Annals of Mathematics and Artificial Intelligence (2021) 89:371–407
https://doi.org/10.1007/s10472-020-09709-z

NSLPCD: Topic based tweets clustering using Node significance based label propagation community detection algorithm

Jagrati Singh[1] · **Anil Kumar Singh**[1]

Published online: 24 September 2020

Abstract
Social networks like Twitter, Facebook have recently become the most widely used communication platforms for people to propagate information rapidly. Fast diffusion of information creates accuracy and scalability issues towards topic detection. Most of the existing approaches can detect the most popular topics on a large scale. However, these approaches are not effective for faster detection. This article proposes a novel topic detection approach – Node Significance based Label Propagation Community Detection (NSLPCD) algorithm, which detects the topic faster without compromising accuracy. The proposed algorithm analyzes the frequency distribution of keywords in the collection of tweets and finds two types of keywords: topic-identifying and topic-describing keywords, which play an important role in topic detection. Based on these defined keywords, the keyword co-occurrence graph is built, and subsequently, the NSLPCD algorithm is applied to get topic clusters in the form of communities. The experimental results using the real data of Twitter, show that the proposed method is effective in quality as well as run-time performance as compared to other existing methods.

Keywords Tweet clustering · Supervised and Unsupervised technique · Label propagation · Keyword co-occurrence · Topic modeling

1 Introduction

The microblogging platform - Twitter has become the most popular communication channel to share information for users. Nearly 500 million tweets per day and 6000 tweets[1] per

[1] https://www.dsayce.com/social-media/tweets-day/(31October2019)

✉ Jagrati Singh
singh.jagriti5@gmail.com

Anil Kumar Singh
ak@mnnit.ac.in

[1] CSED, Motilal Nehru National Institute of Technology Prayagraj, Prayagraj, India

second are generated by 330 million active users[2]. Twitter has various features that make it better from news media websites, blogs, or other traditional information channels like television and newspapers. Users in real-time generate tweets. Due to the limitation of content size (280 characters for a tweet), twitter is called microblog rather than a blog (no restriction on content size). With the brevity guaranteed by a 280-character-tweet limit and the popularity of mobile applications, people do tweet and retweet instantly. Thus, many times Twitter reports the news first and later captured by traditional news media agencies.

Tweets have extensive coverage of real-world events that cover every aspect of daily life. Tweets are user generated content. So, Users can report news related to any event happening around them. Due to the rapid and extensive information diffusion, researchers are interested in analyzing the information to gain knowledge of current trending events. In particular, various research studies are being followed to answer the question, "What is the trending topic right now?". The process of detecting and summarizing hot issues in the form of news information is called topic detection. As an application, timely detection of disaster-related events over the Twitter stream is instrumental in disaster management and decision making to save various people's lives and properties [1, 2].

Most of the existing topic detection approaches have been designed for analyzing news articles containing long sentence structure. However, such traditional methods do not apply to the tweets containing short sentence structure with informal use of language (misspelled keywords, multi-language, abbreviations). Moreover, tweets containing useful information is very low in number compared to the volume of all tweets. Also, it is very difficult to identify the relationship between tweets due to the diversity of vocabulary, thus making it challenging to distinguish topics. Ultimately, a faster topic detection approach is needed because a huge volume of tweets is produced at a very rapid rate.

Among the various existing approaches for trending topic detection in Twitter, feature-pivot based techniques are most suitable. It considers a topic to be a cluster of keywords that co-occur. Recently, graph-based community detection algorithms have been widely used for topic detection in Twitter. Sayyadi and Raschid [3] designed a keyword co-occurrence graph model and apply edge betweenness [4] community detection algorithm with $O(n^3)$ complexity to extract topics in the form of communities from Twitter data. During the graph construction, they filter edges based on keyword co-occurrence frequency to remove noisy keywords. We observed two major issues in this approach. First, the high time-complexity of the community detection algorithm. Second, the cluster splitting problem wherein a cluster representing the topic gets divided into many subtopics.

This article proposes a topic detection approach for Twitter using an improved label propagation community detection algorithm. The proposed approach considers both the accuracy and scalability issues. To overcome the high time-complexity, the label propagation [5] community detection algorithm is extended. Traditional label propagation is good enough for faster detection because of linear time-complexity. However, it is not appropriate for good quality performance due to the random and uncertain nature of the label updating process. So, to handle the random nature of the algorithm, we fixed the node processing order and selected the label associated with the set of high significant nodes when there is more than one highest frequency label present. For handling the problem of cluster splitting, we propose a new edge filtering method to find out subtopics of each detected topic in one community instead of multiple communities. The experimental results demonstrate that the

[2]https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/(31October2019)

proposed approach has superior quality performance as well as run-time performance than the compared state of art approaches.

The rest of the paper is organized as follows. The next section briefly discusses the related work based on topic detection and community detection algorithms. Section 3 describes the proposed graph-based approach in detail. The experimental dataset, evaluation method, and results obtained by comparing the proposed approach with four competitive baseline methods have been presented in section 4. Finally, section 5 concludes with directions for further research.

2 Related Work

The problem that this article focuses, has been addressed by researchers under the two broad categories of topic detection and community detection. We have categorically discussed the related work in the following subsections.

2.1 Topic Detection

A lot of research has been done in this area using Twitter data based on different methods that can be classified into three types:

1. Document-pivot methods: Find groups based on document similarity [6, 7].
2. Feature-pivot methods: Make clusters based on feature similarities like keywords, segments, links [3, 8–11].
3. Probabilistic topic modeling methods: Group the similar patterns based on the statistical behavior of input documents [12, 13].

These techniques have been reviewed in the following subsubsections.

2.1.1 Document-Pivot Techniques

Petrović et al [14] proposed an online first story detection approach for twitter data. They used the Locality Sensitive Hashing (LSH) method to find the nearest neighbor in the Twitter search space. This approach gives the subset of tweets, called a thread, which are all related to the same topic. Osborne et al [15] improved the precision of the system proposed by [14] by utilizing Wikipedia page views to rank the topic threads. Petrović et al [16] improved the accuracy of the system proposed in [14] by handling the problem of the high degree of lexical variation in documents that means semantics of various documents is the same but expressed using different words. They presented a new way of combining paraphrases with LSH to get a more accurate system.

Feng et al [17] proposed an event detection system based on the extended LSH algorithm by adding two kinds of hash functions related to content and location instead of only content. Cluster scoring step is missing because they crawl tweets of some specific domain related events based on keyword search criteria. Hasan et al [18] improved the LSH based event detection system by combining random indexing based term vector model with LSH to capture the semantic correlation between terms. Alsaedi et al [19] proposed an event detection framework to track small scale events of particular locations like terrestrial events and events during riots. Naive Bayes classifier is used to filter out the noisy tweets.

2.1.2 Feature-Pivot Techniques

Mathioudakis and Koudas [9] extract and group the bursty keywords based on co-occurrences in some number of tweets by using the QueueBurst algorithm and identify the origins of frequent tweets from each group to detect the location of an event. To describe an event, frequently cited links are extracted from tweets of each group. Li et al [20] proposed an event detection system named Twevent, which is based on bursty segments (consecutive phrases in Web N-Gram) to detect events. The importance of the segment as an event candidate is detected by utilizing Wikipedia. After detecting event segments, they are clustered into groups based on the similarity between event segments.

Ifrim et al [21] proposed an approach based on selecting a bursty bi-gram and tri-gram for aggressive term filtering. They utilized two-stage hierarchical clustering where first stage groups similar tweets based on cosine similarity and second stage groups the resulting headlines from the first clustering step for solving the problem of topic fragmentation.

Sayyadi and Raschid [3] proposed a topic detection system that is based on the community detection algorithm within the keyword co-occurrence graph. Nodes represent keywords, and edges denote the relationship between keywords if they co-occur in the same document. Keywords are filtered if document frequency is low. An edge should satisfy two conditions: first, keywords co-occur above some threshold, and second, the conditional probability of the occurrences and similarity between keywords is higher than the predefined threshold. Between-ness centrality score is used to find the edges between two communities, and cosine similarity is used to build clusters of similar types of communities. The main drawback of this approach is the higher time complexity of $0(n^3)$.

Zhao et al [22] proposed a summarization framework based on the word dependency graph approach, in which nodes represent keywords and edges represent dependency grammar relation between keywords. Dependency grammar relations between the words are generated using the dependency grammar technique, and the importance score of the keywords is calculated using Hypertext-Induced Topic Search (HITS) algorithm. Maximal marginal relevance algorithm is used to rank the relevant sentences.

Zhang et al [23] proposed a local real-time event detection approach from geo-tagged twitter streams named GeoBurst+. Initially, geo-tagged tweets are grouped based on geographical and semantic proximity to make the event candidates. The Epanechnikov kernel function is used to calculate geographical proximity. For semantic proximity, they built a keyword co-occurrence graph and applied random walk with restart(RWR) to define the similarity between keywords. After that, cluster scoring is done based on geo-topical authority score that is measured by geographical and semantic proximity of each cluster.

Hossny and Mitchell [24] proposed a system based on tracking of each word-pair related to civil unrest events on Twitter. Choi and Park [25] proposed an approach to detect emerging topics from Twitter based on High Utility Pattern Mining (HUPM), which considered the utility as well as the frequency at the same time.

2.1.3 Topic Modeling Techniques

Latent Dirichlet Allocation (LDA) [12] is a probabilistic topic model that considers each document as a collection of keywords containing more number of topics. The authors used the Bayesian inference model to calculate topic distribution per document and keyword distribution per topic. Some limitations are reported while applying it on micro-blogging data like the predefined number of topics, higher time-complexity, and data sparsity problem due to the limitation of characters per document. Mehrotra et al [26] improved the LDA model

to handle data sparsity problem of Twitter data by using pooling schemes like author wise pooling, burst-score wise pooling, temporal pooling and proposed hash-tag based pooling that group tweets into "macro-document" as a pre-processing step.

Zhou and Chen [27] proposed a Location-Time Constrained Topic (LTT) model that is an extension of LDA by adding location and time parameters. The authors also capture the social connections between users by measuring link similarity between two messages. KL-divergence measure is used for content similarity, and the Longest Common Subsequence method is used for link similarity. To expedite the detection process, a new hash-based index scheme named variable dimensional extensible hash is used.

2.2 Community Detection

Community detection in complex networks has become one of the challenges in the era of big data research. Therefore, Complex Network Analysis (CNA) has attracted more and more attention to research communities to find valuable explanations for behavior prediction and functional analysis of complex networks [28]. For example, the analysis of similarities of entities in a given social network (viz., Facebook, Twitter, etc.) represents their specific behaviors. Hence, we detect those groups of entities as a community. Their detection could help to understand the working and structure of complex networks.

Topic-based community detection approaches [29–32] started gaining attention to identify similar topics. Girvan and Newman [33] proposed the concept of community structure that contains more dense connections within the same community compared to different communities. Community detection algorithms can be divided into four major categories:

1. Modularity based algorithms
2. Clique percolation based algorithms
3. Hierarchical partitioning based algorithms
4. Label propagation based algorithms

The research work pertaining to these categories have been reviewed in the following subsubsections.

2.2.1 Modularity based Algorithms

Newman and Girvan [34] proposed the concept of modularity that plays an important role in deciding the quality of community structure. The larger value of modularity indicates a better community structure. Newman [35] proposed an algorithm named FastQ that initialized each node as an individual community and merged the two individuals with the greater increment or the lowest decrease in modularity followed by the greedy approach. The process is repeated until the community structure gets stable. Clauset et al [36] observed that each time merging of communities is a time-consuming process in the FastQ algorithm. Therefore, to overcome the limitation of the FastQ algorithm, a faster CNM algorithm is proposed using balanced binary trees and max heaps data structure to merge two communities quickly.

Blondel et al [37] proposed the Louvain algorithm based on modularity optimization, to maximize the modularity of the whole community structure. Waltman and Van Eck [38] proposed a smart local moving algorithm (SLM) that reapplies the Lovain algorithm on resultant communities after the first phase of the Louvain algorithm. In the next phase, each resultant community is assumed as a node to build a new network. Therefore, it performs better than Louvain by considering more number of iterations.

2.2.2 Clique percolation based algorithms

A clique is a subset of vertices of an undirected graph G such that every two distinct vertices in the clique are adjacent; that is, its induced subgraph is complete. The clique percolation mehod is a popular approach for analyzing the overlapping community structure of networks. The core idea of clique percolation based algorithms is to identify the community as an aggregation of complete sub-graphs named clique linked by a shared node. Palla et al [39] proposed the clique percolation method (CPM) in which edges within the community are more promising to build complete sub-graphs based on the idea of the close relationship between the nodes within the same community. The algorithm needs a user-defined parameter k, indicating the number of nodes present in the search for the clique and k affects the community detection result. In case of a smaller value of k, the large number of communities are eventually detected with a sparse community structure. One of the limitations is the restriction of nodes allocation inside of the complete sub-graph. For sparse real-time networks, conditions of CPM are not suitable.

Kumpula et al [40] proposed the sequential clique percolation algorithm (SCP) based on the idea of CPM, using a serialization method for community detection. In many cases, the SCP is better than the CPM when k is very small, but in case of a large value of k, the time complexity would be quite higher. Lee et al [41] proposed a greedy clique expansion (GCE) algorithm which detects all maximal cliques consist of at least k nodes in the network as seeds, and applies the fitness function to populate the current unpopulated maximum seeds.

2.2.3 Hierarchical partitioning based algorithms

One set of approaches initially assume all nodes in one community, then partition them based on certain criteria. Hierarchical partitioning based community detection algorithms can be divided into two types: Divisive hierarchical method and Agglomerative hierarchical method. The Divisive approach considers top to bottom approach until a single node is treated as a community. In contrast, the Agglomerative approach initially considers a single node as a community and iteratively merges other nodes into a larger community in a bottom to top manner.

