%	90.73%	91.21%	91.63%

Table 7 shows the results of the classification study obtained when using the RF algorithm. According to the experiments conducted in this category, the highest success was obtained when using WordNet ontology and Doc2Vec together, with an accuracy of 93.01%. This success was achieved with Doc2Vec using N = 200 as vector size, W = 8 as the window size, 100 workers, and 25 epochs. The results of the experiments using the SVM algorithm are shown in Table 8. In the experiments, the highest success value, an accuracy of 92.25%, was obtained when using WordNet and Doc2Vec methods together. The parameters used for these methods are given in Tables 6a, 6d and 6f. Finally, in the experiments conducted in the MLP category shown in Table 8, the highest accuracy value of 91.63% was obtained when using the WordNet and Doc2Vec models together, as shown in Table 9. The parameters of the methods with high results in this category are shown in Tables 6d and 6g.

Figure 9 presents the ROC curves for the methods with the highest success among the machine learning classification models. The area values of each class label are shown separately in the ROC curves.

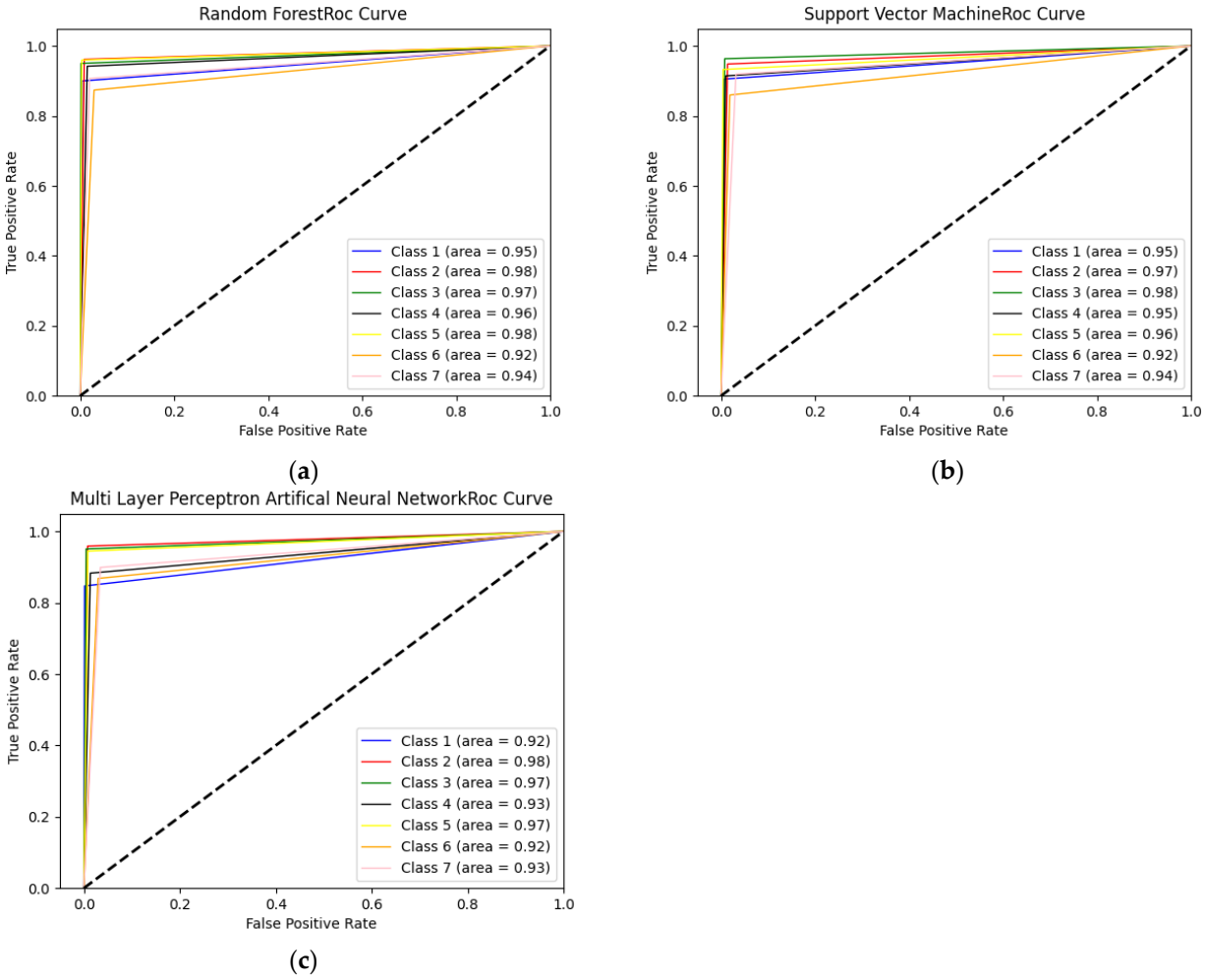

Figure 9. ROC curves for the methods with the highest success among the machine learning classification models. (**a**) WordNet+Doc2Vec+RF Roc Curve; (**b**) WordNet + Doc2Vec + SVM Roc Curve; (**c**) WordNet + Doc2Vec + MLP Roc Curve.

As a result, it was seen in the experiments conducted in the RF, SVM and MLP categories that the Doc2Vec model was effective at increasing the classification success, as well as WordNet. This is because of the semantic preservation of documents achieved by taking the sum and average of all word vectors in the Doc2Vec method. Another finding is that while high success was achieved with low numbers of epochs for the Doc2Vec model, the opposite was true for the Word2Vec model.

5.3. Deep Learning Classifier

The pretrained BERT and DistilBERT were used as deep learning-based classification methods, and experiments were carried out using BERT or DistilBERT alone, as well as with the combined use of BERT + WordNet and DistilBERT + WordNet. The tested parameters included a learning rate ranging from 1e5 to 4e5, a batch size ranging from 4 to 16, a max length ranging from 128 to 512, and 1, 3, 5, 7, and 10 training epochs. After analyzing the experimental results, we determined the optimal parameters of BERT and DistilBERT, as shown in Table 10.

