




Unsupervised tweets categorization using semantic and statistical features

Maibam Debina Devi¹ · Navanath Saharia¹ 

Received: 3 January 2021 / Revised: 11 February 2022 / Accepted: 4 April 2022 /

Published online: 6 May 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Clustering is one of the widely used techniques in information retrieval. This experiment intends to categorize Tweets (based on their content) as representative of social media/user-generated content by exploiting statistical and semantic features. *tf-idf*, being widespread, is employed in combination with a synonym-based weighting scheme. The output of *tf-idf* in the form of the weight vector is transferred to the next phase as input, where based on the word synonyms, the system generate another weighted vector. Both vectors are used as a feature for clustering. The synonym-based feature technique adds semantic importance to the formation of the clusters. Using a density-based categorical clustering algorithm (with 8 as minpoints and 1.5 as epsilon), we categorized the Tweets into clusters. Six clusters are formed from 1K Tweets, which are evaluated manually and found cohesive. The Silhouette coefficient score (0.47) is used to validate the clusters.

Keywords Unsupervised learning · Social blogging · Semantic similarity · *tf-idf* · DBSCAN

1 Introduction

Analyzing documents by determining the latent information, fact identifications, and relationships decoding has gained popularity due to its self-described importance in information retrieval. Clustering and topic modeling are commonly employed to discover the latent topic structure of the underlying documents. Clustering is organizing documents based on cohesiveness while uncovering the topic-space is the primary task of the topic modeling. The

Maibam Debina Devi and Navanath Saharia contributed equally to this work.

✉ Navanath Saharia
nsaharia@iiitmanipur.ac.in

Maibam Debina Devi
debina@iiitmanipur.ac.in

¹ Data Engineering Lab, IIIT Senapati, Manipur, 795002, India

primary variation among the clustering algorithms lies in the metric they used to refer to cohesiveness, such as distance, connectedness, or density. Distance and connectedness-based clustering algorithms have proven track records within the field of affective computing compared to density-based clustering algorithms.

Latched on the impact and penetration of microblogging, researchers are exploring new techniques and tools to analyze short messages including Tweets to get better results against the open and existing problems in research domain, such as, summarization [27], topic modeling [29], entity (such as, mood [7, 12], and event [24]) detection and classification. Extracting information from short-messages using techniques such as, LDA and K-means are not new [29, 40], where distinct categories are used to retrieve tweets, which are then analysed on a fine-grained and coarse-grained basis.

It is widely accepted that the length of a document is proportional to the amount of information it contains, and that retrieval is based on it. It also applies to short messages, where the length of the message has a role. In topic modelling, clustering aims to extract an unified notion from a collection of supporting messages. Due to the variations in message length, such a configuration does not accomplish optimal retrieval of information. Major text retrieval venues, such as, TREC¹, FIRE² and SIGIR³ highlight the impact of message length, demonstrating that longer messages have a greater influence on text retrieval. The length of the message, on the other hand is a significant factor that heavily influences the importance of a phrase in the text. The possibility of repeated terms increases with increasing message length, leading to a high term frequency and impact the contribution of the terms in the query. It also impacts the degree of matching between query and message and eventually dominant over the retrieval of a long message over a short message, which implies that the appropriate selection of features may only lead to an efficient categorization of short messages.

The state-of-the-art Tweet categorization techniques use features as the bag of words, a bag of entities, word embedding, graph [1], lexicon [28], and deep neural network [38], where dependency lies in modules, such as, domain-specific content [9, 28], to construct semantic features. We hardly found reports where the feature generation is entirely dependent upon the data-set. This experiment aims to perform clustering over short texts exploiting statistical and semantical features to reduce the dependency and to prioritize the data-set for feature construction. The contributions of this experiment are as follows:

- Introducing domain-independent synonym-based semantic weighting scheme to classify the short messages.
- Development of a light-weighted Tweet categorization model through hybrid feature techniques, resulting in a combination of statistical and semantic measures.
- Design a good sequence-structure pre-processing for short text data to account for data preservation through removal.

This report is divided into five sections. Section 2, describes the concept, importance, and state-of-the-art of clustering in the field of text mining. Section 3 describe the methodology adopted for this experiment, which include the feature technique (Section 3.1), clustering algorithm and its evaluation technique(Section 3.3). Section 4, explains the experiment

¹Text REtrieval Conference, <https://trec.nist.gov>

²Forum for Information Retrieval Evaluation, <http://fire.irs.ri.res.in>

³ACM's Special Interest Group on Information Retrieval, <https://sigir.org>

working procedure which include the description of the used data-set (Section 4.1), experimental setup and result analysis Section 4.4. Section 5, concludes the article with directions to extend this work in the future.

2 Literature survey

A large amount of work on categorizing Tweets and their related areas have been reported priming a score to quantify the concept and context. As stated earlier, researchers have focused on the areas like summarization, topic detection, and classification [29].

Tweets are often used to express author's emotions, including the celebration, product reviews, knowledge sharing, etc. Emotional feeling or opinion is related to the sentiment, which pulls the attention of different researchers for the sentiment analysis. A good report on Tweet sentiment analysis using neural networks is seen in [13, 18]. Using baseline features, such as, BOW (bag of words) with unigram and bigram [18] works on different data-sets, (STSTd [16], SE2014, STSGd [34], SSTd [19], SED) with GLoVe embedding and ngram as sentiment feature upon convolution neural network, support vector machine to perform the sentiment analysis task. It claims using the neural networks approach outperformed classification over machine learning classifier. Effective word score introduced by Sahni et al. [33] is based on frequent words polarity scores with subjective distant supervision. For the most frequent 2500 words in the data-set, a dictionary is maintained with its polarity score from range -5 to 5. With $N(+x)$ and $N(-x)$, define a total number of words in the Tweet with x negative and positive polarity score with the use of Naive Bayes, SVM, and Maximum entropy as the classifiers. Apart from conventional sentiment analysis, a concept like event detection, aspect categorization, and sarcasm analysis has also recently focused on exploration by a different researcher. Work by [39] is based on event exploration on Twitter data. An unsupervised technique works on 3 levels approaches filtering, extraction, and categorization. With keyword and classifier-based applications for filtering the Tweets, the first step is the binary word feature, event-related feature based on observed frequent words, and event element name entity, time, and location as the supporting features for the classifier-based filtering. By introducing the latent event category model part of extraction and categorization, this experiment results in an event with 4 entities for each category. Recent work on the categorization of the sub-event task, techniques like noun-verb pairs and phrases is also seen in [4] report. Feature extraction like BOW doesn't consider the neighborhood relationship of words in the document highlighted in [23] report, and it explores the concept, fuzzy neighborhood model and applied to the 2 kernel-based clustering algorithms *hierarchical* and *c - mean* to group the real-time Tweets. More recent work on Tweet clustering adopted the ant clustering [15], DHST and attention-DHST [2] based clustering technique for various applications. The following are few observations based on the state-of-the-art unsupervised Tweets categorization.

