Sādhanā (2020)45:125
https://doi.org/10.1007/s12046-020-01359-5

A novel Tag Score (T_S) model with improved K-means for clustering tweets

S POOMAGAL*, B MALAR, J INAMUL HASSAN and R KISHOR

Department of Applied Mathematics and Computational Sciences, PSG College of Technology, Coimbatore, India
e-mail: poomagal.swamy@gmail.com; rsakthimalar@gmail.com; inamulhassan9698@gmail.com; sarankrishi198@gmail.com

MS received 17 July 2018; revised 23 January 2020; accepted 23 February 2020

Abstract. Clustering of tweets is useful for analyzing the attitudes of people towards a particular product. The companies can use this analysis to modify their products to meet the needs of people. Recently, K-means clustering is widely used to cluster the tweets with bag of words as a feature set. The key factors contributing to the quality of clusters and performance of clustering are dimensionality reduction and initial selection of centroids. This paper addresses these issues using a newly proposed Tag Score (T_S) model with improved K-means in which semantically similar features from bag of words are grouped into tags, scores are modified based on sentiment polarity values and the initial centroids are selected with the help of sentiment scores. The performance of the proposed T_S model with improved K-means is compared with T_S model with random K-means and conventional word vectors with random K-means by considering three labeled datasets and three unlabeled datasets. The results show that the proposed method produces significant results in approximately 70% of the cases in terms of purity, F-measure, intra-cluster distance and inter-cluster distance.

Keywords. Clustering; K-means; tweets; sentiment analysis; opinion mining; Sentiwordnet.

1. Introduction

Twitter is one of the popular social media sites wherein people post tweets to show their likes and grievances on a particular policy or product. Companies can analyze these tweets to understand the acceptance or rejection of their products by their customers. This analysis can then be used in decision making to improve their business.

Users who registered with Twitter can only post tweets and the number of characters in each tweet can be at most 280. In addition to the text content, it also has other parts in it such as hash tag, URL and username. Hash tag contains keywords which are used to mention the topic on which the tweet is posted. URL is used for analytical purposes.

Natural language processing and machine learning play a major role in identifying the part of the text content which needs to be considered for analysis and in finding the meaning of the words (synset) present in it with their polarity values (Sentiwordnet) as positive, negative and neutral. This classification is essential in many scenarios like, to get public's opinion on government's policies and to review them based on their feedback. Existing techniques in the literature extract only the words in tweets for clustering and they used Wordnet and Sentiwordnet for finding domain specific tweets [1]. Meaning of the words but not a specific domain [1] can also be considered to reduce the number of words to be considered for clustering. This needs to be done since the corpus is larger which in turn produces more number of features and takes much time to process.

Clustering is an unsupervised learning method which takes the data points as input and produces groups of data points based on similarity among the data points. It can also be applied on data points to detect the outliers [2]. From the existing clustering algorithms, K-means algorithm works well on large datasets. In the realm of natural language processing/sentiment analysis, the selection of initial centroids and the dimensionality of dataset determine the quality of clusters and performance of clustering algorithm. Hence, dimensionality reduction and initial centroid selection play a vital role and need to be addressed.

This paper addresses these issues by introducing a new dimensionality reduction technique called tag score model and a heuristic for selecting initial centroids. Dimensionality reduction technique transforms conventional bag of words model by combining semantically related words into tags and computes a new sentiment score based on the reduced features to find the significance of each word/tag with respect to polarity. Furthermore, the proposed model modifies the existing TFIDF scores using the calculated

*For correspondence

Published online: 15 May 2020

sentiment scores. The modified TFIDF scores are then used to select initial centroids for K-means clustering algorithm.

The key contributions of the proposed work are four fold:

1. It introduces a new T_S model which reduces the number of words by grouping the semantically related words under a tag. Hence, the dataset contains tags and words as features.
2. It also contributes a new method to compute sentiment score for each tag and untagged words through which the strength of clustering is improved. In addition, it presents a new way of modifying TFIDF scores by combining traditional TFIDF scores and sentiment scores.
3. It proposes a new heuristic for calculating initial centroids for K-means clustering algorithm using polarity of the tweets.
4. The performance of the proposed T_S model using improved K-means and random K-means is compared with conventional TF and TFIDF vector values and analyzed through experiments.

The remainder of the paper is organized as follows: Section 2 discusses various techniques in the area of tweet sentiment analysis. Section 3 states the problem. Section 4 explains different stages in the proposed method. Section 5 presents the extensive results of experiments conducted with proper findings. Section 6 concludes the paper with the summary.

2. Related work

Due to the large volume of social media data, it is necessary to develop an automated clustering system for sentiment analysis or opinion mining from it. This clustering task can be done using various parts of the text processed using natural language processing techniques.

Various levels of text part have been analyzed by researchers such as word or phrase level [3], sentence level [4] and document level [5]. Word level analysis is performed to find how much a word can contribute to the polarity of the entire text. The computed polarity values are combined to find the opinion of the entire sentence, whereas entire sentence or document level tells single polarity value for the sentence or document level based on the way the words are arranged in it. These techniques have not used any machine learning algorithm for finding the polarity of the tweets. Instead, they identified the polarity based on the polarity of the words present in the tweets and combined the values to find the polarity of the entire tweet.

Most important section of tweets for sentiment words is noun, adjective, adverb and verb. Karamibekr and Ghorbani [6] used only verbs instead of nouns, adverbs and adjectives for analyzing the sentiments in social media contents. Using verbs, opinion structure is found and sentiment orientation is identified from the formed opinion structure. In this work, opinion structure is formed by finding the opinion verb which is considered as the core of the sentence. There are many ways in which these sentiment words can be extracted namely, machine learning based [7], lexicon based [1] and combination of both [8].

Three methods for extracting opinion word are proposed by Duric and Song [9] which include Hidden Markov Model-Latent Dirichlet Allocation (HMM-LDA) to get syntactic classes of features, syntactic and semantic classes based features and max scores based features. Cruz [1] defined a set of domain-specific resources to extract the opinion words. Domain-specific resource consists of a set of features such as feature-taxonomy, feature clues and dependency patterns. They utilized dictionary-based approach such as WordNet, PMI, and SentiWordNet based classifier for sentiment classification. The lexicon was expanded using random walk algorithm. This work used SentiWordNet to extract sentiment clues and sentiment topics specific to a domain and they have not grouped the words into tags and improved the score of the words. Instead, they considered only the sentiment clues and sentiment topics as features in the vector space model. Zhai *et al* [10] proposed a method to extract different opinion features such as sentiment term, substring, substring-group and key-substring group features. The collection of sentiment words with their respective polarity value can be formed as a lexicon. This is done by first selecting the initial set of seed words and including synonyms and antonyms of them in the list of words [11].

Go *et al* [12] used emoticons and hashtags for constructing the learning set and Davidov *et al* [13] utilized emoticons as class labels to find polarity of the tweets. Lexicons produced from the emoticons and the others produced by only textual features are compared by Boia *et al* [14]. Emoticons are the groups of characters used to express the feelings of users such as representing happiness using :). These emoticons provide greater information in performing sentiment analysis. Hence, the authors have used both emoticons and text and stated the one which provides better results.

