Journal of Theoretical and Applied Information Technology
15th September 2022. Vol.100. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IWVTSA: IMPROVED WORDS VECTORS FOR TWITTER SENTIMENTS ANALYSIS

[1]AMINA SAMIH, [2]ABDERRAHIM GHADI, [3]ABDELHADI FENNAN
[1,2,3] Department of Computer Sciences, Data & Intelligent Systems (DIS) Team, FSTT, Abdelmalek Essaadi University, Tetouan, Morocco

ABSTRACT

Twitter sentiment analysis technology provides methods for polling public opinion on events or products. The majority of current research is aimed at obtaining sentiment features by analyzing lexical and syntactic features. Word embeddings express these characteristics explicitly. This paper proposes a new approach called Improved Words vectors for Twitter sentiments analysis (IWVTSA) to improve the f1 score of twitter sentiment analysis. We introduce a word embeddings method called tweet2vec, which was launched concurrently with doc2vec, to form a sentiment feature set of tweets; these word embeddings are combined with word sentiment polarity score features. For training and predicting sentiment classification labels, the feature set is integrated into four machine learning classifiers (XGBoost, SVM, Logistic Regression, Random Forest). We compare the quality of our model to that of the baseline model, the bag of words algorithm, and the results show that the combination of tweet2vec and Polarized lexicon in our model performs better on the F1-measure for Twitter sentiment classification.

Keywords: *Sentiment Analysis, Tweet2vec, Doc2vec, Polarized Lexicon, Machine learning.*

1. INTRODUCTION

Sentiment analysis [1] is a practical technique that enables businesses, researchers, governments, politicians, and organizations to learn about people's emotions, including "happy, sadness, rage, or a reasonably neutral mood." It plays an essential role in decision-making. *Sentiment analysis* is a technological procedure used to determine the author's sentiment about a topic. It extracts certain words or phrases from a document to evaluate whether it is good, negative, or neutral. Its main advantage is to analyze a large amount of data to understand the general feeling of a community. It was designed to provide insight into how a brand, campaign, or topic is perceived by its audience. These metrics help understand a company's reputation, the perception of competitors in a market, or the overall understanding of how well a chosen message was received.

Because of the ubiquity of social media and the low barrier to sending a message, social media sentiments provide the most up-to-date and comprehensive information [2]. Social media has grown in popularity in recent years, and it now plays a vital role in everyone's daily lives in our modern digital era. The data generated by social media is massive and unstructured, containing real-time public opinion and sentiments in numerous formats and languages. Twitter is a microblogging website that serves as a primary location for people to communicate their opinions and views on politicians and political parties [3].

Any breaking news or emergent events are almost promptly followed, resulting in a spike in Twitter volume and a unique opportunity to elucidate the relationship between election events and widespread attitude. Twitter gives users a venue to transmit, interpret, and share 280-character posts known as tweets. Twitter presently has 326 million monthly active users and is accessible via SMS, mobile devices, and a website interface.

With internet connectivity, numerous social media networking sites provide tremendous information on various topics in real-time from any location, at any time, and from any location. On average, 6000 tweets are generated every second, equating to 500 million tweets every day and 350,000 tweets per minute. As a result, Twitter provides a massive supply of information and data on current trends, people's opinions, and sentiments in real-time, which may be used for data analysis and text analytical study and yield significant insights.

Until recently, sentiment analysis in scientific publications relied nearly solely on the bag-of-words method. [4] investigated several sentiment categorization methods for Twitter data and discovered that they are widely used. Researchers use pre-existing sentiment dictionaries [5] or develop custom, context-sensitive dictionaries [6]. The third set of investigations uses machine learning tools [7].

Some researchers have discovered that evaluating sentiment at the article or speech level produces decent results [6]. However, the bag-of-words method's assumptions and simplifications, such as loss of grammatical structure or context-dependent word meanings, have been often emphasized [8]. [9] provide a history of vector space model development. Mikolov and colleagues [10] demonstrated a more efficient architecture for creating reusable word vector representations from large text corpora, which garnered considerable attention to the word embeddings technique. Word Embedding is a deep learning technique for producing vector representations of words and texts. These techniques have received much interest in text and sentiment analysis because of their ability to capture the syntactic and semantic relationships between words. The most popular are Word2Vec [11] and Doc2vec [12].

[11][13] proposed respectively; the word2vec echnique for obtaining word vectors by training a text corpus and the doc2vec to conduct sentiment analysis on twitter with semi-supervised data in both English and Turkish. The concept of word2vec evolved from the distributed representation of words [10], and doc2vec is an extension to word2vec to learn document-level embeddings.

Several scientists used these methodologies in their sentiment analysis studies. [14,15,16]. A sentiment review statement's word vectors are averaged in [17], and authors extended word embeddings to sentence embeddings. Their findings showed that word embeddings outperformed the bag-of-words model in sentiment classification. In this work, word2vec and doc2vec techniques on sentiment analysis were explored. Word embeddings trained from different corpora were compared using two parallel models, tweet2vec(based on word2vec) and polarized lexicon vs. Doc2vec and polarized lexicon, via different machine learnings classifiers. Experiment results show that the first model(tweet2vec +polirized lexicon) improves the F1 score of twitter sentiment analysis.

The rest of this paper is as follows: The background is provided in Section 2, which includes a brief description of the Twitter sentiment analysis and sentiment classification stages. Section 2 discussed related work on this topic; Section 3 presents our proposed approach and algorithms; Section 4 reports our experiments, showing results and evaluations; Section 5 concludes the paper and highlights future works .