- Data-set specific prioritization of features have not been observed so far, and
- Majority of the reports are domain specific, which may likely to fit into some specific scenario.

This research aims to bridge the gap between the state-of-the-art domain-specific unsupervised Tweets categorization and domain independent unsupervised Tweets categorization using prioritizing data-set specific features. To make the model light weight, the semantic feature extraction approach work on selective words in the data-set that also implies the

confinement of the data-set to some specific topics, likely to achieve fewer categories as preferred for short text data.

3 Methodology

This section describes the feature extraction technique, the clustering algorithm and its evaluation approaches.

3.1 Feature extraction

As mentioned earlier, we employed statistical and semantic features to club similar Tweets. *tf-idf*, a popular frequency-based feature extraction mechanism is used as a representative of statistical feature, and a word synonyms-based feature extraction mechanism is used as a representative of the semantic feature. As depicted in Fig. 1, the output of *tf-idf* in the form of the weighted vector, V_1 is transferred to the next phase as input, where based on the word synonyms, it generates another weighted vector, V_2 . Both the vectors, V_1 and V_2 are used as a feature to the next phase. On being widespread, multiple variants of *tf-idf* were coined overtime to tackle environment-specific limitations. Basic version of *tf-idf* was used in this experiment.

$$tf-idf = \left[\frac{\text{Total count of } t \text{ in } d}{\text{Total words in } d} \right] \times \left[\frac{\log(1+n)}{1+df(t)} + 1 \right]$$

where, n is the number of Tweets in the corpus, which is 1K in our case, and $df(t)$ is the Tweet frequency that implies the number of occurrences of term t in the n Tweets. That is, irrespective of position, order, or occurrences of a term, only the frequency of t is employed to compute the weight independently.

As t is independent, entire words including inflected, derived, or root participate equally in the feature generation process. Thus, each inflected form of a word gets greater than zero weightage if it exists in the corpus. To avoid equal participation of inflected words in the feature generation process, we employed a WordNet-based lemmatizer to normalize the inflected tokens as a part of the pre-processing phase.

Considering the frequency of t with respect to the distribution *tf-idf* generates the vector weight against each t . An inherent filter is employed with *tf-idf* vectorization to remove all terms, which are lacking at least in three Tweets. It is well understood that t with very low frequency (in our case, the occurrence of t at least in three Tweets) can hardly be considered a factor in the categorization process.

The second feature extraction technique aims to establish a semantic bridge among the terms. This part of the experiment greatly relies on WordNet to bridge the statistical and semantic relation. A list is prepared by extracting the synonyms of the top k terms from WordNet. The selection of k entirely depends on the overall weight of the terms generated by *tf-idf* vectorization phase. This step results a list of 270 new words, with $k = 20$, in our case. Among the top 20 words, the word ‘love’ has the highest weight value 24.5595, and the 20th word ‘know’ has the weight value 6.0192. Considering both the highest and lowest, we introduced a new weighting scheme for the synonyms. Synonyms define close semantic relation to the root word (top-weighted k words) in the corpus, With this purpose, we aim to bring words with close semantic closure to define a thin line over Tweets with a different context. The assignment of weight to words with respective root words is part of the semantic weighting scheme. Choosing the value of weight range is one of the

