

Twitter Data Mining for Events Classification and Analysis

Nausheen Azam*, Jahiruddin†, Muhammad Abulaish‡§, SMIEEE, and Nur Al-Hasan Haldar‡

*School of IT, Centre for Development of Advanced Computing (CDAC), Noida, India

†Department of Computer Science, Jamia Millia Islamia (A Central University), Delhi, India

‡Centre of Excellence in Information Assurance, King Saud University, Riyadh, KSA

Abstract—The increasing popularity of the micro-blogging sites like Twitter, which facilitates users to exchange short messages (aka *tweets*) is an impetus for data analytics tasks for varied purposes, ranging from business intelligence to nation security. Twitter is being used by a large number of users for events update and sentiment expression. Since tweets are generally unstructured in nature and do not follow grammatical structures, parsing techniques generally do not work well due to incorrect parts-of-speech assignment to individual words. In this paper, we have proposed an n-gram based statistical approach to identify significant terms and using them for vector-space modelling of the tweets. Thereafter, a social graph generation method is proposed, considering tweets as nodes and the degree of similarity between a pair of tweets as a weighted edge between them. The social graph is decomposed into various clusters using Markov Clustering technique, wherein each cluster corresponds to a particular event. The experiment is carried out using a corpus of 3100 tweets related to Israel-Gaza conflicts, Delhi assembly election, and union budget 2015 . The experimental results are encouraging, showing the efficacy of the proposed social graph generation and event classification methods.

Index Terms—Social network analysis; Twitter data mining, Social graph generation; Markov clustering; Event classification.

I. INTRODUCTION

The recent advancements in Web technologies have attracted a large number of internet users to use online social networks like Facebook and Twitter for varied purposes, including events update and data sharing. As a result, social network applications are emerging as a powerful online tool for users to express and share their views with other users around the globe. Twitter is one such social media application with a large and rapidly growing user base. It has become the most popular micro-blogging social networking website in which users share their views in the form of very short message limited to 140 characters – called “tweets”. Besides events update and data sharing, Twitter is also being used for many other purposes, including product marketing, political campaign, and market research. In addition, Twitter is also being used by the users to express their opinions and views about prominent issues of day-to-day life that may be social, political, or entertainment. Analyzing tweets to spot emerging issues and trends and to assess public opinion concerning topics and events is of considerable interest to various stakeholders, including government, companies, and security agencies.

However, performing such analysis is technically challenging due to unstructured nature of tweets and that the opinions of the users are typically expressed as informal communications and are buried under the pile of vast and largely irrelevant data generated by the millions of users and other online content producers. One of the ground challenges in analyzing Twitter data is their classification on the basis of the events under discussion, which is generally conceptualized using a set of significant terms embedded within the tweets. For example, “Israel-Gaza conflict” event can be conceptualized using the key terms *israel*, *gaza*, *palestinian*, *hammas*, *peace*, etc., whereas *election*, *vote*, *party*, etc. can be used to conceptualize the event “Delhi assembly election”.

In this paper, we present a statistical approach to analyze twitter data using the concepts of social network generation and graph-based clustering. Tweets are tokenized using n-gram technique and analyzed using Latent Dirichlet Allocation (LDA) method to identify significant key terms, which are later on used for social network generation. Finally, Markov Clustering method is applied on the generated social network to identify dense regions (aka communities or cliques), each one representing the set of tweets related to a particular event.

The rest of the paper is organized as follows. Section II presents a brief review of the twitter data analysis techniques. Sections III presents the proposed data mining technique for tweets classification and analysis. Section IV presents the experimental setup and results. Finally, section V concludes the paper with future directions of work.

II. RELATED WORK

Twitter has recently evolved as a popular micro-blogging website and consequently a number of methods are proposed by different researchers to analyze twitter data for varied purposes. Chung and Mustafara [1] examined the predictive power of social media (especially, the Twitter) using sentiment analysis methods and identified conflicting results in the domain of US political elections held in 2010. Cheong and Lee [2] studied the detection of interesting patterns using Self-Organizing Map (SOM) related to 2009 Iran election issue and the iPhone OS 3.0 software launch. Akcora and Ferhatosmanoglu [3] identified the breakpoints in public opinion to extract major news about the events effectively and developed an application where users can view the important news stories and find the related articles on the Web. Bollen et al. [4]

§Corresponding author. E-mail: abulaish@ieee.org, Tel. +91-11-26980014

proposed a method to make precise and useful predictions for stock market. In [5], Thelwall assessed whether popular events are typically associated with increase in sentiment strength.

