Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 5536560, 10 pages
https://doi.org/10.1155/2021/5536560

Research Article

Evaluation of Sentiment Analysis via Word Embedding and RNN Variants for Amazon Online Reviews

Najla M. Alharbi,[1] Norah S. Alghamdi ,[2] Eman H. Alkhammash,[3] and Jehad F. Al Amri[4]

[1]*King Abdulaziz City for Science and Technology, Riyadh, Saudi Arabia*
[2]*College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, 11671 Riyadh, Saudi Arabia*
[3]*Department of Computer Science, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia*
[4]*Department of Information Technology, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia*

Correspondence should be addressed to Norah S. Alghamdi; nosalghamdi@pnu.edu.sa

Received 30 January 2021; Revised 24 March 2021; Accepted 21 April 2021; Published 8 May 2021

Academic Editor: G. Muhiuddin

Copyright © 2021 Najla M. Alharbi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Consumer feedback is highly valuable in business to assess their performance and is also beneficial to customers as it gives them an idea of what to expect from new products. In this research, the aim is to evaluate different deep learning approaches to accurately predict the opinion of customers based on mobile phone reviews obtained from Amazon.com. The prediction is based on analysing these reviews and categorizing them as positive, negative, or neutral. Different deep learning algorithms have been implemented and evaluated such as simple RNN with its four variants, namely, Long Short-Term Memory Networks (LRNN), Group Long Short-Term Memory Networks (GLRNN), gated recurrent unit (GRNN), and update recurrent unit (UGRNN). All evaluated algorithms are combined with word embedding as feature extraction approach for sentiment analysis including Glove, word2vec, and FastText by Skip-grams. The five different algorithms with the three feature extraction methods are evaluated based on accuracy, recall, precision, and F1-score for both balanced and unbalanced datasets. For the unbalanced dataset, it was found that the GLRNN algorithms with FastText feature extraction scored the highest accuracy of 93.75%. This result achieved the highest accuracy on this dataset when compared with other methods mentioned in the literature. For the balanced dataset, the highest achieved accuracy was 88.39% by the LRNN algorithm.

1. Introduction

Thousands of people leave reviews about products on e-commerce sites (e.g., Amazon and eBay) and opinions about services such as restaurants or tourist attractions (e.g., Trip Advisor, Rotten Tomatoes, and Yelp) and the social media (e.g., Facebook and Twitter) which can mention almost anything. Therefore, sharing the reviews and feedbacks of customers about products or services used online will influence new customers' perspectives towards these products, services, organizations, or institutions. Generally, our behaviours, opinions, and perceptions about our choices are influenced by other experiences and opinions. Especially, when we deal with something new, feedback or reviews are requested from experienced people. Thus, imagine millions of users giving out their experiences through online reviews about products and services; this will profoundly impact other people by either encouraging or discouraging them towards trying these products or services. It is highly important to make sure of the authenticity of the seller. Over time, consumers expand their opinions and feelings on the virtual communities, the social networks, and the social media communities. Classifying and categorizing large amounts of unstructured data from the Internet are becoming more challenging tasks; hence, the sentiment analysis, along with Natural Language Processing (NLP)

techniques are flourishing to carry out such tasks providing analysis for the textual data obtained from reviews or surveys. These techniques predict the polarity of the opinions (positive, negative, or neutral) of assisting customers draw a better conclusion about a product. In this research, the sentiment analysis will be carried out for the reviews of a product (i.e., the unlocked mobile phone) on the website amazon.com. In addition, this analysis will help customers making the right decision to purchase or not. Also, companies are able to understand how their customers feel about their products and their level of satisfaction, to sustain their level of comfort, to maintain their performance, and to improve their services.

1.1. Problem Statement. Nowadays, the use of the Internet on electronic commerce websites has grown to the point where customers rely on them for purchasing and selling [1]. Since these websites allow consumers to write their feedback on different goods and services, huge amounts of reviews have become available [2]. Consequently, the need to analyse these reviews to understand the feedbacks of the consumers has increased for both the vendors and the consumers. However, with many comments, it is difficult to read all feedbacks for a particular item, especially for the popular items [3]. This research will evaluate different sentiment analysis approaches for the dataset of the mobile phone reviews in order to predict consumers' satisfaction for a mobile phone reviews by using deep learning algorithms, including five different RNN models to be evaluated based on their performance. Nevertheless, this will also help buyers to make better decisions when considering the purchases of a specific mobile phone.

The structure of paper is as follows. Section 2 displays the required background of the work. Related work is discussed in Section 3. Section 4 defines the research methodology, and, finally, the implementation and the results are shown in Sections 5 and 6, respectively.

2. Background

Sentiment analysis, which can also be referred to as opinion mining or emotion AI, uses NLP along with the text analysis and the computational linguistics to analytically identify, to study, and to categorize subjective information, leading to stating its impacts. The sentiment analysis has been widely used in fields like data mining, web mining, and social media analysis since sentiments are the essential characteristics to judge human behaviours. The sentiment analysis is the area which deals with judgments, responses, as well as feelings that are generated from texts by an artificial intelligent computer algorithm that mainly depends on machine learning techniques. The polarity can be positive, negative, or just neutral. NLP is profoundly involved with the sentiment analysis which is commonly used to feel the voice of customers from reviews, survey responses, social media, and resources for users coming from different area such as marketing and clinical medicine to improve customer service. These opinions of customers are judgements/statements that reveal people's sentiment or attitudes towards a specific product or service. Moreover, it could be thought of as a detecting subjective opinion of speakers or writers on a specific topic or to recognize the dominant feelings of the text. Sentiment analysis can be thought of as a process that uses specific methods or techniques to detect, extract, and analyse subjective information from its language (in a form of text) in order to have a feel about the customer's satisfaction and to draw a conclusion about the overall experience about services or products. Nevertheless, it sometimes provides an arithmetical score formulating the usefulness of the sentiment. To achieve such thing, a system (algorithm) will be built to collect and to assess opinions given after online purchases.

