A Comparison of Sentiment Analysis Methods on Amazon Reviews of Mobile Phones

Sara Ashour Aljuhani[1], Norah Saleh Alghamdi[2]
School of Computing, Dublin City University (DCU)[1]
Dublin, Ireland
Department of Computer Science[1,2]
Princess Nourah bint Abdulrahman University (PNU)
Riyadh, KSA

***Abstract*—The consumer reviews serve as feedback for businesses in terms of performance, product quality, and consumer service. In this research, we predict consumer opinion based on mobile phone reviews, in addition to providing an analysis of the most important factors behind reviews being classified as either positive, negative, or neutral. This insight could help companies improve their products as well as helping potential buyers to make the right decision. The research presented in this paper was carried out as follows: the data was pre-processed, before being converted from text to vector representation using a range of feature extraction techniques such as bag-of-words, TF-IDF, Glove, and word2vec. We study the performance of different machine learning algorithms, such as logistic regression, stochastic gradient descent, naive Bayes and convolutional neural networks. In addition, we evaluate our models using accuracy, F1-score, precision, recall and log loss function. Moreover, we apply Lime technique to provide analytical reasons for the reviews being classified as either positive, negative or neutral. Our experiments revealed that convolutional neural network with word2vec as a feature extraction technique provides the best results for both the unbalanced and balanced versions of the dataset.**

***Keywords*—Bag-of-words; TF-IDF; glove; word2vec; logistic regression; stochastic gradient descent; naive bayes; Convolutional Neural Network; log loss; lime**

I. Introduction

Purchasing a product is an interaction between two entities, consumers and business owners [1]. Consumers often use reviews to make decisions about what products to buy, while businesses, on the other hand, not only want to sell their products but also want to receive feedback in terms of consumer reviews. Consumers reviews about purchased products shared on the internet have great impact [2]. Human nature is generally structured to make decisions based on analysing and getting the benefit of other consumer experience and opinions because others often have a great influence on our beliefs, behaviours, perception of reality, and the choices we make [3]. Hence, we ask others for their feedback whenever we are deciding on doing something. Additionally, this fact applies not only to consumers but also to organizations and institutions. In the last few years, consumer ways of expressing their opinions and feelings have changed according to changes in social networks, virtual communities and other social media communities [4]. Discovering large amounts of data from unstructured data on the web has become an important challenge due to its importance in different areas of life [5]. To allow better information extraction from the plethora of data available, sentiment analysis has emerged to be able to predict the polarity (positive, negative, neutral) of consumer opinion [6]. This in turn would help consumers to better analyse the textual data providing useful information. We study in this research sentiment analysis of mobile phone reviews taken from the Amazon [1] website, and how these reviews help consumers to have confidence that they have made the right decision about their purchases. Also, the research in this work aims to help companies understand their consumers' feedback to maintain their products/services or enhance them. In addition, giving them insights about them in providing offers on specific products to increase their profits and customer satisfaction.

A. Problem Statement

Recently, electronic commerce websites use of the Internet has increased to the point that consumers rely on them for buying and selling [7]. Since these websites give consumers the ability to write comments about different products and services, huge amounts of reviews have became available [8]. Consequently, the need for to analyse those reviews to understand consumers' feedbacks has increased for both vendors and consumers. However, it is difficult to read all the feedbacks for a particular item especially for the popular items with many comments [9]. In this research, we attempt to build a predictor for consumers' satisfaction on a mobile phone product based on the reviews. We will also attempt to understand the factors that contribute to classifying reviews as positive, negative or neutral (based on important or most frequent words). This is believed to help companies improve their products and also help potential buyers make better decisions when buying products.

This paper is structured as follows, Section 2 discusses the required background of the work. The related work in the previous literature is discussed in Section 3. Section 4 and Section 5 explain both methodology and implementation respectively. Section 6 reports the experimental results of various settings while Section 7 discuss the limitations that could be leveraged as future directions. Lastly, Section 8 concludes the findings of the paper.

II. Background

Sentiment analysis involves a combination of natural language processing, computational linguistics and textual analy-

[1] https://www.amazon.com/

sis in order to detect positive, negative or neutral feelings about the subject of the text [10]. It is used in different areas such as marketing, customer services, and amongst others. Sentiment analysis can be performed on both document-level or sentence-level depending on the unit of information being considered. In this project, sentence-level was considered [11]. Sentiment analysis has several applications in different areas including advertisement where sentiment analysis contributes in selecting specific advertisements to be shown on commercial and social media channels according to particular users opinions on particular products [12]. Sentiment analysis can also be utilized for opinion retrieval, i.e. build search systems to search for specific views on specific topics [13].

III. Related Work

Since this work is interested in studying the sentiments of mobile phones reviews on Amazon, the work related to analysing the sentiments of mobile phones or Amazon reviews have been considered in the review. In the following, these researches are reviewed in terms of pre-processing techniques, feature extraction methods, proposed methodologies, and evaluation metrics.

