Twitter has become a leading source of real-time world-wide information and a great medium for exploring emerging events, breaking news and general topics which most matter to a broad audience. On the other hand, the explosive rate of incoming information in Twitter leads users to experience information overload. Whereas, a significant fraction of tweets are about news events, summarizing the storyline of events can be helpful for users to easily access to the relevant and key information hidden among tweets and thereby draw high level conclusions. Storytelling is the task of providing chronological summaries of significant sub-events development and sketching the relationship between sub-events. In this paper, we propose a novel framework to generate a summarized storyline of news events from social point of view. Utilizing the concepts in graph-theory, we identify sub-events, summarize the evolution of sub-events and generate a coherent storyline of them. Our approach models a storyline as a directed tree of social salient sub-events evolving over time. To overcome the enormous number of redundant tweets, we keep distilled information in super-tweets. Experiments performed on a large scale data set from tweets sent during the Iranian Presidential Election (#IranElection) and the results demonstrate the efficiency and effectiveness of our framework.

ACM Reference Format:
Nazanin Dehghani and Masoud Asadpour, 2015. Graph-based Method for Summarized Storyline Generation in Twitter. ACM Trans. Knowl. Discov. Data. V, N, Article A (January YYYY), 19 pages.
DOI: 0000001.0000001

1. INTRODUCTION
There has been a surge of interests in the use of microblogging services during the last few years leading to a spectacular rise in their popularity. Myriad of users have been attracted to such services generating a massive amount of short texts in an unprecedented rate. Popularity of microblogging services is indebted to their large-scale participation through user-contributed short text.

Meanwhile, Twitter with more than 288 million monthly active users is the widely popular microblogging service provider. People tweet for several reasons from daily conversation to URL sharing and news reporting. Each tweet can start a thread of dialogues around a topic. Despite official news agencies and journalists who are late to react upon events, Twitter users can rapidly and easily report what is happening in front of their eyes. This makes them live reporter of the events, happening around the world. Considering the fact that Twitter users produce more than 500 million tweets a day, Twitter is now becoming an information plus social network. Nowadays, Twitter has become a leading source of cutting-edge information and a great medium for
exploring emerging events, breaking news and general topics which most matter to a broad audience.

From another standpoint, the surge of social networks brought down the barriers of broadcast. In Twitter, users can easily diffuse tweets via re-tweet (i.e. re-sending the tweet that has early been published by another user). The unprecedented volume and velocity of incoming information in Twitter and disseminating this information to a wider audience leads to information overload. So, users find themselves overwhelmed by the increasing amounts of data. Therefore, relevant information is often buried in an avalanche of data and processing which is difficult to follow if not possible.

Besides information sharing, users refer to Twitter for getting real-time information on actual events resulted in Twitter search engine responding to more than 2.1 billion query per day.

Whereas, a significant fraction of tweets are about news events, automatic methods are required to organize and summarize storyline of news events so that users can easily access to the relevant and key information hidden among tweets to draw high level conclusions.

Although, Twitter search results presents a simple list of relevant tweets in a chronological order and retrieve relevant parts of answer to user’s query, it is still incapable of providing a coherent picture of the whole answer. Furthermore, browsing and seeking important information in a list of relevant tweets is still confusing. On the other hand, it is difficult for users to express their needs in terms of a few keywords and hence, the retrieved tweets might not be useful. For these reasons, providing a whole picture that covers different angels of the query and relations among them, can serve user’s needs in a more appropriate way. We believe that, preventing users from experiencing information overload while keeping them informed of significant tweets is becoming one of the most challenging issues for Twitter in the near future.

However, heterogeneity of tweets, the limitation of 140 characters in their length and, high redundancy in Twitter stream, coupled with the messiness of tweets with wide variability in quality and writing style generated by a legion of Twitter users make providing a coherent picture of news events and the relationship among them in a well-organized manner quite challenging. Furthermore, proposed solutions should be designed efficiently to cope with the big size of data.

In this paper, we propose a graph-based method for creating a structured summary of events for a set of tweets. Our approach generates a concise set of salient phases related to the user’s query and provide summary for each phase of the events. Moreover, our method explicitly show the relations among sub-events in a way that captures story development. In other words, our approach not only tends to sketch a coherent picture of story development, but also attempts to reveal the relationships among phases of the event. We demonstrate that abstractions and cleaning heuristics mitigate the effect of noisy tweets and leads to cope with big datasets.