Girvan and Newman [33] proposed the hierarchical community detection approach named the Girvan Newman (GN) algorithm that computes the edge between-ness value of all existing edges and repeatedly removes the edge with the highest value of edge between-ness followed by top-down hierarchical approach. The time complexity of the GN algorithm is very high because it needs to compute the edge between-ness value of each edge repeatedly, but the performance is of better quality. Gregory [42] extends the GN algorithm to calculate the overlapped communities by introducing split node between-ness value based on edge between-ness and remove the edges repeatedly based on the more substantial value of split between-ness.

2.2.4 Label propagation based algorithms

Basically, in a complex network, edges represent the propagation of information between nodes. According to community structure, nodes within the community contain the same information, while different community nodes contain different information. This lead to the generation of label propagation based community detection algorithms. Label Propagation Algorithm (LPA) Raghavan et al [5] follows some heuristics to transmit label information iteratively between nodes. It starts by assigning unique label id to each node,

and in each iteration, the node updates its label to the one shared with the highest frequency among neighbors. If there is more than one highest frequency label present, then the algorithm selects a label at random. In this iterative process, nodes of densely connected components of the graph get the same label and form a community. The following (1) does label updating:

$$C_i = \arg\max_{l} \sum_{j \in N^l(i)} 1 \tag{1}$$

where, $N^l(i)$ denote the neighboring nodes of n_i labeled with l and C_i represent community assigned to node n_i.

The biggest advantage of the algorithm is the linear time complexity, so the run-time performance is very high. But it does not support the case of overlapped community structure and it is also very difficult to find the optimal solution while processing large networks due to the random nature of the algorithm. Gregory [42] extends the LPA algorithm by proposing an overlapped community structure, naming it as Community Overlap PRopagation Algorithm (COPRA). Xie and Szymanski [43] also extends the LPA algorithm to support the overlapping community structure by introducing a label storage list for each node. Nodes with more than one label are considered as overlapping nodes.

In a study, Xing et al [44] improved the label propagation algorithm by updating the label based on the influence of degree and edge weight of associated neighbors when the majority of neighboring nodes contain a set of labels instead of one label. Liu et al [45] proposed the edge label propagation algorithm (ELPA) by combining the link community with the execution efficiency of the LPA. Gui et al [46] proposed the label boundary node algorithm (LBN) that handles the random update process in the label propagation to improve the stability of the algorithm. Our proposed approach - NSLPCD handles the randomness nature of the LPA algorithm to improve the quality performance of detected communities.

3 Proposed Method

This article proposes a new approach for faster and precise topics detection in a set of Tweets. The proposed approach consists of several steps required to detect topic communities, which are shown in block diagram of Fig. 1. There are three major steps: The first step builds a graph where nodes represent keywords of tweet text, and edges represent co-occurrence of keyword pairs in the same tweet. The second step applies an improved label propagation community detection algorithm to find communities of different topics. The third step calculates cosine similarity between community keywords and tweets to extract most representative tweets to summarize each topic. The major steps have been described in the following subsections.

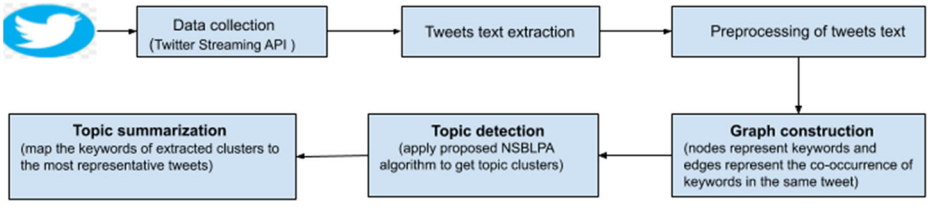


Fig. 1 Overview of proposed approach

3.1 Graph Construction

The construction of the graph is done through the data of the preprocessed dataset. The preprocessing of dataset required normalization of tweets to remove stopwords, punctuation, user mentions, URLs, digits, and other useless symbols. Further, tweets having less than four tokens were removed. The reason for removing such tweets is that usually, very short messages do not convey important information related to the topic. It is very difficult to identify the topic of a tweet through only two or three keywords during the labeling of data to make ground truth data.

When any topic gets popularized, the frequency of some keywords suddenly reaches some peak value. Such keywords are denoted as "topic-identifying keywords". Other co-occurring keywords, which play an important role in describing the topic or tracking the sub-topics (for understanding the whole topic), are denoted as "topic-describing keywords". Most of the existing techniques concentrate on only topic-identifying keywords, which is not sufficient to understand the whole topic.

Two frequency thresholds "$node\_max\_freq$" and "$node\_min\_freq$" are considered to decide the frequency range of the topic-identifying and topic-describing keywords. To determine these threshold values, we need to compute the frequency distribution of the keywords. Then, we select the $node\_max\_freq$ threshold value from the high-frequency range that contains a few most important keywords and $node\_min\_freq$ value from the medium-frequency range that contains more number of the keywords. The frequency of topic-identifying keywords should be greater than $node\_max\_freq$ threshold, and the frequency of topic-describing keywords should lie between $node\_min\_freq$ and $node\_max\_freq$ threshold values. The keywords which occur in lesser frequency (below $node\_min\_freq$ threshold) are removed. Such keywords usually do not play a role in topic detection. These keywords are considered as noisy keywords, which include misspelled keywords, abbreviations, and slang keywords. For example, keywords like *Fig*, *brd*, *gooood*, *Alahabaad*, etc. occurred in low frequency as compared to other correct keywords in the corpus. The quality of performance varies depending on the values of the parameters. Also, the best threshold values for one dataset, differ for another dataset.

There are some hashtag keywords also, which occur in less frequency due to the lexical variation problem but are not considered as noisy. For example, different variants of hashtag like "#GorakhpurTragedy", "#TragedyInGorakhpur", "#GORAKHPURTRAGEDY", "#gorakhpurtragedy" refer to the same topic and all variants play an important role in topic detection. To handle the lexical variation problem, we performed hashtag normalization based on case normalization and syntactic segmentation. Initially, the hashtags that are written in CamelCase notation are processed. For example, "#TragedyInGorakhpur" is segmented as "Tragedy", "In", and "Gorakhpur". Then, the lower case of segments extracted from CamelCase notation hashtags along with other keywords (which are extracted from the corpus) are used for the segmentation of its variants that are not present in CamelCase notation. For example "gorakhpurtragedy" could be segmented into "gorakhpur" and "tragedy". The process of segmentation helps to increase the frequency count of the main keywords, which are used to identify the topic.

Once the keywords are extracted the graph is drawn. Nodes are created corresponding to each of the keywords left after preprocessing. Edges are drawn between any two co-occurring keywords in a tweet. At this stage, the generated graph is very dense, with a large number of edges. Processing such a graph can be expensive for computational cost. Hence, a new edge filtering method is proposed to make the process faster. We consider

only those edges that are drawn either between topic-identifying keywords or between topic-identifying and topic-describing keywords to capture important information related to each topic. The frequency of keywords decides node significance value towards topic detection. Nodes containing topic-identifying keywords have higher significance value compared to nodes that contain topic-describing keywords.

Algorithm 1 : Graph construction.

Input: Tokenized tweets corpus where each line represents one tweet.
Output: Graph G(V, E) where V represents node and E represents edge

1. Extract the keywords from the tweet corpus and store the frequency of each keyword in the dictionary dict_freq
2. Set the $node\_min\_freq$ and $node\_max\_freq$ threshold values and remove the keywords which contain frequency less than $node\_min\_freq$ threshold value and update the dictionary dict_freq
3. For each keyword k_i in dict_freq do:
   (a) Create one node n_i
   (b) If $freq(k_i) > node\_max\_freq$
   (c) node is labeled as topic-identifying keyword node
   (d) Else node is labeled as topic-describing keyword node
4. Create one edge $e_(i, j)$ between each pair of keyword nodes if both keywords present in the same tweet
5. Filter those edges that do not contain at least one topic-identifying keyword node

For a better understanding of the graph construction, its process has been demonstrated on the following set of example tweets related to Virginia protest on 13 and 14 August 2017.

1. *FBI opens civil-rights probe after a car slammed into a crowd of protesters in Virginia.*
2. *Two Virginia State Police troopers were killed when their helicopter crashed and burned near Charlottesville rally.*
3. *Deadly day in Virginia white supremacist rally blamed for dozens of deaths.*
4. *Marco Rubio calls events in Charlottesville rally, Virginia "a terror attack by white supremacists".*

Let us suppose that $node\_min\_freq$ threshold value is 1 and the $node\_max\_freq$ value is 2. *Virginia, Charlottesville, white, supremacist, rally* would be considered as topic-identifying keywords and other keywords like helicopter, protesters, crashed, killed, police, troopers, etc. would be considered as topic-describing keywords. An edge between any pair of topic-describing keywords can be filtered because even without considering these relationships, we can capture these keywords to describe the topic. We can track sub-topics regarding each topic from the collection of tweets.

In the KeyGraph approach, three thresholds - $edge\_min\_df$, $node\_min\_df$ and $node\_min\_prob$ are used. $edge\_min\_df$ threshold is used to filter edges. It is defined as the minimum number of tweets containing both keywords connected with an edge. $node\_min\_df$ threshold is used to filter noisy nodes. $node\_min\_prob$ in another edge filtering threshold, which calculates the co-occurrence probability of connected keywords.

In the considered example, suppose $node\_min\_df$ value is 1 and $edge\_min\_df$ value is 2. After removing stop words, keywords make nodes of the graph. The first criteria of edge filtering yield a graph with edges between *Charlottesville-Virginia, Charlottesville-rally, Virginia-rally, Virginia-white, Virginia-supremacist, white-rally*. We are not applying the second filtering criteria because of a very small dataset. On these parameter values, some important keywords (helicopter, crashed, police, killed) that capture the chain of sub-events occurring within an event are missed. But, the proposed method can track the chain of sub-events within each event. If we set the value of $edge\_min\_df$=1 to capture these keywords, there are chances that true cluster splits into various communities because of the presence of cluster splitting problem in the KeyGraph approach.

To understand the cluster splitting problem, in the considered example, we consider the top two tweets of one topic to make a graph using the Keygraph approach.

1. *FBI, opens, civil-rights, probe, car, slammed, crowd, protesters, Virginia*
2. *Two, Virginia, state, Police, troopers, killed, helicopter, crashed, burned, near, Charlottesville*

Now, nodes are made corresponding to each keyword of both tweets and edges are made if the pair of keyword nodes present in the same tweet. Nodes are labeled as Ti_Ki where Ti represents tweet id and Ki represents keyword id (location in the tweet). In the graph shown in Fig. 2(a), two complete subgraphs (9 and 11 nodes) with one common node (Virginia) are present. The community detection algorithm finds the densely connected components. Hence, each complete subgraph can be considered as a community. But, in the proposed approach, these keywords are divided into two categories: topic-identifying and topic-describing keywords. We consider "Virginia" as a topic-identifying keyword because the frequency of this keyword is higher than other keywords. So, the edges existing between the "Virginia" and other keywords make a star graph that is shown in Fig. 2(b). Here, only one densely connected component exists.

3.2 Extraction of Topic Clusters

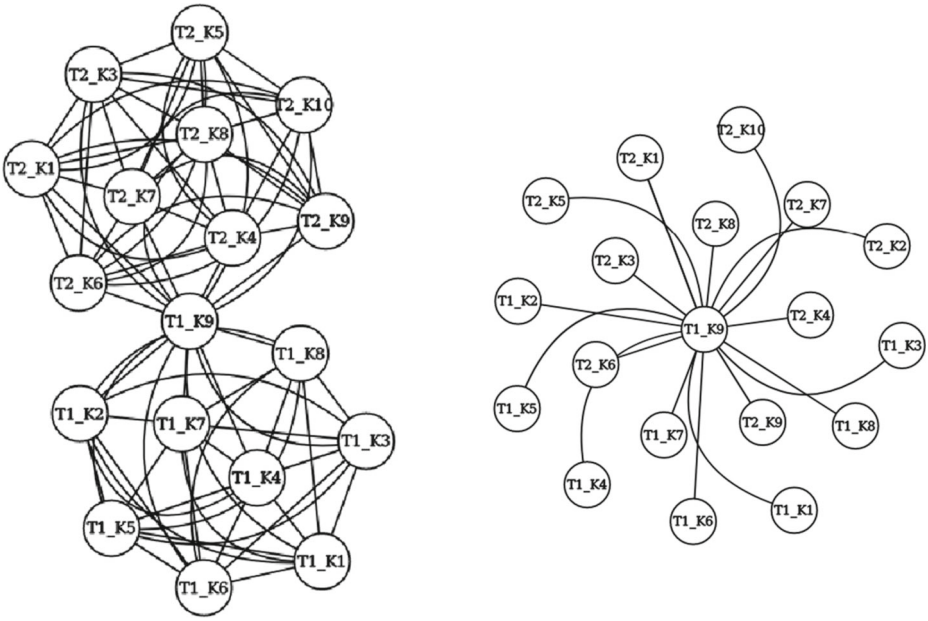
After constructing the graph G(V, E) of keywords where co-occurring keywords depict a topical relationship between keywords, densely connected components are identified by applying an improved LPA algorithm to get topic clusters. LPA is a widely used community detection algorithm due to the linear time complexity algorithm. The main shortcoming of this algorithm is randomness, which degrades the accuracy of the results and affects the stability of the community. To overcome these shortcomings, we fix the node processing order in decreasing order of corresponding keyword frequency . Since highly frequent keywords play a vital role in detecting topics as compared to less frequent keywords, the proposed node updating order makes the LPA more stable.

