applied system innovation

MDPI

Article

A Novel Machine Learning Approach for Sentiment Analysis on Twitter Incorporating the Universal Language Model Fine-Tuning and SVM

Barakat AlBadani , Ronghua Shi and Jian Dong *

School of Computer Science and Engineering, Central South University, Changsha 410083, China; barakat_lt@hotmail.com (B.A.); shirh@csu.edu.cn (R.S.)
* Correspondence: dongjian@csu.edu.cn

Abstract: Twitter sentiment detectors (TSDs) provide a better solution to evaluate the quality of service and product than other traditional technologies. The classification accuracy and detection performance of TSDs, which are extremely reliant on the performance of the classification techniques, are used, and the quality of input features is provided. However, the time required is a big problem for the existing machine learning methods, which leads to a challenge for all enterprises that aim to transform their businesses to be processed by automated workflows. Deep learning techniques have been utilized in several real-world applications in different fields such as sentiment analysis. Deep learning approaches use different algorithms to obtain information from raw data such as texts or tweets and represent them in certain types of models. These models are used to infer information about new datasets that have not been modeled yet. We present a new effective method of sentiment analysis using deep learning architectures by combining the "universal language model fine-tuning" (ULMFiT) with support vector machine (SVM) to increase the detection efficiency and accuracy. The method introduces a new deep learning approach for Twitter sentiment analysis to detect the attitudes of people toward certain products based on their comments. The extensive results on three datasets illustrate that our model achieves the state-of-the-art results over all datasets. For example, the accuracy performance is 99.78% when it is applied on the Twitter US Airlines dataset.

Keywords: machine learning; transfer learning; sentiment analysis; SVM; ULMFiT; US airlines

check for updates

Citation: AlBadani, B.; Shi, R.; Dong, J. A Novel Machine Learning Approach for Sentiment Analysis on Twitter Incorporating the Universal Language Model Fine-Tuning and SVM. *Appl. Syst. Innov.* **2022**, *5*, 13. https://doi.org/10.3390/asi5010013

Academic Editors: Patrícia Ramos and José Oliveira

Received: 28 November 2021
Accepted: 11 January 2022
Published: 14 January 2022

1. Introduction

Internet data grow rapidly, given the preference of citizens to share their views. Through the expansion of social media, people's opinion tools have been updated, and fields such as opinion mining and sentiment analysis have obtained growing demands. Online reviews cannot be overlooked owing to the possible effects that customer feedback may have on companies. A significant number of research practitioners are currently developing structures that can collect information from such feedback to support marketing insight, drive public sentiment, and enhance consumer loyalty. Consequently, opinion analysis was implemented and applied in several study areas and companies. Twitter has been one of the most widely used microblogging services and a fascinating forum for more than 500 million messages per day from about 1.3 billion people. A message on Twitter (similar to a post on Facebook), by a sequence of characters confined to 280-character limit, is posted publicly or by established followers owing to the account's privacy, which is different from other social media websites [1].

However, users struggle to articulate their brains with limited words because they are constrained in character. Twitter is a significant reflection of world affairs, because it is commonly utilized across all classes of people. As a part of the company's review and feedback, Twitter is a key outlet for the new developments appearing on Twitter. These patterns are used to reach commercial objectives, such as viral ads for trend items [2].

Analysis of tweets is important for businesses to establish effective campaign ideas. Twitter users tweet various limitless messages throughout the day that can offer opinions regarding products, companies, items, and public relations. These views may be categorized into three distinct attitudes: negative, positive, and neutral. Each attitude may also be classified into different levels. This method is called a sentiment analysis that evaluates and derives contextual knowledge from raw data.

The study area of emotion analysis and the field of natural language processing (NLP) are interlinked. Extracting useful knowledge from naturally written texts allows NLP to resolve the distance between humanity and machine. The goal of text sentiment analysis (SA) is to extract and analyze knowledge from personal data or product reviews and feedback provided on the internet. As a result of its wide range of industrial and academic applications, as well as the rapid growth of social networks, SA has emerged as a key topic in the field of natural language processing (NLP) in recent years [3,4].

In the literature, three different approaches have been used to solve the problem of SA: the lexicon-based approach, standard machine-based approach, and deep learning-based approach. The lexicon-based method uses a glossary of sentiment terms including enhancement and negation to measure the polarity of each phrase. In the standard machine learning-based approach, the classification of opinions is used as a special case of the issue of classification of documents. However, this method depends on the extraction of knowledge from a statement with an opinion polarization. In addition, this extraction can be annotated individually by terms or automatically through sentiment index-like ratings in comments or emoticons used in tweets. The third approach is deep leaning-based approaches, which has two phases. The term embedding in the text corpus is learned in the first phase [5,6]. The second phase focuses on the use of word embedding to create interpretations of sentences of semantic composition using different deep learning techniques [7].

The ability of any classification method may be enhanced by a technique of integrating several predictor outcomes. The voting system creates a standard strategy by compound classifiers on a stand-alone basis; then, their outputs are integrated into the final decision. In addition, the final polarity label is calculated by the results of each section of the class i.e., the plurality of the polarity labels [7].

This study has three main goals: developing Twitter sentiment detectors that can perfectly and quickly detect the sentiment, improving the accuracy of classification, training, and testing times, computational complexity, and storage requirements of the SVM algorithm, in the context of Twitter sentiment detection.

The rest of the paper contains the following sections: the related work is shown in Section 2, the proposed model is presented in Section 3, the experimental results and evaluation are discussed in Section 4, the discussion is shown in Section 5, and the conclusion in Section 6.

