International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 10, No.1 (Apr-2021)

http://dx.doi.org/10.12785/ijcds/100150

Real-Time Twitter Corpus Labelling Using Automatic Clustering Approach

Itisha Gupta[1] and Nisheeth Joshi[1]

[2]Department of Computer Science, Banasthali Vidyapith, Tonk, Rajasthan, India

Received 8 Jul. 2020, Revised 27 Aug. 2020, Accepted 29 Sep. 2020, Published 21 Apr. 2021

Abstract: In this paper, we present a novel automatic labelling approach for the classification of large amount of unlabelled real-time twitter datasets for textual-based Twitter Sentiment Analysis. The tweets are labelled or classified as Positive, Negative or Neutral using the novel automatic approach. The proposed approach applies an unsupervised clustering technique that would generate clusters based on the underlying patterns (finding similarities between tweets) in the collected twitter corpus. Twitter search API is used to collect real-time English tweets on several topics such as “#Demonetization”, “#lockdown”, and “#9pm9minutes” by the use of search operator. To analyse the sentiment from real-time tweets, labelling of the corpus is required. Manual annotation of large twitter corpus is time and labor-intensive. Moreover, domain experts are needed for labelling of tweets belonging to a particular domain. Thus, in this work, we propose the use of the improved K-mean clustering approach, which is an unsupervised way of labelling corpus, which could then be used for learning supervised models such as SVM for sentiment analysis. To make the corpus ready for clustering and to get quality clusters, we have applied some basic to advanced cleaning operations known as tweet normalization. Furthermore, extensive feature engineering is conducted to generate different types of features including POS-based (Part-of-Speech), ngrams, Twitter-specific, negation, and lexicon-based features from our collected unlabelled twitter corpus. Those features act as input to the K-mean clustering algorithm and help it in identifying patterns from the data for cluster generation. Moreover, we handle an important linguistic phenomenon namely negation before the cluster generation. Our main focus is on handling those negation tweets in which negation presence has literally no sense of negation (negation exception cases). At the end, cluster analysis is done manually to find out the sentiments expressing by tweets in a particular cluster. Accordingly, cluster classification is done and each cluster is assigned one class that is Positive, Negative, or Neutral. The main contribution of this work is the idea of amalgamation of extensive feature engineering and negation modelling with the unsupervised K-mean clustering approach for classification of large unlabelled twitter corpus. A comparative analysis of our proposed approach is done with or without negation exception cases and with random K-mean using only conventional TF-IDF as features. The proposed automatic labelling approach produces substantial results in terms of cluster quality assessed through two evaluation metrics known as inertia and silhouette score.
.

Keywords: Feature Engineering, Negation Handling, Negation Exception Cases, K-mean, Cluster Analysis, Corpus Labelling, Real-Time Tweets, Twitter Sentiment Analysis, Pre-processing

1. INTRODUCTION

Microblogging is one of the popular and widespread broadcasting media amidst the internet world. People frequently use microblogging websites such as Twitter (created in 2006) for sharing their views, opinions, emotions, etc. on any event, product, services, and idea. Thus, the enormous amount of opinionated data is available in digital form on different platforms such as blogs and discussions which is very useful for decision making or feedback. Day to day basis huge amount of tweet gets generated on hot and latest topics. Automatic analysis and reasoning of such data help in deriving meaningful visions, which carries opportunities for users, consumers, and businesses i.e. analysis of such data provides insight into people’s opinion and inclination. One of the effective techniques for analysis of such opinionated tweets is Twitter sentiment analysis (TSA). TSA is used for determining accumulated opinions of people from digital opinionated data. It plays a significant role in formative the opinion of an individual and the influence of that opinion on society [29], [30], [31].

Approaches for TSA are categorized into the three groups namely knowledge-based approach (lexicon-based), statistical approach (machine learning and deep learning), and the hybrid approach (combination of

E-mail: *itishagupta07@gmail.com*, *jnisheeth@banasthali.in*

knowledge-based and statistical approach). In most of the earlier works on TSA, a supervised learning approach has been used that needs labelled corpus for training and sentiment prediction. However, real-time tweets gathered from twitter using API are not labelled or classified readily. It is necessary to label the collected Twitter corpus before performing supervised sentiment analysis on them. Corpus can be labelled or classified either into sentiment classes (such as Positive or Negative) or into emotions such as joy, fear, sadness, and many more.

Manual annotation of the large unlabelled corpus is labor-intensive and requires the knowledge of domain expert. Typically, it is feasible for the small size corpora. Moreover, it is expensive to hire a human annotator for the manual labelling because of the big size real-time Twitter corpora. As the size of real-time Twitter dataset grows, manual labelling becomes more time consuming and expensive. Thus, our solution is to assign a sentiment label (Positive, Negative or Neutral) to the real-time tweet (that is classifying a tweet according to the sentiment expressed by it) through the use of automatic clustering approach. Once the automatic approach is ready, large amount of corpora can be labelled rapidly.

There exist various automatic approaches in the literature for corpus labelling [12], [17], [20], [23], [28]. One of the famous approaches is the distant supervision approach for corpus classification based on the presence of emoticons in a tweet. That is, if a tweet is containing only positive emoticons, then that tweet would be classified as positive. Few works used positive and negative hashtag words such as #happy for corpus labelling. Rule-based classifier (based on the count of positive and negative sentiment words in a tweet) is also an automatic way of corpus classification. As an illustration, if a tweet contains a minimum two positive sentiment words and no negative sentiment words then that tweets is labelled as positive. Nevertheless, such automatic approaches have shown considerable performance, but ignore the context (local context such as negation) in which a word appears due to classifying the corpus based on the presence of positive or negative counts of emoticons or sentiment words. Moreover, there is a possibility that a classifier learns on the corpus labelled through above-mentioned automatic approaches will not generalize. It would simply mimic the rule-based classifier on the labelled training corpus and generate biasness in the performance of classifiers during sentiment analysis.

One way to combat this problem is to use as many features for the corpus labelling rather than using only the count features (such as no. of positive emoticons, no of positive words, and many more). Thus, we perform extensive feature engineering from the unlabelled real-time Twitter corpuses. That is, we have not only extracted count and binary features (Twitter-specific and POS-based) but also real-valued features including ngram and lexicon-based features. Furthermore, an important linguistic phenomenon called as "negation" is also handled. Existing works in the literature consider conventional Bag-of-Word (BOW) approach (collection of words with their occurrences) for clustering of dataset, ignoring the impact of linguistic elements such as negation during clustering. Negation is one of the critical elements that can change the polarity of opinionated text as in "#Demonetization is not a failure". Here, negation cue "not" affecting the polarity of sentiment word "failure" and thus, changing the polarity of this tweet from negative to positive. We address this issue of negation handling during clustering of tweets and make use of two Twitter-specific automatic lexicons [32] that contain score of words in negated context as well as affirmative context. Also, we handle those negation tweets in which negation cue presence has no sense of negation as in "@Dipankar_cpiml @svaradarajan @mkvenu1 Poor earnings a living out of #DeMonetisation queues .... isn't that bad". In this tweet, negation cue "isn't" has no literal sense of negation i.e. it is not affecting the sentiment word "bad". We identify such negation exception cases using linguistic rules and ignore the negation handling in such cases. Handling the negation in addition with negation exception cases would help in generation of quality clusters from the unlabelled real-time Twitter corpuses.

Thus, in this paper, we aim to present an improved unsupervised clustering approach (K-mean) in amalgamation with extensive feature engineering and negation modelling for classification or labelling of real-time unlabelled twitter corpus. In contrast to supervised machine models, K-mean has the ability to classify data points without having firstly trained on the labelled dataset. Clustering is an unsupervised approach used to find sub-groups (clusters) within a dataset based on the underlying patterns [9], [16], [36]. Objects in the same cluster are more similar to each other than the other cluster. Put simply, clustering needs unlabelled data as input and gives clusters as output. It is widely used in several applications such as market segmentation, image segmentation, and many more. Nevertheless, the main aim of clustering is to find structures in the data, but clusters generated by it could be considered as labels for the unlabelled corpus. Then one can train a classifier using those labels as the target. Hence, clustering can be used for classification that is labelling of unlabelled data. Furthermore, clustering has the power to reveal the unforeseen groups in a large dataset that may convey significant information.