2. BACKGROUND

Twitter is a microblogging service that began in 2006 and allows users to share text messages, known as tweets, as well as links to other content such as images, websites, and articles. Every tweet has a maximum length of 140 characters. It can include links to news articles, videos, or images while describing an event or people's opinions about an event or a person. Hashtags can be used in a tweet to indicate relevant topics. Recent statistics show over 320 million active Twitter accounts, with 500 million tweets sent every day1. Twitter sentiment analysis aims to categorize tweets based on whether they are positive or negative. As a result, it's a classification issue [18]. After that, we'll go over the stages of Twitter sentiment classification.

2.1 Phase of data ingestion

Ingests data streams from the Twitter API and other sources [19]. MQTT, RabbitMQ, ActiveMQ, NSQ, ZeroMQ, NiFi, Distributed Log, and Kaf ka are popular open-source options for ingesting data streams into analytics platforms or a data store. Kafka is the most widely used of these tools [20]. It is a distributed streaming platform that streams records and stores them for processing [21]. Kafka is a distributed messaging system that runs as a cluster on servers across multiple data centers, storing streams of records in categories (i.e. topics). A key, a value, and a timestamp make up each record. Records ingested with data ingestion tools are stored in frameworks like Hadoop Distributed File System (HDFS) and Cassandra, which are suitable for further analytics. The most popular framework is Hadoop.

It is an open-source software library written in the JAVA programming language that allows for distributed data processing and parallel processing of large datasets across a cluster of nodes. Hadoop is made up of four major modules: I Hadoop Common, which contains utilities used by other modules; (ii) Hadoop Distributed File System (HDFS), which provides storage capabilities by breaking large files into blocks and storing them in different nodes across a cluster; (iii) Hadoop Map-Reduce, which processes

a large dataset in parallel by each map task working on a subset of the data input (the final output is processed further in the reduce phase); and (iv) Hadoop YARN, which is a [21] and [22].

2.2 Phase of data analysis

The goal is to perform data analytics. Spark and Flink [19] are two engines that can be used for distributed analytics. The former is by far the most popular engine. Spark is an open-source clustering computing framework with features such as implicit data parallelism, fault tolerance, SQL libraries, and stream processing [21]. Spark is a stand-alone application. Spark's cluster manager assigns tasks to workers, one for each partition. Each task performs a unit of work in its own dataset partition and saves the result to disk. Four cluster managers are supported by the Spark framework:

- Preprocessing: consists of three steps that transform raw data into a machine-readable format: data cleaning, data transformation, and data reduction [23]. All URLs, hashtag symbols, and other special characters are removed during the data cleaning process. Stop words (commonly used words such as - the, a, an, in - that a search engine can ignore when retrieving them as a result) are also removed to save space and time [23].

- Feature extraction: This technique aims to extract essential features for training purposes. The more features extracted, the more accurate the classification [24]. The following are some of the features that can be extracted: (1) Sentiment characteristics: these are characteristics that are related to the positivity and negativity of words and emotions (e.g., the number of positive and negative words or emotions in a text). (2) Syntax-based characteristics: question, exclamation, parentheses, and quotation marks, as well as their count in sentences (for example, number of exclamation marks and number of dots). (3)Focus on the logic behind the sentences, such as passive and active forms, with semantic features. (4)Unigram-based features: as seed words for a user-defined input, include hypernyms (i.e., more generals) and hyponyms (i.e., more specifics). (5) N-gram features specify a feature by grouping together an N number of sequential words. Bigram features (BGF) [25] are created when two consecutive words are used. (6) Extraction of words with a high number of occurrences in the text is one of the top features of Top Words.(7) Pattern-based features: extract patterns in sentiments using Part-of-Speech tags (e.g., positive and negative names, positive and negative verbs, positive and negative adjectives, pronouns, etc.).
- Feature extraction (filtering): Used to reduce the size of features in order to improve the speed and accuracy of classification models. The most widely used feature selection method is known as Frequency-Inverse Document Frequency (TF-IDF) [26]. It calculates the most frequently used terms (TF) as well as the frequency with which the term appears in a text (IDF). The terms with the highest TF-IDF (i.e., the product of TF and IDE) scores are those that appear the most frequently and contain the most relevant information on a given topic.
- Classification is the process of dividing a text into distinct groups. The following are some examples of classifiers: Lexicon classifier [27]: a lexicon is a group of words with a predetermined polarity. (iii) Support Vector Machine (SVM) [28]: a tool for data mining tasks such as classification, regression, and novelty detection; and (iv) Nave Bayes (NB) [29]: a simple probabilistic classifier that calculates a set of probabilities by counting the frequency and combinations of values in a given data set; SVMs have been used successfully in a variety of applications, including particle detection, face recognition, and text categorization. There are three main classification types, according to [30]:

✓ Binary classification: Positive (i.e., high positive scores) and negative (i.e., low positive scores) sentiments are classified into two fundamental polarities (i.e., high negative scores). Positive, negative, and neutral sentiments are classified into three groups: positive, negative, and neutral (i.e., no positive or negative scores are included).

✓ Ternary classification: classification of sentiments into three classes: positive, negative, and neutral (i.e., do not include any positive or negative scores).

✓ Multiclass classification: dividing sentiments into multiple predefined categories to extract not only positive or negative sentiments but also feelings and opinions. Furthermore. The classes can be used to categorize texts based on their happiness, enjoyment, or dislike.