The overarching theme of this work is first mitigating the effect of redundant tweets by introducing the concept of super-tweet. Then, a temporal-similarity super-graph is constructed over super-tweets to model both text similarity and temporal order among tweets. To avoid noise and to keep our graph less polluted, we only maintain super-tweets with k-shell value greater than a threshold and discard other nodes with their edges. This threshold keeps super-tweets with high social support intuitively. Communities in residual graph are considered as different phases of the event as sub-events. To summarize each sub-event, the residual graph is extended to the tweet-user graph. Super-tweets with both lexical and social salience are added to the summary. The spanning arborescence of minimum weight in the sub-events graph portrays the storyline of sub-events. Our approach is implemented and tested on a dataset that we have collected in our lab from tweets sent during the Iranian Presidential Election.
Graph-based Method for Summarized Storyline Generation in Twitter

(#IranElection) and the results are reported. This tag was among the 10 top tags in Twitter in year 2009.

Specifically, the contribution of this paper can be listed as follows:

• We introduce the abstraction concept of super-tweet and keep distilled information in super-tweets to reduce the size of dataset and the constructed graph which leads the graph operations to be performed an order of magnitude faster.
• We propose a novel graph-based method for identifying important phases of the event. Sub-events with social support greater than a predefined threshold, $k$, are considered as important.
• We devise a novel graph-based technique to summarize tweets of each phase of the event which not only consider the content of tweets but also aggregate their social aspect.
• We propose a novel graph-based approach for storytelling. Our approach connect sub-events considering both time continuity and semantic similarity resulted in a coherent storyline.
• Our proposed method is examined and validated using experiments which demonstrate both the performance and applicability of them.

The rest of the paper is organized as follows: Section 2 discusses the related work. Section 3 is devoted to present the proposed framework for summarized storytelling. The experiments and results are brought in Section 4. Finally, Section 5 provides some concluding remarks and directions of future researches.

2. RELATED WORKS

In this section, we review the related works in two categories that best line up with our research: storyline generation and multi-tweet summarization.

2.1. Storyline Generation

When information is abundant on an evolving topic, typical summarization techniques alone are incapable of providing a comprehensive and picturesque summaries of the topic. Therefore, researchers turned their attention to study how stories organize and evolve in Twitter. This has opened a new research area on storyline generation (as known as storytelling). The goal of storyline generation is linking tweets into stories and finding meaningful connections among them.

Although, the problem of storytelling has recently been focused in Twitter, it was first formulated in [Kumar et al. 2008] as a generalization of re-description mining. Given a set of objects and a collection of subsets over these objects, they relate object sets that are disjoint and dissimilar by finding a chain of redescriptions between the sets. The concept was later developed as connecting the dots between documents in [Hossain et al. 2011; Hossain et al. 2012; Shahaf and Guestrin 2010]. Hessain et al. [Hossain et al. 2011] used the notions of distance threshold and clique size on a graph model to connect seemingly unrelated PubMed documents. They generalized their work in [Hossain et al. 2012] to construct stories in entity networks and proposed an optimization framework based on concept lattice mining.

Shahaf et al. [Shahaf and Guestrin 2010] focus on the news domain and proposed an algorithm to find coherent chain linking two given news articles together. They also incorporate user feedback to have refined and personalized stories. Moreover, they determined the influence between documents based on a bipartite graph model in which documents and words are nodes and edges weight are obtained by $tf - idf$ weighting schema. Shahaf et al. [Shahaf et al. 2012a; Shahaf et al. 2012b] proposed another noticeable idea in this direction by introducing the concept of metro map for creating structured summaries of information. Their method generates a concise set of doc-
documents maximizing coverage of salient pieces of information. Visualizing the story development in metro map shows the explicit relations among documents. Formulating the problem in an optimization framework provides a mean to study coverage and connectivity between maps. The coverage is then defined as $tf-idf$ values and connectivity between maps is calculated by the number of paths that intersect two maps.

In overall, the aforementioned methods heavily rely on the abundance of content to have acceptable textual reasoning, while in social media like Twitter, we face shortage of textual content. Therefore, in this work, we got help from social content besides textual content to construct storyline in Twitter.

A few studies have dealt with storyline construction in Twitter. In [Dos Santos Jr et al. 2013], a story is modeled as a graph of entities propagating through spatial regions in a temporal sequences. Hence, using spatio-temporal analysis on induced concept graphs, they proposed a method to automatically derive stories over linked entities in tweets. Another attempt for storytelling in Twitter was performed in [Lin et al. 2012]. The authors first model tweets in a multi-view graph having both temporal and similarity relations among tweets. Then, they select the most representative summary using the dominating set approximation. Finally, they apply the Steiner Tree which spans all dominating set members to generate a natural storyline capturing both temporal and structural information of tweets.

Different from these approaches, our method constructs a tree over sub-events by considering social, textual and temporal aspects of tweets. Another distinguished feature of our system is its capability of adding more details and side stories or tuning it in a very abstract level in terms of social support of sub-events. Moreover, the efficient storage and computational cost of our method makes it quite practical in large data sets of tweets for online applications.