Another factor which affects the stability of the algorithm is that in presence of more than one highest frequency label, the LPA selects a label at random.

This is rectified in proposed LPA, which uses the (2) to select the label rather than making a random selection.

$$C_i = \underset{l \in lmax}{\arg\max}\, LS(i, l) \tag{2}$$

where, C_i represents the most significant community label of i^{th} node and lmax represents the number of labels assigned with the highest frequency among neighbors. LS(i,l) represents the significance of the label l during the label updating process of i^{th} node. LS(i,l) is



(a) Keygraph (b) NSLPCD

Fig. 2 Graph construction on sample tweets using Keygraph and NSLPCD approaches

computed using the (3).

$$LS(i,l) = \sum_{j \in N^l(i)} NS(j) \tag{3}$$

where, $N^l(i)$ represents neighboring nodes labeled with l. $NS(j)$ represents the node significance value of node j, which is obtained using the (4)

$$NS(j) = \frac{\sum_{k \in Neigh(j)} w_{jk}}{number\ of\ neighbors} \tag{4}$$

where, w_{jk} represents edge weight, which is count of the number of tweets containing both keywords corresponding to connected nodes j and k.

For a better understanding of readers, NSLPCD is demonstrated on an example set of tweets related to two events: Virginia protest, and Gorakhpur tragedy. These tweets are part of the experimental dataset, but we have taken 5 such tweets as follows for demonstration purposes.

1. *President Trump criticizes white nationalists violence Virginia protest leading backlash.*
2. *Gorakhpur Mp Yogi Adityanath suspends principal BRD medical college Gorakhpur because of 60 children died due to the shortage of Oxygen.*
3. *Three killed and dozens injured after a violent white nationalist rally in Virginia.*
4. *Two Virginia state police troopers died in a helicopter crash near Charlottesville rally.*
5. *Children killed in Indian Gorakhpur hospital due to the oxygen cut bill dispute.*

The input sample graph, as shown in Fig. 3, contains topics keywords with corresponding node Ids and frequency value. As an output, a set of two topic communities (clusters) should be obtained. Fig. 4 shows all the steps of the algorithm execution. At iteration 0, node Ids

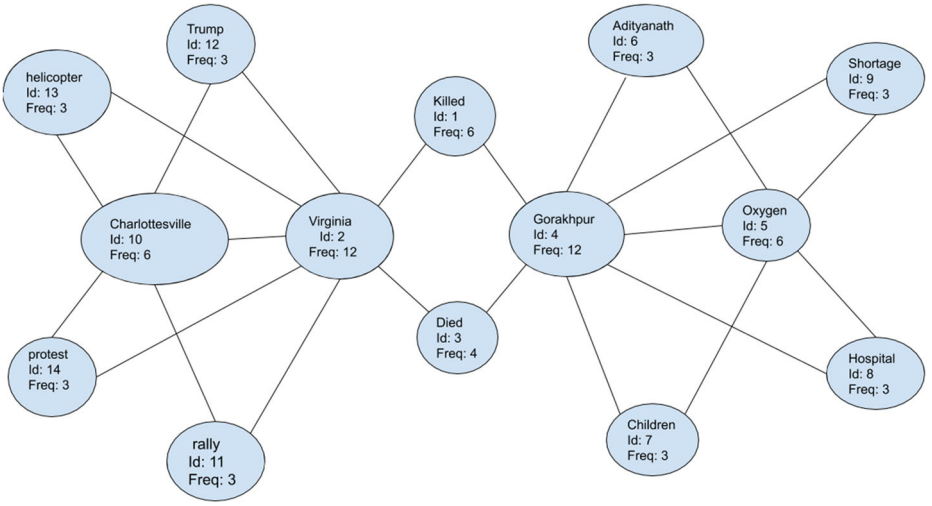
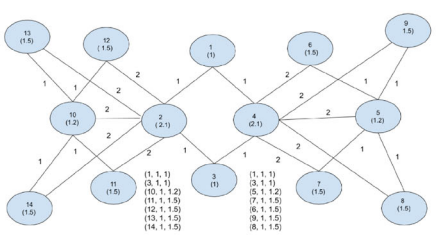


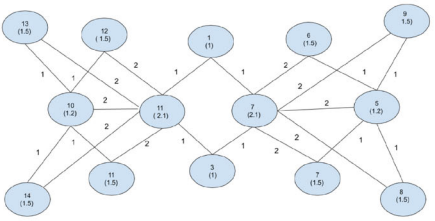
Fig. 3 Sample subgraph generated from Dataset-2

would be assigned as the label Ids of each node according to step 1. In step 2, node updating order is fixed in decreasing order of frequency of keyword as n_2-n_4-n_{10}-n_5-n_1-n_3-n_{11}-n_{12}-n_{13}-n_{14}-n_6-n_7-n_8-n_9 (when freq(i) = freq(j), randomly choose i or j). Step 3 sets the iteration number t=1. For each iteration, the NSLPCD algorithm requires a set of tuple information (l, n, LS(l)) to capture information about neighboring nodes for deciding the label. The parameter l is a label assigned to its neighbors, n is the count of neighbors labeled with l, and LS(l) represents the significance of the label l as given in (3), which is used in case of multiple highest frequency labels. The bracket value inside the node in Fig. 4 represents the node significance value (using (4)) that is required to calculate the significance of lablel l. The edge label represents co-occurrence weight required to calculate node significance value. Now, we start the label propagation process from node n_2. Node n_2 has set of tuple information of seven neighbors as (1, 1, 1), (3, 1, 1), (10, 1, 1.2), (11, 1, 1.5), (12, 1, 1.5), (13, 1, 1.5), (14, 1, 1.5). Here, the count of each label is 1 that generates the situation of a tie. So, step 4(a) selects the label l that has maximum LS(l) value using the (2). Four labels (11, 12, 13, 14) contain the highest label significance value 1.5. Anyone of them can be chosen as a new label for n_2 node. We have arbitrarily chosen 11 as a new label for node n_2.

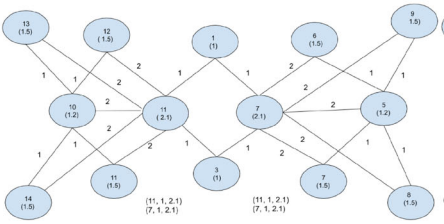
Similarly, node n_4 is updated, and the new label assigned is 7. Updation of the nodes having the same frequency is shown in Fig. 4 simultaneously. In the next phase, node n_1 and node n_3 are updated similarly and shown in Fig. 4(c) and Fig. 4(d). In the next phase, node n_{10} and node n_5 are updated. Node n_{10} has two neighbors containing label 11 and, other neighbors contain different labels. So, label 11 is assigned to node n_{10} as per step 4(b) of the algorithm using (1). Similarly, node n_5 is updated and assigned label 7 as a new label. The update step is shown in Fig. 4(e) and Fig. 4(f). Similarly, all the remaining nodes are updated. After the update of all nodes, we have only two labels 7 or 11 that can be seen in Fig. 4(j). Now, we apply step 5 to check the stopping criteria of the algorithm. In this step, each node is processed to check the label assigned to the node should be greater in number than neighboring labels. Suppose, we select node n_2 with label 11. We can see that six neighbors of n_2 node are labeled with 11 and only one node is labeled with 7. We repeat this process for all nodes and found the same condition. So, there is no need to go for the



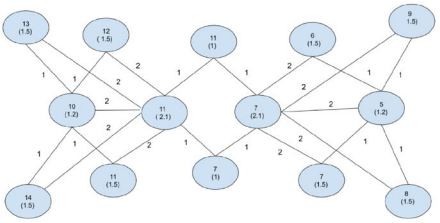
(a) Before n_2, n_4 label updation



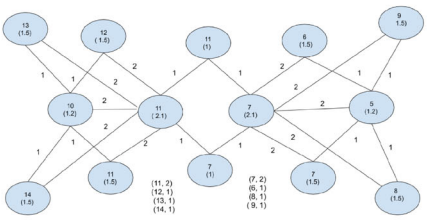
(b) After updation



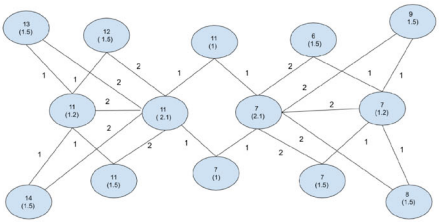
(c) Before n_1, n_3 label updation



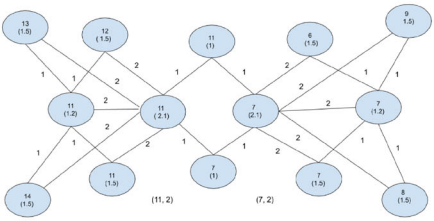
(d) After updation



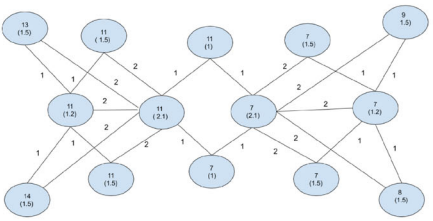
(e) Before n_5, n_{10} label updation



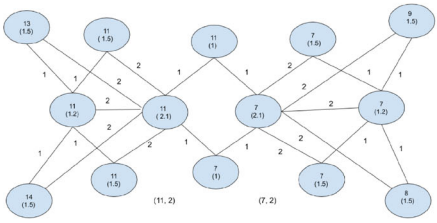
(f) After updation



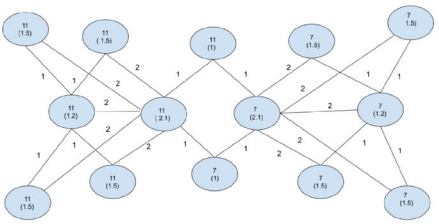
(g) Before n_6, n_{12} label updation



(h) After updation



(i) Before n_8, n_9, n_{13}, n_{14} label updation



(j) After updation

Fig. 4 Node significance based label propagation process

Algorithm 2 : **NSLPCD** - Extract topic clusters.

Input: Graph G = (V, E), and the maximum number of iterations MaxIter.
Output: Set of communities c = $c_1, c_2, \ldots, c_k$ where k is the number of communities.

1. Initialize unique label id to each node in the graph G. For a given node i at iteration 0, $C_i(0) = i$
2. Generate ordered sequence vector N=($n_1, n_2, \ldots, n_i, \ldots, n_n$) based on decreasing order of frequency of keyword set it to N
   Label propagation process:
3. Set iteration number t=1
4. For each node $n_i \in$ N, let $C_i(t)$ = f($C_{i1}(t)$, $C_{i2}(t)$, …, $C_{im}(t)$, $C_{i(m+1)}(t-1)$, …, $C_{ik}(t-1)$), where $n_{i1}, n_{i2}, \ldots, n_{im}$ are the neighboring nodes of node n_i that already have been updated in the current iteration and $n_{i(m+1)}, n_{i(m+2)}, \ldots, n_{ik}$ are the neighboring nodes that are not yet updated in the current iteration. Function f returns the label occurring with the highest frequency among neighbors as per (1):

$$lmax = \arg\max_{l} \sum_{j \in Neigh^{l}(i)} 1$$

   (a) If more than one highest frequency label is present (lmax is not unique label), then assign the label as per (2):

$$C_i = \arg\max_{l \in lmax} LS(i, l)$$

   (b) Else:

$$C_i = lmax$$

5. If C_1, …, C_k are the currently active labels in the network and $Neigh^{C_j}(i)$ is the number of neighbors of node i with nodes of label C_j, then the algorithm is stopped when for every node i with label C_m:

$$Neigh^{C_m}(i) > Neigh^{C_j}(i) \; \forall j$$

   Else, set t=t+1 and go to step 4
6. The nodes having the same label form a community. A community c_j contains the nodes with label C_j where, j ∈ 1, 2,...,k

next iteration t=2. Finally, two topic communities c1 and c2 are obtained, which are labeled with 7 and 11 respectively, according to step 6.

However, if the LPA algorithm is executed on the considered example, then only one community is obtained in most cases. The execution of the LPA algorithm does not require any fixed order. Suppose it first updates node n_1; then, due to equal label significance value of labels 2 and 4, it can get either of them. If label 2 is assigned as a new label; then, the label of n_3 is updated in the same way and gets 2 as a new label. Then, n_4 node is updated by the label of node n_1 that is 2. Proceeding in a similar manner, we get only one community that contains both topics. Cluster merging is a very common problem of the LPA due to its random nature. If a bridge node (which connects two dense components of the graph) gets wrongly labeled due to the random selection of label in case of a tie; then, it can affect the choosing capability of right labels of connecting nodes. This situation can merge the different communities into one., which degrades the cluster quality. To overcome this, NSLPCD has removed the randomness by introducing a new label selection formula

given in (2). This label propagation process is deterministic rather than random. So, output quality is good enough as compared to the traditional LPA algorithm.

3.3 Topic Summarization

The summarization of the whole topic is done through the top five tweets that are most similar (as per cosine similarity) to the extracted topic clusters. For this purpose, the extracted keywords of identified topic clusters and tweets are represented as topic vector f_t and tweet vector v_t, respectively. The cosine similarity between topic vector f_t and tweet vector v_t is computed as per (5):

$$cosine(f_t, v_t) = \frac{f_t.v_t}{||f_t||||v_t||} \tag{5}$$

Algorithm 3 : Topic Summarization.