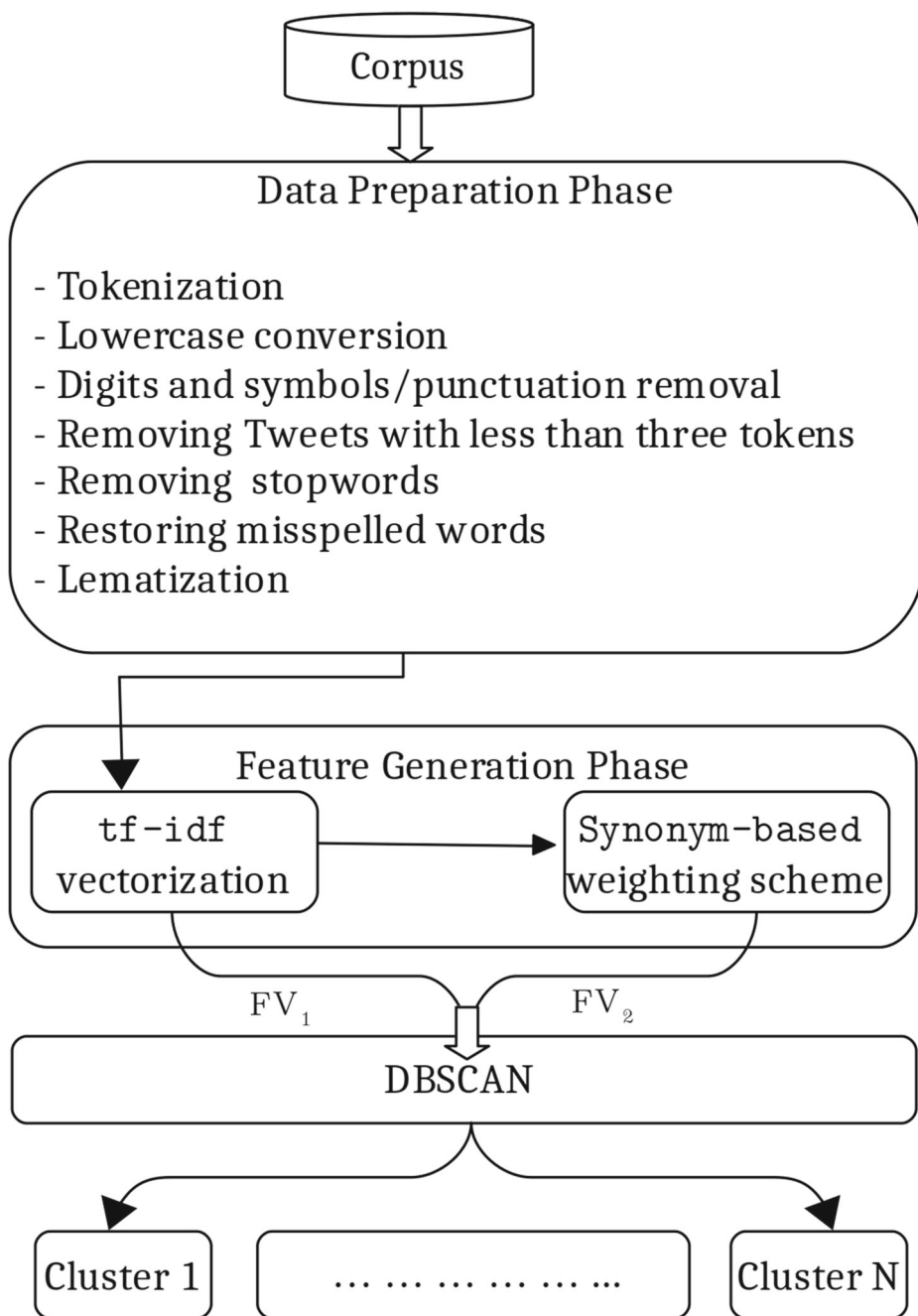


Fig. 1 System architecture

challenging tasks for this experiment, unlike different polarity lexicon exist, which weight range from -5 to +5 normally may not be able to fit for the segregation objective. As it is a synonym-based weight assignment, we aim to introduce a constant variable based on the number of a selected word for mapping synonym. This defined constant variable is made to balance the weight differences of the new words concerning their root words. The value of the constant variable is assigned with references to the weight of the last 20th word. With weight 6.0192 for the last 20th word, the resulting new word weight is formulated as $\text{root word weight} - \text{constant variable}$. Four is used as constant variable minus from all the root word weight as we seek to choose the constant variable less and yet low differences with the 20th word weight in k .

$$\begin{aligned} \text{new word}_i &\in \text{root word}_j \\ \forall \text{ word}_i &\in \text{new word} \\ \text{new word}_{\text{weight}} &= \text{root word}_{\text{weight}} - \text{constant variable} \end{aligned} \quad (1)$$

With illustration to (1), word *dear* is among the 270 new words with reference to root word *love* and it achieved weight of 20.5595. With respect to the above mentioned formula the *love* weight - constant variable that is (24.5595 - 4 = 20.5595). Likewise, for the 270 retrieved synonym-based words, weight is assigned concerning the root word. Concerning the created list, exploration of the statistical feature is performed. For any t in V_1 if it exists in the synonym-list, weight updation is performed by considering the summation of weight from feature one and synonym generated weight. In such a way, we tend to make statistical weight close to semantic, ensuring a better category for similar Tweets. Embedding *tf-idf* with clustering techniques is a well adopted technique. Bafna et al. [5] explores the classification of documents including, research articles, news, and emails using *tf-idf*.

3.2 Clustering

Clustering is one of the widely used techniques to categories similar items. Among various clustering techniques best known popular remains the k-means [17], Density-based [14], Hierarchical [11], and Expectation maximization clustering algorithm [6]. The k-means is a centroid-based vector quantization technique, where instances are assigned to the closest centroid, with the sum of squares as the optimization criterion. An iterative process that partitions data samples into predefined k clusters without overlapping. It aims to minimize the cost function. Equation (2) formulates the minimum cost function for k-mean clustering, where k defines the total number of clusters, n denotes the number of instances, and $\|x_i^j - c_j\|^2$ with centroid c is the distance function for the i^{th} instance. The challenge lies in the selection and initialization of k . Wrong centroid or initialization without domain knowledge may lead to a noisy cluster with higher number of outliers.

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^j - c_j\|^2 \quad (2)$$

Another kmeans variant is the Mini batch k-means algorithm, which works with predefined fix batches refer as a mini-batch. Each iteration takes a random sample of data of batch size, and a cluster update is performed. Selected random samples are assigned to the clusters with references to the previous cluster's centroid location based on gradient descent [30].

This technique outperforms well on the vast dataset as it disallows iteration over the entire dataset.

On the other hand, the hierarchical clustering has two cluster formation techniques, namely agglomerative, which is based on sequentially merging and divisive, which is based on sequential splitting. It begins with the identification of the closest cluster followed by the merging technique. It uses Euclidean distance measurement with different linkage criteria which define starting point of distances computation or construct distance matrix with i^{th} row and j^{th} column define distance between i^{th} and j^{th} elements. For merging the elements (3) formulate the different merging criteria or linkage function. With A and B being the two clusters and x and y being the data samples. Lastly, the result tree diagram is shown with a hierarchical relationship. This procedure required high computational intensity with $O(n^2 \log n)$ as time complexity, which may find it hard to compromise with our system. Besides, data samples may suffer sensitivity to noise or outlier, and identification is one of the challenging factors upon cluster.

$$\begin{aligned} & \max(d(x, y) : x \in A, y \in B) \\ & \min(d(x, y) : x \in A, y \in B) \\ & \frac{1}{|A| \cdot |B|} \sum_{x \in A} \sum_{y \in B} d(x, y) \end{aligned} \quad (3)$$

Another clustering algorithm is Expectation maximization which can accept a different number of clusters and works upon high dimensions. An iterative technique for finding maximum likelihood estimation of a statistical model. With X = observed data, Y = unobserved latent data, and Q = unknown parameter vector, therefore the $L(Q: X, Y) = p(X, Y|Q)$ maximum likelihood estimation is regulated based on observed data marginal likelihood. It begins with latent variable value estimation where random value is assigned for Q and estimates the probability for the conditional distribution of Y. With the obtained value of Y compute a better-estimated Q to optimize the parameter to best data. For this experiment, we preferred to go for a density-based clustering algorithm over others. The factor included in our data-set is the emphasis on a certain subject with unknown categories; it is uncertain about going for a predefined cluster. Choosing density-based clustering may be the right choice to obtain the unbreak cluster and be applicable for any shape. The distribution of feature samples shows high density and for few samples close to each other. Working with the DBSCAN clustering algorithm is one of the simplest and density-based algorithms performed in an unsupervised manner [14]. This technique aims to form a close group for the points based on distance measurement and the number of data points. Unlike other clustering algorithms, this technique could mark the data points as outliers for those in the low-density zone, which assists the formed cluster more meaningful.