2.2. Multi-tweet Summarization

While there has been studies and research conducted for years on the document summarization, tweet summarization is still in its infancy. On one hand, high volume of tweets with redundant and/or irrelevant information in user's timeline and search results emerged the need of tweet summarization. On the other hand, unique characteristics of tweets, from being short and informal to special markup language using hashtag or mention, makes direct use of document summarization techniques almost inefficient.

The concept of Multi-tweet Summarization was first introduced by Sharifi et al. [Sharifi et al. 2010] where they proposed Phrase Reinforcement Algorithm, however, their proposed method outputs a single tweet as a summary of a number of tweets. Later in [Inouye and Kalita 2011] they extended their work to generate multi-tweet summaries by first clustering tweets and then summarizing each cluster exploiting a hybrid $tf-idf$ weighting method.

Characterizing an event by extracting few tweets that best describe its principal subevents has attracted many research interests in recent years. In [Chakrabarti and Punera 2011] Chakrabarti and Punera introduced SUMMHMM in which Hidden Markov Models are trained from previous events to identify subevents based on tweet minimum activity threshold. However, their method has specifically been designed to summarize structured and recurring sport events. Summarizing sport events from tweet streams also drew Kubo et al.’s attention. They produce high quality summaries by exploiting good reporters [Kubo et al. 2013]. Having focused on real-time summarization of events, two papers [Nichols et al. 2012; Zubiaga et al. 2012] have considered sudden increase with respect to the recent tweeting activity and applied different weighting methods to select representative tweets for subevents.
There have also been studies on generating visual summaries of tweets. TweetMotif [O’Connor et al. 2010] groups tweets by frequent terms and characterizes a topic by tag clouds as its label and a set of tweets that contain the label. Marcus et al. [Marcus et al. 2011] also provides a visual summarization of tweets about user-specific events in which relevant tweets along with their geolocation are displayed on a map and positive or negative sentiment of selected topics are colored blue or red, respectively. Similar to Tweetmotif, their summaries are labeled by tag clouds.

Twitter, in addition to an information network, is also a social network and incorporating social aspects can make up for the information shortage in tweets. However, most research has been done on tweet summarization mainly deal with tweet’s text, and rarely pay attention to their social aspect.

In [Liu et al. 2012], XiaoHua et al. concentrate on utilizing social features to better find salient tweets from a given set of tweets. They incorporate social concepts of a tweet in terms of the number of re-tweets it received, number of followers its author has and the readability of tweet’s text. They also add user diversity to their method. However, through experiments we concluded that number of re-tweets, alone, is not good enough to represent the social attention a tweet has received, while a tweet can be re-tweeted by its author or a community of authors multiple times. In fact, number of distinct users that re-tweet a tweet better represents its social salience. On the other hand, the global number of followers a user has is not a good measure to rank users for specific contexts. One may have many followers but might have no active relationship with most of them in a given context.

In contrast, we propose a bipartite user-tweet graph to find social salient tweets and weight them accordingly. Then we select a set of socially salient tweets which cover the textual content of sub-events. Our approach consider those follower/followee relationships that are relevant to the topic. Moreover, the aforementioned endeavors store all the tweets, while we keep distilled information in super-tweets to reduce storage and computation cost.

### 3. OUR PROPOSED FRAMEWORK

Given an event query in terms of user defined keywords about an ongoing event in real world and a collection of relevant tweets, our goal is storytelling as a set of summaries of the principal phases of the event and their relations. The generated storyline has a tree structure where each node is labeled by a summary of an individual phase of the event and edges indicate causal relationship between phases.
A super-tweet $st_i$ is defined as a tuple $(tv_i, rt_i, a_i, date_i)$, where $tv_i$ is the selected tweet’s textual vector, $rt_i$ is the number of duplicated tweets or re-tweets it represents, $a_i$ is the list of unique authors who have posted (or reposted) this tweet and, $date_i$ is the posting time of main tweet.

Accordingly, tweet’s text is preprocessed and represented as a textual vector with the $tf - idf$ weighting scheme. In preprocessing, mentions (i.e. terms like @username) and stop-words are removed and tweets with less than two terms are filtered out. In order to identify redundant tweets, we use the cosine similarity measure.
as a metric and tweets with similarities higher than a predefined threshold $\alpha$ are considered as duplicated tweets or re-tweets. Then, we pick the tweet with the earliest posting time as the representative and discard the duplicates. Afterward, super-tweets are indexed. The next steps of storyline generation work with this index.