Input: Topic vector f_t containing keywords corresponding to each obtained topic community and tweet vector v_t containing keywords corresponding to each tweet in the corpus
Output: Five tweets corresponding to each topic

1. For each topic do
2. For each tweet in the corpus do
3. compute the cosine similarity between topic vector f_t and tweet vector v_t using the formula (4)
4. Sort the tweets based on the computed cosine similarity measure
5. Extract the top five tweets

Algorithm 3 describes the whole topic summarization process. For the demonstration of the topic summarization algorithm, it is applied to the same example tweets and extracted topic community keywords of the previous subsection. As an input, we need topic and tweet vectors firstly. The extracted keywords from the example set of tweets are as follows: {President, Trump, criticizes, white, nationalist, violence, Virginia, protest, leading, backlash, Gorakhpur, Mp, Yogi, Adityanath, suspends, principal, BRD, medical, college, children, died, shortage, Oxygen, three, killed, dozens, injured, violent, rally, two, state, police, troopers, helicopter, crash, near, Charlottesville, indian, hospital, cut, bill, dispute}.

The tweet vectors generated for the tweets are as follows:

v_{t1} =[1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
v_{t2} =[0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
v_{t3} =[0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0]
v_{t4} =[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0]
v_{t5} =[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1]

The keywords extracted from the identified topic clusters:

c_1= Virginia, killed, Charlottesville, protest, rally, helicopter, Trump
c_2= Gorakhpur, oxygen, died, hospital, children, shortage, Adityanath

The topic vectors generated for the identified topic clusters are as follows:

f_{t1} = [0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0]
f_{t2} = [0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0]

Now, we have topic vectors f_t and tweet vectors v_t as an input for the topic summarization algorithm. Next, the cosine similarity is calculated between each topic vector v_t and

each tweet vector f_t to identify the most similar tweets corresponding to selected topic by following step 1, 2, and 3. Suppose, firstly we have considered c_1 topic community keywords to summarize the first topic (Virginia protest). As per step 4, the similarity scores for topic community c_1 are computed as follows:
Cos_sim(f_{t1}, v_{t1}) = 0.35
Cos_sim(f_{t1}, v_{t2})= 0
Cos_sim(f_{t1}, v_{t3}) = 0.37
Cos_sim(f_{t1} v_{t4}) = 0.45
Cos_sim(f_{t1}, v_{t5}) = 0.13

Now, step 5 is applied to obtain three most similar tweets (due to less number of example tweets) with the topic community c_1. Summarized tweets of c_1 topic community are as follows:
1. *President Trump criticizes white nationalist violence Virginia protest leading backlash.*
2 *Three killed and dozens injured after a violent white nationalist rally in Virginia.*
3. *Two Virginia state police troopers died in a helicopter crash near Charlottesville rally.*

Similarly, as per step 4, similarity scores for topic community c_2 are computed as follows:
Cos_sim(f_{t2}, v_{t1}) = 0
Cos_sim(f_{t2}, v_{t2})= 0.60
Cos_sim(f_{t2}, v_{t3}) = 0
Cos_sim(f_{t2}, v_{t4}) = 0.11
Cos_sim(f_{t2}, v_{t5}) = 0.53

Now applying step 5, for extracting the top two most similar tweets describing topic community c_2 are as follows:
1. *Gorakhpur Mp Yogi Adityanath suspends principal BRD medical college because of 60 children died due to the shortage of Oxygen.*
2. *Children killed in Indian Gorakhpur hospital due to the oxygen cut bill dispute.*

4 Experimental Studies

The proposed algorithm NSLPCD is compared with four baseline approaches - LDA, Bi-term Topic Model (BTM), KeyGraph, and Weighted-LPA algorithm. The comparison is made based on the quality of the identified topic clusters and the run-time performance of the algorithms. All the experiments are carried out on a machine with Intel Core i7@4.0GHz quad-core processor and 16GB memory running on the Linux machine. All tweets and the graph are stored in text files. Memory usage (Maximum resident set size) of the running code is 194016 kbyte.

4.1 Compared Algorithms

1. Latent Dirichlet Allocation (LDA): A well-known topic detection algorithm (Blei et al [12]) based on Gibbs sampling (with default parameters $\alpha = 0.5$ for topic distribution, $\beta = 0.01$ for word distribution, default value i=1000 iterations and k = 8)
2. Bi-term Topic Model (BTM): Conventional topic models are based on word co-occurrence patterns at the document level to extract topics. For short documents, the data sparsity problem exists. Cheng et al [47] proposed a different way of modeling

based on co-occurrence patterns at the corpus level. We used $\alpha = 50/k$ for topic distribution, $\beta = 0.01$ for word distribution and k=8 for number of topics. Gibbs sampling was executed for 1,000 iterations.

3. KeyGraph: Sayyadi and Raschid [3] proposed the keyword co-occurrence graph method with an edge between-ness community detection algorithm to extract the topic clusters. Three parameters $node\_min\_df$, $edge\_min\_df$, and $edge\_min\_prob$ are used to filter noisy nodes and edges. We demonstrated this approach for creating the graph for comparison with the proposed method. The values of other parameter values were kept same as for the proposed approach.
4. Weighted LPA: Label propagation for weighted graph named Weighted LPA, where edge weight w_{ij} represents the count of tweets containing both keyword nodes. The label updation is done using the (6):

$$C_i = \arg\max_{l} \sum_{j \in N^l(i)} w_{ij} \tag{6}$$

We build the graph through proposed method and then applied this method to get the topic communities. Performance is not good enough due to the random nature of the algorithm.

5. Node significance based label propagation for clusters detection (NSLPCD): The proposed method is a variant of Weighted LPA, which modifies the label updating formula shown in (2) and (3). Moreover, a fixed order of node processing is considered to improve the performance.

4.2 Dataset

For the experiments, tweets are collected using Twitter Streaming API (Tweepy Python library) from 13th to 16th August 2017. A total of 0.2 M tweets have been placed in the dataset named Dataset-1. We have used the method "api.trends-place(23424848)" provided by Twitter API to collect tweets of a perticular place. Argument of the method shows the location code WOEID (Where On Earth IDentifier) that is a unique 32-bit reference identifier, originally defined by GeoPlanet and now assigned by Yahoo!, that identifies any feature on Earth. The id "23424848" is assigned to India. Due to the lack of ground-truth data, we labeled the subset of collected tweets based on the bootstrapping method for quality performance comparison with existing approaches. The tweets mostly represented one of the 8 events, viz. Virginia Protest, Gorakhpur tragedy, Blue Whale Challenge game, Gurmeet Ram Rahim verdict, Independence day celebration, Janmashtami celebration, Saaho movie promotion, and Football Club Barcelona. To perform labeling, we extracted the initial 500 tweets from Dataset-1, and each of them is labeled manually based on domain knowledge. In these 500 tweets, some are labeled as noisy tweets (out of the domain-knowledge scope), and remaining tweets are classified as one of 8 topics. Since manual labeling is a time-consuming task, we manually selected some most relevant keywords corresponding to each topic, and these seed keywords made a search query to extract tweets containing any of these keywords. We repeated this process three to four times, and finally, the labeled dataset, namely Dataset-2, is prepared. The Dataset-2 contained 11.2 K tweets concerning 8 topics. Statistics of Dataset-2 regarding the number of tweets corresponding to each topic is shown in Table 1.

The whole corpus (Dataset-1) is considered for performing the run-time comparison between the proposed modified LPA (NSLPCD) and Edge-Betweenness community detection algorithm. Fig. 5(a) represents the frequency count of each keyword in Dataset-1. Only

Table 1 Statictics of Dataset-2

Topics	Number of Tweets
	11,242
Virginia Protest	2,321
Gorakhpur Tragedy	1,875
Janmashtami celebration	1,267
Independence Day celebration	1,510
Blue Whale Challenge game	1,334
Gurmeet Ram Rahim verdict	1,328
Saaho movie	737
Football Club Barcelona	870

65,000 unique keywords were present in a total of 0.2 M tweets due to the high number of repeated keywords in social media data. Most of the frequency of the keywords lied below 1000, and very few got a high peak. So, we can set both parameter $node\_min\_freq$ and $node\_max\_freq$ values from this frequency distribution. Fig. 5(b) shows the frequency count of keywords in Dataset-2, which contained only 5,000 unique keywords. Most keywords had frequency below 100, and very few had a frequency above 100. These high peaked keywords had been considered as topic-identifying keywords in the proposed approach. The compared algorithms based on various parameters are inefficient to process real-time data due to a lack of knowledge about data. The proposed approach is more efficient in processing the real-time data by using only two parameters that rely on frequency distribution, while the KeyGraph approach used seven parameters.

4.3 Evaluation Metrics

The Dataset-2 had 8 classes of labeled keywords, each corresponding to one of the eight topics. To compare cluster quality, F-Measure (Larsen and Aone [48]), Rand Index [49] and Normalized Mutual Information (NMI) [48] cluster validity measures have been used. The value of Rand Index is computed using (7).

$$RandIndex = \frac{TP + TN}{TP + FP + FN + TN} \tag{7}$$

Where,
TP:- Number of pair of keywords labeled with the same class in the same community
TN:- Number of pair of keywords labeled with different classes in different communities
FP:- Number of pair of keywords labeled with different classes in the same community
FN:- Number of pair of keywords labeled with the same class in different communities

The value of F-Measure is computed using (8)

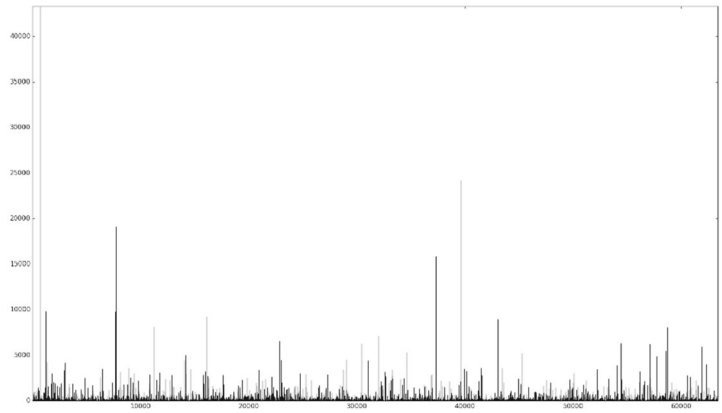
$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{8}$$

Where,

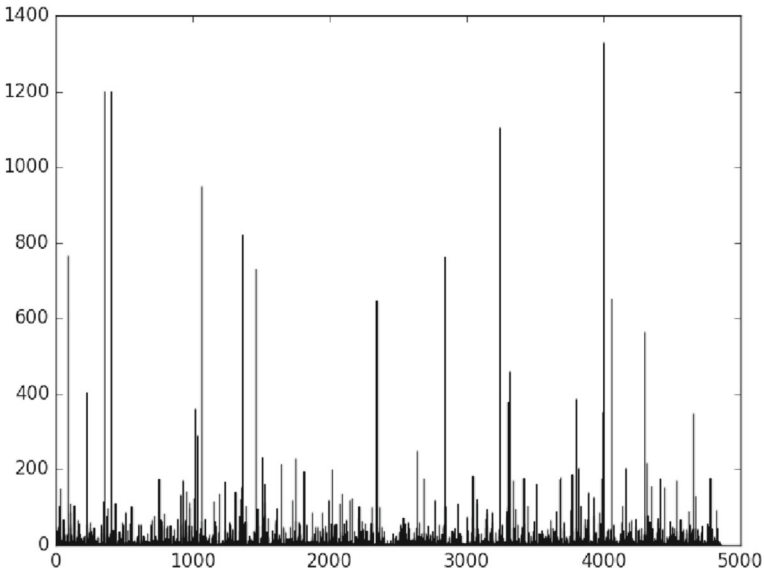
$$Precision = \frac{TP}{TP + FP} \tag{9}$$

and

$$Recall = \frac{TP}{TP + FN} \tag{10}$$



(a) Dataset-1



(b) Dataset-2

Fig. 5 Frequency distribution of keywords on experimental datasets

The value of NMI is based on the contingency values of true classes and output classes. To understand this, let us suppose Y is the collection of all true classes (Y1, Y2, ...), and X is the collection of all output classes (X1, X2, ...). The contingency Table 2 shows the number of keywords corresponding to their true classes and obtained a topic cluster in solution.

Now, NMI can be computed using (11).

$$NMI(X, Y) = \frac{2 * I(X, Y)}{H(X) + H(Y)} \tag{11}$$

Table 2 Contingency table for X and Y

X/Y	Y_1	Y_2	Y_K
X_1	n_{11}	n_{12}	n_1k
X_2	n_{21}	n_{22}	n_{2k}
.			
.			
X_c	n_{c1}	n_{c2}	n_{ck}

Where,
$I(X,Y) = \sum_{i=1}^{|X|} \sum_{j=1}^{|Y|} P(X \cap Y) log(\frac{P(X \cap Y)}{P(X)P(Y)})$,
$H(X) = -\sum_{i=1}^{|X|} P(X) log(P(X))$, and
$H(Y) = -\sum_{i=1}^{|Y|} P(Y) log(P(Y))$.

4.4 Experimental Results

The experimental results have been observed for the comparison of the quality of obtained topic clusters and the run-time performance of the algorithms. The two comparative parameters have been discussed in the following subsections.

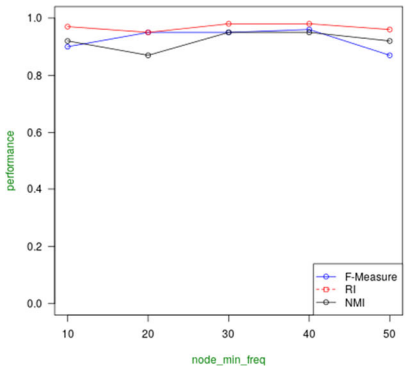
4.4.1 Comparison of Cluster Quality

The primary aim of the experiments was to find the best value for thresholds: $node\_min\_freq$ and $node\_max\_freq$. In order to compute thresholds, the proposed approach is executed with different values of thresholds on Dataset-2, and the value of Precision, Recall, F-Measure, Rand Index, and NMI is observed. Table 3 shows these values for different values of thresholds. Finally, $node\_min\_freq$ having value 40 and $node\_max\_freq$ having value 400 yields the best result to finalize the best values of thresholds. Variation of results on different $node\_min\_freq$ threshold values (10, 20, 30, 40, 50) with 400 value of $node\_max\_freq$ are shown in Fig. 6(a). Results on different $node\_max\_freq$ threshold values (100, 200, 300, 400, 500) with 40 value of $node\_min\_freq$ are shown in Fig. 6(b).