3. Literature Review

Sentiment analyses are done with the help of the machine learning algorithms to classify the textual data detecting the polarity of the reviews about online products (amazon, IMBD, and any other dataset) using either machine learning methods or deep learning methods or in some cases integrating them. In the field of the sentiment analysis using the machine learning algorithms, the researchers in [4, 5] focused on increasing the accuracy of the review classification. They used the unigram and weighted unigram techniques. The definitions are a unigram is one word, a bigram is a sequence of two words, a trigram is a sequence of three words, and so forth. This was done to train the machine with the help of the classifiers and the applied algorithms such as maximum entropy (ME), naïve bias (NB), and support vector machine (SVM). ME classifiers are under the class of exponential models and are categorised as a probabilistic classifier. ME does not take into consideration the independent elements of each other. It is mainly used to work out large quantities of text classification problems and is widely used in the sentiment analysis. However, NB method is a technique for the constructing classifiers for models that designates a class label to the problem examples. They are represented in a vector form varying on the features of the value being characterised, where these labels come from the finite sets. NB is not a standard algorithm but it depends on the used principles. The value of a feature does not relate to the value of another feature making them independent, given the class variable. On the other hand, SVM algorithms are mostly used for classification problems; however, sometimes it could be used in the regression as well. In this algorithm, each feature of the data is plotted in n-dimensional space; the more features present more dimensions. The plot would have with the value of each feature corresponding to a specific coordinate. The classification is performed with finding the hyperplane that distinguishes between the two classes. On the other hand, [5] utilised the technique of "ensemble" machine learning algorithm that combines the predictions from the output of the classifiers NB and SVM together to produce more accurate predictions compared to an individual model. Dealing with the issue of performance, speed execution and accuracy created a better accurate system in contrast with the old systems.

On the other hand, in the field of sentiment analysis using deep learning approaches, a dataset consisting of 10,662 records was used to produce the sentiment analysis from IMDb dataset [6]. This method combined the deep learning and the unsupervised machine learning which in return gave better results and a better analysis with respect to other existing methods [3]. The deep learning aspect used a Convolutional Neural Network (CNN) classifier, on the other hand, the machine learning aspect of the analysis used the K-mean method, which falls under the category of the unsupervised learning. CNN was used to train and to learn from the data, and after the feature extraction, the K-mean clustering algorithm was used to categorize the reviews into the positive or negative clusters. The results showed that the accuracy was acceptable with a low error rate. It was also shown in the research that K-mean algorithm is more effective for the larger datasets. Paper [7] suggested a method to perform a sentiment analysis and to mine customers' reviews from the large datasets of 400,000 reviews. The research involved two parts; first, word2vec was used to convert reviews into another form as vector representation finding similar features for different aspects, and then a technique called common bag of words (CBOW) along with skip-gram was used to integrate with different machine learning techniques, that is, SVM, NB, logistic regression, and "Random Forest" utilising 10-fold cross validation. The results demonstrated that CBOW along with "Random Forest" (with word2vec representation) gave the most superior results; thus, CBOW performed better than the skip-gram. The result showed only the results of the unbalanced dataset because the classification accuracy for the balanced dataset was unacceptably lower than the unbalanced dataset. However, the best accuracy was "Random Forest" with CBOW equal to 90.6622%. In another analysis [6], the dataset of Electronica Shopping site was used; only reviews in the English language were considered. The algorithm would identify if the review is helpful or not, in other words, a negative or a positive review. The deep learning method was used to build the model for the review identification. CNN and RNN classifiers were used with FastText and n-gram models for the feature extraction. It was demonstrated that RNN gave the higher results with 92.6% accuracy.

Additionally, another research that was done that used the method of deep learning that involved different models such as Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), RNN, and CNN, in addition to a hybrid model (LSTM–CNN) [8]. This was done on an IMDb dataset [6]; word2vec was used to extract the data features from the dataset. It was split into 50% positive reviews and 50% negative reviews. The results showed that the combination of the model (LSTM–CNN) outperformed MLP, CNN, and LSTM. The accuracy of (LSTM–CNN) was 89.2%, while the accuracy of CNN was 87.7%, and the accuracy of MLP and LSTM was 86.74% and 86.64%, respectively. The researchers in [9] used texts crawled content consisting of 15,000 hotel reviews and proposed a method to the enhanced word representation method which implements the contribution of sentiment information into the traditional Term Frequency Inverse Document Frequency algorithm (TFIDF) and generates the weighted word vectors. They utilised the word2vec technology to get the distributed representations of words including CBOW and the skip-gram models. Then, the findings are fed into a bidirectional LSTM (BiLSTM) model; next, they integrated the analysis method with their proposed model such as RNN, CNN, LSTM, and NB. The results reported the higher evaluation percentages that include precision, recall, and the F1-scores. RNN falls under the category of deep learning which is a subset of the machine learning under the umbrella of the artificial intelligence. It is most commonly used in NLP and sometimes in speech recognition. It is designed in such a way to be able to recognize the sequential characteristics from data and the used patterns to make predictions about the next output. The current RNNs are only able to extract the preceding information from a sentence; however, [10] suggested a new architecture called Comprehensive Attention-Recurrent Neural Networks (CA-RNN). This architecture will allow RNN to store preceding information and the local context for any part of the developed sequence. Bidirectional Recurrent Neural Networks (BRNN) was utilised to retrieve the future and past information when a convolutional layer is embedded to capture local information. The convolutional layer is where the filters are applied, and the most important parameters are the number and the size of kernels. The typical RNN was swapped with two new RNN variants called LSTM and GRU and LRNN and GRNN, respectively to maximise the effectiveness of this new architecture, such as excluding human intervention in the training process. Several sentiment analyses were done in this research using different datasets (large movie datasets and Stanford sentiment tree bank SST). The results showed that using the CA-RNN method can substantially improve the classification accuracy when compared to the standard RNN methods and the models achieving competitive performance. The researcher used word2vec and random vectors for representation and encoding the word. As a conclusion, the results demonstrated that the new architecture CA-RNN gave a higher accuracy.

