

A Layered Approach for Summarization and Context Learning from Microblogging Data

Muhammad Abulaish
Department of Computer Science
South Asian University, Delhi, India
abulaish@ieee.org

Md. Imran Hossain Showrov
Department of Computer Science
South Asian University, Delhi, India
showrov.cse@gmail.com

Mohd Fazil
Department of Computer Science
Jamia Millia Islamia, Delhi, India
mohdfazil.jmi@gmail.com

ABSTRACT

Twitter, a microblogging online social network, is one of the most popular information sharing and communication platform. The large user-base and users mutual interactions generate massive amount of data that are rich source of information for predictive modeling, sentiment analysis, opinion mining, and other text information processing tasks. Understanding context embedded within text corpus and generating a contextual summary of the corpus is one of the promising research directions in the field of data analytics. In this paper, we present a layered graph-based approach using both content and structural data to analyze and summarize tweets at different levels of granularity. The proposed approach models tweets as a multi-dimensional graph and applies random walk to identify most informative tweets, which are further processed using a graph-theoretic approach, LexRank, to identify most informative sentences for summary generation. Finally, the summary texts are analyzed using TextRank algorithm to identify prominent keywords conceptualizing the context of the underlying corpus. The proposed summary generation and context learning approach is evaluated over four different real-world Twitter datasets using standard information retrieval metrics.

CCS CONCEPTS

• **Information systems** → **Data analytics**; *Similarity measures*; Content ranking; • **Human-centered computing** → **Social network analysis**;

KEYWORDS

Data analytics, Social network analysis, Text mining, Text summarization, Keywords extraction, Context learning.

ACM Reference Format:

Muhammad Abulaish, Md. Imran Hossain Showrov, and Mohd Fazil. 2018. A Layered Approach for Summarization and Context Learning from Microblogging Data. In *Proceedings of the 20th International Conference on Information Integration and Web-based Applications & Services (iiWAS'18)*, November 19–21, 2018, Yogyakarta, Indonesia. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3282373.3282421>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

iiWAS'18, November 19–21, 2018, Yogyakarta, Indonesia

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-6479-9/18/11...\$15.00

<https://doi.org/10.1145/3282373.3282421>

1 INTRODUCTION

In last two decades, online social networks (OSNs) have gained tremendous popularity around the globe. The easy-to-use functionalities and smooth registration mechanism have further helped the development of these platforms. As a result, millions of users have registered on one or more OSNs for real-time communication, entertainment, mutual interactions, and so on. Twitter, a microblogging platform, is one of the OSNs that facilitates its users to share views and thoughts about any incident, event updates, etc. in the form of tweets limited to 280 characters. Twitter has approximately 330 million¹ monthly active users. In other words, among all registered users on Twitter, approximately 330 millions use the platform at least once in a month and generate approximately 500 million tweets every day. The tweets include different types of data such as texts, images, audio, and video, possessing very rich information due to their precise nature.

The massive amount of data generated by the OSNs has opened the door for various research directions, such as text summarization and conceptualization, topic modeling, predictive analytics, link prediction, community analysis, sentiment analysis, and information diffusion [1–4]. Out of these, text summarization and conceptualization is one of the important text analytics task, which is intended to generate summary from a given text corpus and conceptualize it using the prominent keyphrases. Textual contents in social networks are generally informal, short, multilingual, and noisy. Therefore, automatic summary generation and context learning is a challenging task and requires rigorous data pre-processing and development of efficient text information processing techniques.

In this paper, we present a layered approach for summary generation and context learning from Twitter data. The first layer of the proposed approach aims to identify most informative tweets and their sentences for summary generation. The process of most relevant tweets identification models tweets corpus as a multi-dimensional graph in which nodes represent tweets and edges represent their linkages based on common *hashtags* and *mentions*. Thereafter, a *random-walk* algorithm is applied on the constructed multi-dimensional graph to determine the importance score for every node (tweet) of the graph and top-*k* most relevant tweets are considered for further processing. The process of relevant sentence extraction also models the selected tweets using a graph structure in which nodes represent sentences and edges represent their *Cosine* similarities. On the constructed sentence graph, a centrality-based algorithm, LexRank is applied to assign numeric score to each node, representing the relative relevance of the nodes (sentences). Finally,

¹<https://blog.hootsuite.com/twitter-statistics/>

top most relevant sentences are considered to form a summary of the underlying corpus.

The second layer of the proposed approach aims to identify important keywords from the summary sentences. In this process, summary sentences are modeled as word co-occurrence graph in which nodes represent words and edges represent their co-occurrences at sentence level. On the constructed graph, a graph-based ranking algorithm, TextRank is applied to identify important keywords to conceptualize the context of the underlying tweets corpus.

The rest of the paper is organized as follows. Section 2 presents a brief review of the state-of-the-art literatures on text summarization and context learning. Section 3 presents a detailed description of the proposed approach. It also presents a brief description of the datasets used in this study for experimental evaluation of the proposed approach. Section 4 presents a description of the experimental setup and evaluation results. Finally, section 5 concludes the paper, highlighting the future directions of research in the field of text information summarization and context learning.