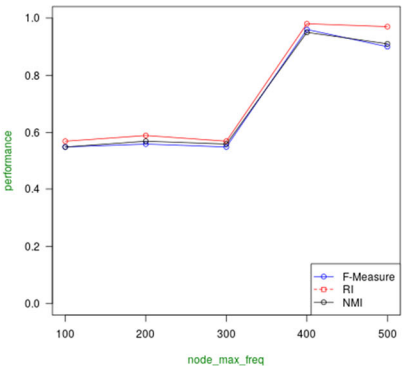
The KeyGraph approach (Sayyadi and Raschid [3]) is analyzed for different parameter values to get the best result in order to compare with the best result of the proposed approach. By changing two parameter values $node\_min\_df$ and $edge\_min\_df$, the best result is obtained on (40,35) which is shown in Table 4. By varying the common parameter value ($node\_min\_freq$ in NSLPCD and $node\_min\_df$ in KeyGraph approach) on best edge filtering threshold in both approaches, validity measures (Precision, Recall, F-Measure, Rand Index and NMI) are compared which are shown in Fig. 7a, b, c, d and e respectively. The proposed algorithm outperforms the KeyGraph approach for all validity measures except Precision on some values. The Precision values of both approaches are nearly equal on average. However, the Recall values of the KeyGraph approach are much lower than the proposed approach because of cluster splitting problem exists in the KeyGraph approach. It is to be noted that other baseline approaches are not graph-based approaches like NSLPCS and KeyGraph approaches, so their comparison is not made for changing values of parameters.

Table 3 Effect of parameters on performance in the proposed approach

node_min_freq	node_max_freq				
	100	200	300	400	500
10	P:0.32 R:0.99 F:0.49 RI:0.36 NMI:0.19	P:0.66 R:0.98 F:0.79 RI:0.91 NMI:0.89	P:0.63 R:0.93 F:0.76 RI:0.89 NMI:0.85	P:0.91 R:0.99 F:0.90 RI:0.97 NMI:0.92	P:0.93 R:0.93 F:0.93 RI:0.98 NMI:0.93
20	P:0.36 R:0.99 F:0.53 RI:0.42 NMI:0.29	P:0.58 R:0.97 F:0.73 RI:0.86 NMI:0.83	P:0.62 R:0.97 F:0.75 RI:0.89 NMI:0.85	P:0.83 R:0.84 F:0.83 RI:0.95 NMI:0.87	P:0.91 R:0.93 F:0.92 RI:0.97 NMI:0.92
30	P:0.36 R:0.99 F:0.53 RI:0.44 NMI:0.34	P:0.39 R:0.98 F:0.56 RI:0.56 NMI:0.52	P:0.39 R:0.99 F:0.56 RI:0.54 NMI:0.52	P:0.95 R:0.94 F:0.95 RI:0.98 NMI:0.95	P:0.88 R:0.91 F:0.90 RI:0.97 NMI:0.91
40	P:0.39 R:0.98 F:0.55 RI:0.57 NMI:0.55	P:0.39 R:0.98 F:0.56 RI:0.59 NMI:0.57	P:0.38 R:0.99 F:0.55 RI:0.57 NMI:0.56	**P:0.96** **R:0.96** **F:0.96** **RI:0.98** **NMI:0.95**	P:0.89 R:0.92 F:0.90 RI:0.97 NMI:0.91
50	P:0.39 R:0.97 F:0.56 RI:0.63 NMI:0.60	P:0.38 R:0.97 F:0.55 RI:0.61 NMI:0.59	P:0.38 R:0.99 F:0.55 RI:0.59 NMI:0.58	P:0.79 R:0.97 F:0.87 RI:0.96 NMI:0.92	P:0.68 R:0.98 F:0.80 RI:0.93 NMI:0.89



(a) Performance of NSLPCD on different $node\_min\_freq$ values and $node\_max\_freq$=400

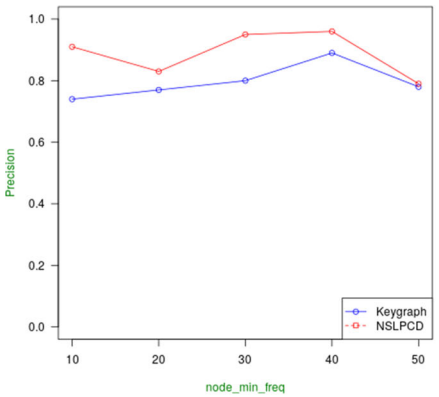


(b) Performance of NSLPCD on different $node\_max\_freq$ values and $node\_min\_freq$=40

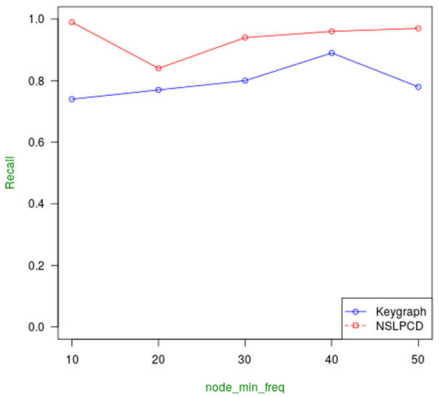
Fig. 6 Performance of NSLPCD on different threshold values

Table 4 Effect of parameters on performance in the KeyGraph approach

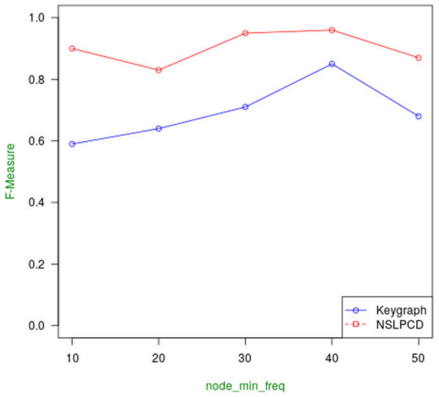
node_min_df	edge_min_df				
	5	15	25	35	45
10	P:0.86 R:0.11 F:0.19 RI:0.85 NMI:0.59	P:0.91 R:0.13 F:0.24 RI:0.86 NMI:0.63	P:0.90 R:0.20 F:0.33 RI:0.88 NMI:0.67	P:0.81 R:0.46 F:0.59 RI:0.90 NMI:0.74	P:0.86 R:0.21 F:0.34 RI:0.87 NMI:0.66
20	P:0.89 R:0.24 F:0.38 RI:0.88 NMI:0.69	P:0.88 R:0.27 F:0.41 RI:0.88 NMI:0.69	P:0.87 R:0.25 F:0.38 RI:0.87 NMI:0.68	P:0.84 R:0.52 F:0.64 RI:0.92 NMI:0.77	P:0.90 R:0.23 F:0.37 RI:0.88 NMI:0.68
30	P:0.85 R:0.53 F:0.65 RI:0.91 NMI:0.76	P:0.84 R:0.52 F:0.64 RI:0.91 NMI:0.76	P:0.84 R:0.62 F:0.71 RI:0.92 NMI:0.79	P:0.86 R:0.61 F:0.71 RI:0.93 NMI:0.80	P:0.86 R:0.63 F:0.73 RI:0.93 NMI:0.81
40	P:0.85 R:0.65 F:0.74 RI:0.93 NMI:0.82	P:0.86 R:0.72 F:0.79 RI:0.94 NMI:0.85	P:0.88 R:0.75 F:0.81 RI:0.95 NMI:0.86	P:0.88 R:0.82 **F:0.85** **RI:0.94** **NMI:0.89**	P:0.89 R:0.66 F:0.76 RI:0.93 NMI:0.84
50	P:0.77 R:0.34 F:0.47 RI:0.89 NMI:0.69	P:0.85 R:0.56 F:0.68 RI:0.92 NMI:0.78	P:0.85 R:0.54 F:0.66 RI:0.92 NMI:0.76	P:0.87 R:0.55 F:0.68 RI:0.92 NMI:0.78	P:0.87 R:0.62 F:0.72 RI:0.93 NMI:0.81



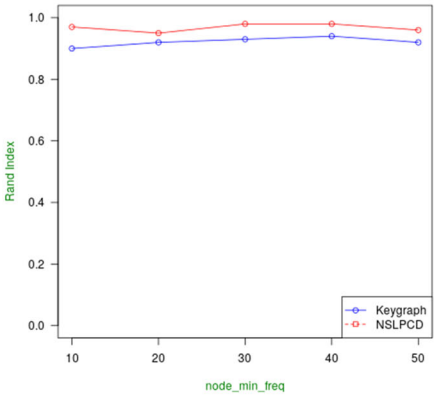
(a) Precision comparison



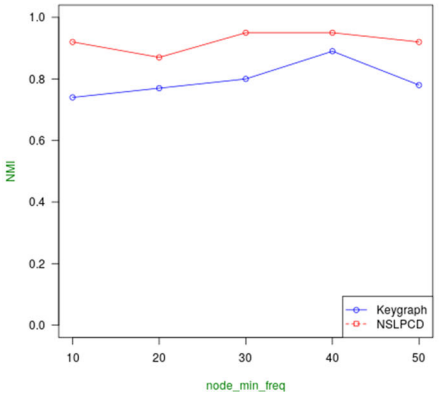
(b) Recall comparison



(c) F-measure comparison



(d) Rand Index comparison



(e) NMI comparison

Fig. 7 Cluster quality performance comparison between NSLPCD and Keygraph approach

The results of eight topics corresponding to the proposed approach (NSLPCD) on best combination threshold value (40,400) and the KeyGraph approach on (40,35) are shown in Tables 5 and 6, respectively. Table 5 contains 8 rows to represent eight topic communities,

Table 5 NSLPCD Topics

Id	Topic Keywords	Topic
1	deal, clause, year, sign, brought, ousmane, second, resmi, standing, story, confirm, transfer, neymar, dembele, fans, hind, forward, plus, jugador, announced, real, move, star, forces, read, valverde, welcome, paid, euro, despite, news, spirit, wait, player, hindustan, freedom, borussia, official, coutinho, place, dortmund, release, joining, agree	Barcelona Football club
2	sentenced, right, trump, crowd, supremacist, dead, unrest, protest, member, resmi, cities, home, rally, nationalist, city, america, police, deadly, trooper, patrolling, march, mcauliffe, state, charlottesville, clashes, helicopter, white, native, troopers, nazis, happened, injured, woman, crash, emergency, bates, driver, protesters, cullen, lives, nationalists, terry, nothing, heather, supremacists, sides, bring, happening, rights, candles, violence, arrested, governor, berke, slams, planned, night, heyer, blood	Virginia Protest
3	google, links, teenagers, challenges, ousmane, trending, link, sarkar, blue, suicide, khan, thank, asks, notice, movie, breaking, please, finally, long, kerala, start, amazing, indian, racist, teenager, dying, films, parents, whatsapp, reading, teen, life, good, committing, bans, government, banned, million, back, action, game, death, facebook, last, whale, killed, benefits, lead, need, challenge, remove, mother, modi	Blue Whale Challenge game
4	beautiful, pakistan, everyone, azaadi, satyagraha, wishing, mataram, tricolor, high, sisters, brave, soilders, love, zinda, armed, pledge, delivered, cherishing, citizens, harsh, indians, nehru, indiansalute, fight, contributing, tricolour, interested, various, hindi, youhappy, speech, bharat, pakistani, conditions, salute, happy, prosperity, mata, brothers, pakistans, afford, india, english, peace, hate, hoisting, assam, jahan, flag, celebrates, shameful, schools, nation, celebrate, welcoming, largest, celebrating, country, proud, harmony, bless, vande, satyagrah, glorious, wish, saheeb, aazadi, virtue	Independence day celebration
5	rahi, manohar, shaant, sarkaar, mafia, dera, sakshi, rapist, haryana, dortmund, punjab, jassi, rohtak, anshul, followers, rape, gurmeet, singh, appeal, public, sachha, sacha, congress, provided, burn, judiciary, pained, harminder, murdered, sentencing, panchkula, dhruv, daughter, exposing, defends, disturbed, maharaj, security, rathee, sauda	Baba Ram Rahim verdict
6	people, gorakhpur, money, lack, sixty, mishra, stern, fail, candle, hospitals, children, principal, media, hospital, supply, yogi, tragedy, supplier, supplies, cause, delhi, rajeev, company, massacre, hours, assures, deaths, political, govt, payment, died, case, purchase, kids, cylinders, oxygen, liquid, released, amid, bill, unpaid, chief, without, jail, shortage, bills	Gorakhpur tragedy
7	family, finalized, sujeeth, related, special, kapoor, wanted, leading, datenone, saaho, film, compensating, announce, anushka, lady, coming, opposite, nitin, actress, lord, pair, america, disappointed, shetty, bigger, congratulations, shraddha, career, always, thing, disappointment, shraddhakapoor	Saaho movie promotion
8	better, iskcon, temple, surat, fortune, birth, krishna, lover, devine, guru, janma, festival, hare, dahi, handi, baris, fuhar, makhan, churane, nandlal, mubarak, tyohar, hardik, shubhkamnaye, sabhi, bhaiyo, yogiadityanath, orders, grand, celebrations, shree	Janmasthami celebration

and each row keywords correspond to only one topic community instead of a mixture of topics. So, the proposed approach obtained the higher values of Precision and recall. Table 6 (KeyGraph approach) contains 11 rows to represent eight topics, i.e., more than one row represents one topic. The Janmashtami celebration topic consists of three communities, and the Independence day celebration consists of two communities. In this case, False Negative would be higher (pair of keywords labeled with the same class in different communities) that decrease the recall value. A common problem with the Keygraph approach is the cluster splitting problem that is explained in Section 3.1 with the help of an example. In both approaches, we get almost the same keywords but the different number of communities. To capture the same keywords, the proposed approach requires fewer edges as compared to KeyGraph approach due to using different edge filtering criteria.