Tracking influence is another important task related to twitter data analysis. Influential users play an important role in the society, and influence tracking may be useful for a number of applications ranging from election to marketing. In [6], the authors analyzed the role of structural features like indegrees, retweets, and mentions for influence tracking and investigated the dynamic nature of user influence across topic and time.

Sentiment analysis and opinion mining is another important field in twitter data mining. Pak and Paroubek [7] proposed a method based on the linguistic analysis of tweets for sentiment analysis on twitter data. Their system is able to determine the positive, negative, and neutral sentiments of a given tweet. In [8], the authors proposed an algorithm to classify tweets as positive or negative. They studied a number of classifiers based on n-gram and Parts-Of-Speech (POS) tag features and reported that multinomial naive Bayes unigram using mutual information outperforms the other approaches.

Another area of research related to twitter data mining is event identification. In [9], the authors proposed a general framework for event identification in social media documents. They used similarity metric learning approaches to produce high quality clustering results. They reported that similarity metric learning techniques yield better performance than traditional approaches that considers text-based similarity. Sakaki et al. [10] proposed a method to identify real-time events. They proposed an algorithm to identify target events in real-time and considered tweets-related features like keywords, number of words and their context for detecting the target events. The main focus of their study is to identify earthquake event. In [11], the authors developed a system to classify tweets into real-world event tweets and non-event tweets. To this end, they have considered temporal, social, topical, and Twitter-centric features for classification of tweets into event and non-event class. In contrast to classifying tweets into only two classes, we consider tweets analysis as a clustering problem in which tweets are classified into various classes based on how many events are described by them.

III. PROPOSED METHOD

In this section, we present the functional details of our proposed tweets mining approach, which aims to classify tweets based their relatedness with various events. Figure 1 presents the work-flow of the proposed method and highlights the functioning details of the various working modules. Tweets crawling aims to retrieve tweets from the server and store them on local machine for analysis. Tweets pre-processing and tokenization process aims to extract tweets contents, filter out unwanted constituents like embedded emoticons and URLs, and tokenize them into 1-grams for further processing. Feature extraction and social network generation identifies significant key terms from the tweets using Latent Dirichlet Allocation (LDA) method and use them to model the tweets as a social network. Finally, Markov clustering is applied on the generated

social network to crystallize it into various clusters, each one representing a particular event. Further details about these functions are presented in the following sub-sections.

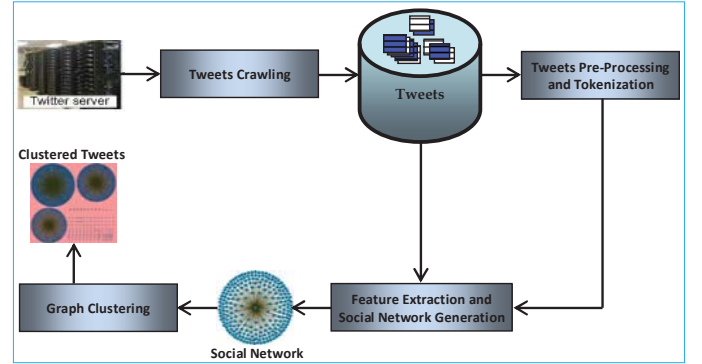


Fig. 1: Work-flow of the proposed twitter data mining technique

A. Tweets Crawling

We have developed a crawler using Twitter Application Program Interface (API) to retrieve tweets from the server and store them on local machine for further processing. In addition to tweets, the crawler also retrieves various users and tweets related structural features and stores them in a structured format. The structural features can be clubbed with the contents to design a better tweets analysis system, which is one of our future directions of work.

B. Tweets Pre-processing and Tokenization

Tweets pre-processing aims to filter out unwanted constituents like special characters, emoticons, URLs etc. associated with each tweets. Thereafter, n-gram technique with the value of n as 1 is applied to tokenize tweets into bag-of-words.