4. Research Methodology

In this section, the used methods and techniques for the classification of mobile phone reviews are discussed and the taken steps during the experiments are clarified. Figure 1 illustrates the phases of this research beginning with the dataset of online reviews until each review is classified into positive, negative, or neutral.

4.1. Preprocessing. The preprocessing methods are used to prepare the data for entering the model and achieving the best outcome. The preprocessing steps included removing Null values, lower cases, spelling corrections, tokenization, stop words, removing punctuation, and lemmatization. Each review in the dataset is labelled based on the review ratings. If the rating is more than three stars, it is labelled as positive, if the rating is equivalent to three stars, it is neutral, and

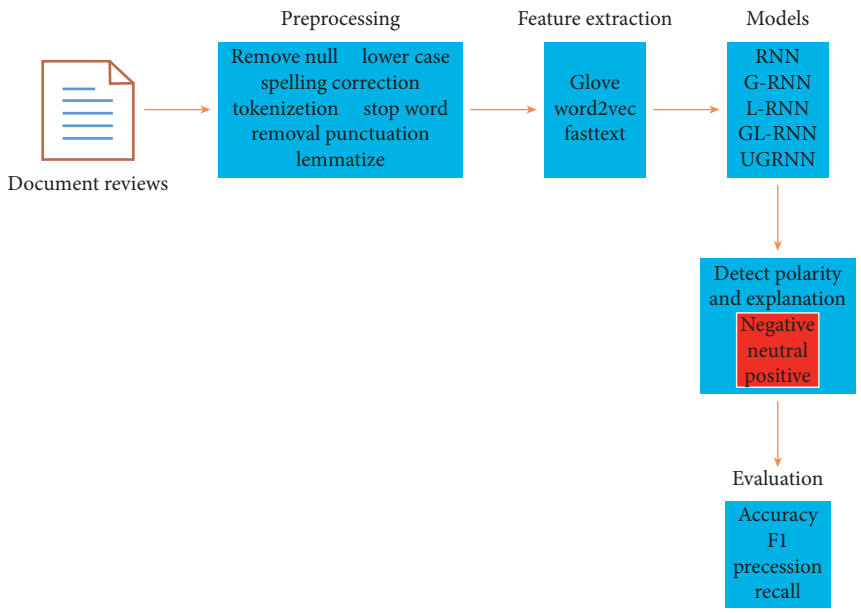

Figure 1: The phases of the proposed approach.

anything less than three stars is considered as negative. Furthermore, the dataset was split into 80% for training and 20% for testing.

4.2. Feature Extraction Word Embedding. There are two main steps involved in the text classification: first, the need to find a word embedding method to convert text into the numerical representations and, second, fitting the numerical representations of the text to the machine learning algorithms or the deep learning architectures for further analysis. However, in this paper, three approaches are deployed, word2vec Embedding, Glove Vector Embedding, and FastText Vector Embedding. The concept of word embedding is converting the text into a vectorized numerical representation. Many machine learning algorithms and virtually all architectures of the deep learning are unable to process strings or plain text in their raw form or to perform any kind of work, such as the classification and the regression. In general terms, they need numbers as inputs. In addition, with the huge amount of data in the text format, it is imperative to extract information from it and to create applications. First, word2vec is a version of the word embedding methods that was introduced in 2013 by Tomas Mikolov, a researcher from Google [11]. It contains vectors of similar words. In other words, it mathematically identifies similarities. word2vec builds vectors that distribute numerical representations of the word features, such as the word context. Additionally, it can make the highly accurate assumptions about the meaning of a word based on the past appearances. Those assumptions can be used to associate a word with other words (e.g., "boy" as "male" and "girl" as "female"). However, the output of the neural network word2vec is a vocabulary in which each object has a vector attached to it that can be fed into a deep learning or simply queried to predict relationships between words. This research applies the skip-gram method, since this method produces more accurate results on the large datasets, with higher dimensions. However, the method of skip-gram which works for the target word is the input used to predict the outputs from the words surrounding the target word based on the context. For example, in the sentence "I have a cute dog," the input would be "cute," and it supposes to have a window size 5 while the output would be "I," "have," and "dog." Second is Glove which is the upbeat word embedding in 2014 from researchers of Stanford's competing [12]. Glove is coined from Global Vectors and it is based on a matrix of factorization techniques on the word-context matrix. Also, this model is a type of the unsupervised learning algorithm that obtains vector representations for the words. First, it constructs a large matrix of cooccurrence knowledge (words × context); that is, for each "word" (the rows), you count how much this word occurs in some "context" (the columns) in a large corpus. Finally, FastText embedding is proposed by Facebook researchers in 2016 as an extension to word2vec [13]. This model is an unsupervised learning algorithm to obtain representations of vectors for words. Facebook makes pretrained models available for 294 languages. So, instead of feeding the single words into the neural network, FastText goes one level deeper. This deeper level consists of the parts of the words and the characters. However, FastText splits words into several subwords (n-grams), for example, the trigrams for the words "products," "pro," "rod," and "duct" (ignoring the beginning and the end of the word boundaries). The word vector embedding for products will be the sum of all these n-grams. After training the neural network, it will have word embeddings for all n-grams given the training dataset. Rare words can now be properly represented since it is highly likely that some of their n-grams also appear in other words.