Since our main goal is labelling or classification of real-time twitter corpus, we have collected real-time tweets on the various hot topics including "#Lockdown" and "#9pm9minutes" through the use of Twitter search API. We have also collected older tweets on "#Demonetization" in order to show the vulnerability of our amalgamated approach (combination of extensive feature engineering, negation modelling, and unsupervised K-mean clustering) on varied domain and older tweets. First of all, the collected corpus is cleaned and normalized by the use of some basic to advanced cleaning tasks from our previous work [14]. Then, we extracted various types of features such as Pos-based, lexicon-based, morphological features, negation, and Tf-Idf features (n-grams) from the cleaned twitter corpus. All the extracted features are concatenated to get a final feature vector, which is then given as input to one of the popular clustering algorithm K-mean, often used for labelling of unlabelled data [18]. Finally, K-mean clustering approach is applied to classify the twitter corpus into three clusters based on the various syntactic and semantic features.

Generated clusters are manually analysed for inspecting the sentiment expressed by tweets in clusters because manual inspection produces more reliable and accurate result. Post manual inspection of each cluster is faster than conducting entire labelling manually as we don’t need to manually analyse all the tweets of a cluster because tweets with in a cluster are more similar to each other (homogenous). Thus, it’s not labor intensive to manually analyse only a few tweets of each cluster for determining sentiments expressed by them. This way helps in assigning each cluster one sentiment class that is Positive, Negative or Neutral. To be more specific, we now have twitter corpus classified into three classes namely Positive, Negative or Neutral. We empirically evaluate the combination of different features in cluster generation. We have observed the best result when all features are given as input to the clustering algorithm. This shows that features play an important role in identifying underlying patterns from the unlabelled corpus through the K-mean clustering method. Moreover, we present experiments for evaluating the significance of modelling negation while clustering so that quality clusters can be obtained.

Following are the main contributions of this paper:

- We present an automatic cascade approach which is an amalgamation of extensive feature engineering, negation modelling, and unsupervised K-mean approach for the labelling or classification of the large unlabelled Twitter data.
- We use the Twitter search API to collect real-time tweets on various topics including “#Demonetization”, “#Lockdown”, and “#9pm9minutes.
- We contribute in presenting heuristics for handling those negation tweets in which negation presence has no sense of negation. This would prevent in misclassification of such tweets in wrong clusters.
- We perform extensive feature engineering to generate various syntactic and semantic features such as POS, negation, Twitter-specific, lexicon-based, and many more that would be used to identify structures within the dataset.
- We demonstrate the use of one of the popular K-mean clustering approach for evaluation of large unlabelled twitter datasets in a fast and objective way to generate clusters (equals to the number of classes or labels that we want for our unlabelled corpus) by the use of various extracted features.
- We inspect each cluster manually outputted by the K-mean for the assignment of sentiment class based on the sentiment expressed by tweets in each cluster. Labelled corpus could be used to train a supervised classifier so that new instances can be predicted.
- We compare and analyze the performance of our proposed K-mean automatic approach (BOW enhanced with local contextual semantics, negation) with random K-mean (conventional BOW model, negation is not considered) and with K-mean when only conventional ngram features (TF-IDF) are extracted for clustering.

The rest of paper is organized as follows: section 2 presents the literature review of earlier work on clustering with classification, section 3 describes the framework of our proposed approach for labelling of twitter corpus, section 4 provides the labelling results and the last section concludes this paper with possible future directions.

2. LITERATURE REVIEW

There exist a large amount of opinionated text in digital form which can provide informative knowledge for strategic decision. There are many sources of such data including blogs, newspapers, social media platforms, etc. Raw data available from such sources are in unformatted form. In order to analyse opinionated data, one needs to classify data either into classes (groups) or need to find hidden structure from it. In the existing literature, several methods have been used for the automatic classification or labelling of the large amount of unlabelled corpus specifically, Twitter corpus. The most popular and widely used approach is the distant supervision approach, in which based on the presence of positive such as :-) and negative emoticon such as :-(, the label is assigned to each training tweets. It is an automatic approach for assigning class labels to text. Reference [25] firstly presented this approach, which used this approach for labelling of data from the Usenet newsgroup. Reference [12] was the first to use the distant supervision approach for the labelling of training tweets containing emoticons. They queried the Twitter API with positive emoticon (“;)”) as well as with negative emoticon such as :(and collected 1.6 million tweets classified into positive and negative classes. Several researchers depend on a distant supervision approach for the collection of training tweets (labelling of tweets) [5], [21], [23], [28].

Another important approach for the creation of the training set is the use of positive and negative hashtag words such as #joy, #disappointed, and many more because people often use the hashtag in a tweet for expressing their sentiment. This technique has been used by several researchers in the past [7], [17], [20].

In contrary with automatic approaches, several researchers used the crowdsourcing method [1], [2], [19] such as Amazon Mechanical Turk or third-party service (Alchemy API) for the labelling of twitter corpus, while few labelled the twitter corpus by themselves [4], [8], [20]. Nevertheless, manual labelling of corpus provides more accurate results, but it is very time consuming and labor-intensive.

Recently, the unsupervised clustering approach was being used for the labelling of the corpus or clustering of unlabelled data [37]. Reference [6] presented a comparative analysis of various unsupervised clustering approaches that have been used in the past for analyzing Twitter data. They presented a comparison of several clustering algorithms, dataset size, clustering features, no. of clusters, and evaluation metrics. Though main aim of unsupervised clustering approach is to find the hidden patterns among dataset, it is being widely used for the classification of unlabelled corpus [9], [10], [13], [16], [18], [22], [24], [26], [34].

Reference [16], for instance, proposed the use of hybrid technique in order to improve SVM performance. They used the K-mean clustering approach for the training subset selection and then hyperparameter tuning was done to optimize the effectiveness of classifier. They evaluated their result on Stanford Twitter Dataset (STS) [12] and the Amazon customer review dataset. Reference [13] presented a comparison of two clustering approach K-mean and Non-Negative Matrix factorization (NMF) on 30000 tweets containing the term “world cup to find topics”. Reference [18] presented a hybrid framework for sentiment analysis of unlabelled Email data. They presented a comparison of three clustering approaches including K-mean, sentiment clustering, and polarity labelling for labelling of unlabelled Email data and several supervised models for sentiment analysis such as SVM, NB, etc. Results showed that K-mean outperformed the other two approaches in the clustering of Email data and SVM performed best in sentiment analysis.

Reference [26], more recently presented a combination of two clustering approach that is K-mean and DENCLUE for twitter sentiment analysis. They observed that a combination of those two algorithms provided effective results than the state-of-the-art methods (e.g., DBSCAN, K-mean) in terms of clustering performance, run time and no. of clusters. In another work [22], authors used the combination of various techniques such as Tf-Idf, Singular Value Decomposition (SDF) (for dimensionality reduction), and artificial bee colony (ABC) (an algorithm used to detect the best initial state of centroids for K-mean) for improving the K-mean performance (41% than normal K-mean). They applied K-mean to generate clusters which were then scored by SentiWordNet [3] for class labelling.

Reference [36] proposed a Tag Score model with improved K-mean algorithm for tweets clustering into positive, negative, or neutral. They grouped semantically similar features from BOW into tags (addressed dimensionality reduction issue), scores of sentiment words were modified and, then, centroids of clusters were chosen based on the sentiment scores.