C. Feature Extraction and Social Network Generation

In this phase, significant key terms are identified as tweets features to represent them into vector-space model, which is the first step for the generation of social graph. For significant key terms identification process, each token of a tweet is considered as a candidate term provided it is neither a stop-word nor containing any special characters. Once the list of candidate terms for each tweet is identified, a term-tweet matrix A of order $m \times n$ is generated, where m is the number of candidate terms and n is the number of tweets. The row of the matrix A represents a term vector and that a column represents a tweet vector. The $(i, j)^{th}$ element of matrix A , a_{ij} , is determined as the weight of the term t_i in j^{th} tweet. The weight of a term t_i in j^{th} tweet, $\omega(t_{i,j})$, is calculated using equations 1 and 2 where, $tf(t_{i,j})$ is the number of times t_i occurs in j^{th} tweet. $|D|$ is the total number of tweets, and $|\{d_j : t_i \in d_j\}|$ is the number of tweets containing t_i . The matrix A is normalized in such a way that the length of the tweet vectors becomes 1.

$$\omega(t_{i,j}) = tf(t_{i,j}) \times idf(t_i) \quad (1)$$

$$idf(t_i) = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|} + 1 \quad (2)$$

Since the term-tweet matrix is generally sparse matrix and the dimension of the term as well as tweet vectors are large, Singular Value Decomposition (SVD) is applied to map the feature set into a low-dimensional space. This increases the efficiency of the proposed method both in terms of memory and computing time requirements. For a given $m \times n$ matrix with $m \geq n$, the SVD decomposes it into an $m \times n$ orthogonal matrix U , an $n \times n$ diagonal matrix S , and an $n \times n$ orthogonal matrix V such that $A = USV^T$. In this decomposition, U represents the term matrix and V represents the tweet matrix. Each row of matrix V represents a tweet vector whose dimension is reduced from m to n in the new feature space.

To assign numeric scores to the candidate terms, we use Latent Dirichlet Allocation (LDA), which is a generative probabilistic model in which documents are represented as random mixtures over latent topics characterized by a distribution over words [12]. For LDA execution, we create a data file using the clusters of the tweets. In this file, the first line contains an integer value k representing the number of clusters (number of documents for LDA). Following this, there are k paragraphs, one for each cluster, containing the list of terms obtained from the tweets belonging to the corresponding cluster.

We have used `JGibbLDA`¹ to execute LDA to generate Θ and Φ matrices. We have set the Dirichlet hyper parameters α and β as 0.1 and 0.5, respectively at the time of LDA execution. The Φ matrix contains the term-topic distributions, i.e., $p(\text{term}_t | \text{topic}_t)$. Each row in this matrix is a topic and each column is a candidate term in the document. The Θ matrix contains the topic-cluster distributions, i.e., $p(\text{topic}_t | \text{cluster}_c)$. Each row in this matrix is a cluster (document) and each column is a topic. We use Φ and Θ matrices to assign a numeric score to each term using equations 3 and 4 in which $|s[l]|$ is the size (number of terms) of the l^{th} cluster, n is the number of topics (we have taken $n = 100$), and k is the number of clusters that is treated as number of documents in this case. After calculating the score of each term, we arrange them in decreasing order of their scores and consider top- n terms as significant key terms. Further details related to the identification of key terms (aka key phrases) can be found in one of our previous works [13].

$$\text{score}(t_i) = \max_{j=1}^n \{\Phi_{j,i} \times \omega_j\} \quad (3)$$

$$\omega_j = \sum_{l=1}^k \Theta_{l,j} \times |s[l]| \quad (4)$$

The top n key terms are used to represent each tweet as an n -dimensional feature vector. The i^{th} element of the feature vector of a tweet is set to 1 if the i^{th} key term is present in the tweet, otherwise it is set to 0. Thereafter, the social network is generated as a weighted graph in which tweets

represent nodes and similarity value between a pair of tweets is considered as a weighted edge between them, provided it is greater than 0. *Cosine similarity* is used to calculate similarity between a pair of tweets. *Cosine similarity* is one of the most popular similarity measures to calculate the similarity between two n dimensional vectors, measuring the cosine of the angle between them. The *cosine similarity* of two n -dimensional vectors a and b can be calculated using equation 5.

$$\text{Cosine}(a, b) = \frac{\sum_{i=1}^n a_i \times b_i}{\sqrt{\sum_{i=1}^n (a_i)^2} \times \sqrt{\sum_{i=1}^n (b_i)^2}} \quad (5)$$