4.3. Machine Learning Algorithms. In this work, we applied deep learning algorithms such as simple RNN. Furthermore, we apply four different variants of RNN, namely, Long

Short-Term Memory Networks (LRNN), Group Long Short-Term Memory Networks (GLRNN), gated recurrent unit (GRNN), and update recurrent unit (UGRNN). Each algorithm will use the three embedding methods individually and the results will be compared based on the evaluation measures suggested: accuracy, recall, precision, and F1-score in order to find which classification along with the embedding method performed the best. The RNNs are types of neural networks, in which the output from the previous step is fed to the current time stage as data. All knowledge (inputs) and outputs of conventional neural networks are independent of each other. Still, in cases such as when the next word of a sentence is to be predicted, it is important to recognize the previous words. Thus, RNN came into being, with the aid of the hidden layer solved in this problem. Long Short-Term Memory (LSTM) networks is an improved RNN network in which it utilises memory cells to make sure the signal is not lost when the sequenced data is processed. LSTM at the high level of RNN processes the sentences one element at a time and preserves them in what is called the memory state. This resolves the vanishing gradient problem. Moreover, the memory cells take into account the current word, cell state, and carry. This method shines when it deals with series suffering from the time lags of different durations. LSTM uses backpropagation to train the model, and it utilises three main layers: the input layer, the output layer, and the LSTM layer, where the LSTM and the input Layers are connected. Gated recurrent unit (GRUs) and LSTM are very similar in how they work. However, GRUs are considered as the new generation of RRN, are much easier to implement, and are less complex in the structure. GRU removes the concept of cell state and replaces it with the hidden layer to transfer information [14, 15]. The hidden layer output the current time step which is calculated using the hidden layer state of the previous time step and the input of the current time step. GLSTM is simply a group of several LSTMs, where LSTM outputs are concatenated. The idea is to split LSTM into several Sub-LSTMs. LSTM, in general, has a complex gated structure which makes it slow. In contrast, updated gate based RNN (UGRNN) has a single gate (update gate). The update gate is responsible for determining if the hidden state at the current time step is carried out from the previous time step or rather updated. In return, this makes it faster with the enhanced efficiency while maintaining the advantages and characteristics of LSTM. The UGRNN's cell was inspired by the LSTM and GRU algorithms, hence combining them together. The concept behind the update gate (a feed-forward highway network) is to determine whether the unit should be integrated or computed [15, 16].

4.4. Evaluation Parameters. The metrics are used to display the classifier that performs well on this test dataset, so we need to be confident that it has the power to generalize well beyond the data from which it was trained. In this paper, different performance evaluation parameters including precision, F-measure, recall, and accuracy are calculated [17].

Precision measures the classifier's accuracy. It is the percentage of the number of correctly predicted positive reviews divided by the total number of predicted as positive reviews:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{1}$$

Recall measures the classifier's completeness. It is the percentage of correctly predicted positive reviews to the actual number of positive reviews on the corpus. Therefore, recall indicates the number of related items we identified:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2}$$

F-measure (or F-score) is defined as the harmonic mean of precision and recall, which combines recall and precision to output a single score. F-measure therefore might have the best value as 1 and the worst value as 0:

$$\text{F} - \text{Measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3}$$

Accuracy is the most important metrics of performance evaluation and is measured as a percentage of the number of correctly predicted reviews to the total number of reviews present in the corpus. However, the accuracy calculates the ratio of inputs in the test set correctly labelled by the classifier:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{4}$$

5. Implementation

5.1. Dataset. The dataset from Amazon.com is a great source to evaluate products based on customer's reviews. When a customer purchases any product online, they will write a comment explaining their opinion on it under the product and they give it a star rating. The dataset consists of 400 K reviews of unlocked mobile phones sold on Amazon.com which is publicly available on Kaggle website. Solution to the problem would be useful for a brand to gain a broad sense of users' sentiment towards a product through online reviews. It contains 6 columns: (1) brand name which depicts the name of the organization, for example, Nokia; (2) product name, for example, Nokia Asha 302; (3) price, that is, the cost of the mobile; (4) rating, that is, the star rating which the costumer gives to the product; (5) reviews, that is, the users' opinion about each product; (6) review votes, that is, the number of consumers who voted the review. Also, to evaluate the model, we divided the dataset to 80% for training data and 20% for testing the data.

5.2. Data Exploration. Before analysing the data, it should be visualised to help understand it, as this will help to draw better conclusions about the results obtained. At the beginning, the statistics are summarised for this paper, and it is found that the total number of reviews was 400,000 with 162,491 unique values and the total brand names amount to

348,669 brands with 384 unique values. The average price in the dataset was 227 $ and the standard deviation is 273 $. Also, the unique values of products are 4,410. Additionally, the most reviewed product was "The Apple phone 4s 8 GB unlocked mobile" with mean rating 3.82 and its standard deviation is 1.55. Figure 2 illustrates the total number of words. The result shows that the positive had the high content words and the most people typing the positive and the neutral reviews for the products. The average reviews were between negative and neutral, while the number of the reviews has abounded with positive ratings. However, the dataset had overstock in comments at positive and negative.