3. RELATED WORK

3.1 Polarized lexicon-based sentiment analysis

Sections Polarized lexicon-based methods and machine learning methods such as deep learning [17] are the two types of sentiment analysis strategies. The polarized lexicon-based sentiment analysis approach is frequently based on lists of positive and negative word and phrase meanings [31]. A dictionary of words with negative and positive sentiment values is required for this strategy. These methods are simple to implement, scalable, and rapid to compute. As a result, they are used to dealing with generic sentiment analysis problems [17]. However, in human-labeled data, lexicon-based techniques rely on human labor [31]. They also rely on discovering the sentiment lexicon utilized in the text analysis [17]. In the polarized lexicon, each word is allocated a polarity score. This score, which is represented by an exact value, reflects the degree (or intensity) of positive or negative on a scale with integer or actual values. For example, on the scale [-2,2], the real -2 denotes total negativity, while the natural 2 denotes absolute positivity. The values less than 0 represent varying levels of negativity: the further the real is from 0 (and closer to 2), the stronger (or more intense) the negativity. Furthermore, the reals bigger than 0 signify varying degrees of positivity: positivity is more important when the reality is closer to 2.

The precise 0 indicates that the term is neither positive nor negative: it has a neutral polarity. Polarity-based techniques have recently been combined with machine learning algorithms due to improved text classification accuracy. Several authors [17,31] reported that machine learning approaches were more accurate than polarity methods. Mudinas et al. [32] enhance sentiment analysis accuracy by combining lexicon-based and Support Vector Machine (SVM) techniques.

Zhang and colleagues [33] successfully combined a lexicon-based method with a binary classifier for sentiment categorization of Twitter data. Kurniawati1 and colleagues [34] used the Particle Swarm Optimization (PSO) methodology with the SVM method for sentiment analysis of Twitter data. In all of these cases, machine learning algorithms improved text classification F-score.

3.2 Word Embedding in sentiment analysis sentiment analysis

, Word Embeddings [35] techniques such as Word2Vec and Doc2vec are continuous vector representations of words that can convert words into meaningful vectors. Vector representations of words can help in text categorization, grouping, and information retrieval. The F1score of the Word2vec and Doc2vec algorithms is affected by the size of the text corpus. In other words, as the text corpus grows, so does the F1score. Severyn and Moschitti [36] learned word embeddings from 50 million tweets using the Word2Vec method and fed the vectors into a deep learning network. [37] utilized word2vec to perform sentiment analysis on 14,640 tweets about US airlines and found a similar range of values. The classifier correctly predicted 75% of the 2750 negative tests in the negative class. The classifier correctly predicted 62 percent of 936 neutral test instances with an F1-score of 56 percent for the neutral class. The classifier correctly predicted 70% of the 706 positive test examples, with an F1 -score of 63% for the positive class. [38] applied Doc2Vec to sentiment analysis by feeding word embeddings into Convolutional neural networks. Because of the limitations and restrictions in particular corpora, researchers choose to use word embeddings vectors as inputs to machine learning models. As a result, enhancing the quality of word embeddings is key, as it plays an important part in sentiment categorization algorithms.

Despite having a low F score, the authors of [39] included Word2Vec vectors in their deep learning model. Applying Word2Vec lowered the F score of sentiment categorization in several datasets. Furthermore, [40] proposed a method for increasing Word2Vec's F score. Their method was tested on two datasets, and the proposed algorithm reduced Word2Vec's F score on one of them. To improve the f1score of tweets embeddings, we suggested building vectorial representations of Twitter sentiments based on the combination of two techniques, "Sentiment2vec" and "Polarized Lexicon" vs. "Doc2vec" and "Polarized Lexicon," and made classification using different Algorithms. The following section presents our proposed approach, methodology, and classifiers used to evaluate our method.

4. PROPOSED APPROACH

We increased the F1score of sentiment classification based on the combination of natural language processing techniques, Polarized lexicon, Tweet2vec (based on word2vec) vs. the combination of natural language processing techniques, Polarized lexicon, Doct2vec in our proposed approach, Improved Words vectors for Twitter sentiments analysis (IWVTSA). Different classifiers were used to make the classification. Figure 1 depicts the primary architecture of the suggested approach.

Journal of Theoretical and Applied Information Technology
15th September 2022. Vol.100. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

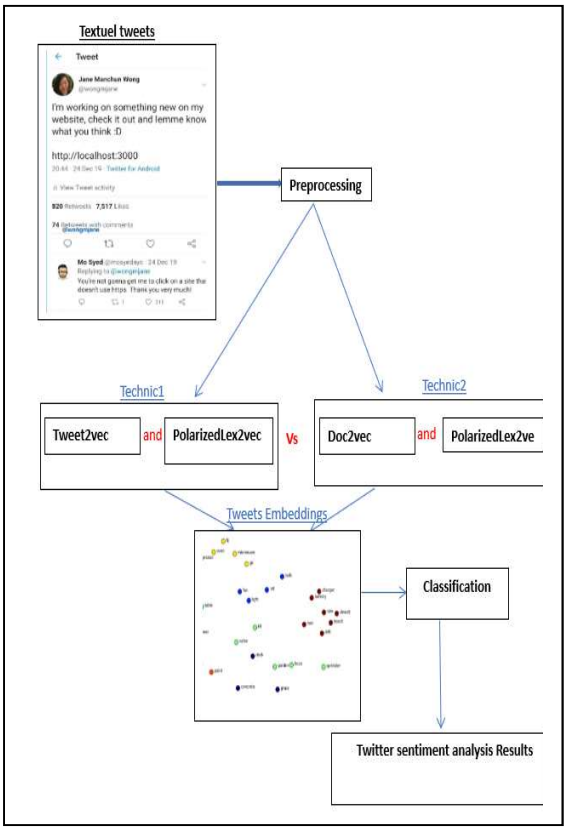

Figure 1: An Overview of IWVTSA Approach

4.1 Preprocessing

Preprocessing takes place before starting the approach procedure; this level includes operations such as tokenization, case folding, and cleaning [41]. Tokenization divides each tweet into smaller components known as tokens or words [42]. Case folding is the process of making all of the characters in a tweet text lowercase [42]. Meanwhile, non-alphabetic letters such as punctuation and digits are used in cleaning. Stemming and filtering are not employed in this investigation because they have not been shown to increase twitter sentiment analysis performance in previous studies.