An analysis of tweets on light rail service in Log Angles is done by Luong *et al* [15] to understand the public opinions. They did the analysis in features of sentiment analysis, topic modeling and the interaction between posters and retweeters. The topics of tweets are identified by using unigram model with K-Medoids clustering algorithm.

Several studies on clustering tweets for identifying the theme of tweets are available in the literature. Clustering-based sentiment analysis has been improved by Li *et al* [8] in two ways. First way is to apply opposite opinion contents processing and non-opinion contents processing techniques to further enhance accuracy. Second way is by using a modified voting mechanism and distance measurement method to conduct fine-grained sentiment analysis. They also proved that this method outperforms other supervised learning approaches.

Luiz *et al* [16] introduced a new algorithm namely C^{3}E-SL by combining classifier and cluster ensembles. This algorithm can refine tweet classifications from additional information provided by clusterers, assuming that similar instances from the same clusters are more likely to share the same class label.

Fernandez *et al* [17] proposed a method for sentiment analysis in online texts based on an unsupervised dependency parsing-based text classification method that leverages a variety of natural language processing techniques and sentiment features primarily derived from sentiment lexicons. These lexicons were created by means of a semiautomatic polarity expansion algorithm in order to improve accuracy in specific application domains.

SVMs were used by Liu *et al* [18] as a sentiment polarity classifier. Unlike the binary classification problem, they argued that opinion subjectivity and expresser credibility should also be taken into consideration. They proposed a framework that provides a compact numeric summarization of opinions on microblogs platforms. They identified and extracted the topics mentioned in the opinions associated with the queries of users, and then classified the opinions using SVM. They worked on twitter posts for their experiment. They found out that the consideration of user credibility and opinion subjectivity is essential for aggregating microblog opinions. They proved that their mechanism can effectively discover Market Intelligence (MI) for supporting decision-makers by establishing a monitoring system to track external opinions on different aspects of a business in real time.

Orkphol and Yang [19] used K-means algorithm for clustering microblogs by selecting the initial set of centroids using artificial bee colony algorithm. Many heuristics [20–22] are introduced to calculate initial centroids based on distance among the data points.

In the existing works on clustering tweets, reducing the feature set by grouping words into tags based on semantic similarity is not done. Instead, words which expresses good sentiment value has been taken as sentiment clue. In addition, many works concentrated on directly finding the polarity of the tweets based on the sentiment of the words present in the tweets but not using machine learning algorithms. Existing works have not concentrated on improving the clustering algorithm by selecting the initial centroids using polarity of the tags formed and the words in the tweets.

These motivated us to propose a new model for reducing the number of features in the collection and for calculating the initial centroids. This paper proposes an improved K-means clustering algorithm for grouping tweets into clusters, based on a newly introduced Tag Score (T_S) model. The basic idea behind this algorithm is that the conventional bag of words are transformed into reduced form of bag of words by combining the related words. Combined words and non-related words are then assigned a sentiment score through which initial centroids are identified. The major objectives of the proposed work are, to reduce the number of features in the corpus and to produce the optimum tweets in the clusters that is to generate right tweets in the right cluster.

3. Problem statement

Given a set of labeled or unlabeled tweets, the proposed method aims to group these tweets into clusters with minimum intra-cluster distance. This is done by first modifying the word vector values using semantic similarity among the words in the collection and sentiment scores of the words. In addition, a new method is proposed to select the initial centroids for K-means clustering algorithm by using the sentiment scores of the words present in the tweets.

4. Proposed T_S model with an improved K-means clustering

The proposed method clusters the tweets as positive, negative or neutral using improved K-means clustering algorithm after tagging the words and scoring the tags and untagged words.

The different stages involved in the proposed scheme are as follows,

i. Preprocessing
ii. Tagging
iii. Word vector formation
iv. Scoring
v. Improved K-means

4.1 *Preprocessing*

Preprocessing of tweets is done to remove unwanted items from the tweets and to retain only the needed text part for analysis. After extracting the text of the tweets, punctuations and stopwords occurring in it are removed. Let $T = \{t_i\}_{i=1}^{n}$ represents n tweets in the tweet set, $WO_i = \{wo_{ij}\}_{j=1}^{nw_i}$ be the set of words in tweet i, Let $W = \{w_i\}_{i=1}^{m}$ be the set of words collected from the tweets after preprocessing.

We have also collected polarity words from the labeled dataset and formed three different sets namely, $PW = \{pw_i\}_{i=1}^{pc}$, $NW = \{nw_i\}_{i=1}^{nc}$ and $NEW = \{new_i\}_{i=1}^{nec}$ representing the set of positive words, negative words and neutral words respectively. These polarity words are extracted from the dataset itself for this work based on the labels of the tweets. When an unlabeled dataset is considered, the polarity words can be collected initially from the internet.

4.2 *Tagging*

This paper introduces a new way of tagging using semantic similarity among words in the tweets to reduce the size of the feature set. This task is done by collecting synonym set for each word containing the lemma, hypernym, hyponym, holonym, meronym and troponym is formed for every word by communicating with Wordnet (2010). Once semantic information for each word is retrieved, semantically similar words are replaced using a common tag. It is performed by comparing synonym set of each pair of words and placing both the words in a same tag when at least one word in their sets is matched. The purpose of doing this is different users may present the same opinion using various words with the same meaning. Hence, all the words with the same meaning can be represented using same tag instead of using them separately. It is mathematically defined as follows.

Let *SS* represents the synsets of *m* words in the tweets, s_i represents the synset of word *i, TAG* has the set of tags, tag_k be the k^{th} tag containing the words with same meaning, wo_{ij} represents the j^{th} word in i^{th} tweet. If s_i of i^{th} word in one tweet is the subset of s_j of j^{th} word in the same tweet or other tweets, then all the occurrences of word *i* and word *j* are replaced by using a single target tag. Now, we formalize the above explanation as follows using Eqs. (1)–(6).

$$SS = \{\ s_i | s_i = synset(w_i)\}_{i=1}^{m} \tag{1}$$

$$synset(w_i) = \left\{ FM \middle| \begin{array}{c} FM \in \{hypernym(f) \cup hyponym(f) \cup holonym(f) \cup \\ meronym(f) \cup Lemma(f)\} \end{array} \right\} \tag{2}$$

$$r_{ij} = s_i \cap s_j \ where\ i \neq j, 1 \leq i \leq m \ and\ 1 \leq j \leq m \tag{3}$$

$$\begin{array}{l} TAG = \{tag_k\}_{k=1}^{x}, \\ \text{where } x \text{ is the number of tags to be generated} \end{array} \tag{4}$$

$$\begin{array}{l} \forall k\ tag_k = \emptyset \\ tag_k = tag_k \cup \{w_i, w_j\} \ \ if\ r_{ij} \geq 1, w_i \notin tag_k \ and\ w_j \notin tag_k \end{array} \tag{5}$$

$$wo_{ij} = \begin{cases} tag_k & if\ wo_{ij} \in tag_k \ and\ 1 \leq k \leq x \\ No\ change & Otherwise \end{cases} \tag{6}$$

Words *i* and *j* are combined into same tag if their synset similarity is greater than or equal to 1 and both the words are not assigned with tags. If any one word is assigned with the tag and the other one is not, then tag is assigned for it.