It was reported in previous studies in which experiments were carried out using the cased and uncased versions of BERT that the uncased version was more successful [65–68]. Therefore, in this study, we selected the BERT-Base uncased pretrained model for fine-tuning. Experiments were performed using the transformers library. We optimized the BERT-Base-Uncased model using the Adam optimizer, and the best accuracy values were obtained with

three and five epochs. In the study, experiments were carried out using different combinations of parameter values, and other optimal parameter values were as shown in Table 10. Batch size was fixed as 4 in both the training and validation set. In the experiments using epoch values of 1, 3, 5, 7 and 10, the highest success value was obtained with three and five epochs, and it was observed that the success decreased when seven epochs were used. In addition, experiments using the same parameters were performed on a smaller, faster and lighter version of BERT, known as DistilBERT.

Table 10. Optimal parameters adapted for BERT and DistilBERT.

Parameters	Parameters Value
Optimizer	Adam
Learning Rate	1e5
Epsilon	1e8
Max Length	256
Batch Size	4
Epochs	3–5

In Table 11, the precision, recall, F1-score and accuracy values of the experimental studies using BERT and DistilBERT are presented. In the experiments conducted using models from the deep learning category, the highest success was achieved with the hybrid use of WordNet and BERT, where an accuracy of 93.77% was obtained. This value represents the highest accuracy result obtained in this study. In order to achieve the highest value possible, fine-tuning was performed, and the parameters in Table 10 were used. In the experiment, batch size was fixed as 4 for both the training and validation set. Additionally, among the epoch values of 1, 3, 5, 7 and 10 used, the highest success values were obtained when using three and five epochs, and it was observed that the success decreased when seven epochs were used. Because it is a powerful model, BERT achieved the highest success rate [55]. BERT uses masked language models to enable deep bidirectional representations to be pretrained. The masked language model randomly masks some tokens from the input, predicting the original of the masked word based only on its context. BERT is very strong for developing an understanding of context-heavy texts. The dataset we used consisted of English long-form news content. Therefore, it was important to analyze it from a semantic and contextual point of view. By using BERT and WordNet together to do this, deeper learning was attained and high success was achieved.