4.2 Tweet2vec(for technic 1)

In this stage, we created Tweet embeddings based on word embeddings. We averaged the vectors of the words in one tweet to obtain tweet embeddings (tweet2vec). The main goal at this step is to determine the word embedding matrix Ww:

$$V_{Tweet2vec}(\omega) = \frac{1}{n} \Sigma W_w^{xi} \qquad (1)$$

Ww (W = W1, W2,...Wn >) represents the word embedding for word xi, which may be learned using the standard word2vec technique [1,11,43].

Word2vec was invented by[11], a well-known and widely used algorithm in learning word embedding that includes two models: Skip-Gram (SG) model and Continuous Bag-of words model (CBOW). The skip-gram model uses a word as input to forecast a target context, whereas the continuous bag of words technique uses a context as input to predict a single word [44]. Figure 2 shows the two word2vec variants. There are three layers in each of these architectures: an input layer, a hidden layer, and an output layer. The neurons in the output layer have softmax activation functions. The Skip-gram architecture is used in this study.

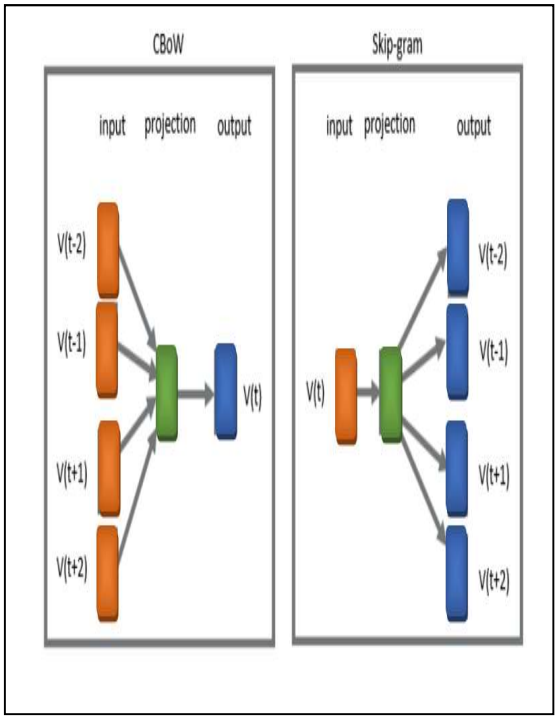

Figure 2: Word2vec architectures

4.3 Doc2vec (for technic 2)

The Doc2Vec is a Word2Vec extension that applies to a whole document rather than individual words. Le and Mikolov created this paradigm, which tries to provide a numerical representation of a document rather than a word representation [12]. Doc2Vec works on the premise that the meaning of a word is also affected by the document in which it appears. The sole difference between the Doc2vec algorithm and the Word2Vec algorithm is the insertion of a document ID, as seen in Figure 3. In this stage; we consider each tweet as a document to launch the doc2vec algorithm.

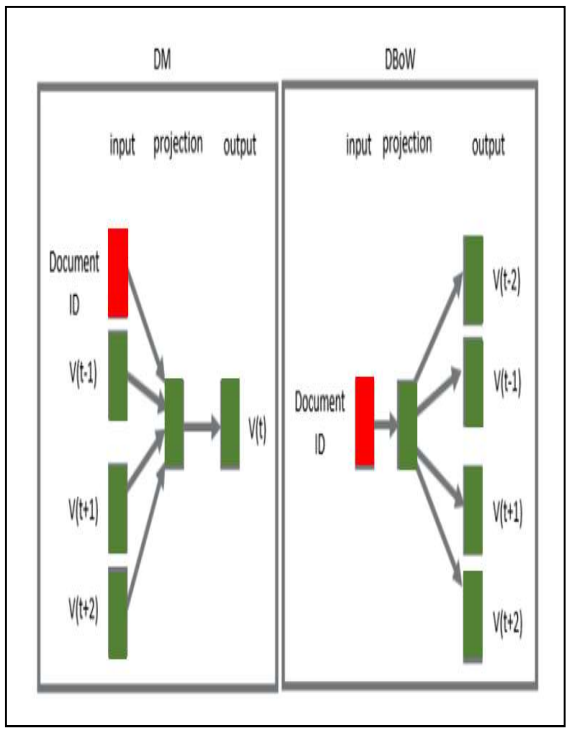

Figure 3: Word2vec architectures

4.4 PolarizedLex2vec (for both technic 1 and Technic 2)

We define *sentiment* as an individual's judgment about an object or subject marked by polarity. A polarity is either positive or negative for us. According to our definition, sentiment is a specific type of polarized opinion.

Sentiment and emotion lexicons are collections of polarity-scored phrases and words that can be used to analyze texts. Each lexicon has words and their values, which are sentiment scores.