Most of the above aforementioned works used the BOW approach (e.g., [35]) to generate the feature vector for the clustering algorithm K-mean. However, considerable performances have been reported by them, but a classifier learn on the training set generated through a simple BOW approach will mimic the word look-up based distribution and might not generalize. Moreover, they ignored the impact of negation on the quality of clusters i.e. plain BOW was used without even considering the impact of local contextual semantics such as negation, intensifiers, and many more. Existing works on tweets clustering have not focused on improving the clusters quality through extensive feature engineering and modelling negation.

This provided motivation to us in proposing an improved K-mean approach for labelling of unlabelled data. In this paper, we propose the use of an improved K-mean approach for identifying the hidden patterns in the unlabelled real-time Twitter corpus based on the extensive feature engineering and negation handling.

3. METHODOLOGY

This section presents a detailed description of our proposed novel automatic approach for the labelling of real-time twitter corpus. The proposed approach cluster the real-time tweets as “Positive”, “Negative”, or “Neutral” using the K-mean clustering algorithm in combination with extensive feature engineering and negation modelling. We start with the collection of real-time tweets on various topics such as “#Demonetization”, “#Lockdown”, and “#9pm9minutes (through the use of twitter search API), followed by the pos tagging and tokenization of generated twitter corpus using CMU Pos tagger [11], designed especially for twitter. CMU tagger is able to identify linguistic peculiarities of tweets such as usernames, URLs, hashtags, emoticons, and many more as discrete entities. Thus, for each input tweet, we have a list of tweet tokens and their corresponding POS tokens. Those tokens would be very useful for extensive feature engineering. Our proposed framework incorporates several phases including corpus collection, data (tweet) pre-processing, negation modelling, feature engineering, clustering for classification, and finally cluster analysis. Fig. 2 epitomizes the workflow of proposed framework.

A. Real-time Twitter Corpus Collection

Twitter corpus is either collection of tweets on a specific topic or may be general tweets. Corpus generation is an essential part for any successful sentiment analysis system. There exist several publically available benchmark Twitter corpuses, which can be used directly for performing sentiment analysis such as most popular SemEval datasets [33], Stanford Twitter dataset [12], and many more. Those publically available corpuses are already labelled and had been used in the existing literature for sentiment analysis purpose. Nevertheless, such corpora were generated long back and contain general tweets. However, we aim to generate real-time Twitter corpuses on hot topics in particular.

Thus, we have collected real-time tweets on three different topics including tweets on “#Lockdown”, “#9pm9minutes” (i.e. light candle or torch at 9 pm for 9 minutes to show unity in India), and “#Demonetization”. It is important to note that, tweets on “#Lockdown” and “#9pm9minutes” are based on latest hot topics. On the contrary, we collected older tweets on topic “#Demonetization” for epitomizing the significance of our proposed approach on older tweets too. That is, our approach for corpus labelling not only works with latest tweets but also for older tweets of past few years.

There are two techniques for collecting real-time Twitter corpus: Search API and Streaming API. Streaming API is basically known as push of tweets as they happen in real-time (goes forward). One can access the real-time tweets using an instant query through streaming API. Firstly, for collection of tweets a connection request is send to the server. Then, server opens the connection and allows the streaming of tweets as they happen. Importantly, it allows only one single connection per Twitter account. Streaming API limitation is that it led the tweets streaming in several languages including some non-Latin alphabets. Moreover, Twitter dataset generated through streaming API form only a small fraction of actual tweets.

On the contrary, Search API is pull of tweets (goes back) commenced by the user. Search API allows the collection of tweets that have already happened. That is, past 7 days tweets can be easily and fastly collected with search query (known as back-filling). For accomplishing different operations in search API, HTTP methods are used (PUT, DELETE, POST, and GET). Also, in search API we can tune the search query based on language, time, or region. Unlike streaming API, search API provides rich set of operators for filtering the search query result based on language, sender location, and many more. Furthermore, using search API more number of tweets can be collected because we can make 15 API requests per minute. Thus, we chose search API for the generation of real-time Twitter corpora. A rate limit is linked with each search query. For handling this rate limit, we continuously sent search queries with a small delay. This helped in generation of broader range of corpora.

Thus, using search API, we collected real-time tweets on varied topics. Complete details on statistics of generated real-time Twitter corpuses are given in section 4.

B. Data Pre-processing (Tweet Normalization)

Tweets are user-generated short messages and often have oddities and quirks. Thus, the tweet is highly unstructured containing a lot of misspelled words, acronyms, and domain-specific entities. It is necessary to clean the tweet by the removal of unnecessary symbols and words which don’t have any semantic orientation such as digits, URLs, whitespaces, stopwords, and many more. The indispensable task of noise removal and normalizing the out-of-vocabulary and Non-English words to their canonical forms is known as data pre-processing. It would prepare the real-time twitter corpus for further analysis and helps in the reduction of feature space too by the removal of unnecessary elements from the tweet. Several early works highlights the significance of data pre-processing before clustering [13], [16], [22], [26], [36].

In our previous work [14], we have implemented a pre-processing framework containing two phases: basic cleaning and tweet normalization. Operations or tasks for the noise removal from tweets come under the basic cleaning phase such as removal of whitespaces, punctuations, stopwords, URLs, numbers, and many more. Tweet normalization phase includes the task of replacing the ill-formed and non-standard words to their canonical forms such as replacement of acronym “lol” by “laughing out loud”. Thus, we are able to get a clean and normalized real-time twitter datasets which is ready for the next phase that is feature engineering.

C. Negation Modelling

Negation has the ability to change the entire semantic orientation of the text. Negation handling consists of three parts: negation cue identification (explicit negation words such as no, not, etc.), scope determination (words affected by negation), and modelling the negation impact. For negation cues identification, we used a base list given by [38] and improved it by the addition of various misspelled cues such as cannt, obtained from the Twitter cluster [39]. We have determined the scope of negation too i.e. words affected by negation are suffixed with tag “_NEG”. As an illustration, consider a negation tweet:

“Suleiman and Aman discover #demonetisation is not a failure”.

After detection of negation scope tweet becomes: “Suleiman and Aman discover #demonetisation is not a failure_NEG”. Thus, token “failure_NEG” would be extracted during feature engineering instead of “failure”.

Furthermore, we exclude few negation tweets from the scope determination procedure which are having explicit negation words but literally, there is no sense of negation [15] as in tweet:

"Now isn't this lovely ! Hazards of #DeMonetization https://t.co/mBwXVxFYKN".

In that negation tweet, the word "isn't" not affecting the semantic orientation of opinionated word "lovely". Hence, negation is ignored in this tweet. Fig. 1 epitomizes the linguistic rules for identification of negation exception cases.

There are actually two situations in which negation cue has no sense of negation (cue act as non-cue) (negation exception cases).

- When negation is a part of negation phrase such as "no one", "by no means", "no more", not just, and many more. Most of the times when negation word (cue) act as a non-cue, pos tag of negation word is either "D" or "!" as in "No ! I am not ready".
- When negation is present in negative rhetoric questions as in tweet:

  "@sudhirchaudhary What the ...? **Isn't** it real dictatorship ? 1st sponsor Dangal in Bangal. And play #DeMonetisation protest politics in capital."

  In such kind of rhetoric questions, negation cue act as non-cue (e.g., "isn't" in the above tweet). For identifying such negative rhetoric questions, we analyse negation tweets and observe few linguistic patterns based on the pos tags of negation token and its neighboring tokens. As an illustration, pattern "V D A" (e.g., "isn't that great") indicates the negative rhetoric question, where "V" is pos tag of negation cue ("isn't") and "D A" is pos tags of next two adjacent tokens ("that great").

Handling the negation before the clustering of tweets prevented the misclassification of negation tweets into wrong clusters. We have conducted series of experiments across all the collected real-time Twitter corpora to validate the significance of modelling negation (especially negation exception cases) before the cluster generation process. It is important to note that, negation scope determination and the identification of negation exception cases are done during tweet normalization phase. However, negated context words are handled during feature engineering phase, where automatic lexicons are utilized for getting the real-valued score of negated context words.