2 RELATED WORKS

Data generated as a result of users interactions and information sharing in online social networks facilitates the researchers to understand various social phenomenon and human behavior such as small-world phenomenon, scale-free property, etc. at large scale, which were earlier not possible. Similarly, user-generated contents are analyzed and modeled at different levels of granularity for event detection, text summarization, and keyphrase extraction [5, 6]. Text information processing and summarization of short and informal microblogging posts is considered as an emerging and fascinating field of research, which is often viewed as an instance of the text summarization and context learning problem. In the early fifties, Luhn et al. [7] proposed an approach for automatic summarization using computational techniques. In the line, other statistics-based summarization and keyword extraction methods were proposed using the concepts of word frequency [7], term frequency-inverse document frequency (*tf-idf*) [8], term co-occurrence frequency, and so on. Similarly, another technique using the positional features (e.g., formatting titles and headings) was presented by Edmundson et al. in [9] for summarizing large documents.

Recently, a number of algorithms, such as SumBasic method proposed by Vanderwende et al. [10] and centroid-based algorithm proposed by Radev et al. [11] were developed for document summarization. In SumBasic method, more frequently occurring words were given higher probabilities than the less frequently occurring words, whereas the method proposed by Radev et al. [11] aimed to generate summary of a single document using a centrality-based measure. Elham et al. [12] presented a comment summarization method for the users' comments on YouTube videos. The proposed approach first clusters the comments and applies a ranking framework for automatically choosing the informative user-contributed comments. Abulaish et al. proposed a lexical and semantic features-based technique in [13] to extract keyphases from text corpus. In another study, Abulaish et al. presented a keyphrase-based tag-cloud generation framework to show the importance of keyphrases, rather than single words [14].

Besides the approaches mentioned above, graph-based models have also been widely used for document summarization with effective and encouraging results, mainly due to the fact that the graph representation of information enables better modeling of the linkages between the documents and accordingly produces more precise results. In this line, researchers have proposed several novel graph-based methods for ranking sentences or keywords and used them for summary construction. Hassan et al. [15] proposed a graph-based approach and applied random walk for term weighting, showing that a specific word feature can be used for context representation. Erkan et al. [16] proposed an algorithm, LexRank, which estimates the comparative importance of sentences based on their graph-based centrality scores. In this method, an adjacency matrix is created using IDF-modified *cosine* similarity measure, and stationary distribution is computed using Markov chain. The sentence similarity graph constructed using LexRank presents a better view of the sentences, compared to the centroid approach. In addition, Mihalcea et al. [17] proposed another graph-theoretic approach, TextRank, which exploits the Google's PageRank algorithm [18] and determines the hierarchical sentences or keywords in a document. On the other hand, Ge Bin et al. [19] used sliding window method to extract topic words, construct spatial vector, and generate an undirected graph. Thereafter, a vector space model was used for calculating edge weights, and weights of document sentences are determined using the compression ratio to determine the topic sentence of a document.

Inouye et al. [20] proposed a hybrid *tf-idf* algorithm which assigns weight to each sentence of a document, reflecting the influence of the sentence within the document. The sentences are arranged based on their weights and top-*k* sentences are considered as the summary of the underlying document. In another method, presented by Inouye et al. [20], tweets are first clustered into a number of groups based on a similarity measure, and thereafter each cluster is summarized by selecting the highly scored tweets determined by *tf-idf* algorithm. Jahiruddin et al. [21] presented a linguistic and latent semantic analysis based technique to extract important keywords and concepts from biomedical text corpus. They have also developed a visualization method to show the extracted concepts from different sources, where concepts are sorted in order of their relevance.

3 PROPOSED SUMMARIZATION AND CONTEXT LEARNING APPROACH

In this section, we present the functioning of the proposed summarization and context learning approach from microblogging data. Starting with a data crawling and preprocessing step, it follows a two-layer approach in which first layer aims to generate corpus summary through identifying top informative tweets and their most relevant sentences from the underlying corpus, whereas the second layer aims to identify most relevant keywords from the summary texts to conceptualize the thematic context of the underlying corpus. Further details about these processes are presented in the following sub-sections.

3.1 Data Acquisition and Preprocessing

We developed a data crawler using Twitter's REST API in Python to crawl tweets and related meta-data. The crawler uses `tweepy`², a Python library, to crawl tweets based on a list of hashtags input by the users. Besides tweets contents, the crawler also retrieves various meta-data like tweet id, tweet creation time, user screen name, and user id associated with the tweets. After crawling, we have applied various data pre-processing steps, such as cleaning, stemming, lemmatization, and smoothing. The cleaning step filtered out URLs, alphanumeric characters, non-English tweets, and punctuation marks from the tweets and converted all characters into lower-case to bring them into a uniform pattern. Thereafter, we applied stemming, lemmatization, and smoothing to convert tweets into record-size chunks for further processing.

3.2 Layer-1: Summary Generation

As stated above, the first layer of the proposed approach aims to identify most relevant tweets and their most representative sentences for summary generation. To this end, it applies a multi-level relevance identification approach. First, it generates a multi-dimensional weighted graph of the tweets and applies a random-walk approach to identify most relevant (top- k) tweets. Thereafter, it models top- k tweets using another graph structure in which nodes represent sentences and edges represent their similarity values, and a graph-theoretic algorithm, LexRank, is applied to identify most relevant sentences to summarize the underlying corpus. Further details about these steps are given in the following sub-sections.

3.2.1 Tweets Graph Generation. This section presents a graph-theoretic model to identify most relevant tweets from a given tweets corpus. To this end, the tweets corpus is modeled as a multi-dimensional graph $G = (V, E_h, E_m)$, where V is the set of nodes representing the tweets, $E_h \subseteq V \times V$ is the set of edges between the nodes (tweets) based on common *hashtags*, and $E_m \subseteq V \times V$ is the set of edges between the nodes (tweets) based on common *mentions*.