After tagging the words, the size of the word set is defined as the number of tags represented as *x* plus the count of remaining words *r*. *r* is defined as in Eq. (7). y_k represents word count of k^{th} tag.

$$r = m - \sum\nolimits_{k=1}^{x} y_k \tag{7}$$

The tagging process results a set of tags and words that are not related to any of the tags.

4.3 *Word vector formation*

Once tagging is done, each resultant tweet has set of tags and the remaining untagged words in it. Word vector is formed by having tweets as rows and tags and words as columns. Importance score for each tag and word in every tweet is calculated. The simplest way to identify the score of a tag or a word is to count the occurrence of it inside each tweet. This is called as Term Frequency (TF). Vector space model generated using these TF values is called as bag of words model. The drawback of this model is that more common words such as "get" may receive more weight. To avoid this problem, an Inverse Document Frequency (IDF) factor is also included which reduces the weight of the words that occur very frequently in the collection and raises the weight of the words that does not occur frequently. In this work, both TF and TFIDF vectors are used.

4.4 *Scoring*

The proposed work introduces a new novel scheme to modify the existing TFIDF values by computing another score for each tag and word based on the sentiment of it. For each tag, three sets of scores namely, positive score set, negative score set and neutral score set are formed. Positive score set includes the sentiment positive score of each word present in the tag returned by Sentiwordnet (2010). Similarly, negative and neutral score sets have the sentiment negative score and sentiment neutral score of each word present in the tag.

Then the positive scores of all the words in the tag which also occur in *PW* are summed up to find positive score of the tag and negative and neutral scores are computed similarly by adding up negative and neutral scores respectively. When the tag has more number of positive words (negative/neutral) occurring in it, positive (negative/neutral) score of the tag is added with *TF* vector and *TFIDF* vector values.

In addition, score for untagged words is found and included in word vectors by first retrieving its sentiment scores from Sentiwordnet (2010), identifying the largest among its three scores and adding it with *TF* and *TFIDF* vector values. The process of finding tag scores and untagged word scores and updating *TF* and *TFIDF* vectors is defined as follows.

4.4a *Word vector updation using tag scores*: Let $TAG = \{tag_k\}_{k=1}^{x}$ be the tags generated during tagging step, tag_k represents the tag *k* with y_k as the number of words in it and it is defined in Eq. (8). pos_score_k, neg_score_k and neu_score_k be the set of positive score, negative score and neutral score sets of the k^{th} tag respectively. The above explanation is mathematically represented using Eqs. (9)–(11).

$$tag_k = \{w_{kj} | w_{kj} \in tag_k\}_{j=1}^{y_k} \tag{8}$$

$$pos\_score_k = \{ps_{kj} | ps_{kj} = sentiposscore(w_{kj})\}_{j=1}^{y_k}. \tag{9}$$

$$neg\_score_k = \{ns_{kj} | ns_{kj} = sentinegscore(w_{kj})\}_{j=1}^{y_k}. \tag{10}$$

$$neu\_score_k = \{nes_{kj} | nes_{kj} = sentineuscore(w_{kj})\}_{j=1}^{y_k} \tag{11}$$

where w_{kj} is the j^{th} word present in k^{th} tag and ps_{kj}, ns_{kj} and nes_{kj} are the positive, negative and neutral scores of j^{th} word present in k^{th} tag respectively.

Let p_count_k, n_count_k and ne_count_k be the number of words in the tag k belonging to positive words list (*PW*), negative words list (*NW*) and neutral words list (*NEW*) respectively. They are computed as in Eqs. (12)–(15).

$$\forall k \; p\_count_k = n\_count_k = ne\_count_k = 0 \tag{12}$$

$$p\_count_k = \sum_{j=1}^{y_k} 1(w_{kj} \in PW). \tag{13}$$

$$n\_count_k = \sum_{j=1}^{y_k} 1(w_{kj} \in NW). \tag{14}$$

$$ne\_count_k = \sum_{j=1}^{y_k} 1(w_{kj} \in NEW) \tag{15}$$

where 1(A) returns 1 if A is true. Otherwise it returns false.

Let p_sum_k, n_sum_k and ne_sum_k be the sum of respective scores of the words in the tag k belonging to *PW*, *NW* and *NEW* respectively. It is calculated using Eqs. (16) and (19). Finally, count and sum values are used to modify *TF* and *TFIDF* scores using Eq. (20). Mathematical representation of the above explanation is presented as follows.

$$\forall k \; p\_sum_k = n\_sum_k = ne\_sum_k = 0 \tag{16}$$

$$p\_sum_k = \begin{cases} p\_sum_k + ps_{kj} & if \; w_{kj} \in PW \; and \; 1 \le j \le y_k \\ No \; change & Otherwise \end{cases} \tag{17}$$

$$n\_sum_k = \begin{cases} n\_sum_k + ns_{kj} & if \; w_{kj} \in NW \; and \; 1 \le j \le y_k \\ No \; change & Otherwise \end{cases} \tag{18}$$

$$ne\_sum_k = \begin{cases} ne\_sum_k + nes_{kj} & if \; w_{kj} \in NEW \; and \; 1 \le j \le y_k \\ No \; change & Otherwise \end{cases} \tag{19}$$

$$TF_{\boldsymbol{ij}} = \begin{cases} TF_{ij} + p\_sum_j \;\; if \;\; p\_count_j = \max(p\_count_j, n\_count_{j,} ne\_count_j), TF_{ij} \ne 0 \; and \; 1 \le j \le x \\ TF_{ij} + n\_sum_j \;\; if \;\; n\_count_j = \max(p\_count_j, n\_count_{j,} ne\_count_j), TF_{ij} \ne 0 \; and \; 1 \le j \le x \\ TF_{ij} + ne\_sum_j \;\; if \;\; ne\_count_j = \max(p\_count_j, n\_count_{j,} ne\_count_j), TF_{ij} \ne \; 0 and \; 1 \le j \le x \end{cases} \tag{20}$$

In the same way, $TFIDF_{\boldsymbol{ij}}$ is also altered by adding p_sum_j, n_sum_j and ne_sum_j with it if it is not equal to 0.