Various work in the literature have focused on the problem of identifying users opinions of different products using Amazon reviews of "Unlocked Mobile Phones" [14], [15]. The work by [14] focused on a specific Brand Name, 'iPhone', to examine algorithms' validity in order to classify online reviews using a supervised model. On the other hand, [15] aggregated 400,000 reviews from various Brand Names. They did their experiments on two steps. First, they used balanced data which means that the number of negative reviews (1 and 2 star) is equal to the number of positive reviews (4 and 5 star), and they removed neutral reviews. Second, while using unbalanced data, they considered (1 and 2 star) as negative reviews and (3, 4, 5) as positive reviews. At the pre-processing phase, [14] did not take emoticon expressions into consideration, they rather focused on reviewers' IDs, they assumed they must be unique and any duplications were eliminated. Also, names with @ sign and blank spaces were rejected. In addition, they applied feature reduction using a filter to remove stop words such as a, an, the, etc. After this reduction, the authors observed that the text was reduced by 8.68%. Moreover, after preprocessing step, the dataset contained 9,500 positive reviews and 9,500 negative reviews, and 2,500 neutral. On the other hand, [15] Used spaCy library to clean the data. They made stemming to utilize only the words roots. They also removed stop words to reduce the number of words, converted the words to lowercase, and removed both punctuation and whitespace. Unigram and weighted unigram were used in [14] as features, and the authors eventually concluded that weighted unigram gave the best results. On the other hand, word2vec [16], CBOW [17] and skip-gram [18] models were used to represent features. In terms of machine learning algorithms utilized, both [15] and [14] used naïve Bayes (NB) [19], [20] and support vector machine (SVM) [21], [22], but [15] added more algorithms such as logistic regression (LR) [23] and random forest (RF) [[24]. In terms of results, [14] achieved the highest accuracy (81.20%) when using SVM and weighted unigram as features. On the other hand, random forest (RF) scored the highest accuracy (90.66%) when used with CBOW as features as reported in the experimental results by [15]. Other authors focused on sentimental analysis of mobile phones reviews from different sources with different languages such as Chinese. The dataset used by [25] was obtained from jeng dong website which is specialized on mobile phone reviews. The authors collected a group of labelled data that consisted of 1,500 positive reviews and 1,500 negative reviews. They also collected real mobile application review. In a similar work proposed by [26], the dataset was gathered from 'we chat' over a span of three years. In addition, the review was scored from one to five where 1 and 2 were considered negative, 3 is considered neutral, while 4 and 5 were considered positive in terms of polarity. At the end, the dataset contained 109,901 positive reviews, 23,654 negative reviews, and 11,688 neutral reviews with percentages of 75.6%, 16.28%, 8.05% respectively. Also, the authors compared between various types of properties about mobile application reviews that made difference between mobile application reviews and PC reviews. Additionally, they used spare of length property which deals with the min and max size of chart. The minimum size is 1 chines chart while maximum size is 6,000 chines chart. Moreover, on short average length, the authors said that while they were reviewing the statistical features, they found that the average chines chart in mobile reviews was 17 while in micro blog it was 45 word. They started with feature selections in many approaches before establishing the algorithms. [25] mentioned variety of N grams. First, character ngram which is based on character sequence. Second, word ngram which considered words sequences. Third, POS (part-of-speech) ngram which considered part-of-speech types sequences. The authors discussed three types of n-POS-gram: i) Noun ngram, ii) a combination of noun and verb ngram, and finally, iii) a mixture of noun, adverb and adjective ngram. In their work, the authors focused on both n-char-gram and n-POS-grams. They developed a feature selection to document frequency method. After that, they used boolean weighted method (TF&IDF) [27] to calculate feature weight. On the other hand, [26] used word count in reviews to make sure there is no repetition within the same review and ngrams were used for features representation. In terms of the utilized machine learning algorithms, in their work [25] applied LIBSVM and SVM algorithms to analyse the sentiment polarity of the review. The result of this paper viewed ngrams using English language limited with one or two words. Yet in Chinese, they used ngram with higher values of n for more accurate results. A high performance was obtained when using 4-grams as reported in their results. In their conclusion, the authors reported that integrating noun, adverb and adjective ngram yields the best results. On the other hand, the authors of [26] showed that using SVM leads to more accurate results to identify positive reviews, and using naïve Bayes is more accurate with negative reviews. Furthermore, the best performance was obtained by using bigrams.

IV. Methodology

In this section, we will present the methodology and techniques used in classifying mobile phone reviews that are adopted by most of the researchers in the field of sentiment analysis. Firstly, we will explain the steps followed during the experiments. Fig. 1 illustrates the phases of this work starting with the reviews dataset till the classification of each review into positive, negative, and neutral.

A. Preprocessing

In the preprocessing step, the reviews were tokenised, spelling mistakes were checked for, and all words were lower-cased. Also, stop-words such as "a, an, with etc." were removed from the data. The tokens were returned to their roots by performing lemmatisation on all tokens. Each review in the dataset was labelled as positive, negative, or neutral based on its star rating in the same way adopted by [14]. The dataset was then split into 70% for training, 15% for development, and 15% for testing.

B. Feature Extraction

The employed dataset is textual, so it needs to be represented in numerical formats to be fed to the machine learning algorithms to build the desired classifiers. To achieve this, different vectorisation techniques are performed including term frequency which involves counting all the occurrences of all the terms in the document or sentence. A term can be expressed as a single word i.e. unigram, or any arbitrary number of words, namely, n-grams [28]. Fig. 2 illustrates how unigrams, bigrams, and trigrams can be formed from a sentence. Term frequency or count vectoriser (BOW) method suffer from a major pitfall, as it takes into account all the terms without taking into consideration the fact that some terms are very frequent in the corpus. Those terms do not capture document specific information since they occur in the majority of the documents. Such a drawback can be tackled by defining a maximum threshold for document frequency. However, the tuning of this threshold can be tricky, therefore, term frequency- inverse document frequency (TF-IDF) [27] is introduced. TF-IDF is a weighting scheme that works by giving low weight to the terms that occur frequently in the given corpus. Inverse document frequency (IDF) is the inverse of the number of times a specific term appeared in the entire corpus. It captures how a particular term is document specific, and when multiplied by term frequency (TF) the result should give a measure of how this term is of particular importance to the document at hand. Equation (1) demonstrates the main formula used for computing the TF-IDF for each term in each document.

$$TF - IDF = TF_{wd} * IDF_W \quad (1)$$

Although TF, and TF-IDF are popular feature representation techniques in various natural language processing tasks [27], they define the vocabulary over a given corpus as a set of unique words, ignoring the semantic and syntactic similarities between those words. For example, in both TF and TF-IDF extraction techniques, the words pretty and beautiful are represented as two different words although they are nearly synonyms. Therefore, distributed words representations, namely word-embeddings were introduced as an alternative features extraction technique [29]. Word-embeddings are extracted from huge corpora using different algorithms including deep learning algorithms [30]. The main idea behind word embeddings is to convert each word to a mathematical vector. In addition, each word will be represented by a vector, words with similar meaning have similar representations and this word is represented as positive and negative decimal number [31]. For example, the representation of Man = [1.0 2.9 0.9 -38 . . .]. Therefore, we can find from this vector the similarity