Table 6 KeyGraph Topics

Id	Topic Keywords	Topic
1	trump, cullen, charlottesville, crash, nationalist, nazi, hate, state, heather, heyer, protest, rally, helicopter, troopers, city, police, arrested, office, white, neo, blaming, update, virginia, governor, supremacist, people, car, bates, related	Virginia Protest
2	yogi, died, largest, cut, money, death, ambulances, hospital, kids, company, massacre, unpaid, sixty, assures, critics, released, shortage, gorakhpur, children, supplier, claimed, action, payment, oxygen, cylinder, tragedy, bills, amid, stern, lack, liquid, due, paid, supplies, principal	Gorakhpur tragedy
3	mataram, citizens, countries, mata, jay, vande, hind, bharat, nation, contributing, pakistan, force, satyagraha, wishing, schools, pakistanis, brave, indian, salute, proud, azaadi, love, happy, tricolour, flag, india, hindustan, hoisting, freedom, celebrate, armed, fight, independence, assam, peace, prosperity, spirit, bless, great, harmony	Independence day celebration
4	tyohar, dahi, janmasthami, handi, makhan, churane, mubarak, fuhar, yogi, adityanath, nandlal	Janmasthami celebration
5	shraddha, nitin, sujeeth, lady, bring, anushka, family, prabhas, announce, disappointed, casted, fans, beautiful, shraddha, kapoor, saaho, joining, actress, heroine, star, movie, neil, films, lead, opposite, paired, welcomes, official	Saaho movie promotion
6	deal, sign, dortmund, dembele, barcelona, confirm, transfer, agree, ousmane, borussia	Football club Barcelona
7	sarkaar, congress, exposing, rahim, derasachhasauda, violence, baba , maharaj, case , gurmeet, ram, shaant, created, murdered, punjab, panchkula, jail, security, benefits, daughter, dhruv, supported, brough, mafia, sentencing, rape, public, disturbed, haryana, victim, rapist, judiciary, convict, burn , pained, chief, political, appeal, Manohar, follow , harminder, march, slams, jassi , Sakshi, candle, aakhri, Delhi	Ram Rahim verdict
8	whatsapp, game, government, committing, suicide, links, life, long, bans, blue, whale, challenge, facebook, teen, killed, notice, google, remove	Blue Whale Challenge
9	english, Nehru, Hindi, shameful, delivered, speech	Independence day celebration
10	guru, shree, birth, fortune, lord, devine, hare, festival, lover, janma, krishna, grand	Janmasthami celebration
11	shubhkamnaye, iskcon, surat, hardik, temple, Janmashtami	Janmasthami celebration

Table 7 LDA Topics

Id	Topic Keywords	Topic
1	punjab, panchkula, shraddha, burn, parents, haryana, benefits, oxygen, prabhas, bjp, political, slams, case, money, singh, great, rahi, kapoor, much, please, finally, see, start, public	Ram Rahim verdict, Saaho movie promotion
2	independence, day, india, salute, delivered, happy, celebrate, love, set, wishing, achaa, sare, hamara, vry, jahan, rapist, forces, satyagraha, azaadi, pakistani, pakistans, armed, brave, youhappy, speech	Independence day celebration
3	aazadi, satyagrah, delhi, oxygen, celebrating, interested, fight, wait, janmasthami, virginia, brought, govt, krishna, orders, supplier, done, shree, celebrations, grand, yogiadityanath, end, paid, payment, released, please,	Janmasthami celebration, Gorakhpur tragedy
4	independence, day, happy, singh, india, jai, hind, freedom, nation, everyone, tricolour, country, indiansalute, hai, flag, citizens, tricolor, virtue, cherishing, ali, tiger, zinda, saheeb, despite,	Independence day celebration
5	oxygen, name, hospital, children, burn, gorakhpur, peace, die, supply, lot, contributing, bless, harmony, indians, prosperity, indian, yogi, kerala, lack, due, raped, gets, died, govt, plus	Gorakhpur tragedy
6	blue, whale, challenge, dembele, barcelona, ousmane, game, shraddha, day, government, happy, prabhas, janmasthami, kapoor, links, bans, asks, remove, whatsapp, google, facebook, dortmund, suicide, saaho, eur	Blue Whale Challenge game, Barcelona Football club
7	ram, rahim, gurmeet, baba, chief, dera, sauda, rape, violence, followers, haryana, sacha, wish, death, days, proud, candle, bjp, sakshi, maharaj, mother, indian, killed, dhruv, mafia	Ram Rahim verdict
8	virginia, state, hindustan, white, charlottesville, police, rally, today, oxygen, next, fake, pakistan, violence, spirit, helicopter, sentenced, candles, crash, trump, killed, year, murdered, one, lal, nationalist	Virginia protest, Gorakhpur tragedy

Resultant topic communities of other compared approaches are shown in Tables 7, 8, and 9. Table 7 shows topics for LDA execution. It contains eight rows because it requires a fixed number of topics, and most of the keywords of each row correspond to one topic, but some keywords belong to other topics. The result of this approach is not better because of the data sparsity problem, which exists in the case of Twitter data. Table 8 contains BTM output, which is better than LDA because of handling data sparsity problem. In Table 9 (Weighted-LPA), only six rows are present corresponding to eight topic communities, i.e., each row contains more than one topic keywords. The first row consists of two topics keywords; the second row consists of two topics; and so on. In this case, each community can be a combination of topics. The main problem with the LPA community detection algorithm is the cluster merging problem due to its random nature. This problem is explained in

Table 8 BTM Topics

Id	Topic Keywords	Topic
1	blue, whale, challenge, game, government, bans, links, asks, google, facebook, whatsapp, remove, suicide, day, committing, teenagers, long, parents, days, notice, kerala, mother, son, killed, death	Blue Whale Challenge game
2	oxygen, hospital, children, gorakhpur, govt, die, supply, yogi, due, lack, indian, hai, supplier, cut, died, brd, paid, deaths, kids, payment, bill, company, tragedy, supplies, money	Gorakhpur tragedy
3	virginia, state, white, police, charlottesville, rally, today, violence, helicopter, killed, trump, crash, governor, heather, heyer, people, troopers, nationalist, year, old, car, one, two, supremacists, trooper	Virginia protest
4	independence, day, happy, singh, india, jai, hind, freedom, nation, everyone, tricolour, country, indiansalute, hai, flag, citizens, tricolor, virtue, cherishing, ali, tiger, zinda, saheeb, despite,	Independence day celebration
5	dembele, barcelona, ousmane, dortmund, eur, borussia, deal, signing, del, transfer, done, por, million, neymar, player, agree, clause, forward, bvb, fee, sign, signed, resmi, euro, official	Barcelona Football club
6	ram, rahim, gurmeet, baba, singh, haryana, chief, dera, burn, punjab, panchkula, sauda, rape, violence, bjp, delhi, rapist, followers, sacha, virginia, case, jail, congress, deeply, pained	Ram Rahim verdict
7	shraddha, prabhas, oxygen, kapoor, saaho, next, leading, lady, please, opposite, anushka, lead, think, big, going, movie, film, star, https, thing, announce, time, actress, much, getting	Saaho movie promotion
8	day, independence, shraddha, india, name, oxygen, fans, set, aazadi, satyagrah, kii, pakistani, pakistans, azaadi, satyagraha, sujeeth, prabhas, aap, men, celebrate, brave, armed, forces, youhappy, janmasthami	Janmasthami celebration, Saaho movie promotion

Section 3.2 with the help of an example. So, precision values are less compared to the proposed approach. An overall comparison of the proposed algorithm with existing approaches is shown in Table 10 and Fig. 8. BTM outperforms LDA and Weighted-LPA. KeyGraph outperforms Weighted-LPA, LDA, and BTM. NSLPCD is superior to the KeyGraph approach in Recall, F-Measure, Rand Index, NMI measures, and has nearly similar Precision. To summarize, NSLPCD outperforms all the compared approaches. The resultant topic clusters in the form of a graph are shown in Fig. 9 in which different color nodes represent different topic clusters. Individual graph communities represent topics that are shown in Fig. 10(a) and 10(b) corresponding to each topic cluster. The most representative tweets of each topic cluster are shown in Table 11.

4.4.2 Running time comparison

The time-complexity of the proposed approach is discussed first. It is followed by the comparison of the execution-time of all the approaches. Let n be the number of tweets in the corpus, d be the max count of unique keywords in a tweet, and m be the count of unique words in the whole tweet corpus. The graph building approach (Algorithm 1) requires n*d operations for the first step since all keywords of each tweet need to be processed for storing the frequency of keywords in the dictionary. The size of the dictionary would be m. In the

Table 9 Weighted-LPA Topics

Id	Topic Keywords	Topic
1	protest, signs, hate, police, better, charlottesville, happened, real, woman, march, emergency, virginia, nationalists, despite, county, arrested, announced, night, release, heyer, signing, right, deal, back, related, sign, ousmane, second, resmi, year, home, confirm, trooper, mcauliffe, state, jugador, nazis, million, neymar, political, news, cities, borussia, violence, berke, slams, plus, planned, dortmund, governor, nationalist, standing, city, story, america, crowd, transfer, dembele, patrolling, clashes, forward, helicopter, white, troopers, lives, dead, spirit, rights, official, signed, agree, sentenced, trump, supremacist, player, rally, native, heather, deadly, forces, place, injured, valverde, driver, protesters, crash, cullen, blood, terry, supremacists, sides, euro, happening, candles, coutinho, joining	Virginia protest, Barcelona Football club
2	bhaiyo, wishing, devine, soilders, mubarak, indiansalute, iskcon, lord, hardik, handi, nation, assam, vande, shameful, birth, schools, celebrate, krishna, earn, fortune, celebrates, bless, pledge, nandlal, hare, tricolor, celebrating, delivered, baris, mata, bharat, conditions, temple, salute, dahi, jahan, shubhkamnaye, country, yogiadityanath, shree, janma, glorious, pakistan, celebrations, love, janmasthami, zinda, cherishing, nehru, festival, indians, tricolour, sare, makhan, hindi, happy, tiger, flag, virtue, welcoming, wish, churane, lover, saheeb, beautiful, citizens, hamara, mataram, sabhi, prosperity, harsh, guru, speech, harmony, grand, brothers, hoisting, sisters, proud, english, tyohar	Independence day celebration, Janmasthami celebration
3	rahi, manohar, ousmane, shaant, sarkaar, mafia, dera, sakshi, rapist, haryana, deeply, punjab, jassi, rohtak, anshul, followers, rape, gurmeet, singh, appeal, public, sachha, sacha, congress, burn, judiciary, pained, harminder, murdered, sentencing, panchkula, dhruv, daughter, exposing, aakhri, defends, disturbed, maharaj, security, rathee, sauda	Ram Rahim verdict
4	supply, brave, azaadi, satyagraha, people, gorakhpur, money, lack, india, sixty, mishra, stern, fail, candle, largest, armed, children, principal, pakistans, dying, hospital, yogi, rajeev, fight, released, indian, gets, supplier, supplies, pakistani, cause, killed, media, delhi, government, afford, youhappy, company, massacre, paid, hours, assures, deaths, tragedy, govt, payment, died, case, purchase, kids, cylinders, oxygen, liquid, getting, amid, bill, without, unpaid, chief, satyagrah, jail, shortage, bills	Gorakhpur tragedy, Independence day celebration
5	google, links, teenagers, challenges, trending, link, sarkar, suicide, khan, thank, asks, notice, dortmund, please, kerala, amazing, racist, teenager, films, parents, whatsapp, reading, teen, life, movie, committing, bans, banned, game, death, facebook, know, whale, aaye, bcoz, great, benefits, challenge	Blue Whale Challenge game
6	virginia, fans, shraddhakapoor, opposite, shraddha, kapoor, disappointment, anushka, shetty, lead, actress, film, neil, nitin, sujeeth, coming, leading, disappointed, saaho, always, star, announce, special, lady, finalized, career, recd, compensating, till, datenone, pair, congratulations, much, bigger, inch	Saaho movie promotion

second step, the frequency threshold condition is checked for each keyword, which requires at most m number of operations. The third step requires the labeling of the node, and it also requires m number of operations. Finally, the edge building step among nodes needs a

Table 10 Cluster quality comparison of the proposed approach with baseline approaches

Approaches	Precision	Recall	F-Measure	Rand Index	NMI
LDA	0.60	0.33	0.42	0.80	0.56
BTM	0.86	0.79	0.83	0.95	0.89
Keygraph	0.88	0.82	0.85	0.94	0.89
Weighted LPA	0.66	0.92	0.77	0.92	0.84
NSLPCD	0.96	0.96	0.96	0.98	0.95

maximum of m^2 number of operations. So, the time complexity of the graph construction algorithm (Algorithm 1) is $O(n * d + m^2)$.