Table 11. Macro averaged scores for BERT and DistilBERT classification. The best score from our study is indicated in bold.

Method	Precision	Recall	F1-Score	Accuracy
BERT	92.49%	91.58%	91.94%	92.32%
WordNet + BERT	**94.31**%	**92.99**%	**93.60**%	**93.77**%
DistilBERT	90.51%	92.57%	91.34%	91.6%
WordNet+DistilBERT	92.71%	92.47%	92.56%	92.5%

In a study using the BERT model, using the pretrained BERT model directly in the classification task did not result in a statistically significant improvement in performance. The importance of using hyperparameters was emphasized, rather than statistical significance [69]. Similarly, in this study, fine-tuning was performed with different parameters after the pretraining stage.

Figure 10 presents the ROC curves obtained when using BERT and WordNet together and when using DistilBERT and WordNet together. The area values of each class label are shown separately in the ROC curves. In BERT, which is a deep learning-based classifier, the highest accuracy value was obtained with the hybrid use of WordNet and BERT. It can be seen from Figure 10a, which shows this method, that the class label field values are closer to each other than in Figure 10b.

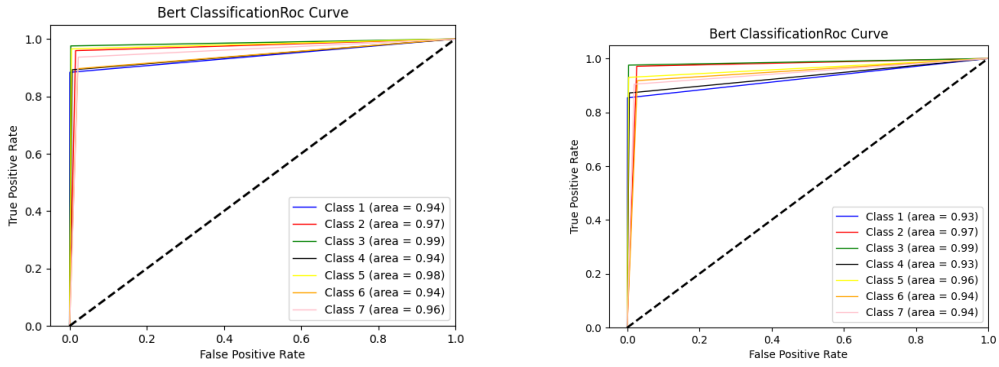

Figure 10. ROC curves formed when using BERT alone and when using WordNet and BERT together. (**a**) BERT+WordNet ROC Curve; (**b**) DistilBERT+WordNet ROC Curve.

6. Conclusions and Future Work

In this paper, we conducted a series of experiments to evaluate the effect of lexical ontology and classification models with the aim of increasing classification success on multi-class and imbalanced datasets.

In order to research the effect of lexical ontology on the success of classification models, we conducted experiments using two different categories of model: machine learning models and deep learning models. In the machine learning category, RF, SVM and MLP models were trained together with the BoW, TF-IDF, Word2Vec and Doc2Vec word embedding methods. Afterwards, these models were retrained with the same word embedding methods using WordNet. The highest success in this category was achieved when using the RF model together with WordNet and Doc2Vec, with an accuracy of 93.01%. Experiments were then performed using BERT, which belongs to the deep learning category. Here, our dataset was first used to train BERT and then it was retrained with WordNet and the experiments were performed. The highest success rate in these experiments, an accuracy of 93.77%, was obtained when using WordNet and the pretrained BERT as a hybrid model. WordNet was used for feature dimension reduction, and lexicography files were used to group words with the same meaning and repeating words, and we showed that using a lexical ontology for dimension reduction increased the classification success rate. It was observed that feature dimension reduction using WordNet increased the success rate on the seven-class imbalanced dataset for both traditional machine learning classification models and the deep-learning-based BERT and DistilBERT classification models. However, BERT was more successful than the other classical methods, because it provides deeper learning. Additionally, DistilBERT, which is a derivative of BERT, provided good results when used in conjuction with WordNet. It also worked faster than BERT. We believe that this study can motivate researchers to conduct document classification research using lexical ontology, and our model can be applied in a variety of text classification tasks, especially in cases where unstructured data are present and there are multiple classes to classify.