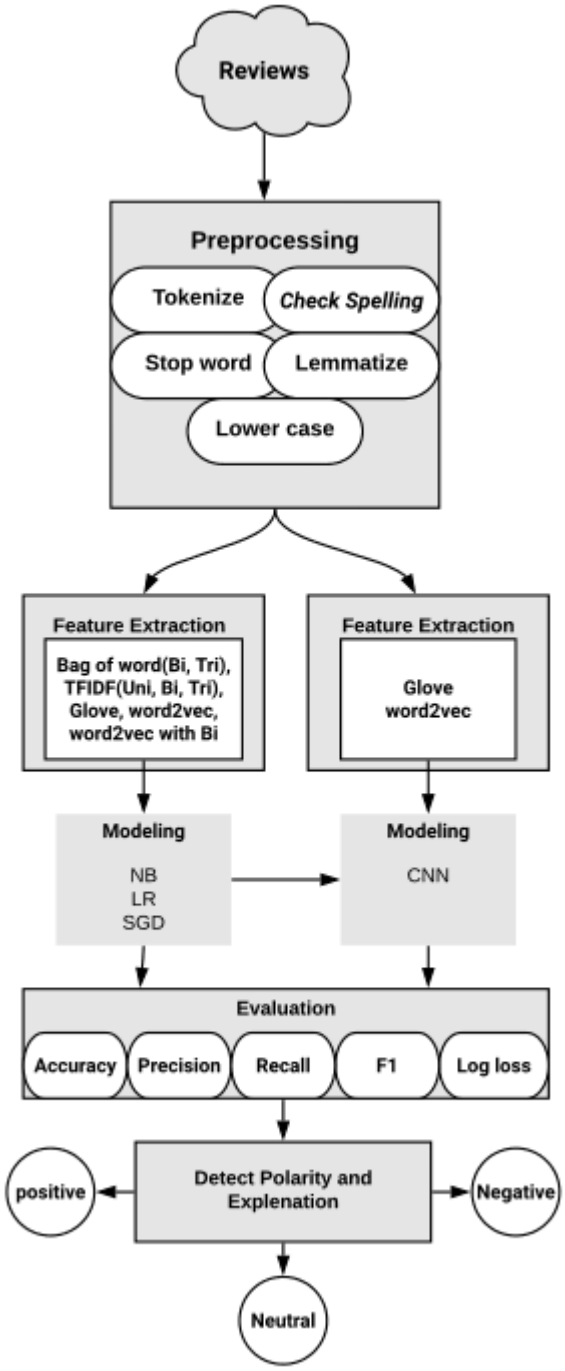

Fig. 1. Phases of the experiments of Amazon website dataset of mobile phone reviews sentiment analysis.

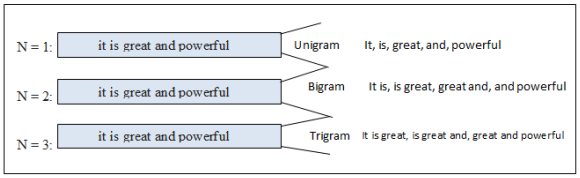

Fig. 2. Unigrams, bigrams, and n-grams extraction from a given sentence

between the words using mathematical equations. To illustrate, when word-embeddings are used to express the words as numerical vectors and different mathematical operations are performed on them as follows: Man + King – Woman = Queen. Egypt + Cairo – Saudi Arabia = Riyadh. Given the two examples above it is illustrated that the learned embeddings capture the information that a king is a man and a queen is a woman. Also that Cairo is the capital city of Egypt and Riyadh is its equivalent for Saudi Arabia. Two different types of word embeddings were included in this study, global vector for word representation (Glove) [32] and word2vec [33][34].

C. Machine Learning Algorithms

In this work, different machine learning algorithms were applied to build a prediction model to assign a polarity for a given review. Logistic regression (LR), naive Bayes (NB), Stochastic gradient descent (SGD) [35], and convolutional neural network (CNN) [36] were experimented. Logistic regression (LR), assigns a probability to each class given an input vector it is originally a binary classification algorithm that can be extended to perform multi-class classification. Naive Bayes is a probabilistic classification algorithm that utilizes the properties of Bayes theorem hypothesis relationship between independent variables [37].There are three types of naive Bayes classifiers: Gaussian, Bernoulli, and multinomial. Gaussian naive Bayes is used when the dataset is continuous, Bernoulli when the dataset is binary, and multinomial with count data. Bernoulli and multinomial naive Bayes are applied in text data classification [38]. Support vector machine (SVM) is a kernel based method, that attempts to find the optimal decision boundary by transforming non-linearly separable data samples to a higher dimension space where there exists a separation hyperplane [21]. Stochastic gradient descent (SGD) [39],[40] is a powerful technique applied to increase the speed and classification capability of SVM and LR. Therefore, it can be effectively applied to large datasets. Also, it works well with text classification and natural language processing. Stochastic gradient descent (SGD) classifier takes as input the sample before predicting the next value, and compares it with the actual value. In addition, it contains a loss function to measure the distance between the predictive and actual value. If the distance is high, then gradient descent (GD) changes the weight of each feature then compares it against each iteration until it reaches to a more similar value to the actual value. If the type of loss function is equal to 'hinge', that means it is used to optimise an SVM, while if loss is equal to 'log' then it is used to optimise an LR model. During the step of changing the weight of the feature, over-fitting problems may arise. So, the classifier compares the prediction value between training and development data. If the value of train is increasing and development is decreasing then an early stop function stops changing the weight. Convolutional neural network (CNN) [41] is a deep learning method which is effective for analysing images and text with huge data volumes. CNN is a supervised algorithm, so it needs labelled data to advance the weights of its convolutional filters. In addition, it receives the data from feature extraction as input then sends it to hidden layers called convolutional layers. These layers are the basis of CNNs. The first layer transforms the input, then the output from this layer sends it to the other layer. It is a sequence until the last layer. This process is called convolutional operation. Moreover, each layer contains filters to detect the pattern. The size of the filter is determined at the beginning to monitor the algorithm and observe how it learns. The size of the filter is determined based on how many characteristics are needed to be detected. If it is large, it means that the size of the filter needs to be increased. Since machine learning models work as a black box, as they take the input, do some processing, then give the output. Lime (Locally interpretable models and effects) [42] is an incredible tool to clarify what classifiers predict. It works by making a line, separating the features and then seeing the strongest feature which are near to the line. Also, Lime adjusts a solitary input test by tweaking the element esteems and watching the subsequent effect on the output. In addition, the output from Lime is a list of interpretabilities. For example, In the medical domain where a patient goes to the doctor and the doctor enters the symptoms to the model, then the model predicts that the patient has flu based on some symptoms such as sneeze, weight, headache, feeling fatigue, age, etc. So, Lime explains why the model predicted flu and gives the reasons, which are sneeze and headache.