4.4b *Word vector updation using word scores:* Let r be the number of remaining untagged words and the scores are computed for these words by retrieving the positive, negative and neutral scores for each word and adding the highest score with respective *TF* and *TFIDF* values. Let $WORDS = \{Word_Z\}_{Z=1}^{r}$ be the remaining untagged words in the word set. Let *psw*, *nsw* and *nesw* be the positive, negative and neutral scores of the untagged words respectively. These are defined in Eqs. (21)–(23).

$$psw_z = sentiposscore(word_z), 1 \le z \le r. \tag{21}$$

$$nsw_z = sentinegscore(word_z), 1 \le z \le r \tag{22}$$

$$nesw_z = sentineuscore(word_z), 1 \le z \le r \tag{23}$$

Once the score values are retrieved from sentiwordnet, the highest score for each word is added with its respective *TF* and *TFIDF* values. Modified *TF* and *TFIDF* scores are obtained as given in Eq. (24) by adding maximum of three scores with the existing *TF* and *TFIDF* scores.

$$\begin{aligned} &TF_{\boldsymbol{ij}} = TF_{\boldsymbol{ij}} + \max(psw_j, nsw_j, nesw_j), where \\ &TF_{\boldsymbol{ij}} \ne 0 \text{ and } 1 \le j \le r \end{aligned} \tag{24}$$

4.5 *Improved K-means algorithm*

Once the word vectors are updated, the proposed improved K-means clustering algorithm is applied to divide the set of tweets as positive, negative or neutral. There are two inputs for K-means clustering algorithm: number of clusters and the set of initial centroids. The number of clusters in this work is fixed as 3 since the sentiments considered are positive, negative and neutral. The proposed method identifies the initial centroids using a new approach which computes a heuristic for each tweet. The heuristic is a set of 3 values one for each cluster and it states the proportion of tags and words contributes for each class/cluster.

This is achieved by counting the number of tags with more number of positive (negative/neutral) words in it and the number of positive (negative/neutral) untagged words. Then the tag count and word count are added to identify the polarity of the tweet. That is, a tweet is assigned with positive polarity value when the respective added count is higher than the remaining two polarity counts. Once the tweets are marked, the average of the vectors of each category is found and used as the centroids. This process is explained as follows.

Let $tag\_pos\_cnt_i, tag\_neg\_cnt_i$ *and* $tag\_neu\_cnt_i$ be the count of positive, negative and neutral tags present in tweet *i*, p_count_k, n_count_k *and* ne_count_k be the number of positive, negative and neutral words present in tag *k* and $word\_pos\_cnt_i, word\_neg\_cnt_i$ *and* $word\_neu\_cnt_i$ be the number of positive, negative and neutral words in tweet *i*. They are represented in Eqs. (25)–(32).

$$\forall k \; tag\_pos\_cnt_i = tag\_neg\_cnt_i = tag\_neu\_cnt_i = 0 \tag{25}$$

$$tag\_pos\_cnt_i = \left\{ tag\_pos\_cnt_i + 1 \;\; if \;\; p\_count_k = max\left(p\_count_k, n\_count_k, ne\_count_k\right), TF_{ik} \neq 0 \;\; and \;\; 1 \leq k \leq x \right. \tag{26}$$

$$tag\_neg\_cnt_i = \left\{ tag\_neg\_cnt_i + 1 \;\; if \;\; n\_count_k = max\left(p\_count_k, n\_count_k, ne\_count_k\right), TF_{ik} \neq 0 \;\; and \;\; 1 \leq k \leq x \right. \tag{27}$$

$$tag\_neu\_cnt_i = \left\{ tag\_neu\_cnt_i + 1 \;\; if \;\; ne\_count_k = max\left(p\_count_k, n\_count_k, ne\_count_k\right), TF_{ik} \neq 0 \;\; and \;\; 1 \leq k \leq x \right. \tag{28}$$

$$\forall k \; word\_pos\_cnt_i = word\_neg\_cnt_i = word\_neu\_cnt_i = 0 \tag{29}$$

$$word\_pos\_cnt_i = \begin{cases} word\_pos\_cnt_i + 1 \;\; if \;\; psw_j = \max\left(psw_j, nsw_j, nesw_j\right), \\ \qquad TF_{ij} \neq 0 \;\; and \;\; 1 \leq j \leq r \end{cases} \tag{30}$$

$$word\_neg\_cnt_i = \begin{cases} word\_neg\_cnt_i + 1 \;\; if \;\; nsw_j = \max\left(psw_j, nsw_j, nesw_j\right), \\ \qquad TF_{ij} \neq 0 \;\; and \;\; 1 \leq j \leq r \end{cases} \tag{31}$$

$$word\_neu\_cnt_i = \begin{cases} word\_neu\_cnt_i + 1 \;\; if \;\; nesw_j = \max\left(psw_j, nsw_j, nesw_j\right), \\ \qquad TF_{ij} \neq 0 \;\; and \;\; 1 \leq j \leq r \end{cases} \tag{32}$$

For each tweet, tag count and word count are summed up for all three sentiments and the tweet is assumed to be of positive polarity if the positive count is higher than negative and neutral counts. Then the tweet vectors of positive polarity are averaged to obtain positive centroid and the same thing is done for negative centroid and neutral centroid using negative polarity and neutral polarity respectively.

5. Experimental analysis

To evaluate the effectiveness of the proposed work, experiments are conducted using three labeled datasets and three unlabeled datasets and the results are reported. The datasets are collected from Kaggle.com. All the datasets considered in this work are the applications of sentiment-based processing since they represent positivity, negativity and neutrality of the services provided or of the schemes introduced. Tweets in airline dataset describe passengers' experience in using a particular airline. Debate dataset include tweets about debate in Ohio and Twitter sanders dataset was created during sentiment analysis competition. Demonetization, Oscars and US army datasets have tweets representing the likes, dislikes and neutral views of citizens about the scheme, about the winners of Oscars and on the army operations in US respectively. Descriptions of labeled and unlabeled datasets are shown in table 1 and table 2 respectively.

5.1 *Set up*

Experiments are conducted by executing K-means in the following three scenarios:

Table 1. Description of labeled datasets.

Dataset	Positive	Negative	Neutral	Total
Airline	1000	1000	1000	3000
Debate	1000	1000	1000	3000
Twitter sanders	449	558	2228	3235

Table 2. Description of unlabeled datasets.

Data set	Number of tweets
Demonetization	14940
Oscars	29427
US army	50000

M1 – Conventional TF and TFIDF with random K-means
M2 – Proposed T_S model with random K-means
M3 – Proposed T_S model with Improved K-means

In each scenario, experiments are conducted five times and the results are presented. The performance of M3 is compared with the average of five runs of M1 and M2. The measures used for comparison are intra-cluster distance (L1), inter-cluster distance (L2), F-measure and Purity. For unlabeled datasets, only L1 and L2 are computed.

5.2 *Results of tagging*

The words from preprocessing step are tagged by looking at the synonyms and the results are presented in table 3. It contains dataset name as the first column (C1), number of distinct words before reduction as the second column (C2), number of tags and distinct words after reduction as the third column (C3), number of tags generated as the fourth column (C4), number of untagged words as the fifth column (C5) and percentage of reduction as the sixth column (C6). Results show that the number of unique words used as features in the proposed model is reduced to half of the traditional bag of words model. This is due to the fact that the T_S model combines the words with similar meaning and reduces the number of words.