2. Related Work

Recently, many researchers in the field of sentiment analysis have used a supervised machine learning algorithm as their clustering module and primary classification, such as the work in [8]. These methods classified and displayed user-created texts that contained the sentiment using n-gram features as well as the bag-of-words (BOW) technique, and they were occasionally combined [9]. The n-gram characteristics have been developed to address the shortcomings of the simple BOW model, such as the fact that it ignores grammatical structures and word order [10]. When using n-gram features, there is a significant disadvantage, particularly when $n \geq 3$, in that the feature distance output is remarkably high dimensional. As a result, feature selection algorithms have been used extensively in recent studies [11] to overcome this disadvantage. Users' sentiments can be extracted from their text using a variety of algorithms, including SVM, Naive Bayes (NB), and artificial neural networks (ANN). Several methods, including that in [12], have demonstrated good performance in this area. When using supervised approaches, one of the challenges that can arise is that they are often slow and require a large amount of time throughout the training process. There have been numerous approaches based on the unsupervised lexicon that have been proposed to address these issues, including those in [8,13]. These methods are quick, scalable, and simple to implement. On the other hand, they rely largely on vocabulary, which has resulted in them being less accurate when compared to their supervised counterparts [13,14]. There have been very few researchers who have taken advantage of the advantages offered by both lexicon-based and supervised-based approaches and then merged them in a variety of ways, as cited by [14,15]. For sentiment analysis, Zhang et al. [16] presented an approach that is divided into two parts at the entity level of tweets. Having a high recall rate is the first step in the process, which is achieved through the use of the supervised approach. The second phase is a high-precision lexical method that is built on the previous stage. According to [17], machine learning methods and lexicon-based sentiment analysis have been combined in a concept-based sentiment analysis model that has been presented. When compared to simply statistical methods, their method was more accurate and justified, and its ability to discern the polarity or strength of sentiment was superior to lexicon-based methods.

2.1. Transfer Learning with ULMFiT

One of the most important techniques commonly used in machine learning applications is transfer learning. This technique aims to convert wisdom knowledge gained from specific tasks to other related tasks.

In 2018, ULMFiT was developed as a natural language processing method (NLP) by Howard and Ruder as part of the fast.ai framework [4]. ULMFiT is used as a transfer learning technique for information extraction solutions. This technique requires a small amount of data in the process of training the model. In addition, ULMFiT can be used with any dataset of different lengths or any document, with a single architecture such as AWD-LSTM; thus, it is called universal. Another great advantage of this technique is its ability to be compatible with any task without engineering for a custom feature [4,18].

2.2. Language Model

ULMFiT can use word embedding, and the entire language model depends on recurrent neural network (RNN) and AWD-LSTM. The technique is based on exclusive training. This model focuses more on typical phrases in addition to becoming accustomed to different relationships among the words [5].

Figure 1 demonstrates the structure of the language model. The first layer is the embedding layer (the black one), which resembles the word embedding. In modern models, the RNN layers also reused the same embedding layer. The RNN layers are the neural network elements with implicit feedback. Words are separately replaced in the embedding layer by the equivalent embedding vector and then fed into the RNN layer as the next input [6]. As shown in Figure 1, the text "People prefer to travel by plane" serves as input and output after shifting by one word; this findings explains the technique of how the model can predict the next word.

2.3. AWD-LSTM

ASGD weight-dropped LSTM or AWD-LSTM uses drop connect and a variant of average-SGD (NT-ASGD) combined with many additional famous regularization approaches. AWD-LSTM is the architecture used by ULMFIT for its language modeling tasks [7,19]. As shown in Figure 2, the repeating LSTM module consists of four interacting layers. The general LSTM unit is formed from a cell, an input gate, an output gate, and a forget gate. LSTM can perform many operations such as language modeling, character-level neural machine translation, and sentiment classification.

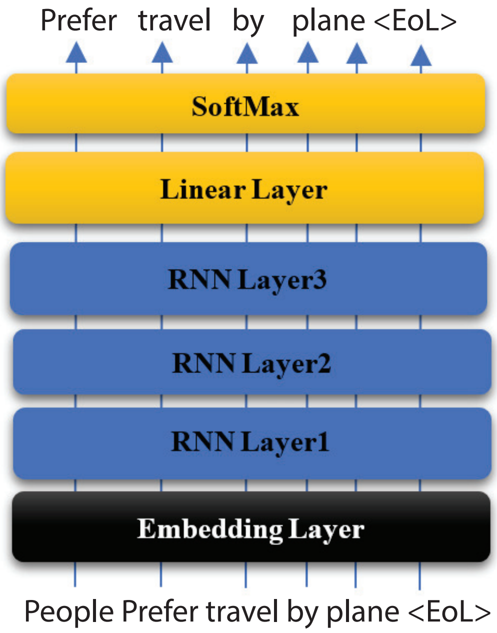

Figure 1. Structure of the language model.

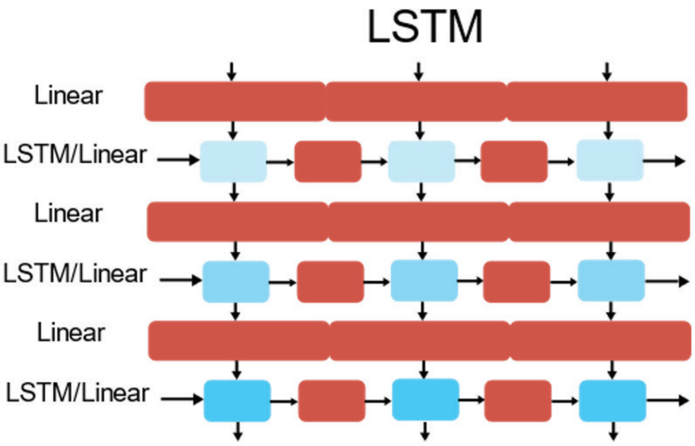

Figure 2. LSTM architecture.

The following Equation (1) is the mathematical formulation to express LSTM:

$$
\begin{aligned}
i_t &= \sigma(W^i X_t + U^i h_{t-1}) \\
f_t &= \sigma(W^f X_t + U^f h_{t-1}) \\
o_t &= \sigma(W^o X_t + U^o h_{t-1}) \\
c_t^{'} &= \tanh(W^c X_t + U^c h_{t-1}) \\
c_t &= i_t \bigodot c_t^{'} + f_t \bigodot c_{t-1}^{'} \\
h_t &= o_t \bigodot \tanh(c_t)
\end{aligned}
\tag{1}
$$

where ($W^i, W^f, W^o, W^c, U^i, U^f, U^o, U^c$) are weight matrices;
(x_t) is the vector input to time step t;
(h_t) is the current exposed hidden state;
(c_t) is the memory cell state;
and ($\bigodot$) is the element-wise multiplication.

2.4. Support Vector Machine (SVM)

The main conception of SVM is to categorize information separately using hyperplane to increase the margin among them to the maximum. SVM is commonly used in learning classification algorithms based on statistical learning theory (SLT), which is a theoretical paradigm for machine learning that draws on statistics and functional analysis. SLT has been successfully applied in various areas, including computer vision, voice recognition, and bioengineering. SVM has high classification accuracy and excellent performance; thus, it is popularly widely used in problems classifications [20].