D. Evaluation Parameters

In this work, the metrics used to test the performance of machine learning classifier are: accuracy, precision, recall, and F1-score [38]. Precision measures the percentage of positive reviews that predict truly divided by the total number of reviews that are classified positively as defined by Equation 2.

$$PR = \frac{tp}{tp + fp} \tag{2}$$

Recall on the other hand measures the percentage of the reviews that classify positively divided by the total number of reviews which are truly positive, as in Equation 3

$$RC = \frac{tp}{tp + fn} \tag{3}$$

F1-score combines both precision and recall as in Equation 4.

$$F1 - score = 2 * \frac{PR * RC}{PR + RC} \tag{4}$$

Lastly, accuracy is defined as the percentage of reviews that are classified correctly divided by the total number of reviews, Equation 5. Where tp, tn, fp, and fn are true positives, true negatives, false positives, and false negatives respectively.

$$ACC = \frac{tp + tn}{tp + tn + fp + fn} \tag{5}$$

Log loss[2] is also used to measure the performance of machine learning algorithms where the forecast input is a probability estimate somewhere in the range between 0 and 1. The objective of our models is to reduce the value of log loss. Therefore, the model performance can judge based on the log loss value, if the result is equal to 1 this means that the model is predicting value far from the actual value and it is not a good model. On the other hand, the model that provides values equal or near to zero is a better model. Moreover, it considers the vulnerability of your forecast dependent on the amount it fluctuates from the actual label. This gives us a more nuanced look into the execution of our model.

[2] scikit-learn.org/stable/modules/generated/sklearn.metrics.log_loss.html

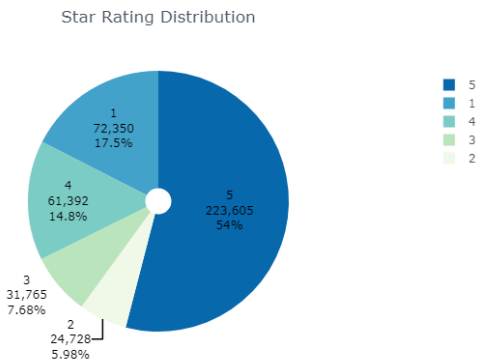

Fig. 3. Illustration of star rating distribution for Amazon website unbalanced dataset of mobile phone reviews.

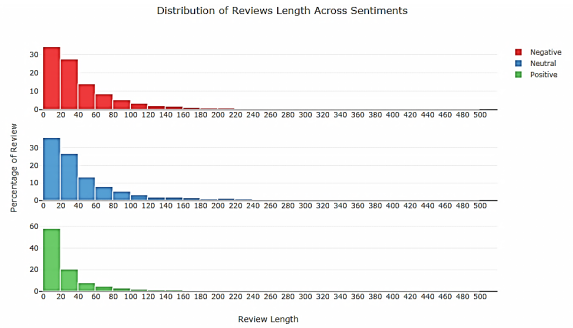

Fig. 4. The length of the review and the sentiment of review for Amazon website unbalanced dataset of mobile phone reviews.

V. Implementation

A. Data Collection

The dataset used for this research was obtained from Kaggle[3]. The data collected from Amazon it is about unlocked mobile phones. It consists of 400,000 reviews and it contains 6 columns: 1) Brand Name: it depicts the name of the organization, e.g. Nokia; 2) Product Name: e.g. Nokia Asha 302; 3) Price: the cost of the mobile; 4) Rating: star rating which the costumer gives to the product; 5) Reviews: the users opinion about each product; 6) Review Votes: the Number of consumers who voted the review. Moreover, to evaluate the model we divided the dataset to 70% for training data, 15% for developing data, and 15% for testing the data.

B. Data Exploration

First, we want to examine the numbers of reviews each star rating contains. To represent this relationship, we used a 'pie chart' or a 'circle chart' diagram. Fig. 3 demonstrates the distribution. In this representation we used plotly and cufflink library. From the 'pie chart' of star ratings among the reviews, we notice that 54% of consumers gave 5, 7.68% gave 3, 17.5% gave 1, 14.4% gave 4, and very few gave a 2-star rating (5.98%).

We studied the distribution of Number of Reviews and the Brand Name. We concluded that Samsung received the highest number of reviews with 57,35k, while Blu and Apple got corresponding number of reviews 50,06K. The reviews are lower for LG. On the other hand, Sony, Posh mobile, and huawei acquired the least reviews. The rest of the brands obtained an average number of reviewers between 19K to 10K.

We also study the relationship between Review Length and Sentiment (Positive, Negative, Neutral). Fig. 4 shows that, negative reviews are longer in general. This is likely to be caused by consumers tending to elaborate in writing to express their feelings when they become angry from a product and tend to write less when they are happy. Moreover, from visualization we observed the distribution of the data is not normal. We evaluated the normality of the data (positive, negative, neutral) using Shapiro test, and we found all classes do not follow normal distribution. So, we applied non-parametric test (Wilcoxon Rank Sum Test) from SciPy library to calculate the medians of the distributions, compare each two classes together and see if it is different than each other or not. Finally, we found 'p value' between (Negative, Neutral) classes = 0.0037, (Negative, Neutral) and (Neutral, Positive) = 0, that means the p value$<$0.05. which clarifies it is a statistically significant variable to identify the polarity.

[3]https://www.kaggle.com/PromptCloudHQ/amazon-reviews-unlocked-mobile-phones

C. Data Preprocessing

Each dataset needs to be prepared before entering it to the machine learning algorithms in order to achieve a high accuracy. Since we deal with textual dataset, it requires appropriate pre-processing. We carried out several steps to clean the data. First, we converted all "Brand Names" to lower case e.g. Samsung is written as "samsung". Second, we dropped the null value in "Reviews" (62 null values). Third, we replaced any "Asus computers" Brand Name to "Asus", "Lg electronics" to "Lg". After that, we used spaCy library, which is a machine learning library. spaCy is very powerful library in the domain of Natural language processing (NLP).

1) Tokenization: the purpose of Tokenization is to split the sentence into separate words based on white space. Each word is called a token. Fig. 5 show the review before and after tokenization.
2) Removing stop words: this involves cleaning stop words (e.g. a, the, about, etc.) that do not add meaning to reviews. There is another kind of stop word (e.g. cell phone, mobile, etc.) which is not built in to the library, specific for the dataset. We removed these special stop words because they are repeated a lot in the corpus, i.e. more than 50%. In addition, the words that repeated less than 4 times in the corpus.
3) Lemmatization of a word: this means returning words to their roots by eliminating all prefixes and suffixes. Fig. 6 illustrates before and after removing stop words and lemmetizing steps. As we can see the words and, the, would, not were removed as they are stop words. Moreover, the word dies is returned to its root die by removing 's' letter.
4) Lower casing: in this step we converted words from upper case to lower case.
5) Punctuation and special characters elimination: Such as coma, full stop, exclamation mark, etc.