The similarity between tweets is a crucial step in the summarization process, and it is a challenging problem for microblogging data due to their short length, informal writing style, and other natural language nuances. In a dataset, different tweets (in terms of string matching) may be considered as similar if they are related to same event (i.e., if they are using similar *hashtags*), and similar tweets (in terms of string matching) may be considered as different if they are related to different events (i.e., if they are using different *hashtags*). Similarly, similarity between tweets can be established based on common *mentions*. Therefore, instead of using string matching algorithms to judge the similarity of tweets, we have considered common *hashtags* and *mentions* between the tweets to establish weighted links between them.

Considering a pair of nodes v_i and v_j having the set of *hashtags* as H_i, H_j respectively, the edge weight between the nodes v_i and v_j based on the *hashtags*, $w_h(i, j)$, is calculated as Jaccard similarity defined in equation 1. Similarly, for a pair of nodes v_i and v_j having the set of *mentions* as M_i and M_j respectively, the edge weight between the nodes v_i and v_j based on the *mentions*, $w_m(i, j)$, is

also calculated as Jaccard similarity defined in equation 2. Finally, an aggregated weight between the nodes v_i and v_j is calculated as a linear combination of $w_h(i, j)$ and $w_m(i, j)$, which is defined in equation 3. In this equation, α and β are constants to assign varying weights to different types of edges. We have given equal importance to both types of edges, i.e., the values of α and β are set to 0.5.

$$w_h(i, j) = \frac{|H_i \cap H_j|}{|H_i \cup H_j|} \quad (1)$$

$$w_m(i, j) = \frac{|M_i \cap M_j|}{|M_i \cup M_j|} \quad (2)$$

$$w(i, j) = \alpha w_h(i, j) + \beta w_m(i, j) : \quad \alpha + \beta = 1 \quad \text{and} \quad \alpha, \beta \geq 0 \quad (3)$$

3.2.2 Tweets Graph Analysis and Relevant Tweets Identification.

This section presents details about analyzing the tweets graph, generated in the previous section, to identify most relevant tweets for further processing and corpus summary generation. In order to determine a relevance score for each node (tweet), we have applied a random-walk algorithm proposed by Hassan et al. [15] on the tweets graph. It is a PageRank-inspired terms weighting algorithm that assigns relevance score to a word, based on the co-occurrence of the words in a corpus when it is modeled as a word-to-word relationship graph. Similar to PageRank, random-walk is a graph-based algorithm and generally applied on weighted graphs using the principle of voting/recommendation. We have used the random-walk algorithm proposed by Hassan et al. [15] for the tweets-relationship graph as constructed earlier and used a variable damping factor variation of the random-walk algorithm, where damping factor for each edge in every iteration is updated using a damping function given in equation 4. In this equation, $d_{w(i, j)}$ is the damping function and $\max_{p, q} \{w(p, q)\}$ represents the edge of the graph with maximum weight. Thereafter, importance score of each node (tweet) is updated using the formula defined in equation 5, where N represents the total number of nodes (tweets) in the tweets graph, d is the damping constant, C is a scaling constant which is taken as 0.95 in our experiment, and $\mathcal{N}(v_i)$ is the connecting/neighbors nodes (tweets) of the node v_i . The value of damping constant d is generally taken as 0.85 [18].

$$d_{w(i, j)} = \frac{w(i, j)}{\max_{p, q} \{w(p, q)\}} \quad (4)$$

$$S(v_i) = \frac{1-d}{|N|} + \sum_{v_j \in \mathcal{N}(v_i)} C \times \frac{d_{w(v_j, v_i)} \times S(v_j)}{|\mathcal{N}(v_j)|} \quad (5)$$

The whole procedure is repeated iteratively, and in each iteration, damping factor and ranking score for each node (tweet) is updated using equations 4 and 5, respectively such that the updated score S of a node (tweet) accumulates the updated weights of its neighboring nodes (tweets). This process is repeated until ranking score distribution of nodes (tweets) converges to a stationary distribution such that the change in score falls below a pre-defined threshold. At this stage, the value at each node (tweet) represents its rank/relevance score, and based on this value top- k nodes (tweets)