5.3 *Comparative study*

For comparison, K-means algorithm is executed with five random sets of centroids and the results are taken. Output of all five runs for all labeled datasets is presented in figures 1–3. TF, TFIDF, TF_SCORE and TFIDF_SCORE in the figures present the outputs of five runs using M1 (TF and TFIDF) and M2 (TF and TFIDF) respectively. It is observed that the results of clustering experiments show that variation of intra-cluster distance is high in TFIDF and low in TFIDF_SCORE. This is due to the fact that the tag score reduces the number of features in clustering as well as the intra-cluster distance consistently across all the five experiments.

Table 3. Reduction in bag of words after tagging.

C1	C2	C3	C4	C5	C6
Airline	1640	744	44	700	54.63
Debate	1854	900	54	846	51.46
Twitter sanders	1849	837	39	798	54.73
Demonetization	2660	928	371	557	65.11
Oscars	3118	1400	351	1049	55.09
Us army	7440	1881	880	1001	74.72

* C1 – Dataset name * C2 – Number of distinct words before reduction
* C3 – Number of tags and distinct words after reduction
* C4 – Number of tags * C5 – Number of untagged words
* C6 – Percentage of reduction

In Airline dataset, the highest variation of intra-cluster distance and inter-cluster distance is obtained in TFIDF and the minimum variation of intra-cluster distance is in TFIDF and inter-cluster distance is in TF model. Based on purity and F-measure, all the models give almost consistent results.

In debate dataset, there is a minimum variation in intra-cluster distance across all five experiments and there is no variation in inter-cluster distance across the experiments. There is a less variation when considering purity and F-measure. Similarly, in Twitter Sanders dataset, there is a reduction in intra-cluster distance and increase in inter-cluster distance in case of both TF_SCORE and TFIDF_SCORE. When purity is considered, all the methods have produced almost same results and in case of F-measure, TF_SCORE give better results in two out of five runs.

Figures 4–6 show the results for unlabeled datasets and we projected only intra-cluster distance and inter-cluster distance as the other two measures require a dataset with labels. In demonetization dataset, there is a degradation in TFIDF, TF_SCORE and TFIDF_SCORE in intra-cluster distance when comparing with TF. In inter-cluster distance, three runs have less variation and other two have higher variation and TFIDF_SCORE produces higher value than other two techniques for four out of five cases.

For Oscars dataset, TFIDF gives almost similar values for five runs and TF and TFIDF_SCORE methods give high intra-cluster distance for two out of five cases when compared to other two methods. When considering inter-cluster distance, three methods except TFIDF give nearby values for many cases. For US army dataset, conventional TFIDF with random K-means is the least performing method in both intra-cluster distance and inter-cluster distance for most of the cases. TFIDF after scoring performs better than the other methods for three cases.

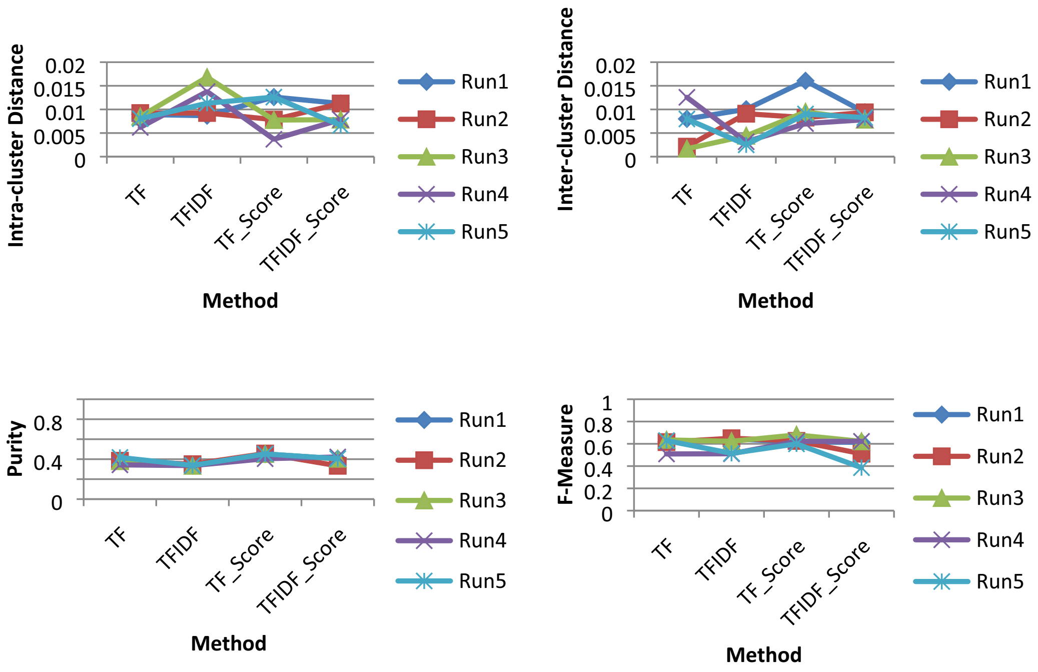

Figure 1. Comparative study of different measures for Airline dataset.

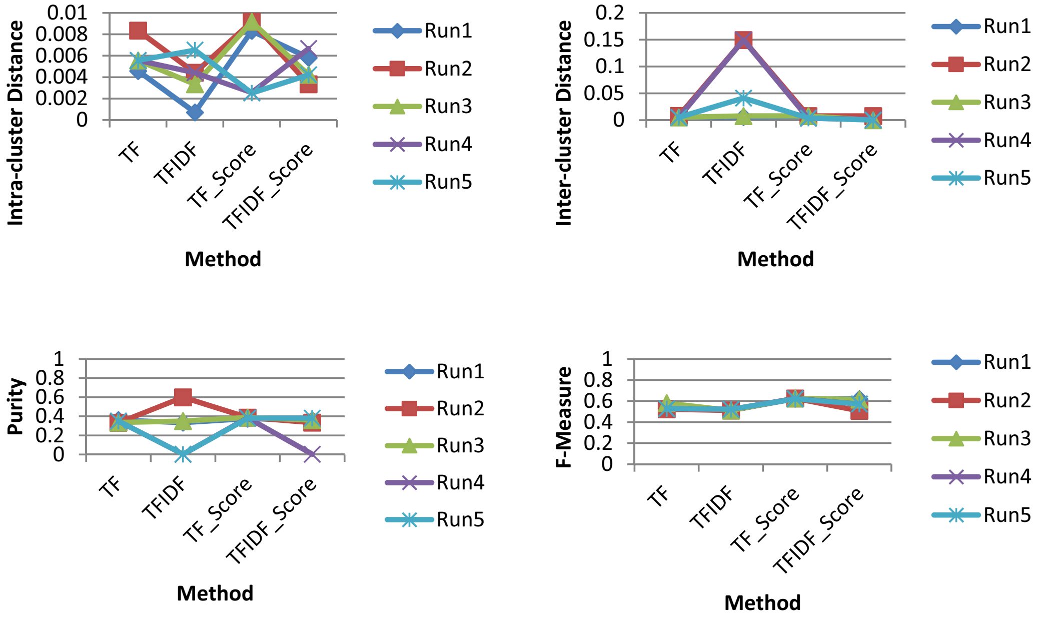

Figure 2. Comparative study of different measures for Debate dataset.