D. Graph Clustering

Once the social network is generated for the complete set of tweets, Markov CLustering (MCL) algorithm is applied to crystallize in into various clusters, each one representing a particular event. The MCL transforms the given graph into directed graph with several weakly connected components, each one resulting into a separate cluster. The MCL is an iterative method that interleaves matrix expansion and inflation steps [14]. Matrix expansion corresponds to taking successive powers of the transition matrix, while matrix inflation makes the higher probability transition and reduces the lower probability transition. It should be noted that MCL does not require the number of clusters k parameter for clustering, rather it requires an inflation parameter $r > 1$. A small value of r results in small number of clusters of larger size, whereas a high value of r generates large number of clusters of smaller size.

IV. EXPERIMENTAL SETUP AND RESULTS

In this section, we present our experimental setup and evaluation results. For experiment, we have crawled 3100 tweets related to three different events *Israel-Gaza conflict*, *Delhi assembly election*, and *union budget 2015* using Twitter's API. Out of these 3100 tweets, 1500 are related to *Israel-Gaza conflict*, 900 are related to *Delhi assembly election*, and remaining 700 are related to *union budget 2015*. Further statistics about these data sets is given in Table I. We have applied the key term extraction process discussed in the previous section and top-100 key terms, as shown in Table II, are considered for vector-space modelling of the tweets. These key terms are used to generate the feature vectors of each tweet. The feature vector of a tweet is a 100-dimensional binary vector in which the values are either 1 or 0, depending on the presence or absence of the respective key term in the tweet.

Finally, social network is clustered using Markov CLustering (MCL) algorithm, for which the value of inflation parameter r is determined empirically. MCL is applied on the social network with different values of r , ranging from 1.2 to 5.2, and finally 1.5 is considered as the optimal one, as shown in Figure 3, for further experimentation. The clustered tweets at $r = 1.5$ is shown in Figure 2, in which each cluster corresponds to a particular event. It can be seen in this figure that besides three bigger clusters there are some isolated nodes

¹<http://jgibblda.sourceforge.net/>

TABLE I: Statistics of the Twitter data sets

Data set category	Tweets-related statistics				User-related statistics		
	Number of tweets	Avg. no. of hashtags	Avg. no. of URLs	Avg. no. of mentions	Avg. no. of followers	Avg. no. of friends	Avg. no. of tweets
Israel-Gaza Conflict	1500	1.30	0.37	0.95	2104.40	1093.84	18865.53
Delhi Assembly Election	900	0.32	0.49	1.03	2352.48	600.97	29707.23
Union Budget 2015	700	0.98	0.71	0.83	1597.59	973.84	28244.10
Total count	3100	0.94	0.48	0.95	2061.99	923.65	24130.86

TABLE II: Significant key terms and their LDA scores

Key term	Score	Key term	Score	Key term	Score	Key term	Score
palestine	584.35	reasons	57.43	results	30.36	budget2015	22.64
gaza	480.41	hammas	56.73	corporate	27.37	prayforgaza	22.61
israel	392.34	free	55.94	responsibility	27.37	attack	22.61
delhi	331.75	stop	55.14	uphold	27.37	injured	22.61
aap	299.56	people	53.56	syria	27.37	rights	21.82
budget	268.91	introspect	53.04	chief	26.70	narendramodi	21.58
kejriwal	198.61	conflict	51.97	victory	26.70	finance	21.40
union	188.27	civilians	47.21	sarkar	26.70	abppensioen	21.03
israeli	147.97	occupation	47.21	pray	26.58	divest	21.03
bjp	138.63	india	46.83	jaitley	26.36	dead	21.03
hamas	132.10	war	45.62	soldiers	25.79	banks	20.23
bedi	96.93	human	44.83	terrorists	25.79	financing	20.23
palestinian	95.61	don	40.60	support	25.79	industry	20.16
gazaundersattack	90.05	polls	39.87	military	25.79	meets	20.12
kiran	87.42	arvindkejriwal	39.14	minister	25.74	freedom	19.44
unionbudget2015	84.67	jews	36.89	mahmoud	24.20	innocent	19.44
arvind	72.79	peace	36.89	media	24.20	party	19.39
killed	66.25	freepalestine	36.10	highlights	23.88	protest	18.65
loss	65.47	kill	35.31	killing	23.41	latest	17.92
dilli	65.47	rockets	33.72	abbas	23.41	govt	17.92
fatwa	64.01	hospital	32.93	voted	23.04	secretary	17.92
modi	63.28	world	32.13	elections	23.04	death	17.85
blame	63.28	illegal	31.34	aapsweep	23.04	netanyahu	17.85
children	62.28	kiski	31.09	tax	22.64	superbudget	17.68
election	58.16	president	30.55	arun	22.64	rail	17.06

that do not belong to any of them. On analysis we found that the tweets corresponding to these nodes are generally not directly related to the events under consideration and they can be considered as outliers.