SVM is one of the effective ML algorithms for regression and classification and sentiment detection [21]. SVM is one of the most popular classifiers due to its capability to achieve higher generality performance whenever the dimension embedding of the input features space is very high. The purpose of SVM is to discover the most suitable classification function to differentiate between members of the two classes in the training data. The SVM classification uses structural risk reduction to create a hyperplane for the exclusion of positives from a group of negative instances. SVM aims to separate data points and evaluate each data point category into a hyperplane. SVM maximizes the margin between support vectors because separating all classes is necessary [22]. SVM is commonly used to solve many problems in the real world, including intrusion recognition, image processing, text classification, etc. Initially, SVM was designed to classify the binary classes. Later, the examples were expanded to multiple classes. In binary classification, the classification task is the data points classification task of a given dataset of instances into two separate sets and defines whether they have some features or not. Many real-world tasks have been used to the binary classification task in its implementation, where the response to some query is either a yes or no: for example, object detection, figuring out relations to a specific class of the instance, face detection, or intrusion detection. The mathematical basis of the SVM algorithm is presented in the binary classification task in two cases: linear divisible and non-linear divisible cases. In a case that is non-linear divisible, there is one or more than one hyperplane that may split the two categories represented by the training data. The well-known query is how to select the most suitable hyperplane that would get the best out of the accuracy performance on test data. The best answer is to exploit the boundary between support vectors that splitting the positive and negative points into the training data. Then, the most suitable hyperplane is the one that evenly splits the boundary between the two classes.

2.5. Long Short-Term Memory (LSTM)

LSTM is an extension of RNN. The vanishing of gradients in the training of vanilla RNN was proposed. It has a special memory mechanism that gives it an advantage over the vanilla RNN. The memory mechanism enables the network to capture long-term dependencies, in particular the LSTM appraoch in [23], which succeeded in minimizing dimensionality and outstanding performing concerning the precise classification of opinion. Reducing the input functions is an important task for the classification of sentiment based on machine learning techniques. Therefore, the suggested method could be a promising solution for better classification with scalability. The proposed approach would be particularly suitable for applications that have a large dataset such as the detection of sentiment for product and service reviews.

Tarasov [24] used a long short-term model for the sentiment of the restaurant reviewers for RNNs. Several approaches are used to compare the gathering results using the following techniques: simple recurrent neural networks, logistic regression, bidirectional RNNs, and bidirectional long-term memory. Deep bidirectional LSTM with numerous hidden layers yielded the best performance across all RNN models.

A vector representation of single words can be considered as a system training parameter to simplify the analysis of the general model text data. Thus, we can initiate values for single words with random values of vector representations to replicate certain variables.

Tai et al. [25] performed the LSTM to solve identifications of the romantic phrases extracted from film reviews and estimation of somaticizing sentence pair's tasks.

Socher et al. [26] presented the Treebank sentiment and recursive neural tensor networks to solve their feeling detection project. In the case of single-sentence positive/negative cataloging, by using recursive neural tensor networks, performance increased from 80% to 85%.

The prediction of fine-grained feeling labels for all sentences was 80.7% higher than the 9.7% increase with a bag of features for approaches, such as SVM and Naive Bayes using the recursive neural tensor network process. Neural networks are effective for managing text data analysis tasks as variants in the LSTM model. The use of these models is an innovative approach to describe the social networking emotions of users.

The idea with LSTM is to have a self-regulating flow of information through the cells that can be forgotten or modified based on the information input to the cell. Therefore, some extra parameters have been added to each recurrent cell to enable the RNN to propagate information and overcome optimization issues. The added parameters, as shown in Figure 3, act as filters to allow the cell to select which information is worth remembering and which is worth forgetting.

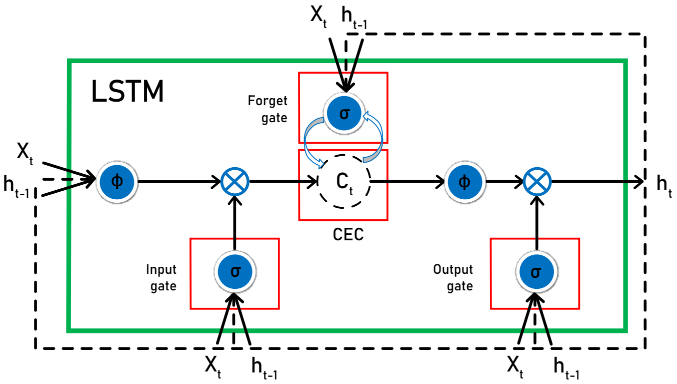

Figure 3. LSTM diagram.

3. Methodology

We used ULMFiT combined with SVM and applied it on different sentiment analysis datasets. We used radian basis function (RBF) kernel as the basic SVM function, such as sigmoid functions, polynomial function, and linear functions.

RBF kernel has two parameters (C and σ). C is the penalty parameter for the error term and kernel function's coefficients. The main function of this parameter is to control the tradeoff between hyperplane smooth decision boundary and the support vector (classifying training points) successfully where sigma (σ) is a non-linear hyperplanes parameter. The gamma value is the indicator of trained dataset fitting; a higher gamma value indicates a fitter trained dataset. Suitable kernel function type and suitable parameters should be selected effectively to obtain SVM with high performance. We used the RBF kernel, and then, it was tuned by grid search with a three-fold (KFOld) cross-validation (GridsearchCV) method.

3.1. ULMFit–SVM Model

ULMFiT is the most effective method in transfer learning, and it can be used with different tasks in NLP. ULMFiT is pretrained by LM on a large public domain for the collection of written texts. We proposed the ULMFit–SVM method. The objective is to use innovative techniques with an SVM classifier instead of SoftMax. For instance, for k nodes of the SotMax layer, the probability distribution is denotes by pi, which is calculated with the following Equation (2):

$$p_i = \frac{\exp(a_i)}{\sum_j^k \exp(a_j)} \tag{2}$$

where a_i indicates the total input to the softmax layer, and the $\hat{I}$ class would be expected to be $\hat{I} = arg_i max_{p_i}$.

After introducing the basic ideas underlying ULMFiT, we can focus on the structure of the actual model. The model can be split into three phases to provide a general overview, as deliberated in Figures 4 and 5:

- General-domain LM pre-training;
- Target task LM fine-tuning;
- Target task classifier.