In future works, we aim to extend these experiments using other BERT models such as ALBERT, RoBERTa, XLNet, etc., with WordNet and various multi-class imbalanced datasets. We will also conduct experiments to address class imbalance in multi-class datasets using various methods. Furthermore, since the Graph Convolutional Network (GCN) has recently achieved successful results in text classification, it will be used together with and independently of BERT in order to measure its success on multi-class imbalanced datasets.

Author Contributions: Conceptualization, I.Y. and A.G.; methodology, I.Y., A.G. and M.Z.; software, I.Y.; validation, I.Y., A.G. and M.Z.; formal analysis, I.Y. and A.G.; investigation, I.Y. and M.Z.; resources, I.Y.; data curation, I.Y.; writing—original draft preparation, I.Y., A.G. and M.Z.; writing—review and editing, I.Y., A.G. and M.Z.; visualization, I.Y.; supervision, A.G.; project administration, A.G.; funding acquisition, I.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by TUBITAK (2011-C Priority Areas Graduate Programme Scholarship).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Available on request.

Acknowledgments: We wish to thank TUBITAK (2011-C Priority Areas Graduate Programme Scholarship) and Turkcell Group Company Digital Educational Technologies Inc. for financial support.

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

References

1. Kadhim, A.I. Survey on supervised machine learning techniques for automatic text classification. *Artif. Intell. Rev.* **2019**, *52*, 273–292. [CrossRef]
2. Kumbhar, P.; Mali, M.A. Survey on Feature Selection Techniques and Classification Algorithms for Efficient Text Classification. *Int. J. Sci. Res.* **2016**, *5*, 1267–1275.
3. Mwadulo, M.W. A Review on Feature Selection Methods for Classification Tasks. *Int. J. Comput. Appl. Technol. Res.* **2016**, *5*, 395–402.
4. Zhang, T.; Yang, B. Big data dimension reduction using PCA. In Proceedings of the 2016 IEEE International Conference on Smart Cloud (SmartCloud), New York, NY, USA, 18–20 November 2016; pp. 152–157. [CrossRef]
5. Lu, Z.; Du, P.; Nie, J.Y. VGCN-BERT: Augmenting BERT with graph embedding for text classification. In *Advances in Information Retrieval, Proceedings of the 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, 14–17 April 2020*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 369–382. [CrossRef]
6. Barbouch, M.; Verberne, S.; Verhoef, T. WN-BERT: Integrating WordNet and BERT for Lexical Semantics in Natural Language Understanding. *Comput. Linguist. Neth. J.* **2021**, *11*, 105–124.
7. Kowsari, K.; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D. Text classification algorithms: A survey. *Information* **2019**, *10*, 150. [CrossRef]
8. Stein, R.A.; Jaques, P.A.; Valiati, J.F. An analysis of hierarchical text classification using word embeddings. *Inf. Sci.* **2019**, *471*, 216–232. [CrossRef]
9. Sen, P.C.; Hajra, M.; Ghosh, M. Supervised classification algorithms in machine learning: A survey and review. In *Emerging Technology in Modelling and Graphics, Proceedings of the IEM Graph 2018, Kolkata, India, 6–7 September 2018*; Springer: Singapore, 2020; pp. 99–111.
10. Han, Q.; Snaidauf, D. Comparison of Deep Learning Technologies in Legal Document Classification. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021; pp. 2701–2704.
11. Kosar, A.; De Pauw, G.; Daelemans, W. Unsupervised Text Classification with Neural Word Embeddings. *Comput. Linguist. Neth. J.* **2022**, *12*, 165–181.
12. Han, H.; Giles, C.L.; Manavoglu, E.; Zha, H.; Zhang, Z.; Fox, E.A. Automatic document metadata extraction using support vector machines. In Proceedings of the 2003 IEEE Joint Conference on Digital Libraries, Houston, TX, USA, 27–31 May 2003; pp. 37–48.
13. Biagioli, C.; Francesconi, E.; Passerini, A.; Montemagni, S.; Soria, C. Automatic semantics extraction in law documents. In Proceedings of the 10th International Conference on Artificial Intelligence and Law, Paris, France, 6–8 July 2005; pp. 133–140.
14. Maynard, D.; Yankova, M.; Kourakis, A.; Kokossis, A. Ontology-based information extraction for market monitoring and technology watch. In Proceedings of the ESWC Workshop End User Apects of the Semantic Web, Heraklion, Greece, 6–10 June 2005.
15. Mohemad, R.; Hamdan, A.R.; Othman, Z.A.; Mohamad Noor, N.M. Ontological-based information extraction of construction tender documents. In *Advances in Intelligent Web Mastering–3, Proceedings of the 7th Atlantic Web Intelligence Conference, AWIC 2011, Fribourg, Switzerland, 26–28 January 2011*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 153–162.
16. Bloehdorn, S.; Basili, R.; Cammisa, M.; Moschitti, A. Semantic kernels for text classification based on topological measures of feature similarity. In Proceedings of the Sixth IEEE International Conference on Data Mining (ICDM'06), Hong Kong, China, 18–22 December 2006; pp. 808–812.
17. Cristianini, N.; Shawe-Taylor, J.; Lodhi, H. Latent semantic kernels. *J. Intell. Inf. Syst.* **2002**, *18*, 127–152. [CrossRef]
18. Dhyaram, L.P.; Vishnuvardhan, B. Random subset feature selection for classification. *Int. J. Adv. Res. Comput. Sci* **2018**, *9*, 317–319. [CrossRef]