For evaluation of the proposed method, we have considered $F_{\alpha=0.5}$ (or F_P at $\alpha = 0.5$) and $F_{B-cubed}$ (or F_B), where F_P is defined as the harmonic mean of *purity* and *inverse purity*, and F_B is defined as the harmonic mean of *B-cubed precision* and *B-cubed recall*, which are defined in the following paragraphs.

Purity and Inverse Purity: If C_i is a cluster (containing the i^{th} tweet) generated by the system, L_j is the actual cluster (containing the j^{th} tweet), and n is the total number of tweets, then *purity* and *inverse purity* are defined using equations 6 and 7, respectively.

$$purity = \sum_i \frac{|C_i|}{n} \times \max Precision(C_i, L_j) \quad (6)$$

$$inversePurity = \sum_i \frac{|L_i|}{n} \times \max Precision(L_i, C_j) \quad (7)$$

B-Cubed Precision and B-Cubed Recall: If C_i is a cluster (containing the i^{th} tweet) generated by the system, L_i is the actual cluster (containing the i^{th} tweet) then precision and recall of the i^{th} tweet are defined using equations 8 and 9,

respectively. Thereafter, the mean of all individual precisions and recalls is taken as the values of B-cubed precision and B-cubed recall, respectively.

$$precision(i) = \frac{|C_i \cap L_i|}{|C_i|} \quad (8)$$

$$recall(i) = \frac{|C_i \cap L_i|}{|L_i|} \quad (9)$$

For evaluation purpose, we have labeled each *Israel-Gaza conflict* related tweet by ‘‘G’’, each *Delhi assembly election* related tweet by ‘‘DE’’, and each tweet related to *union budget 2015* by ‘‘UB’’. We have also developed a Java application to calculate the values of the above-mentioned evaluation metrics in an automatic manner. Evaluation results for different values of the inflation parameter, r , are shown in Table III. Figure 3 is a visual representation of the results shown in Table III. It can be observed from this figure that the values of both F_P and F_B parameters is highest for $r = 1.5$.

V. CONCLUSION AND FUTURE WORK

In this paper, we have presented a Twitter data mining technique for events classification and analysis. LDA is used to identify significant key terms for tweets representation using

TABLE III: Evaluation results of the proposed method for different inflation parameter (r) values

r	No. of connected components	No. of nodes	No. of isolated nodes	Purity	Inverse Purity	F_P	Average B-cubed precision	Average B-cubed recall	F_B
1.2	90	3100	89	0.5048	0.9713	0.6644	0.3909	0.9438	0.5529
1.5	92	3100	89	0.9877	0.9590	0.9732	0.9758	0.9204	0.9473
2.5	99	3100	90	0.9910	0.9355	0.9624	0.9822	0.8762	0.9262
3.5	102	3100	91	0.9910	0.9110	0.9493	0.9828	0.8348	0.9027
4.5	107	3100	95	0.9913	0.8481	0.9141	0.9831	0.7470	0.8490
5.2	112	3100	97	0.9923	0.7432	0.8499	0.9849	0.5954	0.7422