4.5 Classification

. In the last stage, four machine algorithms are used independently for tweets embeddings classification:

- XGBoost(XG) [45] is a more complicated variant of the gradient boosting method. It comes with tree learning techniques and a linear model solver. It is speedy since it can perform parallel processing on a single CPU. It also includes cross-validation and important variable detection methods. To optimize the model, several parameters must be modified. Some of the critical benefits of XGboost are as follows:(1) Regularization: helps to reduce overfitting. (2)Parallel Processing: XGboost considerably accelerates parallel processing. (3) Missing Values: It includes a technique for dealing with missing values. (4)Built-in cross-validation: The user can execute cross-validation at each iteration of the boosting process.
- Support Vector Machine (SVM) [46]: This algorithm often used to tackle classification problems. This study uses this approach to represent each tweet embedding as a data point in d dimensional space (where d is the number of features). The value of each feature is the value of a specific coordinate.
- Logistic Regression (LR)[47] is a set of independent variables that predicts a binary result (1 / 0, Yes / No, True / False). Logistic regression is a subset of linear regression in which the dependent variable is the log of probability, and the outcome variable is categorical. In other words, it predicts the occurrence of an event by fitting data to a logit function. This model understands a vector of variables and evaluates coefficients or weights for each input variable before predicting the class of the provided tweet vector [21].
- RandomForest (RF) [48]: We also use random forest for categorization in this work. A supervised classification algorithm is the random forest algorithm, a decision tree-based ensemble learning technique .To improve robustness over an individual estimator, this Ensemble approach aggregates the predictions of some base estimators built with the decision tree process. Random Forest grows many classification trees, which is known as a forest. We categorized tweets in vectorial form, with each tree providing one vote for its category forecast. The forest selects the category with the most votes. The higher the F score findings are given, the more trees in the random forest.

4.6 Algorithms

The first algorithm uses each tweet as input and uses word2vec to return vectors of the tweet words. As a result, in the second phase, each word vector of the input tweet is extracted from word2vec [11], and the mean of word vectors is determined to calculate the vector of the tweet using tweet2vec. The sentiment scores of each word are retrieved from lexicon-based polarity in the third stage (negative or

positive). Furthermore, we will make them acceptable. If a word (negative or positive) does not appear in this couple, its score is 0. The vectors generated in each stage will be concatenated with vectors generated in previous steps. On the produced vectors, we fit the mentioned machine learning classifiers.

The second algorithm takes out words from the tweets and then creates a LabelSentence object to train Doc2Vec. As a result, in the second phase, each tweet is considered as a document to calculate the vector of the tweet. The sentiment scores of each word are retrieved from lexicon-based polarity in the third stage (negative or positive). Moreover, we will make them acceptable. If a word does not appear in this couple (negative or positive), its score is 0. The vectors generated in each stage will be concatenated with vectors generated in previous steps. On the produced vectors, we fit the mentioned machine learning classifiers independently. in the third stage (negative or positive). Moreover, we will make them acceptable. If a word does not appear in this couple (negative or positive), its score is 0. The vectors generated in each stage will be concatenated with vectors generated in previous steps. On the produced vectors, we fit the mentioned machine learning classifiers independently.

Algorithm 2: Improved Words vectors for Twitter sentiments analysis (Doc2vec+Polarizedlex2vec)

Inputs:

T = {W1, W2,....., Wn} , *Input Tweet T contains n words*
D2Vec = Doc2vec
PL2vec (P, N): *Polarized Lexicon*
P: Positive, N: Negative

Output:

IWVTSA: Improved Words vectors of tweet T for Twitter sentiment analysis.

```
for j=1 to m do
    VTj ⟵ GenerateVector ( Tj )
    Tj ⟵ <Tj , VTj>
end for

for each Wi in T do
        If Wi exist in D2Vec then extract VecWi
                TVi ⟵ VecWi
        endif
        for k=1 to h do
                If Wi in PL2vec then
                    Sik ⟵ FindVector ( Wi )
                end if
            ADD Sik into TVi
        end for
ADD TVi into IWVTSA
Return IWVTSA
end for

//fit classifiers(Xgboost,SVM,LR,RF) on IWVTSA
```

5. EXPERIENCES AND EVALUATION

In this section, we present the datasets(from Kaggle[38]) and experimental assessments that we used to demonstrate the efficacy of our suggested approach.

5.1 Datasets

We used two Twitter sentiment analysis datasets,TtD1[49](3.5 MB) and TtD2[49] (4.74 MB).

Datasets contain tree attributes

- tweeted - unique ID for each piece of text
- Tweet - the text of the tweet
- sentiment -the polarity of the tweet (positive) or (negative)

5.2 Evaluation metrics

Precision, recall, and F1score are commonly used to assess the performance of classification systems. We utilize F1 in this investigation because it represents both recall and precision. The F1 score is utilized as the evaluation metric, and it is the weighted average of both Recall and Precision. As a result, this score considers both false positives and false negatives. It is appropriate for problems with unequal class distribution.

The essential components of the F1 score are:

- True Positives Tweets (TPT) - These are the correctly predicted positive values which mean that the value of the actual class of Tweets is yes, and the value of the predicted class is also yes.
- True Negatives Tweets (TNT) - These are the correctly predicted negative values, which mean that the actual class of Tweets value is no, and the value of the predicted class of Tweets is also no.
- False Positives Tweets (FPT) – When the actual class of tweets is no, the predicted class of tweets is yes.
- False Negative Tweets (FNT) – The actual class of tweets is yes, but the predicted class of tweets is no.

1) Precision = TPT/TPT+FPT

2) Recall = TPT/TPT+FNT

3) F1 Score = 2(Recall * Precision) / (Recall + Precision)

5.3 Results

For each dataset, we have preset train and test sets. Our implementation begins with data cleansing. The text in the evaluation datasets is a very unstructured sort of data, and it contains several sorts of noise. Without any preprocessing, the data is challenging to comprehend. We used a series of cleaning techniques and text normalization to make it noise-free and ready for analysis.

In (technic1), the Word2Vec model is trained on datasets to generate vector representations for the unique terms in each tweet. Then, Tweet2vec generates a vector for each tweet by averaging the vectors (from word2vec) of the words in the target tweet. The length of the resulting vector will be "200." we will repeat the same procedure to obtain vectors for all textual tweets available in the mentioned datasets.