19. Bamatraf, S.A.; Bin-Thalab, R.A. Semantic Classification Model for Twitter Dataset Using WordNet. *Int. Res. J. Innov. Eng. Technol.* **2021**, *5*, 5.
20. Gawade, M.; Mane, T.; Ghone, D.; Khade, P.; Ranjan, N. Text Document Classification by using WordNet Ontology and Neural Network. *Int. J. Comput. Appl.* **2018**, *182*, 33–36. [CrossRef]
21. Elhadad, M.K.; Badran, K.M.; Salama, G.I. A novel approach for ontology-based dimensionality reduction for web text document classification. *Int. J. Softw. Innov.* **2017**, *5*, 44–58. [CrossRef]
22. Demirsoz, O.; Ozcan, R. Classification of news-related tweets. *J. Inf. Sci.* **2017**, *43*, 509–524. [CrossRef]
23. Xue, B.; Zhu, C.; Wang, X.; Zhu, W. The Study on the Text Classification Based on Graph Convolutional Network and BiLSTM. In Proceedings of the 8th International Conference on Computing and Artificial Intelligence, Tianjin, China, 18–21 March 2022; pp. 323–331. [CrossRef]
24. Bouazizi, M.; Ohtsuki, T. A pattern-based approach for multi-class sentiment analysis in Twitter. *IEEE Access* **2017**, *5*, 20617–20639. [CrossRef]
25. Dogra, V.; Alharithi, F.S.; Álvarez, R.M.; Singh, A.; Qahtani, A.M. NLP-Based Application for Analyzing Private and Public Banks Stocks Reaction to News Events in the Indian Stock Exchange. *Systems* **2022**, *10*, 233. [CrossRef]
26. Xue, B.; Zhu, C.; Wang, X.; Zhu, W. An Integration Model for Text Classification using Graph Convolutional Network and BERT. *J. Phys. Conf. Ser.* **2021**, *2137*, 012052. [CrossRef]
27. Vazquez Barrera, A. Neural News Classifier from Pre-Trained Models. Master's Thesis, Universitat Politècnica de València, Valencia, Spain, 2022.
28. Liu, J.; Xu, Y. T-Friedman Test: A New Statistical Test for Multiple Comparison with an Adjustable Conservativeness Measure. *Int. J. Comput. Intell. Syst.* **2022**, *15*, 29. [CrossRef]
29. Labani, M.; Moradi, P.; Ahmadizar, F.; Jalili, M. A novel multivariate filter method for feature selection in text classification problems. *Eng. Appl. Artif. Intell.* **2018**, *70*, 25–37. [CrossRef]
30. Dey Sarkar, S.; Goswami, S.; Agarwal, A.; Aktar, J. A novel feature selection technique for text classification using Naive Bayes. *Int. Sch. Res. Not.* **2014**, *2014*, 717092. [CrossRef] [PubMed]
31. Taieb, M.A.H.; Aouicha, M.B.; Hamadou, A.B. Ontology-based approach for measuring semantic similarity. *Eng. Appl. Artif. Intell.* **2014**, *36*, 238–261. [CrossRef]
32. Salton, G.; Yu, C.T. On the construction of effective vocabularies for information retrieval. *Acm Sigplan Not.* **1975**, *10*, 48–60. [CrossRef]