²<http://tweepy.readthedocs.io/en/v3.6.0/>

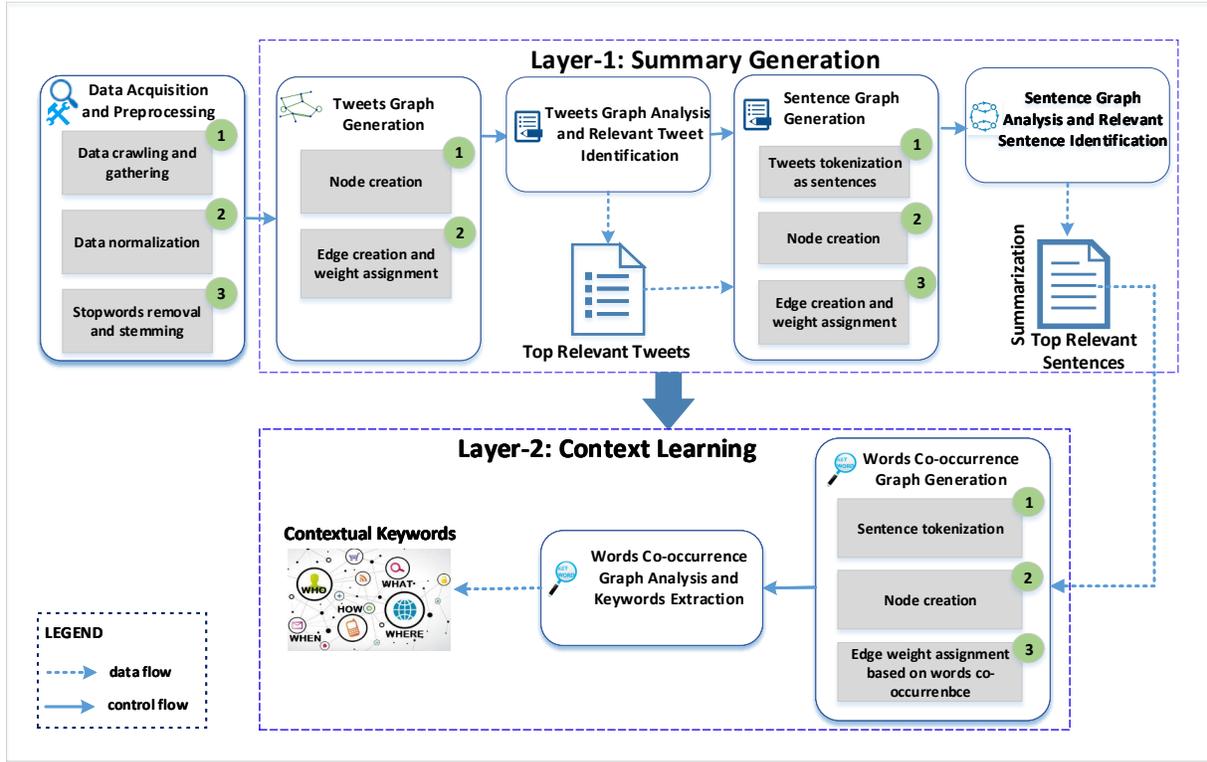


Figure 1: A schematic representation of the proposed approach for summarization and context learning from microblogging data

are selected and considered as the relevant tweets for summary generation.

3.2.3 Sentence Graph Generation. This section aims to model top- k tweets identified in the previous section to identify their most informative sentences for summary generation. To this end, we model the sentences of these tweets as a graph, termed as *sentence graph*, and analyze it using a graph-theoretic algorithm to identify most informative sentences. The top- k tweets are tokenized using period (.) as delimiter and a sentence graph is constructed using the tokenized sentences. The sentence graph is defined as $G_S = (V, E_{S_x, S_y})$, where V represents the set of vertices (sentences) and $E_{S_x, S_y} \subseteq V \times V$ represents the set of edges between the nodes (sentences). Similarity (weight) between a pair of nodes (sentences) of the graph G_S is calculated using *Cosine* similarity defined in equation 6, where S_x and S_y represent the set of words from the respective sentences, $tf_{(w,x)}$ and $tf_{(w,y)}$ represent the term frequency of the word w in the sentences S_x and S_y , respectively, and idf_w is the inverse document frequency of the word w .

$$Sim(S_x, S_y) = \frac{\sum_{w \in S_x \cap S_y} tf_{(w,x)} \cdot tf_{(w,y)} \cdot (idf_w)^2}{\sqrt{\sum_{w \in S_x} (tf_{(w,x)} \cdot idf_w)^2} \times \sqrt{\sum_{w \in S_y} (tf_{(w,y)} \cdot idf_w)^2}} \quad (6)$$

In the sentence graph G_S , an edge between a pair of nodes is created only when the value of $Sim(S_x, S_y)$ is greater than a threshold value δ , as defined in equation 7. The *Cosine* similarity values between all pairs of nodes of the graph G_S are stored in a matrix M_{CS} for efficient processing. Finally, the elements of M_{CS} are normalized using equation 8 to make it a stochastic matrix.

$$M_{CS}(x, y) = \begin{cases} 1, & \text{if } Sim(S_x, S_y) > \delta \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

$$\hat{M}_{CS}(x, y) = \frac{M_{CS}(x, y)}{degree(x)} \quad (8)$$

3.2.4 Sentence Graph Analysis and Relevant Sentence Identification. This section presents an analysis of the sentence graph G_S generated in the previous section to identify most informative sentences for summary generation. To this end, we have applied LexRank, which is a ranking algorithm proposed in [16] to estimate the relative importance of the nodes based on their graph-based centrality measures. LexRank is applied on the similarity matrix \hat{M}_{CS} . It uses power iteration approach based on Markov chain to observe the stationary distribution and keeps on iterating until the values of the matrix are converged. In the iterative procedure, all nodes are initially assigned equal weights, and thereafter, the weights of the nodes are adjusted on the basis of their similarity with other nodes of the graph. The iterative process repeats until the weights of the nodes converge, i.e., the changes in the node weights are below a

threshold value ϵ . A formal description of the LexRank algorithm to determine a numeric score for each node of the sentence graph is presented in algorithms 1 and 2.