Two parameters `epsilon`, and `minpoints` act as the deciding factor for the cluster formation. Parameter `epsilon` represents density checkpoint, which denotes the radius to be created around each data point. To define the minimum radius number of data points to have participated is defined by `minpoints` parameter.

$$N_{\epsilon}(p) = \{q \in D / \text{dist}(p, q) \leq \epsilon\} \quad (4)$$

$$\begin{aligned} & p \in N_{\epsilon}(p) \\ & |N_{\epsilon}(q)| \geq \text{minpoints} \end{aligned} \quad (5)$$

$$\forall(p, q) \text{ if } p \in C \text{ } q \text{ density reachable from } p \\ \text{if } p \text{ is density connected to } q \quad (6)$$

With p as data point, q represent the neighbourhood data point and D as total number of points as per data-set. A point p is density reachable to q is decided based on ϵ , \minpoints and with under certain condition (c.f. (5)). With respect to the parameters and condition given in (6) p, q belong to same cluster C which is non empty subset of D . For this experiment, we find 1.4 as most suitable ϵ value and 8 as \minpoints .

3.3 Evaluation of clustering algorithm

Cluster evaluation require three factors namely, *clustering tendency*, *number of clusters*, *clustering quality*. Data distribution nature reveal our feature extraction technique well satisfies the *clustering tendency* as uniformity in distribution is not observed. To measure the cluster quality, we consider an intrinsic approach, the *Silhouette coefficient* that does not require growth truth and are primarily used in unsupervised learning approaches, where the formation of a cluster relies on its internal information and is measured through the three parameters. a. cluster cohesion, relative measure of objects closeness within the cluster. b. how well enough clusters are separated among them. c. connectivity defines the position of data points to nearest neighbours.

$$s(i) = \frac{b(i) - a(i)}{\max a(i), b(i)}, | C_i > 1 | \quad (7)$$

Silhouette coefficient (7) is considered for cluster validation of this experiment, it measures the average distance between clusters. It range from -1 to +1.

The formula for estimation of silhouette coefficient is define under (7). In which i denote each observation, average dissimilarity within same cluster a_i , $d(i, c)$ define average dissimilarity of i with respect to other cluster and b_i define the minimum $d(i, c)$.

4 Experiment and result analysis

This experiment aims to group similar Tweets by exploiting the inherent features such as word distribution and word similarity. Due to the freedom in expression and human mind-set, the content collected from the social networking sites (as discussed in Section 1) is compact and abbreviated in nature [35], free from grammars [8, 32], and full of internet-jargon [37], which multiplies the challenges associated with the extraction of information. Despite of all odds, content analysis of social networking sites is still considered a trending area of research as it is the only fresh-and-fast source of news and opinions. Although every second users are contributing a huge amount of Tweets, sparsity is a known limitation specifically in new domains due the complexities in identification and annotation of these user generated content. Users often express their opinion (which is influenced by the user's/author's culture, emotion and location) uniquely, boosting the semantic divergence with polysemous words [31]. Synsets of Wordnet, and features of Wikipedia (such as, redirection [36], graph [3], and category mapping) are among few commonly used techniques to analyse the impact of synonym and polysemous words [10, 31]. Our implementation uses Wordnet lemmatizer to extract synonyms. The next section describes the used data-set and pre-processing stages.

4.1 Data-set description

A set of one thousand English language hate-speech Tweets⁴ is used as a primary experimental test-bed, which are extracted from Twitter using the API. This .csv formatted data-set consists of two fields - the first field is the Tweet ID, assigned to each Tweet by the Twitter API, and the second field contains the actual text of the Tweet.

The basic statistics of the data-set are enumerated in Table 1, where *mean* denotes the average token count per Tweet (token count per Tweet over a total count of Tweets). The *Min* count denotes the lowest token count in a Tweet among all the Tweets, whereas *Max* denotes the highest. It is also observed that 25%, 50%, and 75% of the Tweets have frequency scores of 12, 16, and 20, respectively. Standard deviation is calculated using the following formula.

$$\sqrt{\frac{1}{\text{total number of tweets}} \times \sum (\text{word count per tweet} - \text{mean})^2}$$

Table 1 statistic result in the evident differences in Tweet length in the corpus with a max of 32 words. Normalizing the length of the Tweet is one of the objectives to yield an optimal result. With this motive, the pre-processing phase corresponds to handling the task.

The experiment started with cleaning the corpus (depicted in Fig. 1), where the primary aim was to filter out the comparatively weak tokens in generating the optimal solution. This phase comprised extracting the tokens, case conversion, removal of digits, and special symbols, including punctuation markers. For example, tokens such as ‘wimbledon2016’, ‘it’s4u’ and ‘B4’ were translated to ‘wimbledon’, ‘itsu’ and ‘b’ at the end of the punctuation-removal step. The execution sequence followed was the same as stated in Fig. 1, as changing the sequence of execution may change the overall output. The sequences of pre-processing yet affect the data normalization. The tokenization and lowercase conversion tend to maintain the stable position for pre-processing since every stage later is based on words. The lowercase conversion eliminates the case of multiple occurrences of the same word. The digit and symbol removal attempt is made before the word length-based elimination for this data-set, which contains many words with symbols and digit. For example, word #eng20, alt19 maintain to participate feature generation as *eng*, *alt* if changed in order between third and fourth steps of pre-processing. Whereas with our described sequence, such words will not participate in features, eliminating word length after the symbol removal step. With achieved words, the removal of stop-words is done. Although multiple collections of stop-words for the English language are available publicly, the Gensim⁵ the use of a stop-word list was to remove stop-word from the corpus.

The word length of approximately 20% words of English language is less than three⁶ and the contribution of such words in the overall information extraction process are quite less [25]. To understand the semantic and properties of short-length text type, [22] report has to highlight some few importance. It explains the relation of frequent words and their grammatical function, knowing that short-length words normally result in high occurrences in documents/texts. Classifying words into 2 types ‘function words’ and ‘content word’ are performed where function words traditionally happen to be *articles*, *prepositions*, *pronouns*, *numbers*, *conjunctions* and *auxiliary verbs*. We analyze and extract the words that belong

⁴<https://www.kaggle.com/vkrahul/twitter-hate-speech?select=train.E6oV3IV.csv>

⁵<https://radimrehurek.com/gensim/corpora/textcorpus>. <https://html?highlight=stopwords#gensim.corpora.textcorpus>. remove_stopwords; Accessed on: 20 Dec 2020