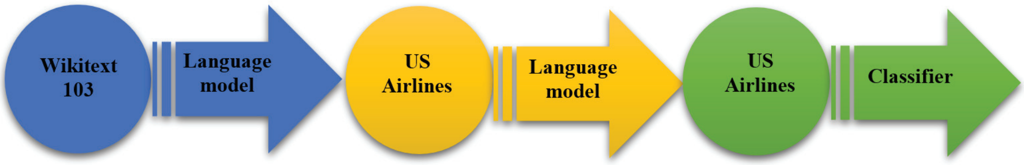

Figure 4. General domain LM pretraining.

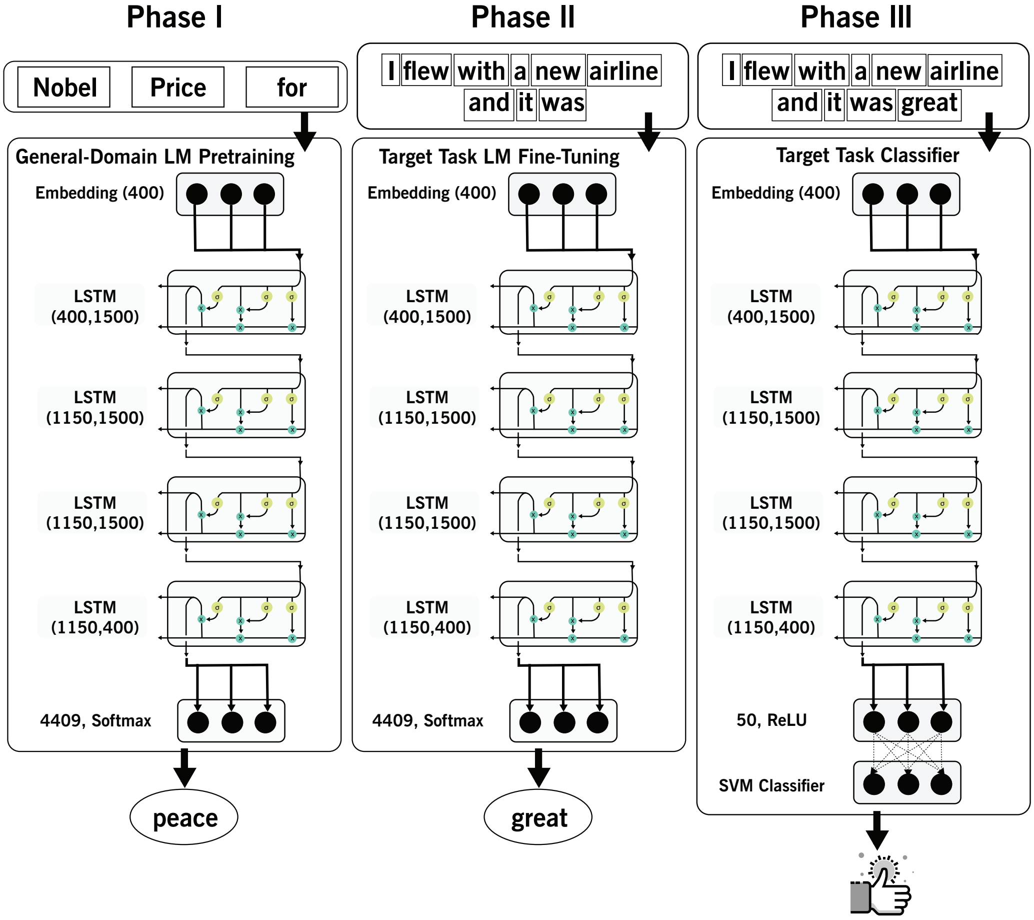

Figure 5. Detailed overview of ULMFiT-SVM.

3.2. Pretrained Phase

The pretrained phase, which is commonly conducted on Wikitext-103, consists of 28,595 English articles. These articles consist of approximately 103 million words [27]. The language model is based on word prediction, where is a large data source is used to obtain the ideal performance.

3.3. Fine-Tuning the Language Model

It is an intermediate phase prior to the classification task, which was used to avoid direct use of the Wikipedia pretrained model and fine-tuning. The aim of this phase is to fine-tune the language model on information from the objective task; classification is implemented to enhance the classification model (deliberated in the third phase) on minor datasets. The subsequent techniques achieved improvement in this phase.

3.3.1. Slanted Triangular Learning Rates (STLR)

The language model is commonly used and applied on a target text that is different from the pretrained model. In this phase, the function of the fine-tuning is to help the model parameters become more suitable for the target text. This is the main function of STLR. The difference between TLR and STLR is the linear short increase and long decay [4].

The initial slight increase in the learning rate is important because coverage of the model is suitable for the measurable factor of the aimed task. The next step is the long decay period that provides more refining of the parameters [5].

The STLR update schedule is given in Equation (3), as follows:

$$\begin{aligned} cut &= (T.cut - frac) \\ p &= \begin{cases} t/cut & \text{, if } t < cut \\ 1 - \frac{t-cut}{cut.(1/cut-frac-1)} & \text{, otherwise} \end{cases} \\ \eta_t &= \eta_{max}.\frac{1+p.(ratio-1)}{ratio} \end{aligned} \tag{3}$$

where

- (T) refers to the count of training iterations (one training iteration is equal to the number of epochs times the number of updates per epoch).
- (cut_frac) refers to the fraction of iterations.
- (cut) refers to the iteration in case of raising or lowering the LR.
- (for $t < cut, p)$ refers to the count of iterations the LR has increased upon the total number of increasing iterations
- $t >= cut, p$ refers to the total count of iterations the LR has decreased upon the total number of decreasing iterations.
- $(ratio)$ states the size of the lowest LR compared with the maximum LR, η_{max}.
- (ηt) refers to the learning rate at iteration t.
- $cut\_frac = 0.1, ratio = 32$ and $eta_{max} = 0.01$.

3.3.2. Discriminative Fine-Tuning (DFT)

DFT requires that various layers in a model seize various types of information and learning rates. The first layers seize the common form of data. The same with the language modeling task, the initial layers seize the maximum common data of the language and require the minimum quantity of fine-tuning. Consequently, moving toward the last layer increases the amount of necessary fine-tuning [7]. Hence, we used different learning rates for each layer as an alternative of using the identical learning level in the entire model. We used fine-tuning for the last layer to select the learning rate. The following formula was used for the lower layers: $(\eta_l - 1 = \eta_l/2.6)$, where η_l is the learning rate of the l-th layer.

3.4. Model Training

The third phase in our model is to train the model with additional dual linear blocks. We used ReLU activation as the transitional layer and then SVM as a supervised algorithm for the classification task and distinguishing various styles of classification because conjoining strong classifiers, such as SVM, with ULMFiT boosts TSD performance.