Algorithm 1: LexRank_Score(S, δ , ϵ)

```

/* S is an array of n sentences,  $\delta$  and  $\epsilon$  are thresholds */
/* Output an array L of sentence scores */
1 Array CosineMatrix[n][n];
2 Array Degree[n];
3 Array L[n];
4 for i  $\leftarrow$  1 to n do
5   for j  $\leftarrow$  1 to n do
6     CosineMatrix[i][j] = Sim(S[i], S[j]);
7     if CosineMatrix[i][j] >  $\delta$  then
8       CosineMatrix[i][j] = 1;
9       Degree[i] ++;
10    end
11   else
12     CosineMatrix[i][j] = 0;
13   end
14 end
15 end
16 for i  $\leftarrow$  1 to n do
17   for j  $\leftarrow$  1 to n do
18     CosineMatrix[i][j] = CosineMatrix[i][j]/Degree[i];
19   end
20 end
21 L = Sentence_Score(CosineMatrix, n,  $\epsilon$ );
22 return L;

```

Algorithm 2: Sentence_Score(M, n, ϵ)

```

/* M is a stochastic, irreducible and aperiodic matrix */
/* n is the matrix size,  $\epsilon$  is a threshold */
/* Output the eigenvector p */
1  $p_0 = \frac{1}{n} \mathbf{1}$ 
2 t=0
3 repeat
4   t=t+1
5    $p_t = M^T p_{t-1}$ 
6    $\delta p_t = ||p_t - p_{t-1}||$ 
7 until  $\delta p_t < \epsilon$ ;
8 return  $p_t$ 

```

After convergence, the elements of p_t represent the relevance scores of the respective sentences. Based on this relevance score values, top- m sentences are selected to represent the summary of the underlying corpus.

3.3 Layer-2: Context Learning

As stated earlier, the aim of this layer is to further analyze the summary texts for keywords extraction and conceptualization of the corpus, highlighting its underlying context. Like previous processes, we model summary texts as a words co-occurrence graph and apply a graph-theoretic approach to identify most relevant words for conceptualizing the context of the underlying corpus. Further details about these processes are presented in the following sub-sections.

3.3.1 Words Co-occurrence Graph Generation. This section aims to model summary texts generated by the first layer of our proposed approach to identify most relevant words (aka keywords) for context representation of the corpus. To this end, we model the summary texts as a graph, termed as *words co-occurrence graph*, and

analyze it using a graph-theoretic algorithm to identify most relevant keywords. The summary texts are tokenized into words and words co-occurrence at sentence-level is considered for generation of the *words co-occurrence graph*. The *words co-occurrence graph* is defined as $G_W = (V, E)$, where V represents the set of nodes representing words, and $E \subseteq V \times V$ represents the set of edges between the nodes. Based on the information contained in G_W , a words co-occurrence matrix M_{WC} is created as a binary-valued matrix using equation 9.

$$M_{WC}(w_i, w_j) = \begin{cases} 1, & \text{if } w_i \text{ and } w_j \text{ co-occur} \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

3.3.2 Words Co-occurrence Graph Analysis and Keywords Extraction. This section presents an analysis of the words co-occurrence graph G_W generated in the previous section to identify most relevant keywords. In many existing literatures on keywords extraction, the researchers have used terms frequency-based weighting schemes to identify keywords in text corpus. Though frequency count is a simplest solution for keywords extraction and work well in many applications, it does not considers the linkages between the words based on their sentence position. Rather, it considers each word as an independent constituent and analyzes accordingly. Therefore, in this study, we have used a graph-based ranking method, TextRank, which is proposed in [17] for keywords extraction from texts corpus. TextRank is a graph-based keywords extraction method based on Google PageRank [18] algorithm to rank the nodes of a words graph. The basic assumption behind TextRank is that if a word co-occurs frequently with other important words, then it is likely to be an important word. The TextRank algorithm is applied over the words co-occurrence matrix M_{WC} to compute the weight of each node (word) of the graph. In this process, the rank $TR(v_i)$ of a node v_i is calculated iteratively using the formula given in equation 10, where $adj(v_i)$ is the set of adjacent nodes of v_i , $O(v_j)$ represents the out-degree of the vertex v_j , N is the total number of words (nodes), and d is the damping constant having any value between 0 to 1. After convergence, the final scores of the nodes are used to identify top-ranked words from G_W and termed as keywords.

$$TR(v_i) = \frac{1-d}{|N|} + d \times \sum_{v_j \in adj(v_i)} \frac{TR(v_j)}{O(v_j)} \quad (10)$$

After extraction of top relevant keywords, we have applied a merging process to concatenate all contiguous occurring words, which can be termed as a keyphrase. For example, in a sample text *two Ghana badminton players didn't return home*, both *badminton* and *players* are selected by the TextRank algorithm as keywords, and merged together to generate a single keyphrase *badminton players* because of their adjacent position in the sentence.

4 EXPERIMENTAL SETUP AND RESULTS

In this section, we present a detailed description of our experimental setup and results. Starting with a brief description of the datasets

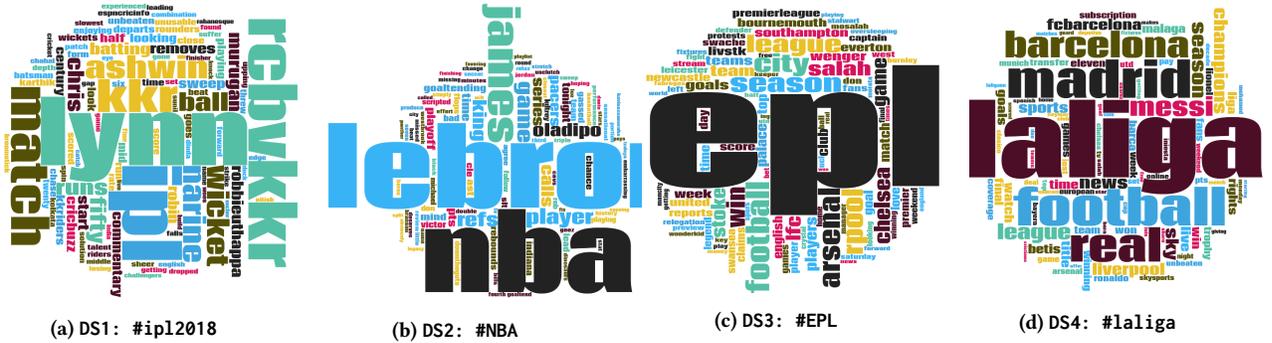


Figure 2: Word clouds representing the relative relevance of the top-100 keywords extracted from each of the four different real-world Twitter datasets