In (technic 2) doc2vec was implemented by labeling each tokenized tweet with a unique ID using Gensim's LabeledSentence () method. As a result, we treated each tweet as a separate document.

Two tweet lexicons based on polarity (positive tweet, negative tweet) were employed to extract and produce the lexicon polarity vectors for both technic one and technic 2. The pre-modeling procedures mentioned above are essential to collect the data in the appropriate form and shape. We generated models on the datasets with prepared feature sets — our created vectors (IWVTSA). The following algorithms (see section E) are utilized to make classification: XG[45], LR[47], SVM[46] , RS[48].

We have compared our approach (Tecnic1 and Tecnic2) with Bag of Words; We extracted bag of words features from our datasets. We analyzed a Group G containing Tweets T(as documents)d1,d2.....dD, and N distinct tokens derived from the Group G.

The N tokens (words) will create a dictionary. T X N will determine the size of the bag-of-words matrix M. The frequency of tokens in tweet T(i) is represented by each row in the matrix M. Let us take this illustration as a simple example.

T1: They are smart. You are also smart.

T2: Sandra is a smart person.

The dictionary created would be a list of unique tokens in the corpus =['They',' You',' Smart',' Sandra',' person']

Here, T=2, N=5.

Table 1: Center THE RE

	They	You	smart	Sandra	person
T1	1	1	2	0	0
T2	0	0	1	1	1

The columns in the preceding matrix can now be employed as features in the construction of above classification models.

Journal of Theoretical and Applied Information Technology
15th September 2022. Vol.100. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

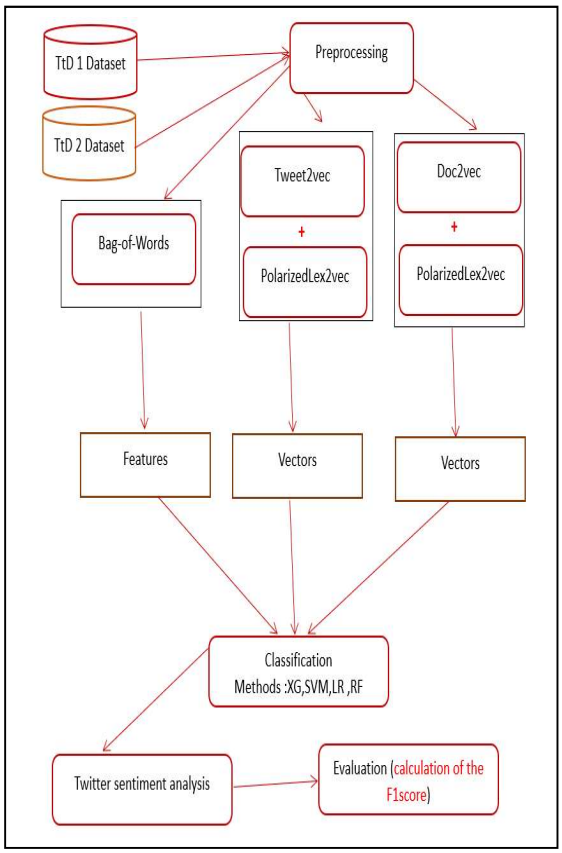

Figure 4: Process Of Experiments And Evaluation

To represent our F1score results, we considered:

- Tweet2vec as T2vec
- Doc2vec as D2vec
- PolarizedLex2vec as PL2vec
- Bag of Words as BW

Table 2: F1-Score Results Using Ttd 1 Dataset

Classifiers	T2vec+ PL2vec	D2vec+ PL2vec	BW
XG	**0.69**	0.37	0.55
SVM	0.66	0.24	0.55
LR	0.55	0.38	0.56
RF	0.53	0.21	0.54

Table 3: F1-Score Results Using Ttd 2 Dataset

Classifiers	T2vec+ PL2vec	D2vec+ PL2vec	BW
XG	**0.70**	0.31	0.51
SVM	0.64	0.22	0.53
LR	0.56	0.37	0.56
RF	0.52	0.20	0.54

It is now time to finish things up. Let us go over everything we have learned so far. First, we cleaned our raw text data, and then we learned about three different sorts of feature-sets using: (Tweet2vec+PolarizedLex2vec), (Doc2vec + PolarizedLex2vec), and (Bag of words) that we extracted from Ttd 1 and TtD 2 datasets. We then used these feature sets to develop models for Twitter sentiment analysis (which can be retrieved from any text data). The tables below provide F1 scores for several models (XG, SVM, LR, RF) and feature sets.

As can be seen, the technic 1 (Tweet2vec+PolarizedLex2vec) with the XGboost model has the highest F1score, and the RF has the lowest F1score. In comparison, XGBoost with technic one features was the best model for this problem. Tweet2vec performed better than Doc2vec and bag of words on all classifiers; This clearly shows the power of word embeddings in dealing with NLP problems.

6. CONCLUSION

This research offered a novel proposition for increasing the F1score of Twitter sentiment analysis . We examined lexicon polarity with both technic 1 based on tweet2vec and technic 2 based on doc2vec ; and features derived from a bag of words using two Twitter sentiment analysis datasets with four machine learning classifiers to assure the F1score of Tweets embeddings (XG, SVM, LR, RF).

According to the trial results, tweet2vec outperformed doc2vec and the traditional bag of words ;in all datasets, technique 1 in our proposed strategy boosted the F1score of twitter sentiment categorization tasks.In summary, the suggested approach has the following advantages:

(a) However, technique 1 in the suggested approach (IWVTSA) improved the F1-score of Twitter sentiment analysis for the first time.