⁶<https://norvig.com/mayzner.html>; visited on: December 20, 2020

Table 1 Basic data-set statistics

Parameter	Raw	Preprocessed
Token count	4497	3439
Mean	15.97	08.08
Max	35.00	17.00
Min	03.00	01.00
Standard deviation	05.35	03.2783
Longest token length	28	28
Count, where token length > mean	126	916
Count, where token length ≤ mean	4371	2523
Count, where token length < 3	209	00

to the part of speech classes stated above with character length three or less (as majority of the words are belong to *function words* class). In addition, most words with a length of 3 or less than 3 do not contribute strong sentiment polarity. Hence, part of pre-processing removal of words length with 3 character lengths or less than 3 is done to generate a clean feature vector. Upon Wikipedia web list of words with different character length out of which it is found that 191 words are having length three characters, 112 words are having 2 character length and 26 with 1 character are mostly with low semantic when comes to sentiment. This factor leads to considering the removal of tokens with a length less than three to generate a clean feature vector. The same study also reported distributional insights of words and letters in texts. The average token length was 4.79 letters for raw corpus and 7.60 letters for only distinct (unique) words. With the references to the statistics explained in Table 1, Fig. 2 shows the word length density over un-processed Tweets. We observed disturbed density distribution and noise word participation, with low word length dominating normal/ordinary meaningful words with its density resulting noise. Hence with the understanding as mentioned above, Fig. 3 shows the distribution and participation of types of words after pre-processed. With the pre-processing phase, we aim to normalize the word length and result in their base form to achieve an optimal result. The response over document length participation to achieve an optimal result is described above, besides eliminating words based on desirable rules applied in many reported work. Here we treat all the words equally and attempt to reduce length based on the character contributed over the individual word and its word length. Example *#hello20welcome* will be normalised to *hellowelcome*. The critical point here is that Norvig's [26] experiment performed on Google book dataset⁷, with having conversational, unstructured, and full of Internet-jargon in the corpus, it gave a refined idea with structural representation about the word frequency concerning word length.

Misspelled and abbreviated tokens are among the common issues in the area of social content analysis. For example, *bihday*, and *frikie* are the misspelled form of *birthday* and *freaky*. As Tweets typically represent a casual and straightforward form of expression, often observed the occurrences of misspelled words. In order to overcome such cases, unique words in the data-set are retrieved, manual analysis is performed, and a dictionary is created. A dictionary against the misspelled words is implemented, aiming to normalize them to their

⁷<http://storage.googleapis.com/books/ngrams/books/data-setsv2.html>; Visited on: December 15, 2020

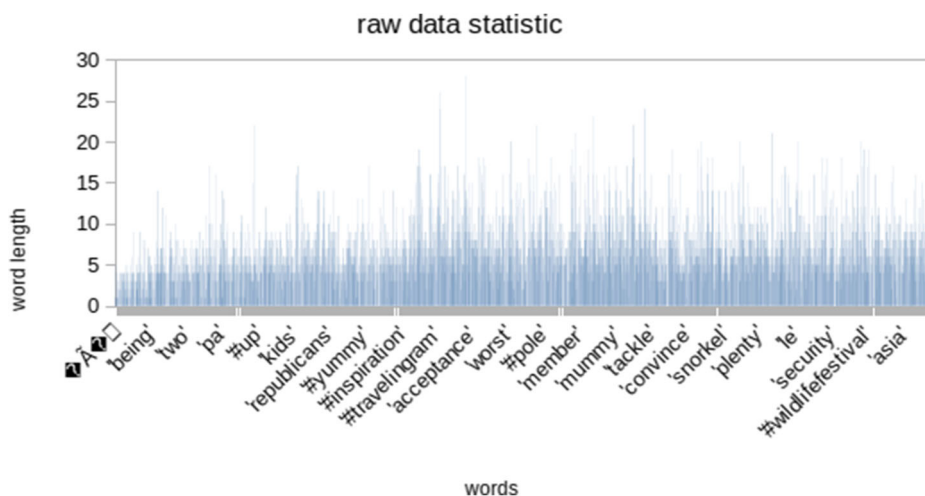


Fig. 2 Words distribution vs word length

original form. With this misspelled word is identified, and auto-correction is made before it passes to the model.

Finding the root form is as complex as restoring out the misspelled words. Lemmatization is preferred over stemming as the semantic similarity is one key aspect in this research. The Wordnet⁸ lemmatizer is used for the process, which reduces the unique word count to 3134 from 3286. Although dictionary-look-up-based spell-checking and exploiting lemmatizer are old and common practices of word normalization, we utilized these techniques as it is. Exploring the advanced solutions to restore the misspelled word to its correct form or advanced solution to extract root form is out of this research's focus. The processing step of an experiment plays a critical role in the model performance enhancement [21]. This experiment contributed to designing a sequential pre-processing step to affect the cleaning process, preserve the importance of data, and ensure the removal is based on the character contribution and length of the word, which may result in noise in model categorization.

4.2 Experimental setup

To perform categorization on unlabelled short-texts, we adapt the clustering approach to group similar Tweets. For each Tweet, the assigned cluster is referred to as the category.

This experiment has relied on two parameters- ϵ and *minpoints* for the best result. For this experiment, a certain range of minpoints and epsilon values are implemented and studied, and response over formed clusters are observed, as deciding the optimal value for the parameters of DBSCAN is one of the challenging task [20]. With estimation of average distance for each point with its respective K nearest neighbors, where K is the *minpoints* value. The average K -distances are plotted (*c.f.* Fig. 4). The point with maximum curvature or slope defines the optimal ϵ value. Therefore, among various values, this experiment yields the optimal meaningful cluster with ϵ as 1.5 and *minpoints* as 8.

⁸<https://wordnet.princeton.edu>; Accessed date: 15 Dec 2020

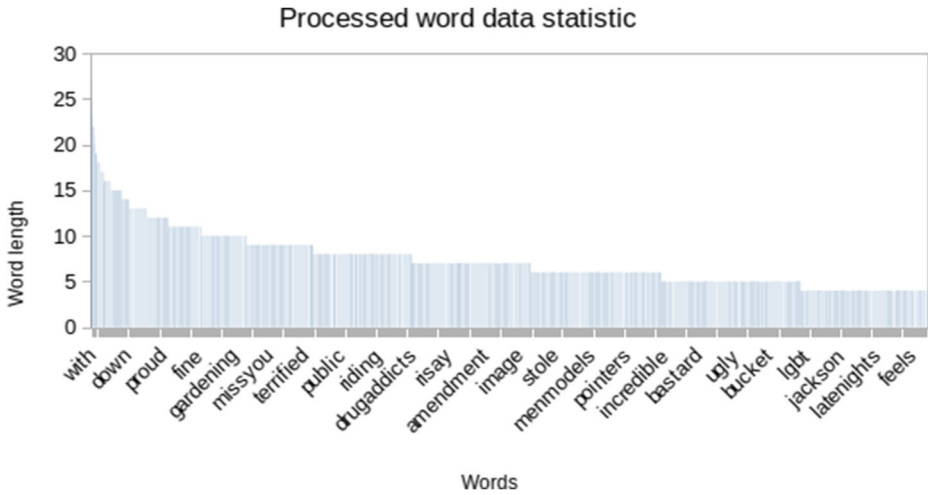


Fig. 3 Processed word distribution vs word length

As per obtained data statistic Table 1 the maximum and minimum Tweets length are 35 and 3, with a difference of 32. Normalization of the Tweet length is performed through the defined pre-processing stage. And it can achieve the normalized length with a maximum of