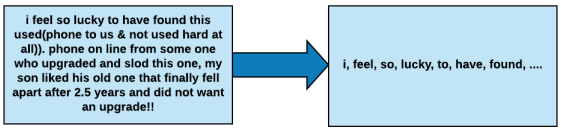

Fig. 5. Tokenization for Amazon website unbalanced dataset of mobile phone reviews.

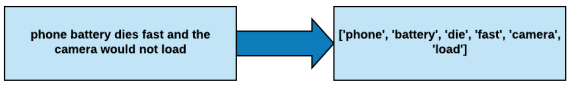

Fig. 6. Lemmatization for Amazon website unbalanced dataset of mobile phone reviews.

D. Features Engineering

In this step we take the "Ready Document" from preprocessing, then vectorize them using the following methods:

- Term frequancy.
- TF-IDF.
- Glove.
- Word2vec.

The result from each method is a matrix that represents all documents in the dataset as vectors. These vectors can be fed to the machine learning algorithms to build classification models.

E. Classification Models

Using all four types of features mentioned in the previous section, LR models were trained. Using unigrams, bigrams, and trigrams for both TF and TF-IDF. NB and SGD models were also trained using all four types of features. Lastly, CNN was trained using documents represented using only word2vec and Glove. The parameters that define the structure of the CNN_word2vec are specified as follows: First, n_unique_words = 10000, here we determined the most 10000 words that are repeated in the corpus. Second, max review length= 100, this shows the length of the vector for each review. Third, n_dim= 100, we determined fixed length of the vector for each word. Fourth, drop_embed= 0.5, it determines the number of neurons work, i.e. when word embedding starts to enforce all neuron to get the same number of jobs. So, in each step only the half of total number of neutrons work. Fifth, k_conv= 3 it is about the dimension of the word, in our project it depended on the Trigram. Sixth, n_conv= 256 it is about the number of filters. Seventh, n_dense= 128 it is about the number of neurons on fully connected layers. After that, we used function sequence to convert each word to its index number. Then we applied pad_sequences function which fills the vector at the begin by zero then the values of the review at the end, it takes two parameters (sequence and max review length). Additionally, sentiment column contains positive, negative, and neutral in one column. However, in order for Keras library to work well, it must separate sentiment column to three columns (positive, Negative, Neutral). To do this step we used np_utils.to_categorical function. Then we used encode function to convert positive to 2, negative to 0, and neutral to 1. After that, we designed Neural Network layers: 1) sequential layers; 2) Embedding layers which take three parameters (n_unique_words, n_dim, input_length=max_review_length); 3) SpatialDropout1D (drop_embed) for the purpose of the number of neurons work; 4) (Conv1D (n_conv, k_conv, activation='relu')): it is about the dimension of the text and activation function relu it is about the mathematical process between neurons; 5) Global Max Pool layer to focus on the most power full words; 6) Dense (n_dense, activation='softmax') we mentioned it above and activation function soft max it is determine the final classes. In addition, to configure the model we used adam optimizer to calculate the distance between the prediction and actual value. Adam optimizer determines the learning rate based on the distance. If the distance large, then adam optimizer will increase the learning rate. Finally, we used 40 epoch to train the data. To build convolutional neural network with Glove as features engineering method we used same parameter we mentioned it in word2vec except we change the value of drop_embed to 0.25 and n_conv to 512, to increase the complexity. Because under fitting problem appeared. Moreover, The same layers and discussed functions in word2vec are being used for the rest of the model to achieved the goal.

To get more insightful results, Lime was applied as follows: At the beginning we start using Lime library by importing lime text to take a review as a text and Lime text explainer to interpret the classifier prediction. Then, we defined variable name class_name that contains the sentiment label 'Negative', 'Neutral', and 'Positive'. After that, we applied visualize_one_exp. This function takes six parameters to visualize the result: First, features: the text we send it to lime. Second, labels: the label of the text. Third, index. Forth, pipeline_obj: it takes two inputs Count victories and our model. Fifth, class_names = bigram_model. classes.tolist(). Sixth, top labels=1: how many interpretable classes are shown, sit explains only positive or negative or neutral or two labels together. Moreover, we used explain_one_instance. This function explains the lime with the same parameter (instance, pipe_line_obj,class_names=bigram_model.classes_.tolist(), top_labels= None). Finally, we used variable exp to save the vale send it from explainer.explain_instance function, this function takes 4-parameter: (i) instance, (ii) pipe_line_obj.predict_proba which predicts the probability for each word in the sentence that effect on the prediction, (iii) num_features= 6 which is about the highest word probability appear, and (iv) top_labels=top_labels).

VI. Experiments and Evaluation

In our research, each review has been classified as positive, negative, or neutral based on the star rating. So, four and five-star ratings are categorized as positive where two and one-star ratings are classified as negative. Finally, three-star rating is classified as neutral. We first, ran experiments on unbalanced data. Second, we applied our experiments to balanced data. Meaning that we took the same number of positive, negative, and neutral reviews. In our dataset, neutral reviews had the lowest number of reviews by 21,000 reviews. Therefore, we used this numbers for each sentiment balanced data.

TABLE I. RESULTS OF LOGISTIC REGRESSION FOR AMAZON WEBSITE UNBALANCED DATASET (DEVELOPMENT SPLIT) OF MOBILE PHONE REVIEWS.

	LL	F	RC	PR	ACC
BOW+B	0.29	91.21	91.71	91.42	91.71
BOW+T	0.28	91.63	92.06	91.77	92.06
TF-IDF+U	0.39	84.29	86.63	85.55	86.63
TF-IDF+B	0.33	87.96	89.47	89.39	89.47
TF-IDF+T	0.34	87.34	88.90	88.76	88.90
Glove	0.48	79.64	82.95	79.39	82.95
word2vec	0.44	81.36	84.56	81.79	84.56

TABLE II. RESULTS OF NAIVE BAYES FOR AMAZON WEBSITE UNBALANCED DATASET (DEVELOPMENT SPLIT) OF MOBILE PHONE REVIEWS.