These unstable variations in the results are due to the random selection of initial centroids. This has been avoided by executing improved K-means by calculating the initial centroids using sentiment scores of the tweets. The result of this execution is compared with average of five runs using conventional TF and TFIDF and TF_SCORE and TFIDF _SCORE. It is presented in tables 4–7. After tagging and scoring, there is a little reduction in intra-cluster distance and an increase in inter-cluster distance. It is further improved using new centroid selection technique. 11 out of

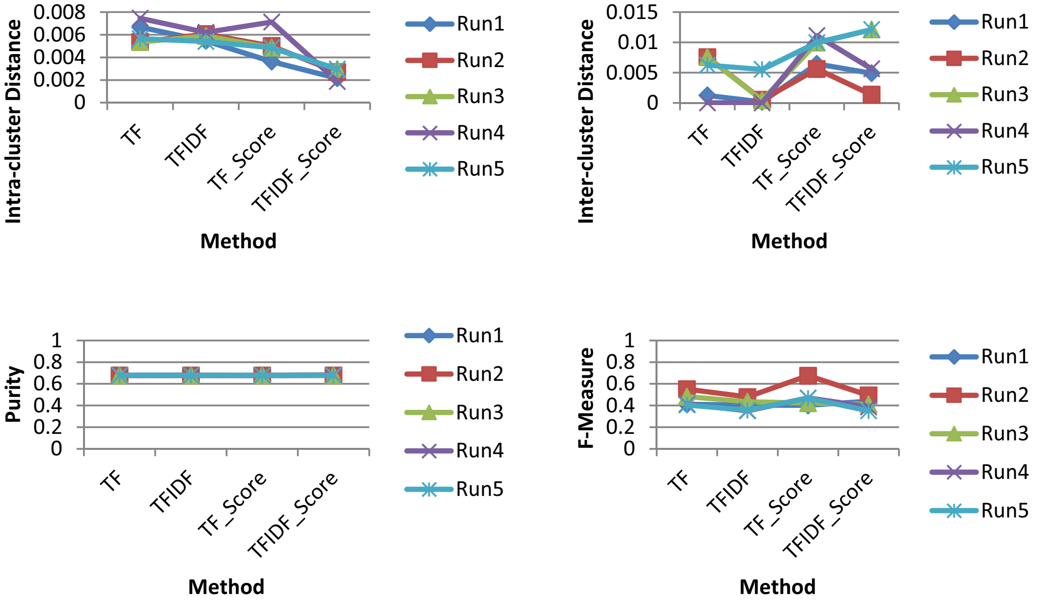

Figure 3. Comparative study of different measures for Twitter sanders dataset.

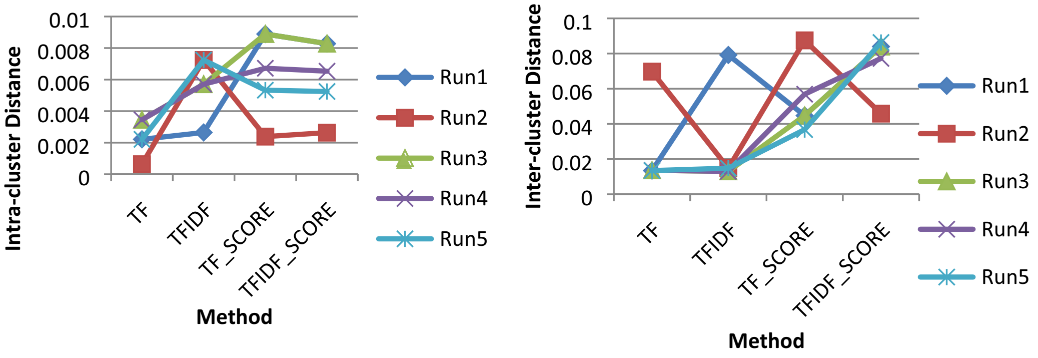

Figure 4. Comparative study of different measures for Demonetization dataset.

12 results from table 4 have shown that the proposed T_S model with improved K-means produced better results than conventional TF and TFIDF vectors. This accounts to 91.6% of the results having lesser intra-cluster distance and higher inter-cluster distance for the proposed method. The proposed Tag Score model combined the similar words into tags and reduced number of features. Furthermore, the new proposed heuristic for centroid selection tries to minimize distance between tweets in cluster and maximize the distance between the clusters.

Purity of the clusters generated by the proposed method is compared with the purity of the clusters produced using initial set of terms. When comparing the results of M1 with M2, the purity gets improved for five out of six cases. That is for 83.33% of the cases, M2 outperforms M1. It is also proved that M3 beats M1 for 83.33% cases and M2 for 66.67% of the cases. Overall, M3 is the winner for four cases, M2 for one case and M1 for one case. It is shown in table 5. When taking F-measure, M2 and M3 have produced better results for five cases and M1 for one case. When M1 and M2 are compared, 83.33% of the results by M2 is higher than M1. In addition, M3 gives highF-measure for the same number of cases. To summarize, M3 wins in three cases, M2 in two cases and M1 in one case. It is presented in table 6. This proves that the proposed T_S model with improved K-means (M3) performs very well. The reason for this improvement is the process of tagging and the inclusion of sentiment scores for altering the word weights and for computing initial centroids.

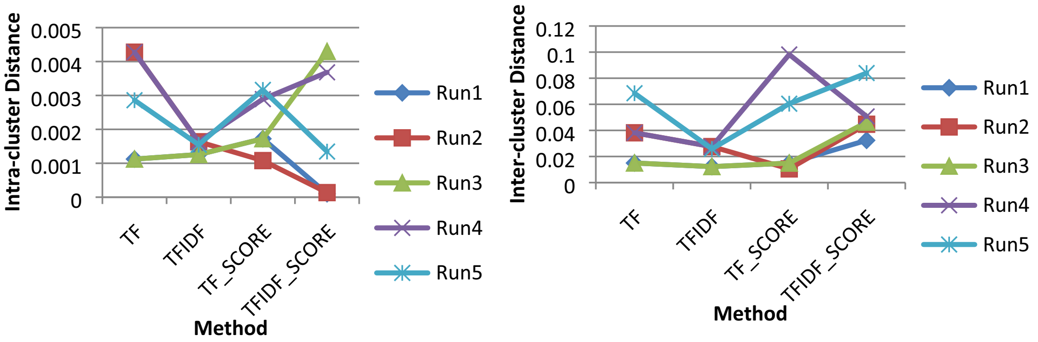

Figure 5. Comparative study of different measures for Oscars dataset.