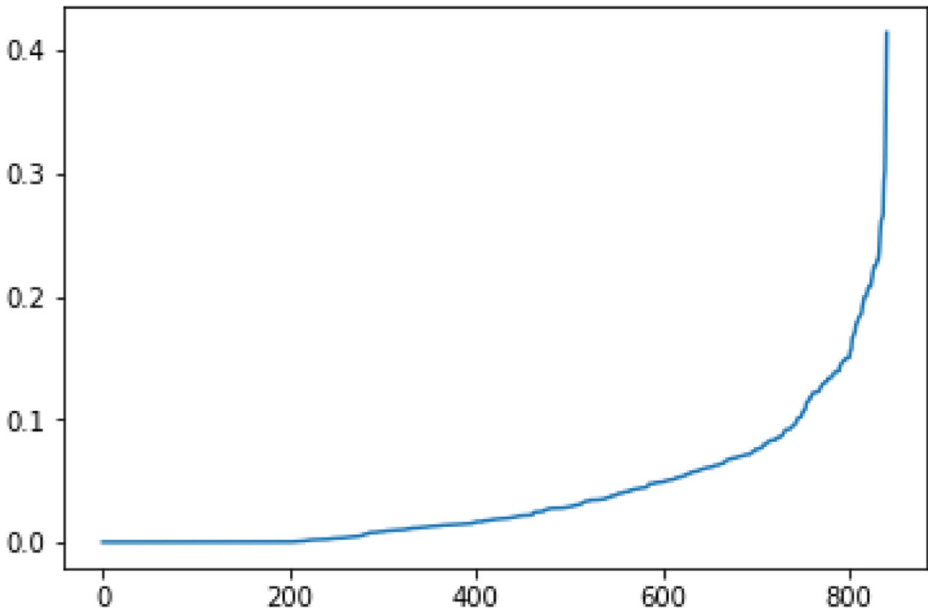


Fig. 4 Point sorted by distance to the 8th nearest neighbor

10 and a minimum of 1 from 3. It is observed for this data-set that many words with frequency count low and such words with their inflectional form exist representing a different word and resulting in its respective occurrence count. To accelerate the frequency count, we consider such words as identical and target to remove the inflectional endings using Word-Net lemmatization resulting in 3134 unique words. This experiment is implemented over Ubuntu 18.04 with i7-8700 CPU @ 3.20GHz processor with graphics UHD graphics 630, 16GB RAM under 64 bit OS type.

4.3 Baseline comparison

We adopted K -means and mini-batch K -means clustering algorithms for the baseline experiment, which are widely used techniques. The clustering process initiates with a random selection of centroid and aims to optimize it until a stable centroid is obtained. It requires predefined K value, which denotes the number of clusters to be formed.

Figure 5 represents the cluster obtained for K -means, mini-batch K -means algorithms. The clusters formed for both K -means and mini K -means are overlapping.

The baseline studies have two challenges: (a) determining the best K value and (b) samples must participate in cluster formation, which means that unrelated samples may be included in the cluster.

4.4 Result analysis and discussion

The final clusters obtained for this experiment are shown in Fig. 6. A total of six different clusters are formed, where each cluster represents a group with a similar subject. These groups consist of tweets with close semantic.

Table 2 result in the tweets with their belonging cluster. Cluster 1 consists of tweets with words *nights*, *watching*, *episodes*, *anytime* which describe semantic close. Likewise, cluster 2 possess words like *beautiful*, *great*, *mediatization*, *service* signifying semantically near and positive attitude. In contrast, cluster 4 result words *blaming*, *conceded*, *racist*, *crime*, *arrest* reveal a negative subject. Therefore, each cluster group represents a different subject with a close semantic feature.