5.3. Data Preprocessing. Preprocessing is a data mining technique which involves transforming the raw data into a clear and understandable format. Preprocessing is considered a crucial step that must be done before importing data to the machine learning algorithms to maximise the performance and the accuracy of the analysis, especially when the dataset being analysed has a textual nature. Several steps were taken during the processing. First, the empty rows form, and the "Reviews" column was dropped (62 null values). Then, all textual data was concerted to the lower case. Furthermore, the natural language toolkit library was used (NLTK), which is a machine learning library within NLP domain. Figure 3 illustrates the sequence of preprocessing for Amazon dataset of mobile phone reviews.

(1) Spelling corrections are to make sure the analysis yields good results, so spelling mistakes have to be accounted for because sometimes spelling mistakes can change the meaning of the sentence. The spellchecker library was used to detect if a word is misspelt and suggest the most appropriate correction.

(2) Tokenization is one of the most commonly used methods when dealing with text data. It is the procedure of converting sensitive data into tokens. In the case of sentiment analysis, text data is tokenized and filtered to remove any unnecessary tokens.

(3) Stop words are words that are considered useless with respect to the sentiment analysis being performed. These words do not help to find the true meaning of the sentence or review; hence, removing them will not impact the results of the model nor the alter precision or recall of the analysis. However, keeping them would increase the size of the index, which will require higher computational power on very large datasets. Two methods were utilised to remove stop words. The first method, which is the most common, used the NITK library identify tokens containing stop words and removing them from the reviews such as (e.g., a, it, is, that, and but). The second method is used to add a word to NLTK stop words collection that is not included in the library and needs to be removed; we use it for the word that had a frequency more than 50% and was removed and discarded. Examples of these words are unlocked, phone, time, mobile, and so on. Moreover, rare words which appeared less than 6 times are also removed and discarded.

(4) Removing punctuation is to remove marks such as comma, full stop, and exclamation mark.

(5) Lemmatization is to return words to their roots by deleting both prefixes and suffixes. This was done using the NLTK library. Lemmatization helps to link words with similar meaning to one word.

5.4. Features Engineering. After preprocessing, the data is vectorized by various types of the feature selection methods, such as FastText, Glove, word2vec, and word2vec skip-gram. In all used models, the data will be read from word2vec, Glove, and FastText, and the fixed parameters used in the modelling include the following:

(i) (max_features) The maximum number of the feature vector that will be used in modelling is equal to 20,000 rows.

(ii) (embedding_dim) The dimensions of the feature vector from "Glove, word2vec, and fast-text" were all similar and were set to 100, to determine fixed length of the vector for each word.

(iii) (validation_split) The mining of test data equals 20% testing and 80% training.

(iv) (maxlen) "The max review length" indicates the output length of the vector for each review, set to 80.

(v) (batch_size) The size of the training data considered in each epoch equal was set to 32.

(vi) (nb_clasees) The number of classes we are classifying is 5 equal stars rating (1, 2, 3, 4, and 5).

The result from each method is a matrix that represents all documents in the dataset as vectors. The results were obtained by vectors which can be fed to the five suggested algorithms, to build classification models.

5.5. Classification Models

5.5.1. RNN Modelling. We used the Keras library initially in the deep learning. We started to set some parameters for building our RNN model. First, we used the RNN layers with 150 hidden units and then used a dense net with Softmax as an activation function to predict the 3 classes; basically, the activation function Softmax determines the final classes. Here, we have kept the dropout as 0.2. We have passed all the type (Glove, word2vec, and FastText) vectors embedding to the model and passed them as an input to the dense layer.

In mathematics, the Softmax function, also known as softargmax or normalized exponential function, is a function that takes a vector of K real numbers as inputs and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers. That is, prior to applying Softmax, some vector components could be negative or greater than one and might not sum to 1; but after applying Softmax, each component

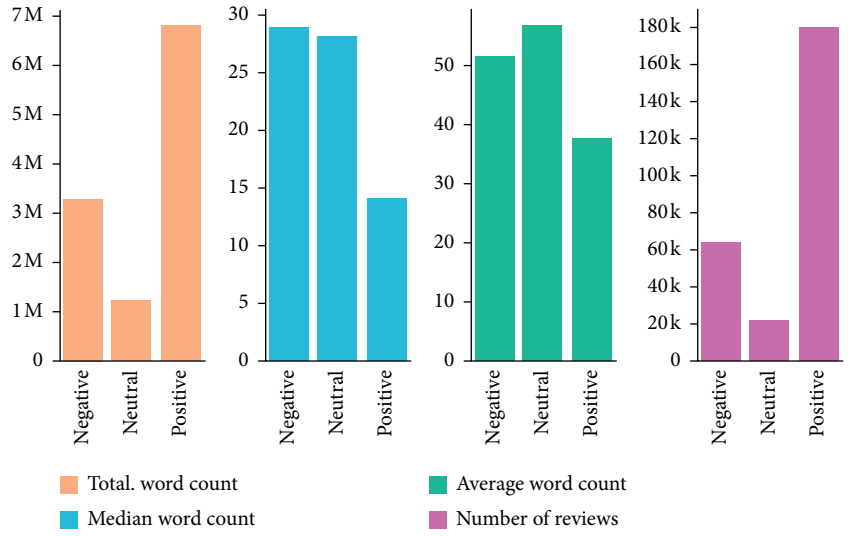

Figure 2: The reviews summary statistics for positive, neutral, and negative rating.