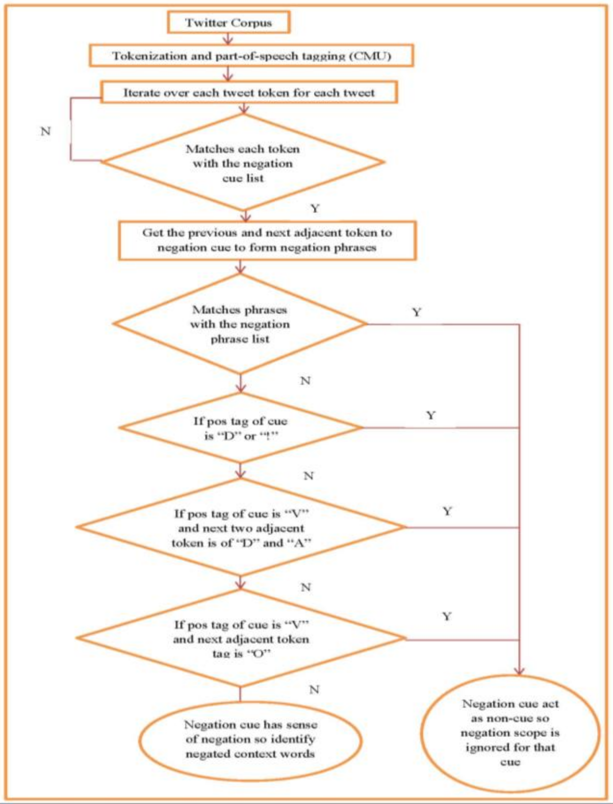

Twitter Corpus
Tokenization and part-of-speech tagging (CMU)
Iterate over each tweet token for each tweet
N
Matches each token with the negation cue list
Y
Get the previous and next adjacent token to negation cue to form negation phrases
Matches phrases with the negation phrase list
Y
N
If pos tag of cue is "D" or "!"
Y
N
If pos tag of cue is "V" and next two adjacent token is of "D" and "A"
Y
N
If pos tag of cue is "V" and next adjacent token tag is "O"
Y
N
Negation cue has sense of negation so identify negated context words
Negation cue act as non-cue so negation scope is ignored for that cue

Figure 1. Procedure for handling negation exception cases

D. Feature Engineering

It is worth to mention that, most of the machine learning algorithms need input in the form of nd-array that is "no. of observations * no. of features". Thus, there is a need for numerical representation of input twitter corpus. In this fold, we have performed extensive feature engineering which led to the generation of several syntactic and semantic features as shown in below table I. We have extracted varieties of features from the unlabelled Twitter corpora such as POS-based features, negation features, lexicon-based features, ngram, and morphological features. We generated binary (presence or absence), count (no. of occurrences), and real-valued features (polarity score of opinionated words from lexicons). Complete description of each type of feature is given in below table I.

It is important to note that, we generated lexicon-based features specifically for handling the negated context words (words affected by the negation presence). We made use of two Twitter-specific automatic lexicons [32], which contain real-valued score of words in negated as well as affirmative context. That is, for each word we get the score from automatic lexicons based on whether that word is under negated or affirmative context. Our negation handling approach is based on the fact that negation doesn't invert polarity every time. Moreover, we generated binary and count features for negated context too i.e. count of number of negated context and presence or absence of negated context as features.

For the ngram features, we evaluated two approaches: TF-IDF and CountVectorizer. TF-IDF approach punishes the frequently occurring words and reward the rare terms. On the contrary, CountVectorizer approach considers each word with its frequency. Thus, frequently occurring words will be given more weightage in CountVectorizer. However, frequently occurring words are not sentiment bearing words such as a, an, the, them, and many more. Thus, we chose the TFIDF for the ngrams representation, which assigns a real-valued score to each term in a tweet. Furthermore, to improve the quality of clusters we have thrown away the ngrams that occur less than 2 times and more than 50% in corpus. This would help in cluster strengthening through dimensionality reduction.

All the feature groups were concatenated to get a single feature vector, which would be then given as an input to K-mean for cluster generation. Those extracted features would help the K-mean in finding out the hidden structure from the unlabelled corpuses.

E. K-mean Clustering for Classification

In this phase, we have used a popular unsupervised feature-based clustering technique known as K-mean (often used for the classification) for the labelling of unlabelled real-time twitter corpus. The purpose of using the clustering approach here is segmentation as well as classification of unlabelled corpus.

K-mean is a good option because it is capable of handling high dimensional data. It is an iterative approach which is used for partitioning the dataset into pre-defined non-overlapping clusters such that points within a cluster are more similar to each other. It is a distance or centroid based algorithm in which we calculate the distances to assign a data point to a cluster. Each cluster is associated with a centroid. The main objective of K-mean is to minimize the sum of distances between data points and their respective centroids. It deals in determining structure in the unlabelled corpus. The steps for K-mean algorithm are as follows:

Let X = {x_1,x_2,x_3,……..,x_n} denote the collection of input data and V = {v_1,v_2,…….,v_c} denote the collection of centroids.

1. Choose the number of clusters ‘c’.
2. Randomly choose *‘c’* data points as cluster centroids.
3. Compute the distance among each data point and centroids of cluster.
4. Allocate the point to the cluster center for which distance from the cluster centroid is minimum of amongst the cluster centroids.
5. Now recompute the new cluster centroid using:

   $$\boldsymbol{v}_i = (1/c_i)\sum_{j=1}^{c_i} \boldsymbol{x}_i$$

   where, *‘c_i’* denotes the no. of data points in i^{th} cluster.
6. Recompute the distance among each data point and newly formed cluster centers.
7. We stop, if no point (data point) was reassigned, otherwise repeat step 4.

The main aim of K-mean is to minimize the below objective function (minimize the distance between each data point and its cluster centroid).

$$argmin_c \sum_{j=1}^{k}\sum_{x\in c_j} d(x,\mu_j)$$

Where k denotes number of clusters, c_j is set of points belong to cluster j and μ_j is centroid of cluster j. d(x, μ_j) is the Euclidean distance (sum of squares with in a cluster, known as inertia).

In this work, we performed extensive feature engineering and negation modeling before using K-mean. Thus, input to K-mean is a large feature vector which is the concatenation of various syntactic and semantic features (see table I). We enhanced our feature vector by handling local contextual semantic known as negation. The aim of using many feature groups apart from TF-IDF (in most of the early works, Tf-Idf is the only feature given as input to K-Mean) is to help the K-mean in finding hidden patterns more accurately. Additionally, classifier trains on the clusters created by K-mean through the use of many feature groups will generalize rather than mimicking.