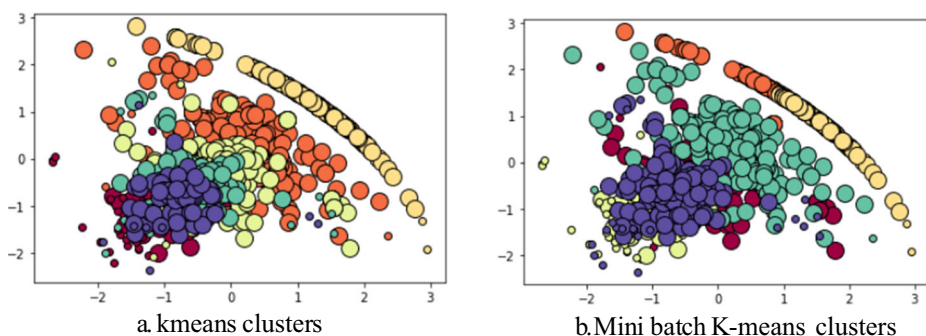


Fig. 5 a kmeans clusters b Mini batch K -means clusters

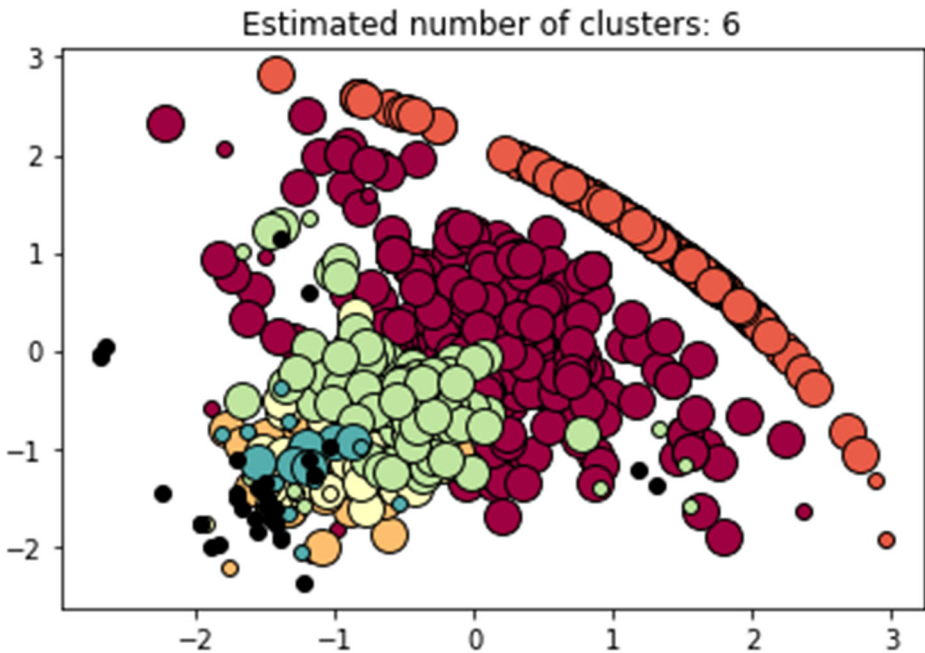


Fig. 6 DBSCAN obtained clusters with noise

The hybrid feature, $tf-idf$, in combination with the synonym-based weighting scheme, is capable of handling syntactic and semantic similarity, which is a major advantage over the baseline approach. The syntactic and semantic similarity features are missing in the baseline approaches. Moreover, the system is not dependent on a predefined cluster count. The quality of the clusters is determined by the Silhouette score, which measures the average similarity of the samples within a cluster and their distance to the other samples. The evaluation score for this experiment is 0.47, which signifies that the quality of the formed clusters are good. This experiment yields 6 different clusters from 842 Tweets post-filtered with 31 Tweets non-participation to any categories obtained, representing noise in cluster formation. Thus, the obtained result is comparable with the state-of-the-art Tweet classification techniques.

The experimental performance shows synonym-based feature topped over frequency weighting scheme works well for the categorization of Tweets. As this hybrid feature depends on the input data-set, the model performance wouldn't be affected when changing the data-set for a different domain, making sense as a domain-independent model. The pre-process stage designed with consideration to unnecessary data lost, where sequences are designed to motive data preservation through removal, which works in a parallel manner. The nature of aggregation of Tweets works well to segregate from other not semantic close samples. Unlike other feature extraction techniques BOW or simple $tf-idf$, which isolate the concept of synonym to overcome such cases and make a domain-independent model, these simple techniques result in a good response over feature design.

Thus, relying on the frequent words of the data-set to define the weight of the synonyms does not only increase efficiency of the categorization, it also makes the system light-weight.

Table 2 Categories of tweets with few processed tweets

C_i	Processed member tweets	TC
C_1	['monday', 'nights', 'finally'] ['watching', 'episodes'] ['think', 'happen', 'anytime']	369
C_2	['beautiful', 'vendor', 'upsideofflorida', 'shopalysas'] ['great', 'panel', 'mediatization', 'public', 'service']	186
C_3	['testing', 'tired', 'annoyed'] ['bamas', 'raising', 'child', 'think', 'advance']	46
C_4	['people', 'blaming', 'conceded', 'rooney', 'knowing'], ['officer', 'viral', 'arrest', 'video', 'chief', 'crime', 'officer']	71
C_5	['morning', 'journey', 'begins', 'travel', 'thejourneybegins', 'hello'] ['little', 'badday', 'coneofshame', 'pissed', 'funny', 'laughs']	120
C_6	['stories', 'happy', 'ending', 'anime', 'story', 'ending', 'like'] ['truthful', 'positive', 'affirmation']	18

$C_i \rightarrow$ Cluster number; TC \rightarrow Total number of Tweets in the cluster C_i

5 Conclusion and future work

This experiment aims to perform categorization of Tweets using DBSCAN algorithm.

To handle semantic similarity along with DBSCAN, we employed hybrid feature, *tf-idf*, in combination with a synonym-based weighting scheme, which is a key loophole in both state-of-the-art light-weight Tweet-categorization techniques and baseline approaches. The output of *tf-idf* in the form of the weight vector is transferred to the next phase as input, where based on the word synonyms, the system generates another weighted vector. Both vectors are used as a feature for clustering. The semantic importance of the clusters is increased by using a synonym-based feature technique. To make the system light-weight, our weighting scheme is depended on the frequent terms of the dataset and weight of the synonyms.

Each Tweet is assigned to a group based on the weight of the terms defined by the hybrid module. Such groups are considered as categories. Total 6 clusters are formed with varying size. Figure 6 represents the tweets belonging to its cluster and shows close collective relation among the tweets. For example, cluster 2 has words like *beautiful*, *great*, *mediatization*, and *service*, all of which are implying a close semantic and positive attitude, whereas cluster 4 contains terms like *blame*, *conceded*, *racist*, *crime*, and *arrest*, which reveal words with a negative subject.

Likewise, cluster 3, cluster 1, cluster 5, and 6 have words like *tired*, *testing*, *raising*, *child*, *nights*, *watching*, *episodes*, *anytime*, *morning*, *travel*, *funny*, *laugh* and *stories*, *ending*, *story*, *truthful*, *positive*, from different tweets among the same group and resulting semantically close relation to representing a subject. In the future, we will be exploring the adaptability of this model with various constraints *size of data-set*, *divergent domain* and analyze the response over and enhancement on it.

Acknowledgements Authors would like to thank anonymous reviewers for their insights and suggestions during preparation of the draft.

Funding The first author acknowledge the financial supports received from TEQIP Phase III, NPIU (Ref. no.: IIITM/ACA-PhD/2017-18/10).

Declarations

Conflict of Interests The authors declare that there is no conflict of interest.