Table 1: Statistics of the crawled datasets

Dataset	Hashtag	#Tweets
DS1	ipl2018	2573
DS2	NBA	2561
DS3	EPL	2551
DS4	laliga	1914

and evaluation metrics in the following sections, we present experimental results and evaluation of our proposed summary generation and context learning method on different real-world datasets crawled from Twitter.

4.1 Datasets

In order to perform experimental evaluation of the proposed approach, we crawled Twitter data using our own crawler discussed earlier in this paper. For data crawling, we considered four *hashtags* namely #ipl2018, #NBA, #EPL, and #laliga representing different sports events, and run the crawler from 24th April 2018 to 30th April 2018. As a result, we generated four different datasets and named them as DS1, DS2, DS3, and DS4 representing the tweets corresponding to the hashtags #ipl2018, #NBA, #EPL, and #laliga, respectively. A brief statistics of the crawled datasets are given in table 1.

4.2 Evaluation Metrics

The performance of the proposed approach is evaluated using standard IR metric *Precision* which is defined as the ratio of the number of correctly extracted keywords to the total number of extracted keywords, as given in equation 11. In this equation TP is the number of *true positives* (i.e., number of correctly identified keywords representing the context of the related event), and FP represents the number of *false positives* (i.e., number of extracted keywords that are not related to the event represented by the underlying dataset). The extracted keywords were manually verified by domain experts from our research team. In this study, we considered an identified keyword as an instance of TP if the keyword semantically matches

with the respective hashtag of the underlying dataset. For example, the keywords *rcbvkk* and *lynn* extracted from the dataset corresponding to the *ipl2018* hashtag can be considered as the instances of TP , because *rcbvkk* represents the match between two teams (RCB and KKR) of the Indian Premier League (IPL), and *lynn* is a popular IPL player. The IPL is a professional cricket league, which is very popular in cricket-playing nations.

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

4.3 Results

In this section, we present some sample results obtained from different real-world Twitter datasets described in the previous section. Table 2 presents a list of top-20 summary sentences and top-20 keywords extracted using our proposed approach from each of the four different datasets. It can be observed from this table that the most of the keywords are relevant to the respective domains and they could be useful to conceptualize the contexts of the underlying corpus. Similarly, figure 2 presents the word-clouds of the top-100 keywords extracted from different datasets in which the font size of a word is directly proportional to its relevance score. It can be observed from these word-clouds that most of the words that are clearly visible due to their larger font-size are very much related to the context of the underlying datasets.

For experimental evaluation of the proposed summary generation approach, we have considered different cut-off values of k to extract top- k sentences as summary texts, and calculated *Precision* for each value of k using equation 11. Table 3 presents the *Precision* values of the summary generation process for $k = 20$, $k = 40$, $k = 60$, $k = 80$, and $k = 100$ on different real-world Twitter datasets. Figure 3 presents a visualization of these results. It can be observed from this figure that *Precision* values are decreased with increasing values of k due to increase in *false positives*. This decrease is due to the fact that relevant sentences are generally placed at top positions in the list.

Similarly, for experimental evaluation of the proposed context learning approach, we have considered different cut-off values of k to extract top- k keywords for conceptualization of the contexts of the underlying datasets, and calculated *Precision* values for each value of k using equation 11. Table 4 presents the *Precision* values of