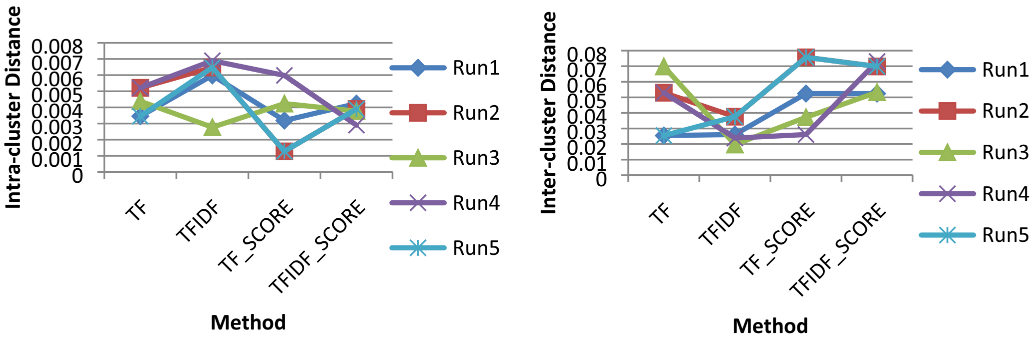

Figure 6. Comparative study of different measures for US Army dataset.

Table 4. Intra- and inter-cluster distances for labeled datasets.

Dataset	Representation	Distance measure	M1	M2	M3	Winner
Airline	TF	**L1**	**0.008176109**	**0.008897361**	**0.007330184**	**M3**
		L2	**0.006464904**	**0.00996642**	**0.012411449**	**M3**
	TFIDF	**L1**	**0.011962227**	**0.008960953**	**0.002058981**	**M3**
		L2	**0.005838849**	**0.008535183**	**0.023384695**	**M3**
Debate	TF	**L1**	**0.014173591**	**0.006354095**	**0.003864335**	**M3**
		L2	**0.005842545**	**0.006004139**	**0.009178070**	**M3**
	TFIDF	L1	0.003886123	0.004854638	0.004418103	M1
		L2	**0.062520253**	**0.003105773**	**0.149481689**	**M3**
Twitter sanders	TF	**L1**	**0.006103641**	**0.005096607**	**0.0010576769**	**M3**
		L2	0.004528902	0.00861677	0.0069023932	M2
	TFIDF	**L1**	**0.005805628**	**0.002535597**	**0.0016207455**	**M3**
		L2	0.001337378	0.007226722	0.0061408248	M2

*Bold values indicate that the respective method has produced better results.

Table 7 shows intra-cluster distance and inter-cluster distance for three unlabeled datasets namely, Demonetization, Oscars and US army. M3 outperforms M1 and M2 in three out of six cases in terms of intra-cluster distance and five out of six cases in terms of inter-cluster distance. M3 is superior to M1 in 66.67% of the cases and M2 in 75% of the cases. When comparing M2 and M1, M2 outperforms M1 for 66.67% of the cases. Major reason for the improvement of M2 over M1 is that the similar words are combined and scored and M3 over M2 is that the initial centroid calculation is done using sentiments of the tweets.

Table 5. Purity for labeled datasets.

Dataset	Representation	M1	M2	M3	Winner
Airline	TF	**0.381921281**	**0.443362241**	**0.459973315**	**M3**
	TFIDF	0.338292195	0.393814434	0.384256170	M2
Debate	TF	**0.346097398**	**0.383855904**	**0.387925283**	**M3**
	TFIDF	**0.258625751**	**0.288192128**	**0.339226150**	**M3**
Twitter sanders	TF	0.678464819	0.678038379	0.678038379	M1
	TFIDF	**0.678282059**	**0.679013098**	**0.681693573**	**M3**

*Bold values indicate that the respective method has produced better results.

Table 6. F-measure for labeled datasets.

Dataset	Representation	M1	M2	M3	Winner
Airline	TF	0.601170263	0.629172257	0.616357157	M2
	TFIDF	0.58833988	0.550849341	0.546521225	M1
Debate	TF	**0.538332616**	**0.624307548**	**0.630253999**	**M3**
	TFIDF	**0.515508832**	**0.57736963**	**0.675676634**	**M3**
Twitter sanders	TF	0.452857042	0.487844342	0.468462485	M2
	TFIDF	**0.405564962**	**0.417014859**	**0.431989771**	**M3**

*Bold values indicate that the respective method has produced better results.

Table 7. Intra- and inter-cluster distances for unlabeled datasets.

Dataset	Representation	Distance measure	M1	M2	M3	Winner
Demonetization	TF	L1	0.002392951	0.006442155	0.005706134	M1
		L2	**0.05411328**	**0.024763684**	**0.063015582**	**M3**
	TFIDF	**L1**	**0.005708995**	**0.00619455**	**0.005174628**	**M3**
		L2	**0.026973832**	**0.075456404**	**0.084827498**	**M3**
Oscars	TF	L1	0.002731517	0.002116495	0.003278659	M2
		L2	0.034962254	0.039794083	0.034718241	M2
	TFIDF	**L1**	**0.001466088**	**0.001912015**	**0.001344756**	**M3**
		L2	**0.021098852**	**0.051613572**	**0.083961219**	**M3**
US Army	TF	L1	0.003537759	0.003189933	0.004343058	M2
		L2	**0.045348671**	**0.053428082**	**0.078973226**	**M3**
	TFIDF	**L1**	**0.005717595**	**0.003738831**	**0.003376451**	**M3**
		L2	**0.028967831**	**0.063663751**	**0.07490214**	**M3**

*Bold values indicate that the respective method has produced better results.

5.4 *Discussion*

Experimental results show that the proposed Tag Score model with improved K-means works better than its predecessors in 24 out of 36 cases. The proposed model incorporates a couple of significant developments in this paper. It combines all the words with similar meaning into tags and proposes a heuristic for selecting initial centroids.

The objective of these significant improvements is not only to minimize the intra-cluster distance but also to maximize the inter-cluster distance, purity and F-measure. Intra-cluster distance is minimized as the distance between the tweets in the same cluster is reduced and inter-cluster distance is maximized as the distance between the tweets in different clusters is increased. From the results, it is proved that when proposed tagging and scoring step is applied using random centroids, there is a little improvement in the quality of clusters. The results are improved significantly by selecting the initial centroids using the proposed heuristic and it outperforms its counterparts. The improvements in the results are due to the fact that sentiment scores of the words are included for altering the word weights and to compute initial centroids.

This method can also be generalized for any domain or application and relevant tweets can be extracted from Twitter using Hashtags. When tweets specific to a domain are passed on to the proposed algorithm, preprocessing returns a set of words explaining the user's view on the respective domain. Tagging groups the words and scoring improves the word vector values based on the sentiment scores. Finally, improved K-means clustering algorithm groups the tweets into clusters. Thus, it can be generalized and used in any crucial applications such as drug review, product review and movie review.

6. Conclusions

This paper proposed a new Tag Score (T_S) model with improved K-means for clustering tweets. Tagging is done by combining the words which have similar meaning and the tags and untagged words are then scored based on the sentiment scores of the words retrieved from Sentiwordnet. The reason for finding meaning of the words in tagging is that the words express similar meaning may be used by different people and the major purpose of finding sentiment score is to boost up the word vector values. Improvement of K-means is done by computing the initial centroids based on the polarity of the tweets. The proposed model is compared with conventional TF and TFIDF values using random K-means and T_S model with random K-means and the results prove that the proposed method produces high quality clusters when compared to conventional TF and TFIDF vectors with random K-means.