The resulting features are delivered to the SVM classifier for the classification process, which was originally formulated for binary classification.

Training data are provided with labels $(x_n, y_n), n = 1, \ldots, N, x_n \in R^D, t_n \in \{-1, +1\}$, and the following Equation (4) show optimizations of SVMs:

$$\begin{aligned} &min_{W\xi_n} \frac{1}{2} W^T W + C \sum_{n=1}^{N} \xi_n \\ &s.t. W^T X_n t_n \geq 1 - \xi_n \forall n \end{aligned} . \tag{4}$$

Data points are penalized by ξ_n, which indicates slack variables. We can include the bias, considering the addition of a scalar value of xn. The following Equation (5) is the accurate unregulated optimization problem: then, data points that exceed the margin and would penalize are obtained as follows:

$$min_W \frac{1}{2} W^T W + C \sum_{n=1}^{N} max(\, 1 - W^T x_n t_n, 0) \;. \tag{5}$$

The goal of the following Equation (6) is known as the main form of the L1-SVM problem through the hinge loss. L1- SVM cannot be differentiable; thus, a popular variation called L2-SVM is obtained as follows:

$$min_W \frac{1}{2} W^T W + C \sum_{n=1}^{N} max(\, 1 - W^T x_n t_n, 0) \;^2. \tag{6}$$

On the contrary, L2-SVM is differentiable and enforces greater loss of those points. The following Equation (7) indicates the class label prediction for X, as follows:

$$arg_t max(\, W^T X) \;_t. \tag{7}$$

We used Radian Basis Function (RBF) kernel as the basic SVM functions such as sigmoid functions, polynomial functions, and linear functions.

The accuracy percentage of our novelist technique is more accurate than that of SVM alone, and at the same time, we could reduce the time used in the training and testing processes [28].

3.4.1. Concat Pooling

The pooled last hidden layer states were used as the first linear layer input. We may lose the information if we consider only the last hidden state of the model because the signal in the classified text sometimes has few words, and this condition can occur in any document. Thus, we concatenated the last hidden state of the model with the max-pooled and the mean-pooled to represent the hidden states over as long as the step is fitted in the GPU memory [4].

3.4.2. Gradual Unfreezing

The benefits of language model pretraining could be nullified as a result of aggressive fine-tuning due to the high sensitivity of the fine-tuning target classifier. Consequently, we proposed gradual unfreezing to conduct fine-tuning classification [5].

- The first unfrozen layer is the last LSTM layer, and then, the model is fine-tuned for one epoch.
- Subsequently, the following lower layer is unfrozen.
- The same procedures of unfreezing are performed on all layers until they are fine-tuned to convergence.

3.4.3. BPTT for Text Classification (BPT3C)

Back-propagation through time (BPTT) is used for text classification and language modeling. This technique divides the text into fixed-length portions. The model starts initializing each portion using the final state of the previous portion. The hidden state of the gradients that is back propagated to the batches is used to contribute to the final prediction. Variable length back-propagation sequences are the most common practical model used [29].

3.4.4. Bidirectional Language Model

After training the backward language model and the forward language model, they are fine-tuned by the classifier individualistically. The final output is the calculation of the average of the classifier prediction results.

3.5. Dataset Overview

We selected the WikiText-103 dataset for our source task, which consists of 28,595 pre-processed Wikipedia articles whose contents sum up to 103 million words. For sentiment analysis, we selected the Twitter US Airline Sentiment dataset. The dataset contains 14,485 tweets regarding the most operated US airlines. The dataset is classified according to the attitudes; the tweets are labeled as either positive, negative, or neutral. Figure 6 and Table 1 show the classifications of the tweets [5].

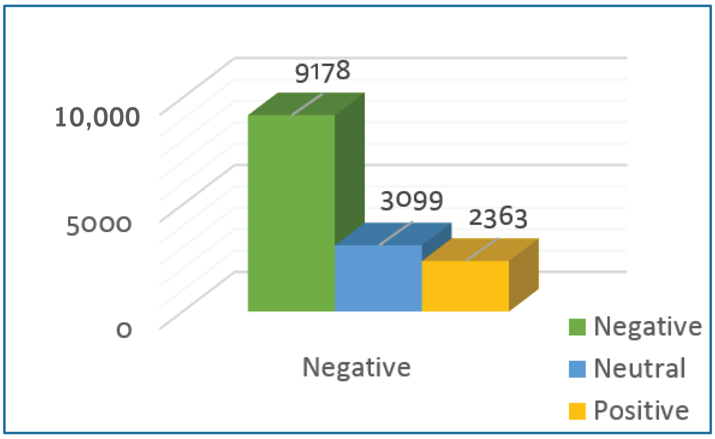

Figure 6. Sentiment distribution in the Twitter US airline sentiment dataset.

Table 1. Twitter sentiment datasets.

	Split	**Twitter US Airlines [30]**	**IMDB [31]**	**GOP Debate [32]**
Positive	Train	1773	18,750	1665
	Test	590	6250	555
Negative	Train	6884	18,750	6357
	Test	2294	6250	2104
Natural	Train	2325	–	2393
	Test	774	–	797
Total	–	14,640	50,000	13,871

We applied the ULMFiT-SVM model on US Airline tweets; Howard and Ruder applied ULMFiT for analyzing sentiments of Yelp and IMDb dataset reviews that contains 25,000 to 65,000 examples. The applied tweets have limitations due to the limited words of the tweet (140 character for each tweet), which is shorter than those in the Howard and Ruder dataset.

3.6. Dataset Preprocessing

Data preparation is regarded as a critical step in the machine learning and data mining processes. The reviews contain incomplete sentences, a high proportion of noise, and weak wording, such as terms that have no applicability, a lot of repetition, imperfect words, and bad grammar. Similarly to structured data, unstructured data have an impact on the results of sentiment classification. Preprocessing the reviews is required in order to maintain a regular structure and limit the likelihood of such issues occurring. In our research, we use cleaning data with filters, partitioning the data into portions for training and testing, and creating datasets with favored words. We employed the following approaches to prepare the data for the analysis: As part of our research, we separated the text into sections based on phrases, words, symbols, and other important aspects, resulting in a list of specific words for each comment. Then, we used each word in each comment as a feature for our training classifier, which was then applied to the entire comment stream. We also delete stop words that have no significance, such as prepositions, and words that do not provide any emotional value (or, also, able, etc.).