(b) Using polarized lexicons in our research improved our suggested approach's F1score in all datasets.

(c) Any future enhancements to trained word embeddings (emoji embeddings, hashtag embeddings, etc.) will increase the F1-score of the proposed method.

As a result, our suggested approach can serve as the foundation for machine learning-based social media sentiment analysis tools. We intend to increase the quality of evaluation in the future by incorporating additional information such as emoji embeddings,

tags, or hashtag embeddings using more complex machine learning techniques.

As Recommender systems recently use the analysis of users reviews , sentiment analysis based recommendation is a main future direction for this research.

REFERENCES:

[1] A. ,Samih, A. Ghadi, & A.Fennan . (2021). ExMrec2vec: explainable movie recommender system based on Word2vec. Int. J. Adv. Comput. Sci. Appl.

[2] L. Yue, W. Chen,Xi. Li, and al. A survey of sentiment analysis in social media. Knowl Inf Syst 60, 617–663 (2019). https://doi.org/10.1007/s10115-018-1236-4.

[3] A. Sharma, & A. Daniels (2020). Tweets Sentiment Analysis via Word Embeddings and Machine Learning Techniques. arXiv preprint arXiv:2007.04303.

[4] I. Mozetič, M.Grčar, & J. Smailovič, (2016). Multilingual twitter sentiment classification: The role of human annotators. PloS One, 11(5), e0155036. doi:10.1371/journal.pone.0155036 Müller, W. C. (1993). Executive–Legislative relations in Austria: 1945– 1992. LegislativeStudies Quarterly, 18(4), 467– 494. doi:10.2307/439851.

[5] J. Kleinnijenhuis, F. Schultz, D. Oegema, & W. ,Van Atteveldt, (2013). Financial news and market panics in the age of high-frequency sentiment trading algorithms. Journalism, 14(2), 271–291. doi:10.1177/1464884912468375.

[6] L. Aaldering, & R. ,Vliegenthart, (2016). Political leaders and the media. Can we measure political leadership images in newspapers using computer-assisted content analysis? Quality and Quantity, 50(5), 1871–1905.

[7] W.,Van Atteveldt, J., Kleinnijenhuis, N., Ruigrok, & S. , Schlobach, (2008a). Good news or bad news? Conducting sentiment analysis on dutch text to distinguish between positive and negative relations. Journal of Information Technology and Politics, 5(1), 73–94. doi:10.1080/19331680802154145.

[8] J., Grimmer, & B. M. Stewart, (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. Political Analysis, 21(3), 267–297. doi:10.1093/pan/mps028.

[9] , P. D Turney., & P. Pantel, (2010). From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research, 37, 141–188.

[10] T. Mikolov, K. Chen, G. Corrado, , & J. Dean, (2013). Efficient estimation of word representations in vector space. CoRR. Retrieved from http://arxiv.org/abs/1301.3781 .

[11] T. Mikolov, I. Sutskever, K. Chen, , G. S. Corrado, & J.Dean, (2013, December).Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems (pp. 3111–3119), Lake Tahoe, CA.

[12] Q. Le, & T. Mikolov (2014, June). Distributed representations of sentences and documents. In International conference on machine learning (pp. 1188-1196). PMLR.

[13] M. Bilgin , and L Şentürk, "Sentiment Analysis on Twitter Data with Semi-Supervised Doc2Vec," in Proceedings of 2nd International Conference on Computer Science and Engineering, Antalya, pp. 661-666, 2017.

[14] Y. Arslan, D. Küçük, & A. Birturk, (2018, June). Twitter sentiment analysis experiments using word embeddings on datasets of various scales. In International Conference on Applications of Natural Language to Information Systems (pp. 40-47). Springer, Cham.

[15] E. M. Alshari, A. Azman, S. Doraisamy, N. Mustapha, & M. Alkeshr, (2018, March).Effective method for sentiment lexical dictionary enrichment based on Word2Vec for sentiment analysis. In 2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP) (pp. 1-5). IEEE.

[16] E. M. Alshari, , A. Azman, S.Doraisamy, N. Mustapha, , & M. Alkeshr, (2017, August).Improvement of sentiment analysis based on clustering of Word2Vec features. In 2017 28th international workshop on database and expert systems applications (DEXA) (pp. 123-126).IEEE.

[17] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, M. Stede, Lexicon-based methods for sentiment analysis, Computational Linguistics. 37 (2011) 267-307.

[18] A. Giachanou and F. Crestani, "Like It or Not: A Survey of Twitter Sentiment Analysis Methods," ACM Comput. Surv., vol. 49, no. 2, Jun. 2016. https://doi.org/10.1145/2938640

[19] S. Ge, H. Isah, F. Zulkernine, and S. Khan, "A scalable framework for multilevel streaming data analytics using deep learning," Proc. - Int. Comput. Softw. Appl. Conf., vol. 2, pp. 189–194, 2019. https://doi.org/10.1109/compsac.2019.10205

[20] G. Wang et al., "Building a replicated logging system with apache kafka," Proc. VLDB Endow., vol. 8, no. 12, pp. 1654–1655, 2015. https://doi.org/10.14778/2824032.2824063 .

[21] "Learn Hadoop - Big Data Analysis Framework." [Online]. Available: https://www.tutorialspoint.com/about/index.htm. [7] A. S. Hashmi and T. Ahmad, "Big Data Mining: Tools & Algorithms," Int. J. Recent Contrib. from Eng. Sci. IT, vol. 4, no. 1, pp. 36–40, 2016

[22] "Spark 101: What Is It, What It Does, and Why It Matters | MapR.".

[23] J. Akilandeswari and G. Jothi, "Sentiment classification of tweets with non-language features," Procedia Comput. Sci., vol. 143, pp. 426–433, 2018.