33. Bond, F.; Lim, L.T.; Tang, E.K.; Riza, H. The combined WordNet bahasa. *NUSA: Linguist. Stud. Lang. Around Indones.* **2014**, *57*, 83–100.
34. Alrababah, S.A.A.; Gan, K.H.; Tan, T.P. Mining opinionated product features using WordNet lexicographer files. *J. Inf. Sci.* **2017**, *43*, 769–785. [CrossRef]
35. Chebotko, A.; Lu, S.; Atay, M.; Fotouhi, F. Efficient processing of RDF queries with nested optional graph patterns in an RDBMS. *Int. J. Semant. Web Inf. Syst.* **2008**, *4*, 1–30. [CrossRef]
36. Miller, G.A. *WordNet: An Electronic Lexical Database*; MIT Press: Cambridge, MA, USA, 1998.
37. Dogru, H.B.; Tilki, S.; Jamil, A.; Hameed, A.A. Deep learning-based classification of news texts using doc2vec model. In Proceedings of the 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), Riyadh, Saudi Arabia, 6–7 April 2021; pp. 91–96.
38. Kang, M.; Ahn, J.; Lee, K. Opinion mining using ensemble text hidden Markov models for text classification. *Expert Syst. Appl.* **2018**, *94*, 218–227. [CrossRef]
39. Luhn, H.P. A statistical approach to mechanized encoding and searching of literary information. *IBM J. Res. Dev.* **1957**, *1*, 309–317. [CrossRef]
40. Arroyo-Fernández, I.; Méndez-Cruz, C.F.; Sierra, G.; Torres-Moreno, J.M.; Sidorov, G. Unsupervised sentence representations as word information series: Revisiting TF–IDF. *Comput. Speech Lang.* **2019**, *56*, 107–129. [CrossRef]
41. Sparck Jones, K. A statistical interpretation of term specificity and its application in retrieval. *J. Doc.* **1972**, *28*, 11–21. [CrossRef]
42. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 1–9.
43. Ren, Y.; Wang, R.; Ji, D. A topic-enhanced word embedding for twitter sentiment classification. *Inf. Sci.* **2016**, *369*, 188–198. [CrossRef]
44. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1188–1196.
45. Breiman, L. Machine learning. *Random For.* **2001**, *45*, 5–32.
46. Wang, Y.; Pan, Z.; Zheng, J.; Qian, L.; Li, M. A hybrid ensemble method for pulsar candidate classification. *Astrophys. Space Sci.* **2019**, *364*, 1–13. [CrossRef]
47. Rustam, Z.; Yaurita, F. Insolvency Prediction in Insurance Companies using Support Vector Machines and Fuzzy Kernel cMeans. *J. Phys. Conf. Ser* **2018**, *1028*, 012118. [CrossRef]
48. Rustam, Z.; Zahras, D. Comparison between support vector machine and fuzzy c-means as classifier for intrusion detection system. *J. Phys. Conf. Ser.* **2018**, *1028*, 012227. [CrossRef]