3.7. Word Embedding

The neural networks or any machine learning algorithm cannot deal with language in its raw form. Thus, converting the language into a type of numerical representation for processing is necessary.

The function of word embedding is to represent words in the form of real-valued vectors in a predefined vector space. Such a dense representation is clearly superior to the high dimensionality of sparse word representations.

The vectors can also recognize semantic relationships among words. For example, it can recognize the relationship between "man" and "woman" as well as that between "king" and "queen" as "male–female" relationships. It can recognize the relationship among the different verb forms such as walk, walks, walked, and walking.

In our AWD-LSTM, the vectors demonstrating the respective words are prepared in the embedding layer and updated while training the neural network. We used the embedding size of 400, indicating that one word is embodied in a 400-dimensional vector space.

The text should be prepared by deleting unnecessary texts as follows:

1. Extra spaces, tab characters, newline characters, and other characters should be removed and replaced with regular characters.
2. To tokenize the data, we use the spaCy library. Since spaCy does not have a parallel/multicore tokenizer, the fast.ai package is used to offer this feature. This parallel version of the spacy tokenizer takes advantage of all of the cores on your computer's CPUs and is significantly faster than the serial version.

The main function of tokenization is to split the text into separate tokens to assign a unique index for each token. This finding indicates that we convert the text into integer indexes that can be used by our model.

The next step is numericalizing, which indicates transforming tokens into numbers. The list of all the tokens is called vocabulary. The process of numericalization is very simple and easy as follows:

- Making a list of all the words that appear in the same order.
- Replacing each word with its index into that list.

We do not focus on all the vocabulary list because the unique words that are not repeated and appear only once are unimportant and will be ignored. The word should appear at least twice to ensure that they are accurately spelled. The model cannot learn anything from the world if it does not appear frequently.

3.8. Evaluation Metrics

We used the Twitter US Airline Sentiment dataset to test and validate the advantage of our technique in enhancing the SVM classification outcomes for text classification.

The possible performance metrics were applied to evaluate the performance of the used techniques. We used the attribute values obtained from the training and testing processes of the Twitter US Airlines sentiment dataset to evaluate the performance metrics of the following attribute values: positive, negative, and neural. Then, the performance metrics were calculated; they may seem negligible but extremely divisive; that is, we should measure accuracy when dealing with the language model.

Accuracy (AC) is the percentage of accurate classifications of the total records used in performing the test, and accuracy expresses about how far the model can predict the next word accurately.

4. Performance Evaluation

We perform our experiments on a Laptop 81L Legion Y7000P with Intel(R) Core(TM) i7-8750H CPU at 2.20_GHZ, 6_cores and RAM of 32 GB. Our implementations are processed under Windows with Python 3.7 version.

The LIBSVM package (Python version 3.23) is used to apply the SVM classifier on the selected dataset for the process. We applied Grid Search for the best parameter selection, CV. We selected the Twitter US Airline Sentiment dataset to evaluate our approach in terms of performance.

4.1. Evaluation Based on Testing Data

We evaluated the performance of the proposed model in three classes using testing data and compared the results with other approaches on the same Kaggle datasets [7] to validate the advantage of our suggested technique. Table 2 displays that the accuracy of our suggested model is better than that of others, thereby demonstrating that our model has greater efficiency than others.

Table 2. Comparison of the accuracy of our ULMFiT-SVM model with other models.

Method	Accuracy
Support Vector Machine (SVM)	78.5%
Bag-of-words SVM	78.5%
Deep Learning Model with Dropouts in Keras	77.9%
SIS-ULMFiT [7]	84.1%
(ULMFiT-SVM) [Ours]	99.78%

4.2. Effect of Hyper-Parameters and Hidden Units Number Setting in Our Model Efficiency

Based on the theoretical study of the ULMFit-SVM, the parameters and hidden unit number setting in our model demonstrate that the hidden unit numbers and parameters are the most important parameters determining classification accuracy and training time speed. As a result, optimizing hyperparameters is a critical problem in the development and design of an effective learning model for network sentiment detection.

An AWD-LSTM language model with embedding size 400, three layers with hidden activations of 1150 for each layer, and a batch size of 70 BPTT were employed in this study. Dropouts of 0.4 were applied to layers, 0.3 were applied to RNN layers, 0.4 were applied to input embedding layers, 0.05 were applied to embedding layers, and a weight dropout of 0.5 was applied to the RNN hidden-to-hidden matrix.

The classifier has a hidden units size of 50, which is a large number. For the LM and the classifier, we used Adam with $beta1 = 0.7$ instead of the usual $beta1 = 0.9$ and $beta2 = 0.99$, and we utilized 64 batches, a base learning rate of 0.004 and 0.01 for fine-tuning the LM and the classifier, respectively. In Table 3, we display the values of the hyper-parameters of our model for sentiments that have been tested.

Using a k-fold cross-validation strategy, we could also increase the overall performance of the single SVM to find the optimal RBF kernel parameters ($C = 5.6569$ and $sigma = 1.0667$) and to fine-tune our approaches' hyperparameters.

Table 3. Tested values of hyper-parameters for our model for our sentiment model.

Hyper-Parameter Name	Meaning	The Best Value
em-sz	Embedding vector size	0.77
nh	Hidden activations number	0.000005
nl	Number of layers	3
bs	Batch size	32
$\beta1$	Optimal bias	0.8
$\beta2$	Optimal bias	0.99
C-GAMMA	SVM parameters	5.6569–1.0667

5. Discussion and Additional Comparisons

In addition, we demonstrate the superiority of our model by comparing its detection accuracy with that obtained from other classification methods in comparable investigations, as discussed in Table 4.

Table 4. Performance comparisons for ULMFiT-SVM-based Twitter US Airlines, IMDB, and GOP debate datasets with several related approaches.

Dataset	Used Model	Accuracy
Twitter US Airlines [30]	SVM only [33]	78%
	RNN/LSTM (ULMFiT) [34]	77.8%
	LSTM, CNN [35]	79.64%
	MultinomialNB [36]	±80%
	ABCDM [37]	±92.75%
	ULMFit-SVM (Ours)	99.78%
IMDB [31]	ToWE-SG [38]	90.8%
	ULMFiT [7]	95.4%
	BERT large fine-tune UDA [39]	95.8%
	RCNN [40]	84.70%
	ULMFit-SVM (Ours)	99.71%
GOP Debate [32]	SIS-ULMFiT [41]	55.034%
	ULMFit-SVM (Ours)	95.78%

Daniel Langkilde in 2017 used linear SVM classification of sentiment in tweets about airlines. The accuracy result was 78%, and the author suggested that achieving a higher score with more tuning or a more advanced approach is possible [42]. Wesley Liao in 2019 explored the Twitter US Airline sentiment dataset and attempted to predict tweet sentiment using a language model and RNN via Fast.ai's library for ULMFiT.