[24] M. Bouazizi and T. Ohtsuki, "Sentiment analysis in twitter: From classification to quantification of sentiments within tweets," 2016 IEEE Glob. Commun. Conf. GLOBECOM 2016 - Proc., 2016. https://doi.org/10.1109/glocom.2016.7842262

[25] P. Barnaghi, P. Ghaffari, and J. G. Breslin, "Opinion Mining and Sentiment Polarity on Twitter and Correlation between Events and Sentiment," Proc. - 2016 IEEE 2nd Int. Conf. Big Data Comput. Serv. Appl. BigDataService 2016, pp. 52–57, 2016. https://doi.org/10.1109/bigdataservice.2016.36

[26] A. Prabhat and V. Khullar, "Sentiment classification on big data using Naïve bayes and logistic regression," 2017 Int. Conf. Comput. Commun. Informatics, ICCCI 2017, 2017. https://doi.org/10.1109/iccci.2017.8117734

[27] A. P. Rodrigues and N. N. Chiplunkar, "A new big data approach for topic classification and sentiment analysis of Twitter data," Evol. Intell., no. 0123456789, 2019. https://doi.org/10.1007/s12065-019-00236-3

[28] F. Paquin, J. Rivnay, A. Salleo, N. Stingelin, and C. Silva, "Multi-phase semicrystalline microstructures drive exciton dissociation in neat plastic semiconductors," J. Mater. Chem. C, vol. 3, pp. 10715–10722, 2015. https://doi.org/10.1039/c5tc02043c

[29] F. Paquin, J. Rivnay, A. Salleo, N. Stingelin, and C. Silva, "Multi-phase semicrystalline microstructures drive exciton dissociation in neat plastic semiconductors," J. Mater. Chem. C, vol. 3, pp. 10715–10722, 2015. https://doi.org/10.1039/c5tc02043c.

[30] M. Byrkjeland, F. Gørvell de Lichtenberg, and B. Gambäck, "Ternary Twitter Sentiment Classification with Distant Supervision and Sentiment-Specific Word Embeddings," pp. 97–106, 2019. https://doi.org/10.18653/v1/w18-6215

[31] K. Ravi, V. Ravi, A survey on opinion mining and sentiment analysis: Tasks, approaches and applications, Knowledge-Based Systems. 89 (2015) 14-46.

[32] A. Sadeghian and A. R. Sharafat, "Bag of words meets bags of popcorn," 2015.

[33] A. Mudinas, D. Zhang, M. Levene, Combining Lexicon and Learning based Approaches for ConceptLevel Sentiment Analysis, In: Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining. 5 (2012).

[34] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu, B. Liu, Combining Lexicon-based and Learning-based Methods for Twitter Sentiment Analysis, Hewlett -Packard Development Company.(2011).

[35] A. ,Samih, A. Ghadi, & A.Fennan , (2020, December). Translational-Randomwalk Embeddings-Based Recommender Systems: A Pragmatic Survey. In International Conference on Advanced Intelligent Systems for Sustainable Development (pp. 957-966). Springer, Cham.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[36] A. Severyn, A. Moschitti, Twitter Sentiment Analysis with Deep Convolutional Neural Networks, In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. (2015) 959-962.

[37] J. Acosta, N. Lamaute, M. Luo, E. Finklestein and A. Cotoranu, Sentiment Analysis of Twitter Messages Using Word2Vec, Proceedings of Student-Faculty Research Day, CSIS, Pace University, Pleasantville, New York, 2017, 7pp.

[38] V. R. K. R. Bhaskar Dhariyal, "Sentiment analysis via Doc2Vec and Convolutional Neural Network hybrids," IEEE Symposium Series on Computational Intelligence (SSCI), pp. 666 - 671, 2018.

[39] Y. Zhang, B. Wallace, A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification, arXiv, 1510.03820v4. (2015).

[40] M. Kamkarhaghighi, M. Makrehchi, Content Tree Word Embedding for document representation, Expert Systems with Applications. 90 (2017) 241-249.

[41] M. ,Fauzi, Ali, D. Cahyo Utomo, E. Sakti Pramukantoro, and B. Darma Setiawan. "Automatic Essay Scoring System Using N-Gram And Cosine Similarity For Gamification Based E-Learning.".

[42] Pramukantoro, E. Sakti, and M. Ali Fauzi. "Comparative Analysis of String Similarity and Corpus-Based Similarity for Automatic Essay Scoring System on E-Learning Gamification." In Advanced Computer Science and Information Systems (ICACSIS), 2016 International Conference on, pp. 149-155. IEEE, 2016.

[43] A. Samih, A. Ghadi, & A.Fennan.: Deep graph embeddings in recommender systems: a survey. J. Theor. Appl. Inf. Technol. 99(15) (2021). https://doi.org/10.5281/zenodo.5353504.

[44] Q., Chen, & M.Sokolova, (2018). Word2vec and doc2vec in unsupervised sentiment analysis of clinical discharge summaries. arXiv preprint arXiv:1805.00352.

[45] T. Chen, T. He, M. Benesty,V. Khotilovich, Y. Tang, H. Cho, & K. Chen, (2015).Xgboost: extreme gradient boosting. R package version 0.4-2, 1(4), 1-4.

[46] V. Jakkula, (2006). Tutorial on support vector machine (svm). School of EECS,Washington State University, 37(2.5), 3.

[47] R. E. Wright, (1995). Logistic regression.

[48] V. Pal, (2005). Random forest classifier for remote sensing classification. International journal of remote sensing, 26(1), 217-222.

[49] Kaggle datasets, https://www.kaggle.com/datasets.