49. Rustam, Z.; Faradina, R. Face recognition to identify look-alike faces using support vector machine. *J. Phys. Conf. Ser.* **2018**, *1108*, 012071. [CrossRef]
50. Rustam, Z.; Ruvita, A.A. Application support vector machine on face recognition for gender classification. *J. Phys. Conf. Ser.* **2018**, *1108*, 012067. [CrossRef]
51. Rampisela, T.V.; Rustam, Z. Classification of schizophrenia data using support vector machine (SVM). *J. Phys. Conf. Ser.* **2018**, *1108*, 012044. [CrossRef]
52. Nadira, T.; Rustam, Z. Classification of cancer data using support vector machines with features selection method based on global artificial bee colony. In Proceedings of the AIP Conference Proceedings, Bali, Indonesia, 26–27 July 2017; p. 020205.
53. Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
54. Esteva, A.; Kuprel, B.; Novoa, R.A.; Ko, J.; Swetter, S.M.; Blau, H.M.; Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **2017**, *542*, 115–118. [CrossRef]
55. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
56. Panchal, G.; Ganatra, A.; Kosta, Y.P.; Panchal, D. Behaviour Analysis of Multilayer Perceptrons with Multiple Hidden Neurons and Hidden Layers. *Int. J. Comput. Theory Eng.* **2011**, *3*, 332. [CrossRef]
57. Zainal-Mokhtar, K.; Mohamad-Saleh, J. An oil fraction neural sensor developed using electrical capacitance tomography sensor data. *Sensors* **2013**, *13*, 11385–11406. [CrossRef]
58. Nozza, D.; Bianchi, F.; Hovy, D. What the [mask]? making sense of language-specific BERT models. *arXiv* **2020**, arXiv:2003.02912.
59. Jśnior, E.A.C.; Marinho, V.Q.; dos Santos, L.B. NILC-USP at SemEval2017 Task 4: A Multi-view Ensemble for Twitter Sentiment Analysis. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 611–615.
60. Rustam, Z.; Ariantari, N.P.A.A. Support Vector Machines for classifying policyholders satisfactorily in automobile insurance. *J. Phys. Conf. Ser.* **2018**, *1028*, 012005. [CrossRef]
61. Dong, R.; Schaal, M.; O'Mahony, M.P.; Smyth, B. Topic extraction from online reviews for classification and recommendation. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI 13), Beijing, China, 3–9 August 2013; pp. 1310–1316.
62. Farkiya, A.; Saini, P.; Sinha, S.; Desai, S. Natural language processing using NLTK and WordNet. *Int. J. Comput. Sci. Inf. Technol.* **2015**, *6*, 5465–5469.
63. Chiorrini, A.; Diamantini, C.; Mircoli, A.; Potena, D. Emotion and sentiment analysis of tweets using BERT. In Proceedings of the EDBT/ICDT Workshops 2021, Nicosia, Cyprus, 23–26 March 2023; Volume 3.
64. Kishan Yadav. Available online: https://www.kaggle.com/datasets/kishanyadav/inshort-news?select=inshort_news_data-1.csv (accessed on 13 January 2023).
65. Yang, Y.; Uy, M.C.S.; Huang, A. FinBERT: A pretrained language model for financial communications. *arXiv* **2020**, arXiv:2006.08097.
66. Dumitrescu, S.D.; Avram, A.M.; Pyysalo, S. The birth of Romanian BERT. *arXiv* **2020**, arXiv:2009.08712.
67. Jahan, M.S.; Beddiar, D.R.; Oussalah, M.; Arhab, N. Hate and Offensive language detection using BERT for English Subtask A. In Proceedings of the FIRE 2021: Forum for Information Retrieval Evaluation, Gandhinagar, India, 13–17 December 2021.
68. Keya, A.J.; Wadud, M.A.H.; Mridha, M.F.; Alatiyyah, M.; Hamid, M.A. AugFake-BERT: Handling Imbalance through Augmentation of Fake News Using BERT to Enhance the Performance of Fake News Classification. *Appl. Sci.* **2022**, *12*, 8398. [CrossRef]
69. Gasmi, K. Improving BERT-Based Model for Medical Text Classification with an Optimization Algorithm. In *Advances in Computational Collective Intelligence, Proceedings of the 14th International Conference, ICCCI 2022, Hammamet, Tunisia, 28–30 September 2022*; Springer International Publishing: Cham, Switzerland, 2022; pp. 101–111.