The extraction of topic clusters using an NSLPCD (Algorithm 2), has a similar time complexity of label propagation O(m^2) [5], as NSLPCD is an improved LPA. The modification improves the quality of topic clusters obtained through the normal processing but in a fixed order and does not affect the time-complexity. So, the time complexity of Algorithm 2 is $O(m^2)$. The topic summarization algorithm (Algorithm 3) requires (n * d * k) number of operations where k is the number of topic communities obtained. Hence, the running time can be summarized as the following:
Graph construction: O(n * d + m^2)
Topic clusters detection: O(m^2)
Topic summarization: O(n * d * k)
Total Running Time: O(n * d * k + m^2)
It must ne noted that the KeyGraph approach uses edge between-ness community detection algorithm, which runs in O(m^3) [4] time complexity.

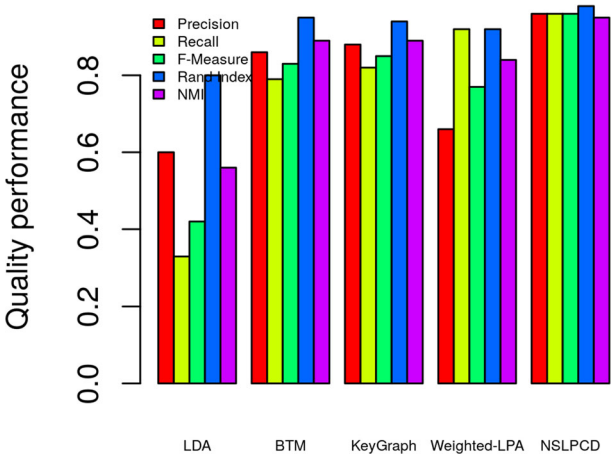
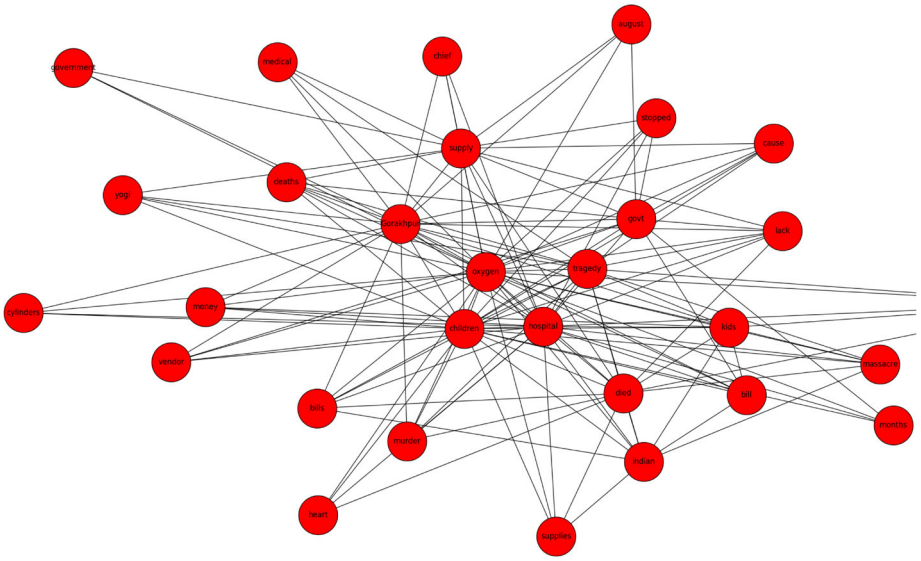
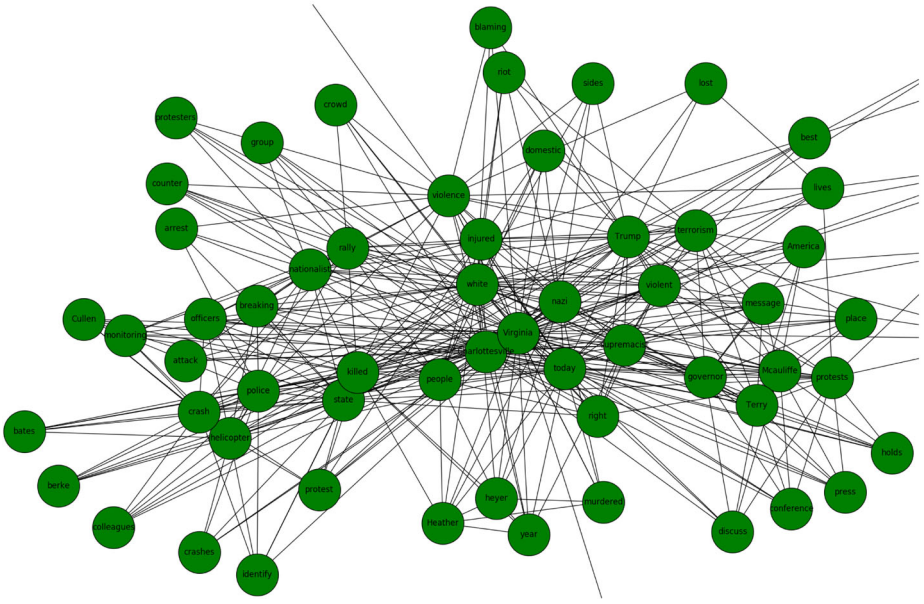


Fig. 8 Quality performance comparison



(a) Gorakhpur tragedy



(b) Virginia protest

Fig. 9 Graph representing detected communities using NSLPCD

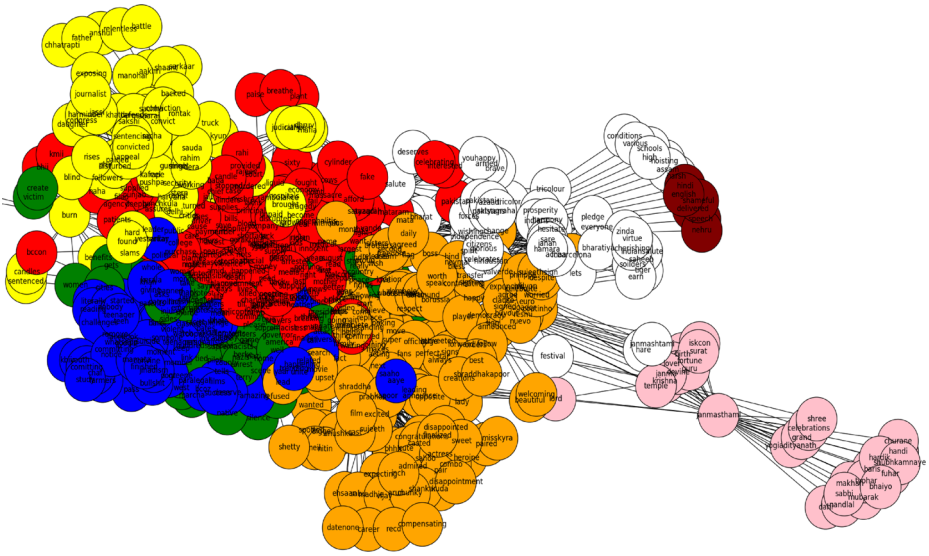


Fig. 10 Visualization of topic communities

Next, we compare the execution time of the proposed approach and other baseline approaches (Keygraph, LDA, BTM, and Weighted LPA) on Dataset-2 (a subset of Dataset-1). Table 12 shows the running time of all compared approaches, and it is also presented in the form of a diagram for proper visualization in Fig. 11. The proposed approach takes 4.8 seconds that is minimum in comparison to other approaches to run 11.2 k tweets. Weighted-LPA approach takes nearly equal time 4.7 seconds because it is also based on a faster LPA community detection algorithm, but the quality of clusters obtained is poor. The KeyGraph approach (8.4 s) is better than LDA (47.2 s) and BTM (41.5) in run-time performance. The analysis of the comparison between the Keygraph approach and LDA is presented very well in Sayyadi and Raschid [3] research work.

Building graphs in KeyGraph and NSLPCD approaches require the same number of operations. We explicitly compared the running time performance of the NSLPCD and KeyGraph approach by comparing community detection time only. In the experiment, the same graph was used to run edge between-ness centrality and improved label propagation (NSLPCD) for the comparison of running time on Dataset-1 (n=0.2 M). The size of the dataset is increased by reducing the parameter values. Table 13 shows the running time of both approaches on different parameter values. The number of nodes and edges is represented on the X-axis of Fig. 12(a) and Fig. 12(b), respectively, and Y-axis shows the running time of both approaches. We stopped edge between-ness algorithm after 1000 nodes and 1 Million edges due to slow response while NSLPCD performed well on up to 7000 nodes and 7 Million edges. As the number of nodes exceeds 1000 and edges exceeds 1 Million, the execution time of the KeyGraph approach would be significantly greater than NSLPCD. The execution time of NSLPCD increased in a linear fashion, whereas the run-time of the KeyGraph approach increased in a nonlinear fashion.

Table 11 Summarization of detected topic communities using NSLPCD

Topic	Representative Tweets
Virginia protest	Woman killed at white supremacist rally in Virginia, USA https://t.co/vm5xAe2pKH https://t.co/S67TfoLmoM RT @CNN: Two Virginia State Police troopers killed when their police helicopter crashed and burned near Charlottesville rally RT @nytimes: Virginia's governor declared a state of emergency as white nationalists and neo-Nazis clashed with counterprotesters RT @PoliticalShort: RIP Lieutenant H. Jay Cullen and Trooper-Pilot Berke M.M. Bates. Virginia state troopers killed in today's RT @DrewLiquerman: R.I.P: State Police officers H. Jay Cullen and Trooper-Pilot Berke M.M. Bates Thank you for your service to protests
Gorakhpur tragedy	@Joydas: 67 Lakhs: Amount due to Oxygen Supplier of Hospital in Gorakhpur that caused death of over 40 Kids 40 Crore: #GorakhpurChildrenTragedy caused by a Govt which has no money for Oxygen but crores for PM's advertisements, https://t.co/kMbdcuxWoo Its supply of liquid oxygen was disrupted over an unpaid bill, officials said. https://t.co/VsLTP3WNLl RT @Riteishd: 30 children die in a hospital due to lack of oxygen. Are you kidding me?? It's murder, it's massacre!!! My heart just died Despicable If Oxygen Shortage Caused Deaths In Gorakhpur: Yogi Adityanath
Blue Whale Challenge	RT @UnSubtleDesi: I don't get it. Kids are committing suicide because of this 50 day long blue whale challenge. How do parents not notice Blue Whale Challenge: Centre asks Google, Facebook and other #SocialMedia platforms to remove https://t.co/XpJVMY363w Blue Whale Challenge: Google, Facebook, WhatsApp asked to remove links https://t.co/EhKPMg4Ee3 'Blue whale challenge killed my son,' says Kerala mother days after teen's death https://t.co/MH0KcKIBVm #Kerala #IndiaAt71 Farmers are committing suicide because of life challenges. Youth is comitting suicide because of blue whale challenge.
Independence Day	RT @sanjaybafna: If women can fight for freedom of India they can surely fight for their independence Aazadi Satyagrah RT @khaleejtimes: #PakistanIndependenceDay: Patriotism at heart, young #Pakistani expats says he is proud to be a #Pakistani in #IndiaAt71 happy Independence Day to all Indian people RT @dna: This salute to Tricolour by Assam school is simply the best! #IndiaAt71 #HappyIndependenceDay Happy #Janmashtami. May Lord Krishna bless all with peace, prosperity & well-being

Table 11 (continued)

Topic	Representative Tweets
Janmashtami celebration	RT @StarGoldIndia: May the auspicious day of the birth of Shree Krishna shower you with blessings. Happy #Janmashtami from #Janmashtami ki hardik badhai sabhi Bhai bahano ko. Jai Shri Krishna #Yogi Orders #Grand #Janmashtami #Celebrations In #UP Read More: https://t.co/NOBJH8vJ5e May this festival bring happiness in your life and may Lord Krishna bless all of you. Happy #Janmashtami everyone Jai Shree Krishna #Janmashtami Shri Krishna Janmashtami ki sabko Hardik Shubhkamnaye Jai Shree Krishna Jai Shree Radhe #ShahRukhKhan
Football club Barcelona	RT @SkySportsNews: BREAKING: @FCBarcelona agree deal to sign Borussia Dortmund forward Ousmane Dembele. https://t.co/GdoEhkkBnj RT @TransferRelated: DONE DEAL: Barcelona have announced the signing of Ousmane Dembele from Borussia Dortmund for Ousmane Dembele: Barcelona agree deal for Borussia Dortmund forward https://t.co/qdFJICzstb https://t.co/WFeggVlUpj Barcelona agree deal to sign Ousmane Dembele from Borussia Dortmund; becomes second-most expensive player in the world
Ram Rahim verdict	30 dead as #DeraSachaSauda followers run riot in Haryana, Punjab https://t.co/sL1oqLR6QQ https://t.co/ZYwSqQXHPT RT @htTweets: Dera chief Gurmeet #RamRahimSingh convicted of raping two women followers, sentencing on Aug #RamRahimSingh #RamRahimVerdict #Panchkula #Haryana #RamRahim Convicted #DeraSachaSauda How many support BJP MP Sakshi Maharaj conviction? RT @SirJadejaaaa: 33 Dead, 300 Injured.#Haryana #KhattarMustResign. #RamRahimSingh RT @SSarkar4: The ridiculousness of some blind followers of a rapist. #Ram RahimSingh #RamRahimVerdict #Panchkula #DeraSachaSauda
Saaho movie	RT @Rahulrautwrites: . @ShraddhaKapoor has been finalized as the leading lady opposite #Prabhas in his next #Saaho. Directed by RT @UV_Creations: Happy to announce the leading lady of Saaho, The beautiful @ShraddhaKapoor. Here's welcoming her to #Saaho RT @KajalfanRavi: Welcome to #TeluguFilmIndustry! All d bst for #Saaho #SaahoWelcomesShraddha RT @ActorPRABHA: @UVCreations @ShraddhaKapoor Happy to announce the leading lady of Saaho, The beautiful @ShraddhaKapoor Happy to announce the leading lady of Saaho, The beautiful @ShraddhaKapoor Here's welcoming her to #Saaho family.