Table 2: Top-20 sentences and keywords extracted from different datasets

Dataset	Hashtag	Top-20 Sentences	Top-20 Keywords
DS1	#ipl2018	rcbvkk r ipl, uthappa slogging look smooth ipl rcbvkk r, lynn finally found patch solution spin woes sweep, rcbvkk r ipl, lynn upping antekkr rcbvkk r ipl, fixxxd message ipl rcbvkk r, committed sweep shots rcbvkk r ipl, lynn build partnership, lynn robin uthappa, rcbvkk r ipl, robin uthappa failing convert start, dropped rcbvkk r ipl, uthappa, lynn nitish rana, lynn ashwin, rcbvkk r robin uthappa falls murugan ashwin, uthappa rcbvkk r ipl, batting exceptionally slow rcbvkk r ipl, nail biter lynn karthik crease, chris lynn ball fifty slowest twenty format	lynn, ipl, rcbvkk r, match, uthappa, kkr, ashwin, narine, wicket, ball, runs, chris, fifty, murugan, batting, rob-bieuthappa, start, removes, robin, cricbuzz
DS2	#NBA	rosie branson scripted, scripted nba, change nba, king nba, king flops nba, cheating called king nba, flops nba, chyambition agree dante diable king james change scenery nba, agree dante diable king james change scenery nba, nba garbage, nba scripted wwe follow lebron ratings, consistency, legendary effort lebron james pts reb asts, insane performance king james, domclare king lebron james doing king lebron james type nba, lebron king, lebron closest cav, losses nba championship lebron james wilt chamberlin, lilbthebasedgod lebron nba lil, lebron simply amazing	lebron, nba, james, game, player, refs, king, oladipo, calls, series, pacers, time, playoff, pts, tonight, goaltending, ast, mind, cle, indiana
DS2	#EPL	bonus deposits via skrill neteller, dmightyangel eplliverpool pts matcheschelsea awayhome brightonmanu pts matcheshome arsenalaway westhambrighton awayhom, eplliverpool pts matcheschelsea awayhome brightonmanu pts matcheshome arsenalaway westhambrighton awayhome watfordtot ptshome watfordhome newcastleaway westbromhome leist citychelsea pt-saway swanseahome huddersfieldhome liverpoolaway newcastle wow, betdeal xbet stronger, footballtips xbet stronger, footballtips, live beinsport, live, multisporttips xbet watch europa league games live bonus deposits, topic watch live, footballtips xbet europa league games live bonus deposits via, watch, recall-formen watch, profit hunters xbe watch europa league games live bonus deposits, profit hunters xbet utd vs arsenal live bonus deposits via skri, footballtips xbet watch europa league games live bonus deposits, betting deals xbet stronger, multisporttips xbet utd vs arsenal live bonus deposits via skri, damn wilfried zaha fire, footballtips xbet utd vs arsenal live bonus deposits via skri	epl, football, arsenal, liverpool, season, city, win, league, vs, live, salah, chelsea, game, lfc, stoke, team, players, teams, wenger, palace
DS2	#laliga	sad admit ive watched laliga, covered, stats covered, wouldnt bara, stats, sad times laliga, barcatalkpod bara talk supporter, apparently guard honor cwc btw partake, achieved brilliant management, congrats andrearadri laliga, fcbarcelona quiet, bara won laliga unbeaten, realmadriden laliga, repeat times receive maximum bonu, fcbarcelona scored, inevitable memes arrived fcbarcelona wrapped laliga, modrics reply tell mller simple scored, thesportalk reporter mller told doesnt rma won, deulofeu england talk animated bara movie fcbarcelonab fc, theriteammas won laliga match	laliga, football, madrid, real, barcelona, messi, league, season, champions, news, sky, liverpool, la, fcbarcelona, sports, th, live, watch, goals, title

the context learning process for $k = 20$, $k = 40$, $k = 60$, $k = 80$, and $k = 100$ on different datasets. Figure 4 presents a visualization of these results. It can be observed from this figure that the *Precision* values are decreased with increasing values of k due to increase in

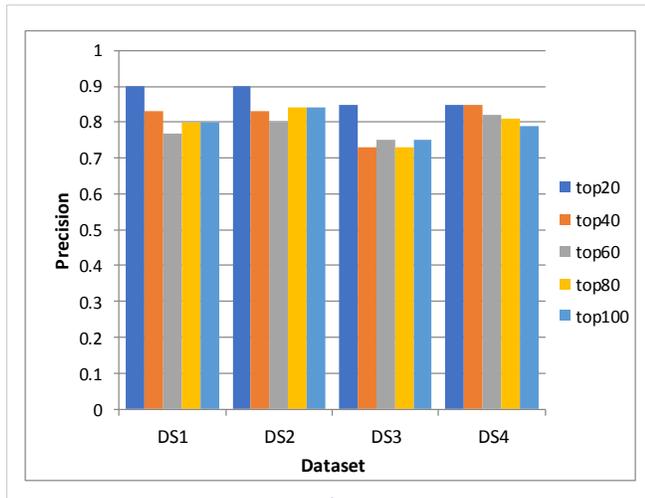
false positives. This decrease is due to the fact that relevant keywords are generally placed at top positions in the list.

Table 3: Performance evaluation results of the summary generation process on four different real-world Twitter datasets

Dataset	Top-k Sentences				
	k=20	k=40	k=60	k=80	k=100
DS1	0.90	0.83	0.77	0.80	0.80
DS2	0.90	0.83	0.80	0.84	0.84
DS3	0.85	0.73	0.75	0.73	0.75
DS4	0.85	0.85	0.82	0.81	0.79

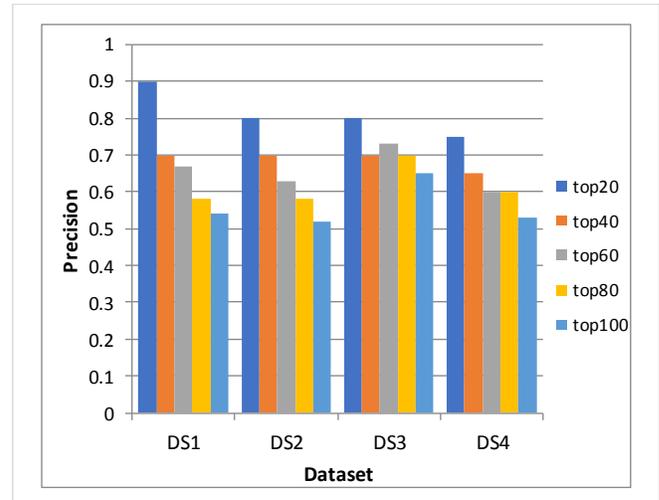
Table 4: Performance evaluation results of the context learning process on four different real-world Twitter datasets

Dataset	Top-k Keywords				
	k=20	k=40	k=60	k=80	k=100
DS1	0.90	0.70	0.67	0.58	0.54
DS2	0.80	0.70	0.63	0.58	0.52
DS3	0.80	0.70	0.73	0.70	0.65
DS4	0.75	0.65	0.60	0.60	0.53