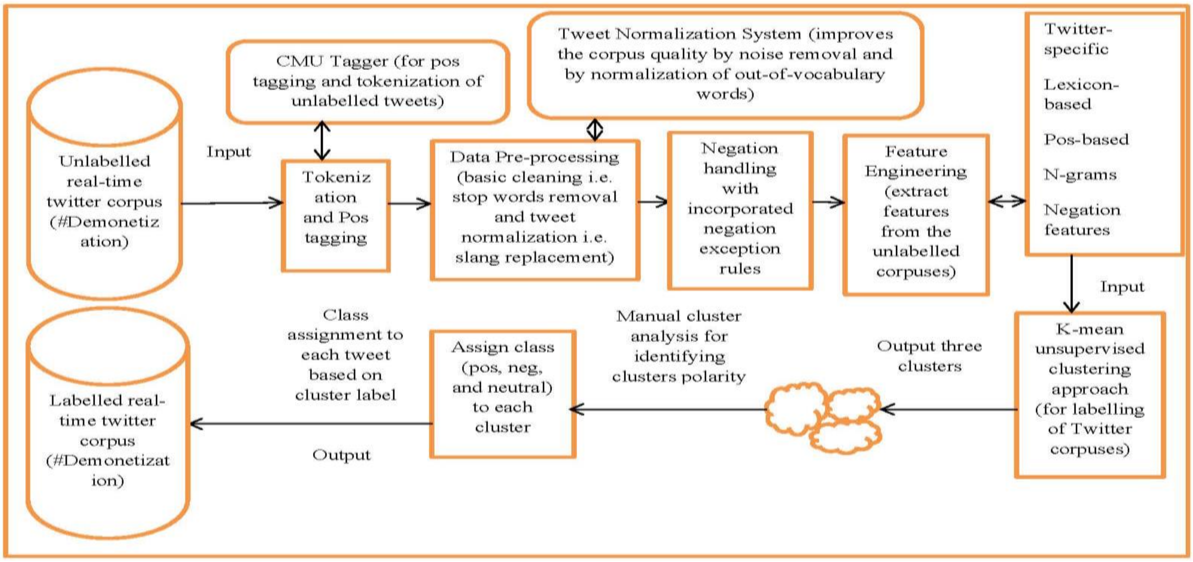

Unlabelled real-time twitter corpus (#Demonetization)
Input
CMU Tagger (for pos tagging and tokenization of unlabelled tweets)
Tokenization and Pos tagging
Tweet Normalization System (improves the corpus quality by noise removal and by normalization of out-of-vocabulary words)
Data Pre-processing (basic cleaning i.e. stop words removal and tweet normalization i.e. slang replacement)
Negation handling with incorporated negation exception rules
Feature Engineering (extract features from the unlabelled corpuses)
Twitter-specific
Lexicon-based
Pos-based
N-grams
Negation features
Input
K-mean unsupervised clustering approach (for labelling of Twitter corpuses)
Output three clusters
Manual cluster analysis for identifying clusters polarity
Assign class (pos, neg, and neutral) to each cluster
Class assignment to each tweet based on cluster label
Output
Labelled real-time twitter corpus (#Demonetization)

Figure 2. Proposed framework for automatic labelling of real-time Twitter corpus (e.g., #Demonetization)

TABLE I. FEATURE VECTOR GENERATED FROM THE UNLABELLED TWITTER CORPUS

Feature Group	**Features**	**Description**
N-grams (unigrams + bigrams)	Tf-Idf feature vector (drop the terms that occur less than 2 times and more than 50%).	Normalize the count of a token based on the no. of document in which it appears. It penalizes the most occurring term and reward the rare term.
Morphological features	• No. of hashtag words • Count and presence of elongated words • Count of emoticons • Count of exclamation, question mark, no. of tokens having only exclamation, question mark, count and existence of exclamation, question at the tweet end • No. of capitalized words • Presence of slang 21-dimensional feature vecor is generated	Twitter-specific and generated based on the linguistic peculiarities of a tweet i.e. hashtag, emoticons, specific punctuation like exclamation and question, all capitalized words, and many more.
Pos-based features	No. of existences of each unique pos tag generated by the CMU Tagger.	Part of speech features such as noun, adjective, adverb, etc. Helps in context identification.
Negation features	• Count the no. of occurences of negated context • Presence or absence of negated context	Affects lexicon-based and n-gram features. Help the clustering technique in finding the negated context patterns like “not good_NEG” so that good would not be considered as positive sentiment bearing word if tagged with “ _NEG”.
Lexicon-based features ➢ Twitter-specific automatic lexicon-based features (S140 and NRC-Hashtag)	• Count of tokens with non-zero sentiment score • Sum of score • Maximum of score • Score of the last token Generated for all positive (4 features), negative (4 features), and all tokens (4 features). Thus, 12-dimensional feature vector is generated)	Automatic lexicons are specifically used to provide the score to word under negated context based on the fact that negation doesn’t reverse polarity every time. They are having real-valued scores for unigrams and bigrams in negated as well as affirmative context. Thus, each n-gram is given two scores: one in affirmative context and another in negated context.
➢ Manual lexicon-based features (Bing-Liu, NRC-Emoticon, and MPQA)	• Sum of positive score of words in negated context • Sum of negative score of words in negated context • Sum of positive score of words in affrimative context • Sum of negative score of words in affirmative context Above 4 features are repeated for hashtag words. This gives us total 8 features. Those 8 features are generated for all—caps, lowercase and unique pos tags.	There is no real valued score for NRC-Emoticon and Bing-Liu. Put simply, Bing-Liu indicate a word as positive or negative. NRC-Emotion indicate emotions (e.g., sad, happy, etc.) also in addition with positive and negative polarity. We used a score of +1 for positive word and -1 for negative word in Bing-Liu and NRC-Emotion lexicon. MPQA indicates strength of polarity too so we used +1/-1 for weak intensity and +2/-2 for strong intensity words.

	For instance, in case of “#Demonetization” dataset 100 dimensional feature vector is generated through manual lexicons (21*4+8+8 = 100). Here 21 indicates the no. of unique pos tags identified by CMU tagger in “#Demonetization” corpus. For “#Lockdown” corpus 22*4+8+8 = 104 dimensional feature vector is generated because CMU tagger identified 22 unique pos tags.	

The output of the proposed K-mean approach is the three clusters, each having tweets that are more similar to each other. The reason for getting only three clusters is that we want to label the real-time tweets as Positive, Negative or Neutral.

F. Cluster Analysis for Class Assignment

This is the last and final phase of our framework in which generated clusters are inspected manually so that each cluster can be assigned to one of three classes namely Positive, Negative, or Neutral. We have analysed the sentiments expressed by a few tweets belonging to each cluster. There is no need to analyse each tweet of a cluster because tweets in a cluster are more similar to each other i.e. they will express the same kind of sentiment. Analysis of a few tweets respective to a cluster will give us the idea of whether a tweet is positive, negative or neutral. Fig. 3 epitomizes the manual analysis procedure for clusters. Accordingly, each cluster is assigned to one of three classes. For instance, if some of the tweets of a cluster are expressing the positive sentiment, then that cluster is assigned class “Positive”. Put simply, all the tweets of a positively assigned cluster will be labelled as “Positive”. In the end, we get a labelled real-time twitter corpus with each tweet labelled as Positive, Negative or Neutral.

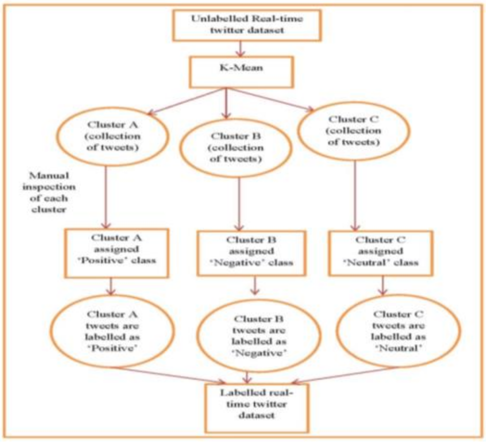

Unlabelled Real-time twitter dataset
K-Mean
Cluster A (collection of tweets)
Cluster B (collection of tweets)
Cluster C (collection of tweets)
Manual inspection of each cluster
Cluster A assigned 'Positive' class
Cluster B assigned 'Negative' class
Cluster C assigned 'Neutral' class
Cluster A tweets are labelled as 'Positive'
Cluster B tweets are labelled as 'Negative'
Cluster C tweets are labelled as 'Neutral'
Labelled real-time twitter dataset

Figure 3. Procedure for manual analysis of clusters

4. EVALUATION

Several experiments are undertaken in this work in order to show the effectiveness of our proposed automatic K-mean clustering approach cascaded with extensive feature extraction and negation modelling for the classification (labelling) of unlabelled twitter corpus.