References

1. Agarwal V (2015) Research on data preprocessing and categorization technique for smartphone review analysis. *Int J Comput Appl* 131(4):30–36
2. Ali A, Zhu Y, Zakarya M (2021) Exploiting dynamic spatio-temporal correlations for citywide traffic flow prediction using attention based neural networks. *Inf Sci* 577:852–870
3. Aouicha MB, Taieb MAH, Hamadou AB (2016) Lwcr: multi-layered Wikipedia representation for computing word relatedness. *Neurocomputing* 216:816–843
4. Arachie C, Gaur M, Anzaroot S, Groves W, Zhang K, Jaimes A (2020) Unsupervised detection of sub-events in large scale disasters. In: *AAAI Conference on Artificial Intelligence*, vol 34, pp 354–361
5. Bafna P, Pramod D, Vaidya A (2016) Document clustering: Tf-idf approach. In: *International Conference on Electrical, Electronics, and Optimization Techniques*. IEEE, pp 61–66
6. Bradley PS, Fayyad U, Reina C et al (1998) Scaling em (expectation-maximization) clustering to large databases. Technical Report MSR-TR-98-35, Microsoft Research

7. Chen J, Yan S, Wong K-C (2020) Verbal aggression detection on Twitter comments: Convolutional neural network for short-text sentiment analysis. *Neural Comput Appl* 32(15):10809–10818
8. Clark E, Araki K (2011) Text normalization in social media: progress, problems and applications for a pre-processing system of casual english. *Procedia-Soc Behav Sci* 27:2–11
9. Coteló JM, Cruz FL, Enríquez F, Troyano JA (2016) Tweet categorization by combining content and structural knowledge. *Inf Fusion* 31:54–64
10. Daouadi KE, Rebaï RZ, Amous I (2021) Optimizing semantic deep forest for tweet topic classification. *Inf Syst* 101:101801
11. Day WHE, Edelsbrunner H (1984) Efficient algorithms for agglomerative hierarchical clustering methods. *J Class* 1(1):7–24
12. Devi MD, Saharia N (2020) Exploiting topic modelling to classify sentiment from lyrics. In: *International Conference on Machine Learning, Image Processing, Network Security and Data Sciences*, pp 411–423
13. Dos Santos C, Gatti M (2014) Deep convolutional neural networks for sentiment analysis of short texts. In: *International Conference on Computational Linguistics*, pp 69–78
14. Ester M, Krieger H-P, Sander J, Xu X et al (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the International Conference on Knowledge Discovery and Data Mining*. AAAI Press, Portland, pp 226–231
15. Firdaus DH, Suyanto S (2020) Topic-based tweet clustering for public figures using ant clustering. In: *3rd International Seminar on Research of Information Technology and Intelligent Systems*, pp 476–481
16. Go A, Bhayani R, Huang L (2009) Twitter sentiment classification using distant supervision. CS224N project report, Stanford 1(12)
17. Hartigan JA, Wong MA (1979) Algorithm as 136: A k-means clustering algorithm. *J R Stat Soc* 28(1):100–108
18. Jianqiang Z, Xiaolin G, Xuejun Z (2018) Deep convolution neural networks for Twitter sentiment analysis. *IEEE Access* 6:23253–23260
19. Jianqiang Z, Xueliang C (2015) Combining semantic and prior polarity for boosting Twitter sentiment analysis. In: *International Conference on Smart City*. IEEE, pp 832–837
20. Link A-K (2018) Challenges for dbSCAN: Closely adjacent clusters and varying densities
21. Meetei LS, Singh TD, Borgohain SK, Bandyopadhyay S (2021) Low resource language specific pre-processing and features for sentiment analysis task. *Lang Resour Eval*:1–23
22. Miller GA, Newman EB, Friedman EA (1958) Length-frequency statistics for written english. *Inf Control* 1(4):370–389
23. Miyamoto S, Suzuki S, Takumi S (2012) Clustering in tweets using a fuzzy neighborhood model. In: *International Conference on Fuzzy Systems, Brisbane*, pp 1–6
24. Mojiri MM, Ravanmehr R (2020) Event detection in Twitter using multi timing chained windows. *Comput Inf* 39(6):1336–1359
25. Munková D, Munk M, Vozár M (2013) Influence of stop-words removal on sequence patterns identification within comparable corpora. In: *International Conference on ICT Innovations*. Springer, pp 67–76
26. Norvig P (2013) English letter frequency counts: Mayzner revisited or etoain srhldcu. <https://norvig.com/mayzner.html>
27. O'Connor B, Krieger M, Ahn D (2010) Tweetmotif: exploratory search and topic summarization for Twitter. In: *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, George Washington University, pp 384–385
28. Park S, Kim Y (2016) Building thesaurus lexicon using dictionary-based approach for sentiment classification. In: *International Conference on Software Engineering Research, Management and Applications*. IEEE, pp 39–44
29. Rosa KD, Shah R, Lin B, Gershman A, Frederking R (2011) Topical clustering of tweets. *Proceedings of the ACM SIGIR workshop on Social Web Search and Mining, Analysis under crisis*, vol 63
30. Ruder S (2016) An overview of gradient descent optimization algorithms. [arXiv:1609.04747](https://arxiv.org/abs/1609.04747)
31. Rudrapal D, Das A (2017) Measuring the limit of semantic divergence for english tweets. In: *Recent Advances in Natural Language Processing*, Varna, pp 618–624
32. Saharia N (2015) Detecting emotion from short messages on Nepal earthquake. In: *International Conference on Speech Technology and Human-Computer Dialogue*. IEEE, Bucharest, pp 1–5
33. Sahni T, Chandak C, Chedeti NR, Singh M (2017) Efficient Twitter sentiment classification using subjective distant supervision. In: *International Conference on Communication Systems and Networks*, pp 548–553
34. Saif H, Fernandez M, He Y, Alani H (2013) Evaluation datasets for Twitter sentiment analysis: a survey and a new dataset, the sts-gold

35. Singh TD, Singh TJ, Shadang M, Thokchom S (2021) Review comments of manipuri online video: Good, bad or ugly. In: International Conference on Computing and Communication Systems, vol 170. Springer, Shillong, p 45
36. Tang G, Xia Y, Wang W, Lau R, Zheng F (2014) Clustering tweets using wikipedia concepts. In: Proceedings of the Language Resources and Evaluation Conference, Reykjavik, pp 2262–2267
37. Teodorescu H-N, Saharia N (2015) An internet slang annotated dictionary and its use in assessing message attitude and sentiments. In: International Conference on Speech Technology and Human-Computer Dialogue. IEEE, Bucharest, pp 1–8
38. Vosoughi S, Vijayaraghavan P, Roy D (2016) Tweet2vec: Learning tweet embeddings using character-level CNN-LSTM encoder-decoder. In: ACM SIGIR conference on Research and Development in Information Retrieval, pp 1041–1044. <https://doi.org/10.1145/2911451.2914762>
39. Zhou D, Chen L, He Y (2015) An unsupervised framework of exploring events on Twitter: Filtering, extraction and categorization. In: AAAI conference on Artificial Intelligence, vol 29
40. Zou L, Song WW (2016) LDA-TM: A two-step approach to Twitter topic data clustering. In: International Conference on Cloud Computing and Big Data Analysis, pp 342–347

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.