Table 12 Execution time with Dataset-2

Approaches	Execution time
LDA	47.2 s
BTM	41.5 s
Keygraph	8.4 s
Weighted-LPA	4.7 s
NSLPCD	4.8 s

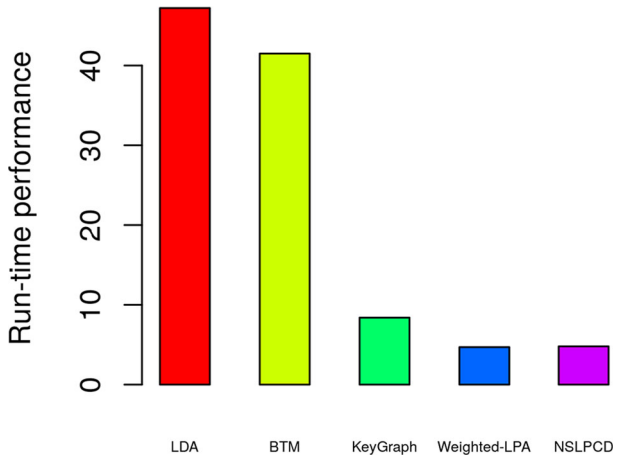


Fig. 11 Execution time comparison

Table 13 Running time comparison between NSLPCD and Edge between-ness community detection algorithm

(node_min_freq, node_max_freq)	Nodes	Edges	NSLPCD	Edge between-ness
(500, 5000)	507	2,35,032	0.11 s	6.20 s
(400, 4000)	655	4,35,967	0.19 s	6.38 s
(300, 3000)	866	6,84,536	0.27 s	25.97 s
(200, 2000)	1,339	14,19,749	0.69 s	59.82 s
(100, 1000)	2,555	28,81,188	1.60 s	138.60 s
(90, 900)	2,823	31,16,507	2.26 s	327.70 s
(80,800)	3,097	33,58,591	2.30 s	939.98 s
(70, 700)	3,437	37,02,369	3.24 s	3853.15 s
(60, 600)	3,858	41,88,003	3.35	-
(50, 500)	4,361	47,30,882	4.32 s	-
(40, 400)	5,090	52,99,187	5.8 s	-
(30, 300)	6,160	59,76,898	8,89 s	-
(20, 200)	7,915	70,31,488	16,23 s	-

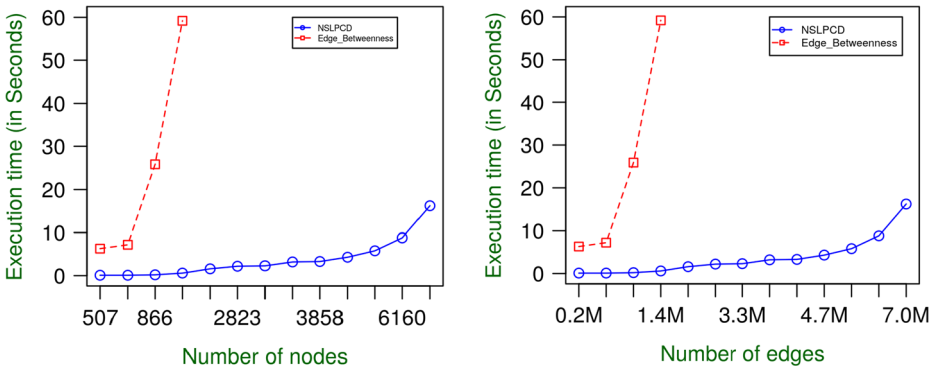


Fig. 12 Run-time performance comparison between NSLPCD and Edge between-ness community detection algorithm

5 Conclusion and Future Work

Twitter has experienced an explosive increase in both users and the volume of information in recent times which has attracted great interest from both industry and academia. Tweets being short and containing noisy data in large volume poses challenges on topic detection task. This article presented a new graph analytical method to detect the most popular topics from Twitter data in a faster manner. Conceptually, the proposed method is similar to other keyword co-occurrence topic modeling approaches, but the basic difference is that it incorporates keyword co-occurrence explicitly. Nowadays, the continuously increasing rate of incoming tweets demands a faster algorithm that can detect events immediately after happening. The proposed algorithm - NSLPCD is capable of fulfilling this demand, without compromising accuracy. To detect the topic, NSLPCD modifies the processing order of nodes. The label updating formula of NSLPCD improves the quality of performance. Experimental results show that the performance of NSLPCD is better than the KeyGraph approach due to high recall value. We also compared the cluster quality and the run-time performance of NSLPCD with LDA, BTM, and Weighted-LPA, and obtain the best results. A comparison of execution time is made between edge between-ness and node significance based label propagation community detection algorithm on different sizes of the datasets. The running time of NSLPCD seems to increase very slowly, and edge between-ness seems to increase very rapidly. The application areas include detection of disease outbreak like COVID-19, emergency management during disasters, or stock market fluctuation through tweets clustering. The future directions of work has enormous possibilites. Twitter analytics can detect emerging real-time events from Twitter feeds timely by adding time and location parameters in the topic. The future may also see the Twitter as newsrooms favorite channel. Twitter analytics can also be used for predicting stock market behavior based on events and related tweet and their sentiments.

References

1. Sakaki, T., Okazaki, M., Matsuo, Y.: Tweet analysis for real-time event detection and earthquake reporting system development. IEEE Trans. Knowl. Data Eng. **25**(4), 919–931 (2013)

2. Li, R., Lei, K.H., Khadiwala, R., Chang, K.C.-C.: Tedas: A twitter-based event detection and analysis system. In: Data engineering (icde), 2012 ieee 28th international conference on, IEEE, pp 1273–1276 (2012)
3. Sayyadi, H., Raschid, L.: A graph analytical approach for topic detection. ACM Transactions on Internet Technology (TOIT) **13**(2), 4 (2013)
4. Newman, M.E.J.: Analysis of weighted networks. Physical review E **70**(5), 056131 (2004)
5. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Physical review E **76**(3), 036106 (2007)
6. Becker, H., Naaman, M., Gravano, L.: Beyond trending topics: Real-world event identification on twitter. ICWSM **11**(2011), 438–441 (2011)
7. Sankaranarayanan, J., Samet, H., Teitler, B.E., Lieberman, M.D., Sperling, J.: Twitterstand: news in tweets. In: Proceedings of the 17th acm sigspatial international conference on advances in geographic information systems, ACM, pp 42–51 (2009)
8. Kim, H.-G., Lee, S., Kyeong, S.: Discovering hot topics using twitter streaming data social topic detection and geographic clustering. In: Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on, IEEE, pp 1215–1220 (2013)
9. Mathioudakis, M., Koudas, N.: Twittermonitor: trend detection over the twitter stream. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, ACM, pp 1155–1158 (2010)
10. O'Connor, B., Krieger, M., Ahn, D.: Tweetmotif: Exploratory search and topic summarization for twitter. In: ICWSM, pp 384–385 (2010)
11. Papadopoulos, S., Kompatsiaris, Y., Vakali, A.: A graph-based clustering scheme for identifying related tags in folksonomies. In: International Conference on Data Warehousing and Knowledge Discovery, Springer, pp 65–76 (2010)
12. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research **3**, 993–1022 (2003)
13. Diao, Q., Jiang, J., Zhu, F., Lim, E.-P.: Finding bursty topics from microblogs. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, Association for Computational Linguistics, pp 536–544 (2012)
14. Petrović, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to twitter. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, pp 181–189 (2010)
15. Osborne, M., Petrovic, S., McCreadie, R., Macdonald, C., Ounis, I.: Bieber no more: First story detection using twitter and wikipedia. In: SIGIR 2012 Workshop on Time-aware Information Access (2012)
16. Petrović, S., Osborne, M., Lavrenko, V.: Using paraphrases for improving first story detection in news and twitter. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pp 338–346 (2012)
17. Feng, X., Zhang, S., Liang, W., Liu, J.: Efficient location-based event detection in social text streams. In: International Conference on Intelligent Science and Big Data Engineering, Springer, pp 213–222 (2015)
18. Hasan, M., Orgun, M.A., Schwitter, R.: Twitternews: real time event detection from the twitter data stream. PeerJ PrePrints **4**, e2297v1 (2016)
19. Alsaedi, N., Burnap, P., Rana, O.: Can we predict a riot? disruptive event detection using twitter. ACM Transactions on Internet Technology (TOIT) **17**(2), 18 (2017)
20. Li, C., Sun, A., Datta, A.: Twevent: segment-based event detection from tweets. In: Proceedings of the 21st ACM international conference on Information and knowledge management, ACM, pp 155–164 (2012)
21. Ifrim, G., Shi, B., Brigadir, I.: Event detection in twitter using aggressive filtering and hierarchical tweet clustering. In: Second Workshop on Social News on the Web (SNOW), Seoul, Korea, 8 April 2014, ACM (2014)
22. Zhao, S., Gao, Y., Ding, G., Chua, T.-S.: Real-time multimedia social event detection in microblog. IEEE Transactions on Cybernetics, 3218–3231 (2017)
23. Zhang, C., Lei, D., Yuan, Q., Zhuang, H., Kaplan, L., Wang, S., Han, J.: Geoburst+: Effective and real-time local event detection in geo-tagged tweet streams. ACM Transactions on Intelligent Systems and Technology (TIST) **9**(3), 34 (2018)
24. Hossny, A.H., Mitchell, L.: Event detection in twitter: A keyword volume approach. In: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, pp 1200–1208 (2018)
25. Choi, H.-J., Park, C.H.: Emerging topic detection in twitter stream based on high utility pattern mining. Expert Syst. Appl. **115**, 27–36 (2019)

26. Mehrotra, R., Sanner, S., Buntine, W., Xie, L.: Improving lda topic models for microblogs via tweet pooling and automatic labeling. In: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, ACM, pp 889–892 (2013)
27. Zhou, X., Chen, L.: Event detection over twitter social media streams. The VLDB journal **23**(3), 381–400 (2014)
28. Jin, D., Liu, D.-Y., Yang, B., Liu, J., He, D.-X., Tian, Y.: Fast complex network clustering algorithm using local detection. Dianzi Xuebao(Acta Electronica Sinica) **39**(11), 2540–2546 (2011)
29. Cruz, J.D., Bothorel, C., Poulet, F.: Community detection and visualization in social networks: Integrating structural and semantic information. ACM Transactions on Intelligent Systems and Technology (TIST) **5**(1), 11 (2013)
30. Nguyen, T., Phung, D., Adams, B., Tran, T., Venkatesh, S.: Hyper-community detection in the blogosphere. In: Proceedings of second ACM SIGMM workshop on Social media, ACM, pp 21–26 (2010)
31. Pathak, N., DeLong, C., Banerjee, A., Erickson, K.: Social topic models for community extraction. In: The 2nd SNA-KDD workshop, 8, p 2008 (2008)
32. Hashimoto, T., OKAMOTO, T.etsuji.K.UBOYAMAbHiroshi., SHIN, K.: Topic extraction from millions of tweets based on community detection in bipartite networks. Information Modelling and Knowledge Bases XXIX **301**, 395 (2018)
33. Girvan, M., Newman, M.arkEJ.: Community structure in social and biological networks. Proceedings of the national academy of sciences **99**(12), 7821–7826 (2002)
34. Newman, M.arkEJ., Girvan, M.: Finding and evaluating community structure in networks. Physical review E **69**(2), 026113 (2004)
35. Newman, M.arkEJ.: Fast algorithm for detecting community structure in networks. Physical review E **69**(6), 066133 (2004)
36. Clauset, A., Newman, M.arkEJ., Moore, C.: Finding community structure in very large networks. Physical review E **70**(6), 066111 (2004)
37. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment **2008**(10), P10008 (2008)
38. Waltman, L., VanEck, N.J.: A smart local moving algorithm for large-scale modularity-based community detection. The European Physical Journal B **86**(11), 471 (2013)
39. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435**(7043), 814 (2005)
40. Kumpula, J.M., Kivelä, M., Kaski, K., Saramäki, J.: Sequential algorithm for fast clique percolation. Phys. Rev. E. **78**(2), 026109 (2008)
41. Lee, C., Reid, F., McDaid, A., Hurley, N.: Detecting highly overlapping community structure by greedy clique expansion. arXiv preprint arXiv:1002.1827 (2010)
42. Gregory, S.: Finding overlapping communities using disjoint community detection algorithms. In: Complex networks, Springer, pp 47–61 (2009)
43. Xie, J., Szymanski, B.K.: Towards linear time overlapping community detection in social networks. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, pp 25–36 (2012)
44. Xing, Y., Meng, F., Zhou, Y., Zhu, M., Shi, M., Sun, G.: A node influence based label propagation algorithm for community detection in networks. Sci. World J. **2014**, 1–13 (2014)
45. Liu, W., Jiang, X., Pellegrini, M., Wang, X.: Discovering communities in complex networks by edge label propagation. Scientific reports **6**, 22470 (2016)
46. Gui, Q., Deng, R., Xue, P., Cheng, X.: A community discovery algorithm based on boundary nodes and label propagation. Pattern Recogn. Lett. **109**, 103–109 (2018)
47. Cheng, X., Yan, X., Lan, Y., Guo, J.: Btm: Topic modeling over short texts. IEEE Trans. Knowl. Data Eng. **26**(12), 2928–2941 (2014)
48. Larsen, B., Aone, C.: Fast and effective text mining using linear-time document clustering. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 16–22 (1999)
49. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association **66**(336), 846–850 (1971)