A. Corpus

Since this work aims in labelling of real-time twitter dataset, we need to collect the real-time tweets from twitter. First of all Twitter API authentication is required to generate authentication credentials. Authentication keys (API keys) are generated upon logging in the Twitter account. To establish a connection with twitter stream API keys are needed. Twitter API allows pulling each and every tweet on a certain topic. We used python library “Tweepy” to make a connection with Twitter API. Tweepy provides a convenient way of accessing API with language Python. It contains several classes and functions that epitomize API endpoints and it handles various low-level details such as HTTP request, rate limit, encoding, decoding, and many more.

We used Twitter search API (see section 3.A for details on search API) for downloading real-time English tweets related to keyword “#Demonetization”, ‘#Lockdown’, and ‘#9pm9minutes’ with English language filtering operator. Tweets on “Demonetization” were collected from 31/12/2016 to 20/03/2017. We collected total of 19615 tweets on the topic “#Demonetization”. We also collected tweets on current hot trending topics such as tweets on “#Lockdown” and “#9pm9minutes”. Tweets on “#Lockdown” were collected from 27/03/2020 to 6/04/2020 and tweets on “#9pm9minutes” were collected from 4/04/2020 to 6/04/2020. We gathered 18365 tweets on #Lockdown and 6358 tweets on #9pm9minutes. Search API results into a JSON object containing tweet text and several associated metadata (its data about a tweet like data, time, user, etc.). We have extracted only tweet text from the JSON object and saved it into three different text files, one for each real-time dataset. Table II presents the statistics of unlabelled real-time twitter corpus that we have generated and fig. 4 portrays the real-time Twitter corpus generation procedure.

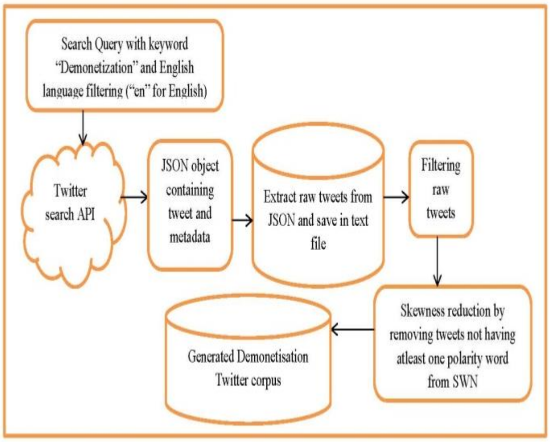

Search Query with keyword "Demonetization" and English language filtering ("en" for English)
Twitter search API
JSON object containing tweet and metadata
Extract raw tweets from JSON and save in text file
Filtering raw tweets
Skewness reduction by removing tweets not having atleast one polarity word from SWN
Generated Demonetisation Twitter corpus

Figure 4. Real-time twitter corpus generation

TABLE II. STATISTICS OF UNLABELLED REAL-TIME TWITTER DATASETS

Dataset	Size (No. of tweets)
Demonetization	19615
Lockdown	18365
9pm9minutes	6358

B. Experiments

In this section, different experimentation results are presented that determines the performance of our proposed automatic approach for clustering and labelling of tweets.

1) Results of Real-Time Twitter Corpus Labelling

Our first set of experiment presented the cluster generation results from the real-time Twitter corpus (collected in above sub-section 4.A). We performed a series of experiments with the K-mean clustering approach and combinations of several feature groups. We observed the best clusters when all features are given as input to K-mean, showing the significance of using more features with K-mean. It is important to notice that, we set the "n_clusters" hyperparameter value of K-mean to 3 because our goal is to classify the real-time twitter dataset into three classes namely "Positive", "Negative", or Neutral". Table III shows the statistics of three clusters (number of tweets per cluster) generated by K-mean for each real-time Twitter dataset.

TABLE III. POPULATION OF EACH CLUSTER (NO. OF TWEETS PER CLUSTER) FOR REAL-TIME TWITTER CORPUS

Dataset	First cluster (Cluster 0)	Second cluster (Cluster 1)	Third cluster (Cluster 2)	Total
Demonetization	4976	8865	5774	19615
Lockdown	4924	8252	5189	18365
9pm9minutes	1614	2339	2405	6358

Finally, cluster analysis was done manually for the assignment of class to each cluster. Table IV shows the result of a class assignment to each cluster for all the three twitter datasets. For instance, from table IV we observed that for the "Demonetization" dataset cluster 0 (first cluster) was assigned "Positive" class, cluster 1 (second cluster) was assigned "Neutral class, and cluster 2 (third cluster) was assigned "Negative" class. Put simply, all the tweets of cluster 0 in "demonetization" corpus were labelled as "Positive", cluster 1 tweets were labelled as "Neutral", and cluster 2 tweets were labelled as "Negative". Table V presents the final labelling of unlabelled twitter corpus based on the classes assigned to clusters and fig. 5 portrays the graphical representation of labelled real-time corpora. It is worth noting that all three real-time twitter datasets are unbalanced, so balancing of datasets is required before using it for further analysis i.e. sentiment analysis.

TABLE IV. MANUAL CLASS ASSIGNMENT TO EACH CLUSTER FOR THE REAL-TIME TWITTER DATASETS

Dataset	Cluster	Classes
Demonetization	Cluster 0 (4976)	Positive
	Cluster 1 (8865)	Neutral
	Cluster 2 (5774)	Negative
Lockdown	Cluster 0 (4924)	Negative
	Cluster 1 (8252)	Neutral
	Cluster 2 (5189)	Positive
9pm9minutes	Cluster 0 (1614)	Negative
	Cluster 1 (2339)	Neutral
	Cluster 2 (2405)	Positive

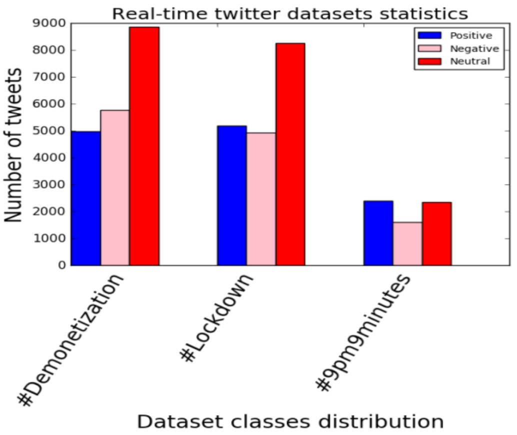


Figure 5. Labelled real-time Twitter corpus statistics

TABLE V. CLASS (LABEL) ASSIGNMENT TO TWEETS (STATISTICS OF LABELLED REAL-TIME TWITTER DATASET)

Dataset	# Positive tweets	# Negative tweets	# Neutral tweets	Total
Demonetization	4976 (25.37%)	5774 (29.44%)	8865 (45.19%)	19615
Lockdown	5189 (28.25%)	4924 (26.81%)	8252 (44.94%)	18365
9pm9minutes	2405 (37.83%)	1614 (25.38%)	2339 (36.79%)	6358

We also provided an interesting visualization of generated clusters in the word cloud form. Word cloud is an interesting technique for textual data representation such that each token (word) size is directly proportional to its importance or frequency. Word cloud helps for analysis of data for the microblogs and other social media sites. We have generated word clouds for each cluster with respect to each real-time twitter dataset. As an illustration, fig. 6 shows the word cloud for positive cluster of "#Demonetization" twitter corpus. We have observed big size words such as success, positive, great, thanks demonetisation, good, benefit, etc. in the below fig. 6, which make sense to be in positive tweets.