List of symbols

T	Tweet set
t_i	Tweet i
n	Number of tweets
WO_i	Set of words in tweet i before preprocessing
wo_{ij}	j^{th} word in i^{th} tweet
nw_i	Number of words in tweet i before preprocessing
W	Set of words after processing
w_i	i^{th} word in the collection
m	Number of words after preprocessing
PW	Set of positive words collected from labeled dataset
NW	Set of negative words collected from labeled dataset
NEW	Set of neutral words collected from labeled dataset
pw_i	i^{th} word in PW
nw_i	i^{th} word in NW
new_i	i^{th} word in NEW
pc	Number of words in PW
nc	Number of words in NW
nec	Number of words in NEW
SS	Synsets of m words in the tweets
s_i	Synset of i^{th} word
r_{ij}	Count of values occurring in both synsets of words i and j
TAG	Set of tags generated
tag_k	k^{th} tag
x	Number of tags generated
uw	Number of untagged words
y_k	Number of words in k^{th} tag
pos_score_k	Set of positive sentiment scores of k^{th} tag
neg_score_k	Set of negative sentiment scores of k^{th} tag
neu_score_k	Set of neutral sentiment scores of k^{th} tag
ps_{kj}	Positive sentiment score of j^{th} word in k^{th} tag
ns_{kj}	Negative sentiment score of j^{th} word in k^{th} tag
nes_{kj}	Neutral sentiment score of j^{th} word in k^{th} tag
p_count_k	Number of positive words in k^{th} tag
n_count_k	Number of negative words in k^{th} tag
ne_count_k	Number of neutral words in k^{th} tag
p_sum_k	Sum of the scores of positive words in k^{th} tag
n_sum_k	Sum of the scores of negative words in k^{th} tag
ne_sum_k	Sum of the scores of neutral words in k^{th} tag
$TF_{\boldsymbol{ij}}$	Term Frequency of j^{th} word in i^{th} tweet
$TFIDF_{\boldsymbol{ij}}$	Term Frequency/Inverse Document Frequency of j^{th} word in i^{th} tweet
psw_z	Positive score of z^{th} untagged word
nsw_z	Negative score of z^{th} untagged word
$nesw_z$	Neutral score of z^{th} untagged word
$tag\_pos\_cnt_i$	Number of positive tags of i^{th} tweet
$tag\_neg\_cnt_i$	Number of negative tags of i^{th} tweet
$tag\_neu\_cnt_i$	Number of neutral tags of i^{th} tweet
$word\_pos\_cnt_{\boldsymbol{i}}$	Number of positive untagged words of i^{th} tweet
$word\_neg\_cnt_{\boldsymbol{i}}$	Number of negative untagged words of i^{th} tweet
$word\_neu\_cnt_{\boldsymbol{i}}$	Number of neutral untagged words of i^{th} tweet

References

[1] Cruz F L, Troyano J A, Enriquez F, Ortega F J and Vallejo C G 2013 Long autonomy or long delay? The importance of domain in opinion mining. *Exp. Syst. Appl.* 40(8): 3174–3184

[2] Han J and Kamber M 2010 *Data mining: concepts and techniques*, MA, USA, Elsevier

[3] Tetlock T, Saar-Tsechansky M and Macskassy S 2008 More than words: quantifying language to measure firms' fundamentals. *J. Finance* 63(3): 1437–1467

[4] Mohammad E B and Arman K 2017 Sentence-level sentiment analysis in Persian. In: *Proceedings of 3rd*

International Conference on Pattern Recognition and Image Analysis (IPRIA), pp. 1–8

[5] Salima B, Fatiha B and Ghalem B 2016 Sentiment analysis at document level communications in computer and information science. In: *International Conference on Smart Trends for Information Technology and Computer Communications*, pp. 159–168

[6] Karamibekr M and Ghorbani A A 2012 Verb oriented sentiment classification. In: *Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, Macao, China*, pp. 327–331

[7] Lin C, He Y, Everson R and Ruger S 2012 Weakly supervised joint sentiment-topic detection from text. *IEEE Trans. Knowl. Data Eng.* 24(6): 1–12

[8] [8]Li S K, Guan Z and Tang L Y 2012 Exploiting consumer reviews for product feature ranking. *J. Comput. Sci. Technol.* 27(3): 635–649.

[9] Duric A and Song F 2012 Feature selection for sentiment analysis based on content and syntax models. *Decis Support Syst.* 53(4): 704–711.

[10] Zhai Z, Xu H, Kang B and Jia P 2011 Exploiting effective features for Chinese sentiment classification. *Exp. Syst. Appl.* 38(8): 9139–9146

[11] Niles I and Pease A 2003 Linking lexicons and ontologies: mapping WordNet to the suggested upper merged ontology. In: *Proceedings of the 2003 International Conference on Information and Knowledge Engineering (IKE 03), Las Vegas*, pp. 23–26

[12] Go A, Bhayani R and Huang L 2009 Twitter sentiment classification using distant supervision. *J. Process.* 150:1–6

[13] Davidov D, Tsur O and Rappoport A 2010 Enhanced sentiment learning using twitter hashtags and smileys. In: *Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING'10*. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 241–249

[14] Boia M, Faltings B, Cristian C and Pu P 2013 Is worth a thousand words: how people attach sentiment to emoticons and words in tweets. In: *Proceedings of the International Conference on Social Computing*, Alexandria, VA, UAS, pp. 345–350

[15] Luong T B, Thuy and Douglas H 2015 Public opinions of light rail service in Los Angeles, an analysis using Twitter data. In: *Proceedings of the iConference 2015*, California, USA, pp. 1–4

[16] Luiz F S, Coletta N, Adia F F, daSilva E R, Hruschka and Estevam R H 2014 Combining classification and clustering for tweet sentiment analysis. In: *Brazilian Conference on Intelligent Systems*, pp. 210–215

[17] Fernandez G M, Alvarez L T, Juncal M J, Costa M E and Gonzalez C F J 2016 Unsupervised method for sentiment analysis in online texts. *Exp. Syst. Appl.* 58: 57–75

[18] [18]Liu H, He J, Wang T, Song W and Du X 2013 Combining user preferences and user opinions for accurate recommendation. *Electron. Commer. Res. Appl.* 12(1): 14–23

[19] Orkphol K and Yang W 2019 Sentiment analysis on microblogging with K-means clustering and artificial bee colony. *Int. J. Comput. Intell. Appl.* 18(03):1–22

[20] Yedla M, Pathakota S R and Srinivasa T M 2010 Enhancing K-means clustering algorithm with improved initial center. *Int. J. Comput. Sci. Inf. Technol.* 1(2): 121–125

[21] Murat E, Nazif C and Sadhullah C 2011 A new algorithm for initial cluster centers in k-means algorithm. *Pattern Recognit. Lett.* 32(14): 1701–1705

[22] Baswade A M and Nalwade P S 2013 Selection of initial centroids for k-means algorithm. *Int. J. Comput. Sci. Mob. Comput.* 2(7): 161–164