This model achieved an accuracy of 77.8% [43]. Anjana Tiha in 2019 performed sentiment analysis with LSTM and CNN. The accuracy of the LSTM and CNN model was 79.64%. Carlo Lepelaars in 2019 predicted the Twitter US Airline Sentiment dataset with Multinomial Naive Bayes (NB). Multinomial NB is a good method for text mining and is useful to set a benchmark for sentiment analysis. This technique is easy to implement with sklearn's "MultinomialNB" class [36]. Table 5 and Figure 7 show additional performance comparisons for ULMFiT-SVM based on different datasets.

Table 5. Additional performance comparisons for ULMFiT-SVM based on different datasets.

Evaluation Type	Accuracy Rate
Twitter US Airlines [30]	99.78%
IMDB [31]	99.71%
GOP Debate [32]	95.78%

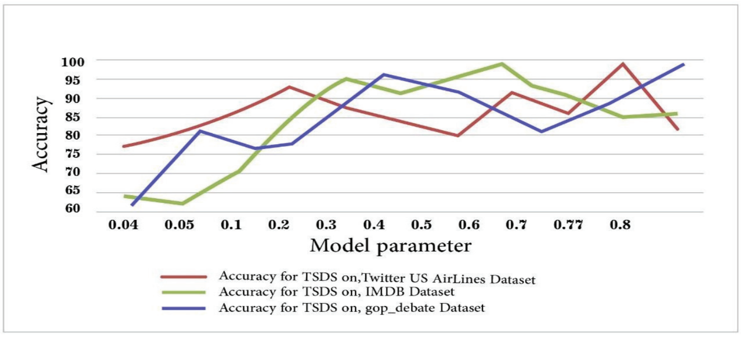

Figure 7. Comparison of the accuracy of our TSDSs based on different datasets.

Moreover, Table 6 shows that our proposed technique can decrease the training and SVM testing time, which is essential for evaluating the efficiency of sentiment detection systems.

Table 6. Training time, testing time, and number of support vectors (nSV) of ULMFit-SVM in comparison with SVM for binary classification.

Technique	Time of Training (s)	Time of Testing (s)	nSV
SVM	901.095	18.533	3649
ULMFit-SVM	682.10	4.321	3448

We also report the F1-score for our model. The F1-scores achieved by ULMFit-SVM are 99.01%, 98.67%, and 95% on the Twitter US Airlines, IMDB, and GOP debate datasets, respectively.

6. Conclusions

In this paper, we propose a novel ULMFit-SVM model to improve the sentiment analysis performance. The proposed model introduces an effective deep learning architecture that combines the universal language model fine-tuning with a support vector machine. The extensive results on three real-world datasets demonstrate that the proposed model increases detection efficiency and accuracy. For example, the model demonstrates an accuracy rate of 99.78% on Twitter US Airlines, 99.71% on IMDB, and 95.78% GOP debate. In this study, the sentiment analysis was restricted to document level. In this investigation, we did not take into account the sentiment at the aspect level. This part will be completed in the future.

Author Contributions: Conceptualization: B.A. and J.D.; methodology: B.A. and J.D.; software: B.A. and R.S.; validation: B.A. and J.D.; investigation: B.A.; resources: J.D.; data curation: B.A. and J.D.; writing—original draft preparation: B.A.; writing—review and editing: R.S. and J.D.; visualization: B.A.; supervision: B.A. and J.D.; project administration: J.D.; funding acquisition: J.D. and R.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China, under grant numbers 61971450 and 61801521, in part by the Natural Science Foundation of Hunan Province, under grant number 2018JJ2533, and in part by the Fundamental Research Funds for the Central South Universities, under grant number 2018gczd014 and 20190038020050.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable

Data Availability Statement: Data are contained within the article.

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

Abbreviations

The following abbreviations are used in this manuscript:

ULMFiT	Universal Language Model Fine-tuning
SVM	Support Vector Machine
SLT	Statistical Learning Theory
NLP	Natural Language Processing
AWD	ASGD Weight-dropped
LSTM	Long Short-term Memory Networks
RNN	Recurrent Neural Network
RBF	Radian Basis Function
DFT	Discriminative Fine-tuning