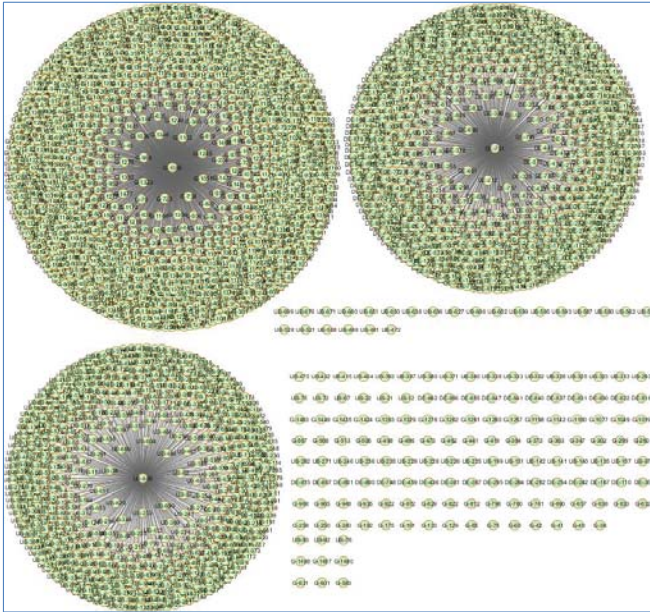


Fig. 2: Clustered tweets using MCL for $r = 1.5$

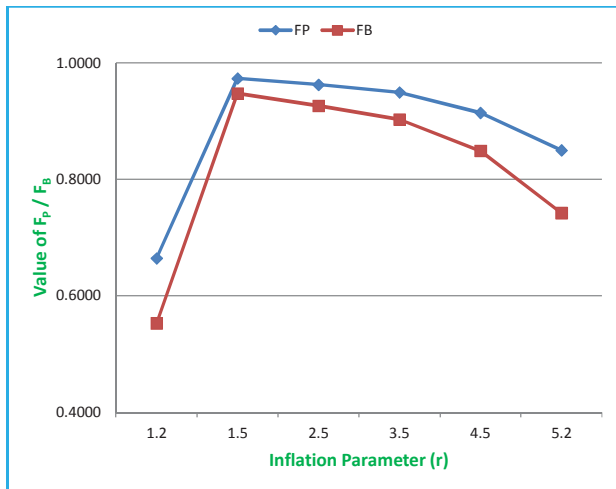


Fig. 3: Visualization of F_P and F_B measures for different inflation parameter (r) values

the vector-space model. We have also proposed a social network generation method, which models tweets as a weighted graph in which the weight of an edge represents the topical similarity of the tweets. Finally, Markov clustering is used to

crystallize the social network into various clusters, each one representing a particular event. Since Twitter API provides various structural features, development of a hybrid approach to analyze twitter data using structural and content-related features could be a promising area of future research.

ACKNOWLEDGMENT

This work was funded by the National Plan for Science, Technology and Innovation (MAARIFAH), King Abdulaziz City for Science and Technology, Kingdom of Saudi Arabia under the Award Number 12-INF-2533.

REFERENCES

- [1] J. Chung and E. Mustafaraj, "Can collective sentiment expressed on twitter predict political elections?" in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011, pp. 170–171.
- [2] M. Cheong and V. Lee, "A study on detecting patterns in twitter intratopic user and message clustering," in *Proceedings of the 2010 20th International Conference on Pattern Recognition*, 2010, pp. 3125–3128.
- [3] C. G. Akcora, M. A. Bayir, M. Demirbas, and H. Ferhatosmanoglu, "Identifying breakpoints in public opinion," in *Proceedings of the First Workshop on Social Media Analytics*, 2010, pp. 62–66.
- [4] J. Bollen, H. Mao, and X.-J. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, 2011.
- [5] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment in twitter events," *Journal of the American Society for Information Science and Technology*, vol. 62, no. 2, pp. 406–418, 2011.
- [6] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi, "Measuring user influence in twitter: The million follower fallacy," in *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, 2010, pp. 10–17.
- [7] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, 2010, pp. 1320–1326.
- [8] A. Go, L. Huang, and R. Bhayani, "Twitter sentiment analysis," Stanford University, Stanford, California, USA, CS224N - Final Project Report, 2009.
- [9] H. Becker, M. Naaman, and L. Gravano, "Learning similarity metrics for event identification in social media," in *Proceedings of the third ACM international conference on Web search and data mining*, 2010, pp. 291–300.
- [10] 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*, 2010, pp. 851–860.
- [11] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: Real-world event identification on twitter," in *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, 2011, pp. 438–441.
- [12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2003.
- [13] M. Abulaish, Jahiruddin, and L. Dey, "Deep text mining for automatic keyphrase extraction from text documents," *Journal of Intelligent Systems*, vol. 20, no. 4, pp. 327–351, 2011.
- [14] S. van Dongen, "Graph clustering by flow simulation," Ph.D. Thesis, University of Utrecht, Utrecht, Netherlands, 2000.