	LL	F	RC	PR	ACC
BOW+B	0.67	87.00	87.87	86.98	87.87
BOW+T	0.73	87.47	88.53	87.94	87.53
TF-IDF+U	0.46	80.17	83.36	82.12	83.34
TF-IDF+B	0.43	82.83	85.82	86.48	85.82
TF-IDF+T	0.34	87.34	88.90	88.76	88.90
Glove	0.75	54.61	67.66	45.78	67.66
word2vec	0.71	54.72	76.71	68.81	67.71
word2vec+B	0.67	55.80	68.17	68.59	68.17

A. Results Obtained by Unbalanced data

Tables I, II, III and IV show the results for LR, NB, SGD, and CNN respectively using different features extraction techniques. Both BOW and TF-IDF have three variations depending on the number of grams used where U represents unigram, B represents bigram, and T represents trigrams. For logistic regression, bag-of-words with trigrams provided the highest accuracy and log loss value with 92.06%, 0.28 respectively. For naive Bayes we can observe that best performance is obtained by TF-IDF (Bigram) at 85.82% accuracy and 0.43 log loss. In contrast, Bag-of-words (Trigram) achieved higher accuracy at 88.53% but its log loss value is far from its actual value, it is 0.73. because when the value is near to zero, its near to actual value. However, Glove, Word2vec, and Word2vec with Bigram did not give good results because these methods study semantic between the word and measure similarity.

As shown in Table III, it can be observed the best performance for SGD is Bag-of-words (Trigram) with 89.61% accuracy. Moreover, we did not get any result from log loss since SGD is not a probabilistic algorithm. Table IV shows that CNN with word2vec achieved an accuracy of 92.73% and Log loss of 0.23. We can observe that the Log loss value is very near to zero. So, the performance of the model is very high, and probability of the error is very low. Additionally, CNN with Glove achieved 90.51% accuracy and 0.29 log loss value. A likely reason for this low result is that Glove has been applied on a per-trained model and the language is formal, while in reviews the language is informal. Finally, we found that CNN with word2vec achieved the best result comparing with Glove algorithms.

All the previous results were obtained using the development split of the dataset. Therefore, to obtain the test results, best settings of all of the four algorithms were applied on the test split, the results are illustrated in Table V. However, CNN achieved the best results with word2vec. In addition, all algorithms provided the lowest results with Glove feature extraction. As shown by Table V, CNN with word2vec achieved best result by 92.72 accuracy and 0.23 log loss.

TABLE III. RESULTS OF STOCHASTIC GRADIENT DESCENT FOR AMAZON WEBSITE UNBALANCED DATASET (DEVELOPMENT SPLIT) OF MOBILE PHONE REVIEWS.

	F	RC	PR	AC
BOW+B	87.86	89.07	88.90	89.07
BOW+T	88.56	89.61	89.43	89.61
TF-IDF+U	81.14	84.74	82.82	84.74
TF-IDF+B	81.59	85.13	84.66	85.13
TF-IDF+T	81.43	85.00	84.58	85.00
Glove	80.11	83.58	80.24	83.58
word2vec	80.79	84.49	81.90	84.94
word2vec+B	81.21	84.70	82.91	84.70

TABLE IV. RESULTS OF CONVOLUTIONAL NEURAL NETWORKS FOR AMAZON WEBSITE UNBALANCED DATASET (DEVELOPMENT SPLIT) OF MOBILE PHONE REVIEWS.

	LL	F	RC	PR	AC
Glove	0.23	92.00	92.00	92.00	92.73
word2vec	0.29	90.00	90.00	90.00	90.51

B. Results Obtained by Balanced data

Tables VI, VII, VIII and IX show the results for LR, NB, SGD, and CNN respectively using different features extraction techniques. Both BOW and TF-IDF have three variations depending on the number of grams used where U represents unigram, B represents bigram, and T represents trigrams.

The best results of all of the four algorithms are reported in Table X. It is observed that We CNN with word2vec provided best accuracy with 79.60% and a log loss of 0.52.

C. Lime Results Analysis

To show positive reviews using Lime, Fig. 7 provides some insight into the possible reasons behind classifying the review as positive. We can notice that the model detects the words ‘great’ and ‘love’ with the highest probability effect by 0.09, the word ‘nice’ at 0.04. To illustrate, the words in dark green give higher effect than the words in lighter color. To represent negative reviews using Lime, Fig. 8 demonstrates why the model predicted the review as a negative one. The word mess with the highest probability is equal to 0.09, crack by 0.08, and the word damage with 0.04 probability. However, the light blue color in the figure means that it does not have effect as much as mess and crack. Also, there is neutral word with total probability 0.08 but the total probability of negative

TABLE V. FINAL RESULTS FOR AMAZON WEBSITE UNBALANCED DATASET (TEST SPLIT) OF MOBILE PHONE REVIEWS.

Setting	LL	F	RC	PR	ACC
BOW+T $\rightarrow$ LR	0.3	91.24	91.72	91.44	91.72
TF-IDF+B $\rightarrow$ NB	0.43	82.77	86.69	86.54	85.69
BOW+T $\rightarrow$ SGD	-	88.49	89.51	89.25	89.51
word2vec $\rightarrow$ CNN	0.23	92.46	92.37	92.37	92.72

TABLE VI. RESULTS OF LOGISTIC REGRESSION FOR AMAZON WEBSITE BALANCED DATASET (DEVELOPMENT SPLIT) OF MOBILE PHONE REVIEWS.

	LL	F	RC	PR	ACC
BOW+B	78.84	0.61	78.96	78.87	78.96
BOW+T	79.37	0.60	79.52	79.44	79.52
TF-IDF+U	71.27	0.67	71.53	71.20	71.53
TF-IDF+B	75.89	0.62	76.04	75.88	76.04
TF-IDF+T	76.90	0.61	77.01	76.89	77.01
Glove	66.35	0.76	66.66	66.29	66.66
word2vec	66.01	0.76	66.44	65.89	66.44
word2vec+B	66.44	0.75	66.73	66.33	66.73

TABLE VII. RESULTS OF NAIVE BAYES FOR AMAZON WEBSITE BALANCED DATASET (DEVELOPMENT SPLIT) OF MOBILE PHONE REVIEWS.