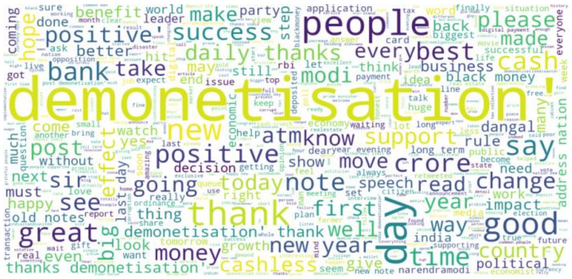


Figure 6. Positive cluster word cloud for "Demonetization" corpus

2) Comparative Study

Our next set of experiment is to present a comparison of our proposed improved K-mean model with or without negation exception rules and also with K-mean algorithm, when negation is completely disregarded. We also presented a comparative analysis with random K-mean when only conventional TF-IDF features are extracted for clustering. This would prove the advantage of extensive feature engineering for the K-mean clustering. It is important to note that, we evaluated the quality of clusters generated by our improved K-mean approach through two metrics: inertia and silhouette score. As an illustration, fig. 7 shows the silhouette plot for our improved K-mean model on 9pm9minutes dataset. Red line shows the average silhouette score.

Inertia basically defines the sum of the distances of all the data points with in a cluster from that cluster centroid. It is the mean squared distance between each sample and its cluster centroid. It gives the sum of intracluster (with in a cluster) distances. For a good clustering algorithm, there should always be less distances between the points with in a cluster. Thus, a low value of inertia is desirable. Minimizing the value of inertia will improve the K-mean algorithm performance.

Silhouette analysis is used to find out the degree of separation between clusters. Its value ranges from +1 to -1, where high value means a data point is well belonged to its own cluster. Negative value means data point is wrongly assigned to cluster. A value of 1 means clusters is well separated and distinguished. Thus, higher is the silhouette score, better is the cluster.

Silhouette Coefficient = (x-y)/ max(x,y)

Where, x indicates the mean intercluster distance and y indicates the mean intracluster distance. For instance,

Experiments were performed in four scenarios:

M1: K-mean when negation not considered at all but all features are taken.

M2: Random K-mean with conventional TF-IDF features only.

M3: K-mean without negation exception rules.

M4: Proposed K-mean in cascade with extensive feature engineering and negation modelling with incorporated negation exception rules.

All experiments were conducted for each real-time Twitter dataset and results are presented in terms of inertia and silhouette score as shown in below table VI. It is important to note that, for calculating silhouette score, 9000 samples were taken from "Demonetization" and "Lockdown" each because both the datasets are quite large in size and it is computationally expensive to calculate silhouette for the entire set as silhouette coefficient needs to be calculated for each point.

From the experimental results, we observed that our model (M4) outperformed M1, M2, and M3 in all the six cases in terms of cluster quality assessed through metric inertia and silhouette score. M1 and M2 models act as baseline for comparative analysis. We observed that our model and M3 model outperformed the baseline models in all the cases. The only difference between our model and M3 model is the removal of negation exception rules. Reason for the improvement of M4 over M3 is the handling of negation exception tweets, which prevents the misclassification. This proves the advantage of handling negation exception cases while cluster generation. Moreover, our model outperformed the M2 model (only TF-IDF features taken) in all the six cases, which once again proved the significance of extensive feature engineering for K-mean rather than using only ngrams features (TF-IDF representation).

Reason for improvement of M3 over M2 is that several varieties of features are extracted for M3 rather than only TF-IDF. Reason for improvement of M3 over M1 is that negation modelling (negation exception cases are ignored however) was done for M3 model, while it was completely ignored in M1. Among the baselines, M1 outperformed the M2 in four out of 6 cases. Reason is that in M1 model several syntactic and semantic features were used for clustering. On the contrary, in M2 model only conventional TF-IDF was used as features for K-mean.

TABLE VI. TABLE: PERFORMANCE COMPARISON OF OUR IMPROVED K-MEAN WITH RANDOM K-MEAN MODELS

Dataset	Comparison Measure	M1	M2	M3	M4 (Our Model)	Winner
Demonetization	Inertia	11644443.101	9717999.438	7315058.311	**7121231.237**	**M4**
	Silhouette	0.01354591	0.01009559	0.03589304	**0.03892406**	**M4**
Lockdown	Inertia	10860228.209	9140360.733	6642548.961	**6391414.709**	**M4**
	Silhouette	0.01988364	0.01165497	0.04875417	**0.04940619**	**M4**
9pm9minutes	Inertia	1106012.190	1820733.503	987796.993	**979154.480**	**M4**
	Silhouette	0.23885369	0.11685138	0.24572806	**0.24826107**	**M4**

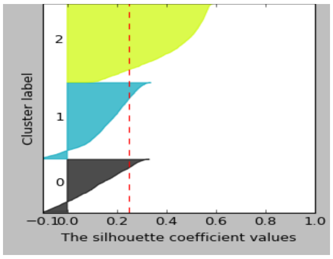


Figure 7. Silhouette plot for our improved K-mean model on 9pm9minutes dataset

5. CONCLUSION AND FUTURE WORK

In this work, we have proposed a novel improved automatic framework (amalgamation of extensive feature engineering, negation modelling, and K-mean) for the labelling of real-time Twitter datasets into three classes namely Positive, Negative, or Neutral for the textual-based Twitter Sentiment Analysis. We chose Search API for the collection of real-time tweets on different topics because search API led to the collection of broader datasets. Tweets were gathered on latest hot topics such as “#Lockdown” and “#9pm9minutes”. Also, we generated corpus containing older tweets on “Demonetization” to prove the vulnerability of our automatic labelling approach on older tweets too.

Extensive feature engineering was conducted to extract varieties of syntactic and semantic features (such as POS-based, Twitter-specific, negation, lexicon-based, and ngrams features) from the collected corpora that would help in finding the hidden structure from the unlabelled corpora. Moreover, we handled one critical aspect of NLP namely “negation” before performing clustering. Based on the fact that, negation presence doesn’t necessarily mean negation, we identified two such scenarios (known as negation exception cases) and presented linguistic rules for them.

Each generated cluster was manually inspected for determining whether the tweets in a particular cluster are expressing positive, negative, or neutral sentiment. We considered only few tweets for manual inspection among each cluster based on the fact that tweets with in a cluster are more similar to each other. That mean, each cluster was labelled as Positive, Negative, or Neutral. Finally, based on the manual inspection each tweet was labelled with one of three classes namely Positive, Negative, or Neutral. That is, if a cluster was labelled as “Positive” during the manual inspection, then all the tweets belonging to that cluster were labelled as “Positive”.

At the end, comparative analysis of our proposed approach (improved K-mean) was done with or without negation exception rules, with conventional TF-IDF feature model using random K-mean, and with K-mean when negation is disregarded at all. Evaluations were done on the basis of clusters quality assessed through inertia and silhouette score. Results showed that our proposed model generated quality clusters when compared to K-mean without negation exception rules, without negation at all, and conventional TF-IDF using random K-mean.

In the future, we shall aim for the optimization of feature vectors for dimensionally reduction and cut down of computational cost. Moreover, we would explore the process of twitter sentiment analysis on labelled real-time twitter dataset in order to show the effectiveness of labels generated by K-mean.

REFERENCES

[1] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R.J. Passonneau, “Sentiment analysis of twitter data,” in Proceedings of the Workshop on Language in Social Media (LSM 2011), June 2011, pp. 30-38.

[2] M.Z. Asghar, A. Khan, S. Ahmad, M. Qasim, and I.A. Khan, “Lexicon-enhanced sentiment analysis framework using rule-based classification scheme,” PloS one, vol. 12, no. 2.

[3] S. Baccianella, A. Esuli, and F. Sebastiani, “Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining,” in proceedings of the Lrec, May 17-23, 2010, vol. 10, no. 2010, pp. 2200-2204, Valletta, Malta.

[4] A. Bakliwal, J. Foster, J. van der Puil, R. O'Brien, L. Tounsi, and M. Hughes, “Sentiment analysis of political tweets: Towards an accurate classifier,” in NAACL workshop on language analysis in social media, Association for Computational Linguistics, June 13, 2013, Atlanta, GA.

[5] K.Z., Bertrand, M. Bialik, K. Virdee, A. Gros, and Y. Bar-Yam, “Sentiment in new york city: A high resolution spatial and temporal view,” arXiv preprint arXiv:1308.5010, August 22, 2013.