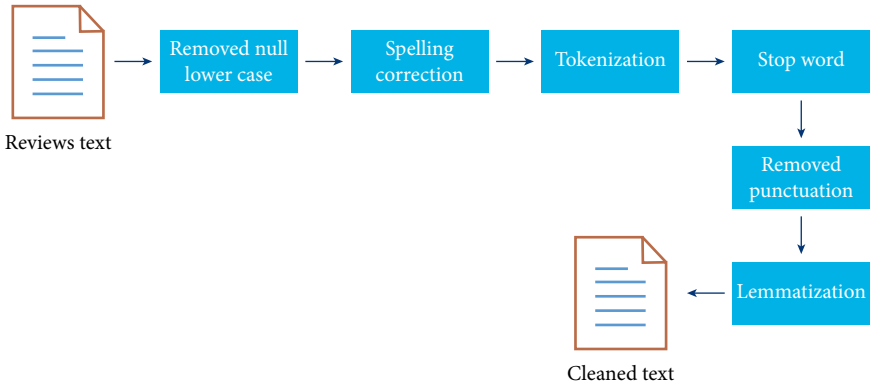

Figure 3: The steps of the preprocessing.

will be in the interval {\display style (0, 1)}(0, 1), and the components will add up to 1, so that they can be interpreted as probabilities. Furthermore, the larger input components will correspond to larger probabilities. Softmax is often used in neural networks, to map the nonnormalized output of a network to a probability distribution over predicted output class. Additionally, we used Adam optimizer to configure the model to determine the distance between the prediction and the actual value. Adam optimizer calculates the learning rate by the distance. If the distance is large, then Adam optimizer will increase the learning rate. Finally, we used 30 epochs to train the data.

5.5.2. LSTM Based RNN (LRNN). In neural networks, Softmax is also used to map a network's nonnormalized output to distribution of probability over predicted output classes. For this method, we have also tried to input the original text with word2vec, Glove, and FastText embedding. The data has been trained for 30 epochs in experiments using LSTM. An epoch is a hyperparameter which is defined before training a model. One epoch is when an entire dataset is passed both forward and backwards through the neural network only once. Adam optimizer has been used to optimize the parameters, the learning rate is 0.01, and the batch size is 32. To prevent overfitting, a dropout rate of 0.2 was set in the LSTM layer with 150 hidden units. Adam is an adaptive method of learning rates. It computes individual learning rates for various parameters. The name comes from adaptive moment estimation, and the reason it is named is that Adam uses first and second gradient estimates to change the learning rate for each neural network weight. In preparation, the amount that the weights are adjusted is referred to as the "learning rate" or step size.

5.5.3. GRU Based RNN (GRNN). In this model, we used the GRU block cell to create a basic GRU with 150 hidden units in this paper, the cell with output size 100 and dropout rate 0.2. The output received is passed to RNN with the sequence length of 62 and 31 for training and testing data, respectively. Finally, we have passed all the (Glove, word2vec, and FastText) vectors embedding to the model, respectively and passed them as an input to the dense layer is applied with ReLu activation function, and the output is passed into the Softmax classifier for the final sentiment classification.

5.5.4. Group LSTM Based RNN (GLRNN). Group LSTM based RNN paper has used Group LSTM cell with output size of 150 and dropout rate of 0.2. A group LSTM cell consists of one LSTM subcell per group, where each subcell operates on an evenly sized subvector of the output. The output received is passed to RNN. Finally, the dense layer is applied with ReLu activation function; ReLU is the most commonly used activation function in neural networks; ReLU is usually a good first choice. The output is passed into the Softmax classifier for the final sentiment classification.

5.5.5. Update Gate Based RNN (UGRNN). GRU-LSTM based RNN has used a network architecture in which both LSTM and GRU cell were used with an output size of 150 and dropout rate of 0.2. This cell is a combination of LSTM and GRU units, wherein there is only one gate. To determine whether the unit should be computing instantaneously or integrating, this is the recurrent idea of the feed-forward highway network. The output received is passed to RNN with the sequence length of 62 and 31 for training and testing data, respectively. Finally, the dense layer is applied with Relu activation function, and the output is passed into the Softmax classifier for the final sentiment classification.

6. Experiments and Evaluation

Firstly, we identified different experiments and discovered the best model for predicting review polarity. These experiments compared feature engineering efficiency; for example, FastText, Glove, and word2vec use skip-gram with each model, before comparing between the models themselves and determining which method had the best performance. The algorithms used in our experiments are RNN, LRNN, GRNN, GLRNN, and UGRNN. However, each review was classified in this paper as positive, negative, or neutral based on the star rating (label). Thus, ratings of four and five stars are classified as positive whereas ratings of two and one stars are classified as negative, and three-star rating is classified as neutral. The first experiment was done on unbalanced data having 4 K reviews. The second analysis was conducted on balanced data. This means that we are solving the problem of unbalanced dataset using a combination of two techniques: undersampling and oversampling. However, the oversampling was used to increase the size of rare samples. Rather than getting rid of abundant samples, new rare samples are generated using repetition (synthetic minority oversampling technique). This was used on neutral reviews which was the lowest class containing only 21,000 reviews. So, we increase the size of neutral to negative reviews by oversampling to achieve 64165 reviews. Also, undersampling was used in order to balance the dataset by reducing the size of the abundant class. This method is used when the quantity of data is sufficient; consequently, we used undersampling with the positive reviews because it was higher than negative and neutral reviews. Note that the number of positive reviews was 18,0686. So, we are undersampling the positive reviews to 64,165 to equal negative. Finally, the balanced dataset consists of positive, negative, and neutral each having 64,165 reviews. The total balanced dataset contains 192,495 reviews.