	LL	F	RC	PR	ACC
BOW+B	1.1	72.28	72.98	72.96	72.98
BOW+T	1.3	70.79	72.10	72.99	72.10
TF-IDF+U	0.72	69.82	69.84	69.83	69.84
TF-IDF+B	0.63	74.08	74.34	74.18	74.34
TF-IDF+T	0.62	74.42	74.90	74.94	74.90
Glove	1.06	49.11	50.03	51.34	50.03
word2vec	1.04	61.96	61.85	63.51	61.85
word2vec+B	1.01	64.05	63.93	65.20	63.93

TABLE VIII. RESULTS OF STOCHASTIC GRADIENT DESCENT FOR AMAZON WEBSITE BALANCED DATASET (DEVELOPMENT SPLIT) OF MOBILE PHONE REVIEWS.

	F	RC	PR	AC
BOW+B	76.66	76.95	76.87	76.95
BOW+T	76.92	77.18	77.12	77.18
TF-IDF+U	70.05	70.74	70.28	70.74
TF-IDF+B	73.90	74.47	74.33	74.47
TF-IDF+T	74.74	74.74	74.58	74.74
Glove	68.68	68.82	68.57	68.82
word2vec	65.97	66.96	66.15	66.96
word2vec+B	69.60	70.15	69.70	70.15

words is highest. To highlight neutral reviews using Lime, we observed from Fig. 9 that the word 'fine' appeared as a positive with 0.17 probability, 'freeze' with 0.33 and 'okay' with 0.32 as a negative word. The words 'freeze' and 'okay' got also probability 0.43, 0.25 prospectively as neutral word. Hence, the total of neutral words is the highest so the model predicted it as neutral.

VII. BENCHMARKING

We also compare our work with some other work. In this paper, we involved dividing the data into three parts. First, training the data with 70%. Second, testing with 15%. Third, development with 15%. [15] has chosen to divided the data into two parts. Where, 80% of the data is training and the 20% left is for the testing, While, [14] has only worked on part of the dataset where only 21,500 were useful for training and 3,000 for testing. Moreover, our work and [15] both have worked on balanced and unbalanced data unlike [14]. A slight difference between our experiments and [15] is that our work categorized both the balanced and unbalanced into, five and four star ratings as positive, one and two as negative, and three as neutral. Meanwhile, [15] has categorized balanced and unbalanced data separately. Where, one- and two-star ratings as negative, four and five as positive, and three has been cancelled off for balance data. For unbalanced data comprised

TABLE IX. RESULTS OF CONVOLUTIONAL NEURAL NETWORKS FOR AMAZON WEBSITE BALANCED DATASET (DEVELOPMENT SPLIT) OF MOBILE PHONE REVIEWS.

	LL	F	RC	PR	AC
Glove	0.51	80.00	80.00	80.00	79.91
word2vec	0.60	76.00	76.00	76.00	76.51

TABLE X. FINAL RESULTS FOR AMAZON WEBSITE BALANCED DATASET (TEST SPLIT) OF MOBILE PHONE REVIEWS.

Setting	LL	F	RC	PR	ACC
BOW+T → LR	0.66	77.27	77.47	77.34	77.47
TF-IDF+T → NB	0.62	74.42	74.90	74.94	74.90
BOW+T → SGD	-	75.75	76.05	75.94	76.05
word2vec → CNN	0.52	79.57	79.55	79.55	79.60

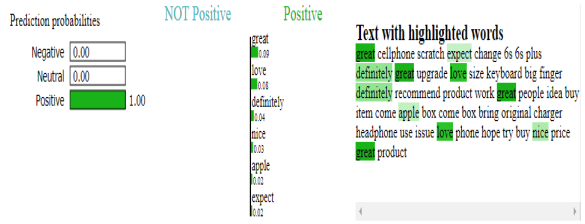

Fig. 7. Positive interpretation for Amazon website unbalanced dataset of mobile phone reviews.

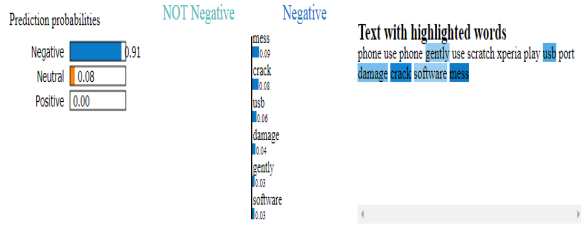

Fig. 8. Negative interpretation for Amazon website unbalanced dataset of mobile phone reviews.

one- and two-star rating as negative a three, four, and five as positive. Also, another point we have in common with one of the papers is that [14] and our work used the same algorithm (naive Bayes) with TF-IDF (unigram). On the other hand, [15] applied different deep learning methods such as, CBOW and skip-gram. To sum up, our work cannot be directly compared to either of the results because of the difference in the data division.

VIII. LIMITATIONS AND FUTURE WORK

In this study, we implemented four types of algorithms with a variety of feature extraction. Some algorithms that remain to be applied in future work include LSTM, KNN, and Maximum entropy. Then, we will compare the result to the result we performed in this current study. Also, we intend to add Arabic language to increase the scope of the research. Our research has some limitations: NLP is relatively a new topic, and highly advanced; hence, it needs a lot of research to understand the field and how it works. Furthermore, we faced some problems with computer memory causing experiments to be highly time consuming. We also used Google Colab to increase the performance, but it did not give us the expected speed.

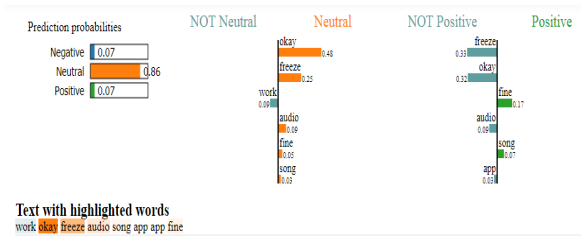

Fig. 9. Neutral interpretation for Amazon website unbalanced dataset of mobile phone reviews.