References

1. Asr, F.T.; Taboada, M. The data challenge in misinformation detection: Source reputation vs. content veracity. In Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), Brussels, Belgium, November 2018; pp. 10–15.
2. Mukherjee, S. Sentiment analysis. In *ML. NET Revealed*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 113–127.
3. Tompkins, J. Disinformation Detection: A review of linguistic feature selection and classification models in news veracity assessments. *arXiv* **2019**, arXiv:1910.12073.
4. Hepburn, J. Universal Language model fine-tuning for patent classification. In Proceedings of the Australasian Language Technology Association Workshop, Dunedin, New Zealand, 11–12 December 2018; pp. 93–96.
5. Katwe, P.; Khamparia, A.; Vittala, K.P.; Srivastava, O. A Comparative Study of Text Classification and Missing Word Prediction Using BERT and ULMFiT. In *Evolutionary Computing and Mobile Sustainable Networks*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 493–502.
6. Shu, K.; Bhattacharjee, A.; Alatawi, F.; Nazer, T.H.; Ding, K.; Karami, M.; Liu, H. Combating disinformation in a social media age. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2020**, *10*, e1385. [CrossRef]
7. Howard, J.; Ruder, S. Universal language model fine-tuning for text classification. *arXiv* **2018**, arXiv:1801.06146.
8. Chauhan, U.A.; Afzal, M.T.; Shahid, A.; Moloud, A.; Basiri, M.E.; Xujuan, Z. A comprehensive analysis of adverb types for mining user sentiments on amazon product reviews. *World Wide Web* **2020**, *23*, 1811–1829. [CrossRef]
9. Liu, B. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*; Cambridge University Press: Cambridge, UK, 2020.
10. Zhao, W.; Peng, H.; Eger, S.; Cambria, E.; Yang, M. Towards scalable and reliable capsule networks for challenging NLP applications. *arXiv* **2019**, arXiv:1906.02829.
11. Georgieva-Trifonova, T.; Duraku, M. Research on N-grams feature selection methods for text classification. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2021; Volume 1031, p. 012048.
12. Chaturvedi, I.; Ong, Y.S.; Tsang, I.W.; Welsch, R.E.; Cambria, E. Learning word dependencies in text by means of a deep recurrent belief network. *Knowl.-Based Syst.* **2016**, *108*, 144–154. [CrossRef]
13. Basiri, M.E.; Kabiri, A. HOMPer: A new hybrid system for opinion mining in the Persian language. *J. Inf. Sci.* **2020**, *46*, 101–117. [CrossRef]
14. Abdar, M.; Basiri, M.E.; Yin, J.; Habibnezhad, M.; Chi, G.; Nemati, S.; Asadi, S. Energy choices in Alaska: Mining people's perception and attitudes from geotagged tweets. *Renew. Sustain. Energy Rev.* **2020**, *124*, 109781. [CrossRef]
15. Cambria, E.; Li, Y.; Xing, F.Z.; Poria, S.; Kwok, K. SenticNet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual, 19–23 October 2020; pp. 105–114.
16. Zhang, L.; Ghosh, R.; Dekhil, M.; Hsu, M.; Liu, B. *Combining Lexicon-Based and Learning-Based Methods for Twitter Sentiment Analysis*; Technical Report HPL-2011; HP Laboratories: Palo Alto, CA, USA, 2011; Volume 89.
17. Sharaf Al-deen, H.S.; Zeng, Z.; Al-sabri, R.; Hekmat, A. An Improved Model for Analyzing Textual Sentiment Based on a Deep Neural Network Using Multi-Head Attention Mechanism. *Appl. Syst. Innov.* **2021**, *4*, 85. [CrossRef]
18. Singh, J.; Singh, G.; Singh, R. Optimization of sentiment analysis using machine learning classifiers. *Hum.-Cent. Comput. Inf. Sci.* **2017**, *7*, 1–12. [CrossRef]
19. Dong, J.; Ding, C.; Mo, J. A low-profile wideband linear-to-circular polarization conversion slot antenna using metasurface. *Materials* **2020**, *13*, 1164. [CrossRef]
20. Jakkula, V. Tutorial on support vector machine (svm). *Sch. EECS Wash. State Univ.* **2006**, *37*, 121–167.
21. Suthaharan, S. Support vector machine. In *Machine Learning Models and Algorithms for Big Data Classification*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 207–235.
22. Pisner, D.A.; Schnyer, D.M. Support vector machine. In *Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 101–121.
23. Hope, T.; Resheff, Y.S.; Lieder, I. *Learning Tensorflow: A Guide to Building Deep Learning Systems*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2017.

24. Tarasov, D. Deep recurrent neural networks for multiple language aspect-based sentiment analysis of user reviews. In Proceedings of the 21st International Conference on Computational Linguistics Dialogue, Sydney, NSW, Australia, July 2015; Volume 2, pp. 53–64.
25. Tai, K.S.; Socher, R.; Manning, C.D. Improved semantic representations from tree-structured long short-term memory networks. *arXiv* **2015**, arXiv:1503.00075.
26. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.Y.; Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1631–1642.
27. Yu, F.; Liu, Q.; Wu, S.; Wang, L.; Tan, T. A Convolutional Approach for Misinformation Identification. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 3901–3907.
28. Czapla, P.; Howard, J.; Kardas, M. Universal language model fine-tuning with subword tokenization for polish. *arXiv* **2018**, arXiv:1810.10222.
29. Zhang, J.; Cui, L.; Fu, Y.; Gouza, F.B. Fake news detection with deep diffusive network model. *arXiv* **2018**, arXiv:1805.08751.
30. Rane, A.; Kumar, A. Sentiment classification system of twitter data for US airline service analysis. In Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 23–27 July 2018; Volume 1, pp. 769–773.
31. Maas, A.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 142–150.
32. Abdul-Mageed, M.; Novak, P.K. Deep Learning for Natural Language Sentiment and Affect. Available online: http://kt.ijs.si/dlsa/2018-09-14-ECML-DLSA-tutorial.pdf (accessed on 14 October 2021).
33. Rathi, M.; Malik, A.; Varshney, D.; Sharma, R.; Mendiratta, S. Sentiment analysis of tweets using machine learning approach. In Proceedings of the 2018 Eleventh International Conference on Contemporary Computing (IC3), Noida, India, 2–4 August 2018; pp. 1–3.
34. Can, E.F.; Ezen-Can, A.; Can, F. Multilingual sentiment analysis: An rnn-based framework for limited data. *arXiv* **2018**, arXiv:1806.04511.
35. Wang, J.; Yu, L.C.; Lai, K.R.; Zhang, X. Dimensional sentiment analysis using a regional CNN-LSTM model. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Berlin, Germany, 7–12 August 2016; pp. 225–230.
36. Singh, R.; Singh, R.; Bhatia, A. Sentiment analysis using Machine Learning technique to predict outbreaks and epidemics. *Int. J. Adv. Sci. Res.* **2018**, *3*, 19–24.
37. Basiri, M.E.; Nemati, S.; Abdar, M.; Cambria, E.; Acharya, U.R. ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis. *Future Gener. Comput. Syst.* **2021**, *115*, 279–294. [CrossRef]
38. Xie, Q.; Dai, Z.; Hovy, E.; Luong, M.T.; Le, Q.V. Unsupervised data augmentation for consistency training. *arXiv* **2019**, arXiv:1904.12848.
39. Benesty, J.; Chen, J.; Huang, Y. Automatic Speech Recognition: A Deep Learning Approach. 2008. Available online: https://www.microsoft.com/en-us/research/publication/automatic-speech-recognition-a-deep-learning-approach/ (accessed on 12 October 2021).
40. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent convolutional neural networks for text classification. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
41. Aldayel, H.K.; Azmi, A.M. Arabic tweets sentiment analysis—A hybrid scheme. *J. Inf. Sci.* **2016**, *42*, 782–797. [CrossRef]
42. Rani, S.; Singh, J. Sentiment analysis of Tweets using support vector machine. *Int. J. Comput. Sci. Mob. Appl.* **2017**, *5*, 83–91.
43. Agarwal, A.; Yadav, A.; Vishwakarma, D.K. Multimodal sentiment analysis via RNN variants. In Proceedings of the 2019 IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering (BCD), Honolulu, HI, USA, 29–31 May 2019; pp. 19–23.