6.1. Results Obtained by Unbalanced Data. Tables 1–5 represent the results for RNN GRU, LSTM, GLSTM, and UGRNN, respectively, using the three feature extraction techniques. Among the five algorithms used on the unbalanced datasets, it appeared that GL-RNN algorithm had the best performance and GRU-RNN had the worst. For this dataset, it was observed that GL-RNN and LSTM-RNN had similar performances; this can be related to the fact that they have similar architectures. Overall, FastText word embedding feature extraction yielded the highest accuracies when compared with other feature extraction method apart from UG-RNN, while Glove yielded the lowest accuracies in all algorithms. The highest accuracy obtained was FastText feature extraction used on the Group LSTM based RNN with accuracy of 93.75%. The lowest accuracy obtained was Glove feature extraction used with the GRU-RNN algorithm with accuracy of 53.87%. The average accuracies for Glove, word2vec, and FastText feature extraction methods are 75.1%, 82.6%, and 83.7%, respectively. From the average accuracies with respect to the 5 algorithms proposed on the unbalanced data, it is evident that FastText performed the best and Glove.

6.2. Results Obtained by Balanced Data. From the first glance, it is shown that for the balanced dataset, the three feature extraction methods scored similarly with respect to the five algorithms proposed. However, the maximum accuracy obtained was 88.39% which was achieved by the LSTM–RNN algorithm using the FastText feature extraction method. Group LSTM–RNN using FastText feature extraction had a 0.01% difference from LSRM RNN algorithm; again, this can be related to the fact that both algorithms share a similar architecture. Overall, the Glove method had the worst accuracies among four algorithms but scored the highest in use with the GRU–RNN algorithm. The average accuracies for Glove, word2vec, and FastText extraction methods with respect to the five algorithms are 71.4%, 80.1%, and 79%, respectively. In contrast to the unbalanced data, where the average accuracy was the highest for the FastText method, the balanced data had word2vec as the highest average accuracy. Tables 6–10 represent the results for RNN GRU, LSTM, GLSTM, and UGRNN, respectively, using the three feature extraction techniques.

6.3. Benchmarking. To compare our work with some other related work, in this paper, the results are obtained by previous analysis that was conducted in different ways, which was discussed earlier in the literature [Section 2]. Various ways and machine learning algorithms were used on

Table 1: Results obtained from unbalanced data analysis for RNN algorithm by the three feature extraction methods.

	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Word2vec	71.86	67.55	71.86	67.59
Glove	68.99	59.59	68.98	61.62
FastText	75.88	74.47	75.88	74.86

Table 2: Results obtained from unbalanced data analysis for LSTM based RNN algorithm by the three feature extraction methods.

	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Word2vec	92.27	92.21	92.26	91.98
Glove	83.56	82.59	83.56	82.60
FastText	93.63	93.67	93.63	93.47

Table 3: Results obtained from unbalanced data analysis for GRU based RNN algorithm by the three feature extraction methods.

	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Word2vec	65.99	49.89	65.99	56.14
Glove	53.87	29.02	53.87	37.72
FastText	70.04	63.19	70.04	65.93

Table 4: Results obtained from unbalanced data analysis for GLSTM based RNN algorithm by the three feature extraction methods.

	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Word2vec	92.29	92.32	92.29	92.00
Glove	86.57	86.03	86.58	85.92
FastText	93.75	93.86	93.75	93.54

Table 5: Results obtained from unbalanced data analysis for UGRU based RNN algorithm by the three feature extraction methods.

	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Word2vec	90.68	90.58	90.68	90.29
Glove	82.49	81.33	82.49	81.23
FastText	85.39	85.27	85.39	85.08

Table 6: Results obtained from balanced data analysis for RNN algorithm by the three feature extraction methods.

	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Word2vec	77.98	06.08	77.98	11.28
Glove	62.31	58.55	62.31	59.04
FastText	70.16	59.27	70.16	63.27

Table 7: Results obtained from balanced data analysis for LSTM based RNN algorithm by the three feature extraction methods.

	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Word2vec	87.43	87.19	87.43	86.75
Glove	69.23	63.64	69.23	63.35
FastText	88.39	88.25	88.39	87.79

Table 8: Results obtained from Unbalanced Data Analysis for GRU BASED RNN algorithm by the three feature extraction methods.

	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Word2vec	61.04	66.69	61.04	58.33
Glove	66.86	71.08	66.86	67.99
FastText	61.04	66.69	61.04	58.33

Table 9: Results obtained from balanced data analysis for GLSTM based RNN algorithm by the three feature extraction methods.

	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Word2vec	87.38	87.57	87.38	86.58
Glove	80.67	80.45	80.67	79.02
FastText	88.38	88.45	88.38	87.79

Table 10: Results obtained from balanced data analysis for UGRNN based RNN algorithm by the three feature extraction methods.

	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Word2vec	86.61	86.48	86.61	85.80
Glove	77.94	78.03	77.94	75.20
FastText	87.25	87.26	87.25	86.57

the same dataset used in this research. Out of the three researches done, it appears that FastText feature extraction used with the group LSTM based RNN method yielded the highest accuracy of 93.75%. The work in [10] was able to achieve an accuracy of 92.73% using Glove feature extraction with CNN algorithm; however, Glove feature extraction gave the worst results when used under the RNN algorithms. The lowest accuracy was achieved by [4] with weighted unigrams feature extraction under the SVM algorithm with accuracy of 81.2%. On the other hand, the work in [7] was able to achieve an accuracy of 90.7% using continues bag of words feature extraction method under the random forest algorithm. It is evident that the best results were achieved by the method proposed, FastText feature extraction under GLRNN.