[6] K.A. Crockett, D. Mclean, A. Latham, and N. Alnajran, “Cluster Analysis of twitter data: a review of algorithms,” in Proceedings of the 9th International Conference on Agents and Artificial Intelligence (Vol. 2, pp. 239-249). Science and Technology Publications (SCITEPRESS)/Springer Books, 2017.

[7] D. Davidov, O. Tsur, and A. Rappoport, “Enhanced sentiment learning using twitter hashtags and smileys,” in Proceedings of the 23rd international conference on computational linguistics: posters, August 23-27, 2010, pp. 241-249, Beijing, China

[8] I. El. Alaoui, Y. Gahi, R. Messoussi, Y. Chaabi, A. Todoskoff, and A. Kobi, “A novel adaptable approach for sentiment analysis on big social data,” Journal of Big Data, vol. 5, no. 1, pp. 12, 2018.

[9] D. Ferraretti, G. Gamberoni, and E. Lamma, “Unsupervised and supervised learning in cascade for petroleum geology,” Expert Systems with Applications, vol. 39, no. 10, pp. 9504-9514, 2012.

[10] V. Friedemann, “Clustering A Customer Base Using Twitter Data,” CS-229, 2015.

[11] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, et al., “Part-of-speech tagging for twitter: Annotation, features, and experiments,” in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2, June 2011, pp. 42-47. Association for Computational Linguistics.

[12] A. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision,” CS224N project report, Stanford, vol. 1, no. 12, 2009.

[13] D. Godfrey, C. Johns, C. Meyer, S. Race, and C. Sadek, “A case study in text mining: Interpreting twitter data from world cup tweets,” arXiv preprint arXiv:1408.5427, 2014.

[14] I. Gupta and N. Joshi, “Tweet normalization: A knowledge based approach,” in 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS), December 2017, pp. 157-162, Dubai, UAE.

[15] I. Gupta and N. Joshi, “Enhanced Twitter Sentiment Analysis Using Hybrid Approach and by Accounting Local Contextual Semantic,” Journal of intelligent systems, vol. 29, no. 1, pp. 1611-1625, 2019.

[16] K. Korovkinas, P. Danėnas, and G. Garšva, “ SVM and k-Means Hybrid Method for Textual Data Sentiment Analysis,” Baltic Journal of Modern Computing, vol. 7, no. 1, pp. 47-60, 2019.

[17] E. Kouloumpis, T. Wilson, and J. Moore, “Twitter sentiment analysis: The good the bad and the omg!,” in Fifth International AAAI conference on weblogs and social media, July 2011, Barcelona, Spain.

[18] S. Liu and I. Lee, “Email sentiment analysis through k-means labeling and support vector machine classification. Cybernetics and Systems, vol. 49, no. 3, pp. 181-199, 2018.

[19] S.M. Mohammad, X. Zhu, S. Kiritchenko, and J. Martin, “Sentiment, emotion, purpose, and style in electoral tweets. Information Processing & Management,” vol. 51, no. 4, pp. 480-499, 2015.

[20] S. Mukherjee and P. Bhattacharyya, “Sentiment analysis in twitter with lightweight discourse analysis,” in Proceedings of COLING 2012, December 2012, pp. 1847-1864.

[21] A. Muhammad, N. Wiratunga, and R. Lothian, “Contextual sentiment analysis for social media genres,” Knowledge-based systems, vol. 108, no. 92-101, 2016.

[22] K. Orkphol and W. Yang, “Sentiment Analysis on Microblogging with K-Means Clustering and Artificial Bee Colony,” International Journal of Computational Intelligence and Applications, vol. 18, no. 03, 1950017, 2019.

[23] A. Pak and P. Paroubek, “Twitter as a corpus for sentiment analysis and opinion mining,” in proceedings of the LREc, May 17-23, 2010, vol. 10, no. 2010, pp. 1320-1326, Valletta, Malta.

[24] R.H. Patil and S.P. Algur, “Classification Connection of Twitter Data using K-Means Clustering,” IJITEE, vol. 8, 2019.

[25] J. Read, “Using emoticons to reduce dependency in machine learning techniques for sentiment classification,” in Proceedings of the ACL student research workshop, June 2005, pp. 43-48.

[26] H. Rehioui and A. Idrissi, “New Clustering Algorithms for Twitter Sentiment Analysis,” IEEE Systems Journal, vol. 14, no. 1, pp. 530-537, 2019.

[27] D.A. Shamma, L. Kennedy, and E.F. Churchill, “Tweet the debates: understanding community annotation of uncollected sources,” in Proceedings of the first SIGMM workshop on Social media, October, 2019, pp. 3-10.

[28] J. Spencer and G. Uchyigit, “Sentimentor: Sentiment analysis of twitter data,” in SDAD@ ECML/PKDD, pp. 56-66, 2012.

[29] E. Cambria, “Affective computing and sentiment analysis,” IEEE Intelligent Systems, vol. 31, no. 2, pp. 102-107, 2016.

[30] E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall, “Sentiment analysis is a big suitcase,” IEEE Intelligent Systems, vol. 32, no. 6, pp. 74-80, 2017.

[31] M. Ebrahimi, A.H. Yazdavar, and A. Sheth, “Challenges of sentiment analysis for dynamic events,” IEEE Intelligent Systems, vol. 32, no. 5, pp. 70-75, 2017.

[32] S. Kiritchenko, X. Zhu, and S.M. Mohammad, S. M., “Sentiment analysis of short informal texts,” Journal of Artificial Intelligence Research, vol. 50, pp. 723-762, 2014.

[33] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, “SemEval-2016 task 4: Sentiment analysis in Twitter,” in proceedings of the 10th international workshop on semantic evaluation (SemEval 2016), 2016, pp. 1-18. San Diego, California.

[34] P. Ray and A. Chakrabarti, “A mixed approach of deep learning method and rule-based method to improve aspect level sentiment analysis,” Applied Computing and Informatics, 2019.

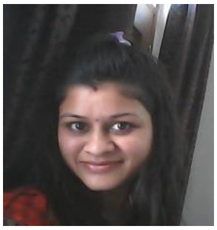
[35] M.I. Zul, F. Yulia, and D. Nurmalasari, “Social Media Sentiment Analysis Using K-Means and Naïve Bayes Algorithm,” in 2018 2nd International Conference on Electrical Engineering and Informatics (ICon EEI), October 2018, pp. 24-29, IEEE.

[36] S. POOMAGAL, B. MALAR, J.I. HASSAN, and R. KISHOR, “A novel Tag Score (T_S) model with improved K-means for clustering tweets,” Sadhana, vol. 45, no. 1, 2020.

[37] S. Wu, Y. Liu, J. Wang, and Q. Li, “Sentiment Analysis Method Based on Kmeans and Online Transfer Learning,” CMC-COMPUTERS MATERIALS & CONTINUA, vol. 60, no. 3, pp. 1207-1222, 2019.

[38] I.G. Councill, R. McDonald, and L. Velikovich, “What's great and what's not: learning to classify the scope of negation for improved sentiment analysis,” In Proceedings of the workshop on negation and speculation in natural language processing, July 2010, pp. 51-59, Association for Computational Linguistics.

[39] O. Owoputi, B. O’Connor, C. Dyer, K. Gimpel, N. Schneider, and N.A. Smith, “Improved Part-of-Speech tagging for online conversational text with word clusters,” In Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies, June 2013, pp. 380-390.



Itisha Gupta is a research scholar in the department of computer science at Banasthali Vidyapith, Rajasthan, India. She did MCA (Masters of computer applications) from Gurgaon Institute of Technology and Management, Gurgaon, India. Her areas of interests are Data analysis, Natural Language Processing and machine learning

Nisheeth Joshi is an Associate Professor in the department of computer science at Banasthali Vidyapith, Rajasthan, India. He primarily works in Machine Translation, Information Retrieval and Cognitive Computing. He has over 12 years of teaching experience.