IX. Conclusion

Reviews are essential for both individuals and companies. Consumers used them to make good decisions prior to buying a specific product and companies benefit from them to know their consumers' satisfaction about products. In this research we studied sentiment analysis of mobile phone reviews using different types of machine learning classifiers, such as Logistic Regression (LR), Naïve Bayes (NB), Stochastic Gradient Decent (SGD) and deep learning algorithms such as Convolutional Neural Networks (CNN). These algorithms are applied using different feature extraction approaches. For example, Bag-of-words with (Bigram, Trigram), TF-IDF with (Unigram, Bigram, Trigram), word2vec, word2vec with Bigram, and glove. We evaluated them with different classification methods such as bag-of-words revealing that when the size of 'n' in n-gram increases, the accuracy will also increase, and Log loss value will decrease. On the other hand, our Bigram approach provided best results with TF-IDF in unbalanced data, and Trigram in balanced data. Moreover, word2vec deep learning feature extraction provided better accuracy than Glove because glove used a pre-trained model, and the language the text written was formal, while in our corpus the reviews were written in informal language. Also, CNN with word2vec achieved the best accuracy (92.72%), and log loss value (0.23) compared to all other algorithms for unbalanced data. While in balanced data CNN with word2vec methods achieved the best result compared to other algorithms with (79.60%) accuracy and (0.52) log loss. Finally, we applied Lime technique to interpret the reasons behind classifying the reviews as positive, negative or neutral. From the statistical analysis, it was concluded that the length of a review is a significant variable to identify the polarity, therefore, it can be included as a feature to the machine learning algorithms.

References

[1] C. Heller Baird and G. Parasnis, "From social media to social customer relationship management," *Strategy & leadership*, vol. 39, no. 5, pp. 30–37, 2011.

[2] A. J. Flanagin, M. J. Metzger, R. Pure, A. Markov, and E. Hartsell, "Mitigating risk in ecommerce transactions: perceptions of information credibility and the role of user-generated ratings in product quality and purchase intention," *Electronic Commerce Research*, vol. 14, no. 1, pp. 1–23, 2014.

[3] Y. Kim and J. Srivastava, "Impact of social influence in e-commerce decision making," in *Proceedings of the ninth international conference on Electronic commerce*, pp. 293–302, ACM, 2007.

[4] R. Zhang and T. T. Tran, "Helping e-commerce consumers make good purchase decisions: a user reviews-based approach," in *International Conference on E-Technologies*, pp. 1–11, Springer, 2009.

[5] R. Bolden and J. Moscarola, "Bridging the quantitative-qualitative divide: the lexical approach to textual data analysis," *Social science computer review*, vol. 18, no. 4, pp. 450–460, 2000.

[6] B. Pang, L. Lee, *et al.*, "Opinion mining and sentiment analysis," *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.

[7] J. Riegelsberger, M. A. Sasse, and J. D. McCarthy, "Shiny happy people building trust?: photos on e-commerce websites and consumer trust," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 121–128, ACM, 2003.

[8] J. P. Singh, S. Irani, N. P. Rana, Y. K. Dwivedi, S. Saumya, and P. K. Roy, "Predicting the "helpfulness" of online consumer reviews," *Journal of Business Research*, vol. 70, pp. 346–355, 2017.

[9] Z. Zhang, "Weighing stars: Aggregating online product reviews for intelligent e-commerce applications," *IEEE Intelligent Systems*, vol. 23, no. 5, pp. 42–49, 2008.

[10] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005.

[11] C. Zhang, D. Zeng, J. Li, F.-Y. Wang, and W. Zuo, "Sentiment analysis of chinese documents: From sentence to document level," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 12, pp. 2474–2487, 2009.

[12] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 83–92, ACM, 2014.

[13] J. Serrano-Guerrero, J. A. Olivas, F. P. Romero, and E. Herrera-Viedma, "Sentiment analysis: A review and comparative analysis of web services," *Information Sciences*, vol. 311, pp. 18–38, 2015.

[14] 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.

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

[16] Y. Goldberg and O. Levy, "word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method," *arXiv preprint arXiv:1402.3722*, 2014.

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

[18] D. Guthrie, B. Allison, W. Liu, L. Guthrie, and Y. Wilks, "A closer look at skip-gram modelling.," in *LREC*, pp. 1222–1225, 2006.

[19] A. McCallum, K. Nigam, *et al.*, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, vol. 752, pp. 41–48, Citeseer, 1998.

[20] I. Rish *et al.*, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, pp. 41–46, 2001.

[21] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*, pp. 137–142, Springer, 1998.

[22] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[23] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of machine learning research*, vol. 9, no. Aug, pp. 1871–1874, 2008.

[24] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[25] L. Zheng, H. Wang, and S. Gao, "Sentimental feature selection for sentiment analysis of chinese online reviews," *International journal of machine learning and cybernetics*, vol. 9, no. 1, pp. 75–84, 2018.

[26] L. Zhang, K. Hua, H. Wang, G. Qian, and L. Zhang, "Sentiment analysis on reviews of mobile users," *Procedia Computer Science*, vol. 34, pp. 458–465, 2014.

[27] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, vol. 242, pp. 133–142, Piscataway, NJ, 2003.

[28] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Systems with Applications*, vol. 57, pp. 117–126, 2016.

[29] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *International Conference on Machine Learning*, pp. 957–966, 2015.

[30] D. Bollegala, T. Maehara, and K.-i. Kawarabayashi, "Unsupervised cross-domain word representation learning," *arXiv preprint arXiv:1505.07184*, 2015.

[31] M. Dragoni and G. Petrucci, "A neural word embeddings approach for multi-domain sentiment analysis," *IEEE Transactions on Affective Computing*, vol. 8, no. 4, pp. 457–470, 2017.

[32] D. Mackay, "Glove.," May 7 1907. US Patent 852,972.

[33] C. Cerisara, P. Kral, and L. Lenc, "On the effects of using word2vec representations in neural networks for dialogue act recognition," *Computer Speech & Language*, vol. 47, pp. 175–193, 2018.

[34] A. Khatua, A. Khatua, and E. Cambria, "A tale of two epidemics: Contextual word2vec for classifying twitter streams during outbreaks," *Information Processing & Management*, vol. 56, no. 1, pp. 247–257, 2019.

[35] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186, Springer, 2010.

[36] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[37] H. Zhang, "The optimality of naive bayes," *AA*, vol. 1, no. 2, p. 3, 2004.

[38] 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.

[39] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: Primal estimated sub-gradient solver for svm," *Mathematical programming*, vol. 127, no. 1, pp. 3–30, 2011.

[40] S. Lu and Z. Jin, "Improved stochastic gradient descent algorithm for svm,"

[41] I. Banerjee, Y. Ling, M. C. Chen, S. A. Hasan, C. P. Langlotz, N. Moradzadeh, B. Chapman, T. Amrhein, D. Mong, D. L. Rubin, *et al.*, "Comparative effectiveness of convolutional neural network (cnn) and recurrent neural network (rnn) architectures for radiology text report classification," *Artificial intelligence in medicine*, 2018.

[42] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, ACM, 2016.