7. Conclusion

This paper aimed to evaluate different deep learning models to predict the polarity of textual reviews of mobile phones from Amazon.com. Five different variations of RNN algorithms were used, RNN, LSTM-RNN, GLSTM-RNN, GRU-RNN, and UG-RNN, and then compared concerning three-word embedding feature extraction methods: Glove, word2vec, and FastText. Word embedding plays a crucial role in text classification by transforming text into vectorized numerical representations which allows us to use it as an input to the machine learning algorithm.

The most challenging part is text classification, since the meaning of words must be understood while taking ambiguity of the human language into account. The data was visualised to gain a better understanding; then after that, it was processed and prepared to use as input to the five different RNN

algorithms. However, the data was processed further to become a balanced dataset to solve the problem of having too many positive reviews compared to neutral and negative. Furthermore, previous research focused on unbalanced datasets and managed to achieve good results. Out of the five algorithms, GLSTM based RNN with the FastText feature extraction was able to yield the best results when the evaluation is in terms of the accuracy, precision, recall, and F1-score, for the unbalanced datasets with the accuracy of 93.75%, while LSTM–RNN also with FastText feature extraction method yielded the best results for the balanced dataset with accuracy of 88.39%. A conclusion can be drawn here that they scored the highest since they share similar architecture; however, in the balanced dataset, the GLSTM and LSTM-RNN algorithms had a 0.01% difference in accuracy making them very similar for such analysis. The unbalanced dataset yielded better results probably due to its larger size. The results were then compared to previous attempts in the literature; as a conclusion, the aim of this paper was successfully fulfilled and all objectives were met; nevertheless, a better model was made that surpasses the results and reliability of previous attempts where the maximum accuracy that was achieved was 92.75% using the Glove feature extraction with CNN algorithm. On a side note, the Glove feature extraction yielded the lowest results when used with the RNN algorithm. This shows that a feature extraction method or a machine learning cannot be judged individually in terms of sentiment analysis; thus, the algorithm has to be tailored for the needs, input, and output of the algorithm.

Data Availability

The dataset from Amazon.com (public data) is a great source to evaluate products based on customer's reviews. When a customer purchases any product online, they will write a comment explaining their opinion on it under the product and give it a star rating. The dataset consists of 400 K reviews of unlocked mobile phones sold on Amazon.com which is publicly available on Kaggle website. Solution to the problem would be useful for a brand to gain a broad sense of users' sentiment towards a product through online reviews. It contains 6 columns which are Brand name, product name, price, rating, reviews, and review votes. Moreover, we use 80% of the data as data training and 20% as data testing.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Taif University Researchers Supporting Project (TURSP-2020/211), Taif University, Taif, Saudi Arabia.

References

[1] K. Korovkinas, P. Danėnas, and G. Garšva, "SVM and k-means hybrid method for textual data sentiment analysis," *Baltic Journal of Modern Computing*, vol. 7, no. 1, pp. 47–60, 2019.

[2] H. M. Kumar, B. S. Harish, and H. K. Darshan, "Sentiment analysis on IMDb movie reviews using hybrid feature extraction method," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 5, no. 5, 2019.

[3] B. S. Lakshmi, P. S. Raj, and R. R. Vikram, "Sentiment analysis using deep learning technique CNN with KMeans," *International Journal of Pure and Applied Mathematics*, vol. 114, no. 11 2, pp. 47–57, 2017.

[4] A. S. Rathor, A. Agarwal, and P. Dimri, "Comparative study of machine learning approaches for Amazon reviews," *Procedia Computer Science*, vol. 132, pp. 1552–1561, 2018.

[5] J. Sadhasivam and R. B. Kalivaradhan, "Sentiment analysis of amazon products using ensemble machine learning algorithm," *International Journal of Mathematical, Engineering and Management Sciences*, vol. 4, no. 2, pp. 508–520, 2019.

[6] K. Q. Anh, Y. Nagai, and L. M. Nguyen, "Extracting customer reviews from online shopping and its perspective on product design," *Vietnam Journal of Computer Science*, vol. 6, no. 1, pp. 43–56, 2019.

[7] B. Bansal and S. Srivastava, "Sentiment classification of online consumer reviews using word vector representations," *Procedia Computer Science*, vol. 132, pp. 1147–1153, 2018.

[8] N. M. Ali, A. El Hamid, M. Mostafa, and A. Youssif, "Sentiment analysis for movies reviews dataset using deep learning models," *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, vol. 9, no. 2/3, 2019.

[9] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on BiLSTM," *IEEE Access*, vol. 7, pp. 51522–51532, 2019.

[10] Y. Zhang, M. J. Er, R. Venkatesan, N. Wang, and M. Pratama, "Sentiment classification using comprehensive attention recurrent models," in *Proceedings of 2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 1562–1569, Vancouver, BC, Canada, July 2016.

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

[12] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, 2014 October.

[13] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[14] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proceedings of 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 324–328, IEEE, Wuhan, China, 2016 November.

[15] J. Collins, J. Sohl-Dickstein, and D. Sussillo, "Capacity and trainability in recurrent neural networks," in *Proceedings of ICLR 2017*, Toulon, France, April 2017.

[16] J. Qu, X. Gu, and L. Zhang, "Improved UGRNN for short-term traffic flow prediction with multi-feature sequence inputs," in *Proceedings of 2018 International Conference on Information Networking (ICOIN)*, pp. 13–17, Chiang Mai, Thailand, January 2018.

[17] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentimental reviews using machine learning techniques," *Procedia Computer Science*, vol. 57, pp. 821–829, 2015.