**Figure 3: Visualization of the performance evaluation results of the summary generation process on four different real-world Twitter datasets**

5 CONCLUSION AND FUTURE WORKS

In this paper, we have presented different graph-based models to represent microblogging data at different levels of granularity. We have also presented an application of various graph data analysis techniques, such as *random walk*, LexRank, and TextRank to extract informative components from our proposed graph-based textual data models. The first two algorithms (*random-walk* and LexRank) have been used to identify relevant tweets and their most

**Figure 4: Visualization of the performance evaluation results of the context learning process on four different real-world Twitter datasets**

representative sentences for summary generation. On the other hand, the third algorithm (TextRank) is used to identify prominent keywords without using Parts-Of-Speech (POS) tagging or parsing of the summary texts to conceptualize the contexts of the underlying corpus. The proposed approach for summary generation and context learning seems very useful to analyze microblogging data, which is generally noisy and informal due to their uncontrolled generation by the naive and casual users. Currently, we are in the process of evaluating the proposed approaches over large datasets. The application of the proposed approaches for topic modeling and tracking to monitor the temporal evolution of various events in online social media seems one of the promising areas of research. Similarly, extending the proposed approaches for Hadoop platform to deal with the huge size of user-generated-contents is also one of the promising future directions of research.

REFERENCES

- [1] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. 1, pp. 993–1022, 2003.
- [2] S. Y. Bhat and M. Abulaish, "Hoctracker: Tracking the evolution of hierarchical and overlapping communities in dynamic social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 4, pp. 1019–1032, 2014.
- [3] G. B. YomiTov, S. Ashtar, D. Altman, M. Natapov, N. Barkay, M. Westphal, and A. Rafaeli, "Customer sentiment in web-based service interactions: Automated analyses and new insights," in *Proceedings of the 27th International Web Conference*, (Lyon, France), pp. 1689–1697, ACM, 2018.
- [4] A. Kamal and M. Abulaish, "Statistical features identification for sentiment analysis using machine learning techniques," in *Proceedings of International Symposium on Computational and Business Intelligence*, (Delhi, India), pp. 178–181, IEEE Computer Society, 2013.
- [5] N. Azam, Jahiruddin, M. Abulaish, , and N. A.-H. Haldar, "Twitter data mining for events classification and analysis," in *Proceedings of 2nd International Conference on Soft Computing and Machine Intelligence*, (Hong Kong), pp. 79–83, IEEE Computer Society, 2015.
- [6] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th international conference on World Wide Web*, (Raleigh, USA), pp. 851–860, 2010.
- [7] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, 1958.

- [8] X. Wenhai and W. Youkui, "A chinese keyword extraction algorithm based on tfidf method," *Information Studies: Theory & Application*, vol. 2, pp. 298–302, 2008.
- [9] H. P. Edmundson, "New methods in automatic extracting," *Journal of the ACM*, vol. 16, no. 2, pp. 264–285, 1969.
- [10] L. Vanderwende, H. Suzuki, C. Brockett, and A. Nenkova, "Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion," *Information Processing & Management*, vol. 43, no. 6, pp. 1606–1618, 2007.
- [11] D. R. Radev, S. Blair-Goldensohn, and Z. Zhang, "Experiments in single and multi-document summarization using mead," in *Proceedings of the 24th International Conference on Research and Development in Information Retrieval*, (New Orleans, USA), ACM, 2001.
- [12] E. Khabiri, J. Caverlee, and C. F. Hsu, "Tsummarizing user-contributed comments," in *Proceedings of the 5th International Conference on Weblogs and Social Media*, (Barcelona, Spain), pp. 534–537, AAAI Press, 2011.
- [13] M. Abulaish and T. Anwar, "A supervised learning approach for automatic keyphrase extraction," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 11, pp. 7579–7601, 2012.
- [14] M. Abulaish and T. Anwar, "Keyphrase-based tag cloud generation framework to conceptualize textual data," *International Journal of Adaptive, Resilient and Autonomic Systems*, vol. 4, no. 2, pp. 72–93, 2013.
- [15] S. Hassan, R. Mihalcea, and C. Banea, "Random walk term weighting for improved text classification," in *Proceedings of the International Conference on Semantic Computing*, (California, USA), pp. 421–439, IEEE Computer Society, 2007.
- [16] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, no. 1, pp. 457–479, 2004.
- [17] R. Mihalcea and P. Tarau, "Textrank: Bringing order into texts," in *Proceedings of the International Conference on Empirical Methods in Natural Language Processing*, (Barcelona, Spain), ACL, 2004.
- [18] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *Proceedings of the 7th International Conference on World Wide Web*, (Brisbane, Australia), pp. 107–117, ACM, 1998.
- [19] B. Ge, F. Li, F. Li, and W. Xiao, "Subject sentence extraction based on undirected graph construction," *Computer Science*, vol. 38, no. 5, pp. 181–185, 2011.
- [20] D. Inouye and J. K. Kalita, "Comparing twitter summarization algorithms for multiple post summaries," in *Proceedings of the 3rd International Conference on Social Computing*, (Boston, USA), pp. 298–306, IEEE Computer Society, 2011.
- [21] Jahiruddin, M. Abulaish, and L. Dey, "A concept-driven biomedical knowledge extraction and visualization framework for conceptualization of text corpora," *Journal of Biomedical Informatics*, vol. 43, no. 6, pp. 1020–1035, 2010.