It is important to note that the procedure of filtering redundant tweets and compressing their information into super-tweets can be performed off-line. Furthermore, this procedure reduces the size of index dramatically and makes the construction and process of temporal similarity graph quite fast. In fact, our method keeps distilled information in super-tweets to reduce storage and computation cost. The data structure of super-tweets can incrementally be updated by entering new relevant tweets. A newcomer tweet is first checked to determine whether it is duplicate/re-tweet or it is a novel tweet by measuring its cosine similarity to the previously known super-tweets. In case of duplicated tweet, its super-tweet's information is updated and then the newcomer tweet is discarded. Otherwise a new tuple is generated and its fields are initiated accordingly. We also maintain the similarity scores among super-tweets in a two-dimensional matrix called super-tweet similarity matrix. Entering a new super-tweet adds one row to the similarity matrix.

Note that, when user requests the storyline of an event, relevant tweets to the user's query are first retrieved using a retrieval method. Our challenge is to generate the storyline from hundreds of thousands of relevant tweets. Discussion about the retrieval methods is out of the scope of this paper.

3.2. Temporal-Similarity Super-graph Construction Module

This module is dedicated to construction of a temporal similarity super-graph over super-tweets for the purpose of storing semantic and temporal information among super-tweets. Having super-tweets, in this step, we construct a weighted temporal similarity super-graph as follows.

Definition 3.2. A temporal-similarity super-graph $G = (V, W, E)$, where $V$ is a set of nodes which represent super-tweets, $W$ specifies the number of retweets, and $E$ is a set of directed edges, which represents the similarities between tweets. The direction of the edge is from the earlier super-tweet to the later one according to their start time of posting.

This way, both similarity and temporal information is encoded in the graph. This means, there exist an edge from super-tweet $st_i$ to $st_j$ if and only if their texts similarity is greater than a threshold $\beta$ and the first tweet in $st_i$ has been posted earlier than the first tweet in $st_j$.

3.3. Sub-event Detection Module

After construction of the temporal-similarity super-graph, we should identify different phases of the event. In the sub-event detection module, the general idea is to perform a community detection method and consider each community as a sub-event. Although we discard redundant tweets, the graph is still too noisy and applying community detection may produce inaccurate results. Our method overcome this issue by pruning nodes with $k$-shell value less than a threshold $K$.

3.3.1. $k$-Shell Decomposition. Among many centrality measures calculating the importance of nodes in a graph, degree centrality and clustering coefficient [Wasserman 1994] can only characterize the local information about nodes. Computational complexity of betweenness centrality [Freeman 1977] is very high due to the need
Fig. 3. Virtual degree calculation according to super-tweet’s weight and degree in super-graph

to calculate the shortest paths. In 2010, Stanley et al. [Stanley and Makse 2010] introduced the $k$-shell decomposition method pointing out that nodes with large $k$-shell values are very important nodes that constitute the core of the network. A $k$-shell is a maximal connected subgraph in which every node’s degree is at least $k$. The maximality indicates that the node belongs to a $k$-shell but not to any $(k + 1)$-shell. This method is often used to identify the core and periphery of graphs; the shells with higher indices lie in the core. It is efficiently implemented in $O(m)$, where $m$ is the number of edges.

As we seek important super-tweets, we can keep super-tweets with large $k$-shell values located in the core and discard the others. However, since in super-graph, each node is a super-tweet which represents a set of tweets, we propose a variant of $k$-shell decomposition for weighted graphs to be applied to the similarity super-graph. As mentioned earlier, the reliability of our method is based on the social hypothesis that popular events attract attention of many people and so they are re-tweeted more than ordinary tweets. On the other hand, they are more likely reported in different narrations while having the same connotations. The $k$-shell structure serves the purpose of identifying significant sub-events with $k$ serving as a social support threshold.

From another standpoint, $k$-shell structure empower us to provide a hierarchical storyline in which the granularity can be tuned by the value of $k$ i.e. starting from the maximum $k$-shell in the graph and gradually decreasing the value of $k$ results in more detailed events or side stories, gradually added to the storyline.

Applying the $k$-shell structure can also accelerate the procedure of community detection in our graph. Considering the fact that for those news events that last for long time, similarity graph might have millions of super-tweets; Performing community detection to extract its sub-events may take several hours. However, in our proposed method, performance of the community detection procedure on the extracted $k$-shells is not only fast but also more accurate.

To apply a $k$-shell decomposition method on our weighted super-graph, one can imagine each super-tweet $st_i$ is a representative of a clique of size $rt_i$, since in the similarity graph, there should be an edge between a tweet and its re-tweets. Therefore, we set a virtual degree to each super-node according to the following equation:
Graph-based Method for Summarized Storyline Generation in Twitter

VirtualDegree(st_i) = \begin{cases} w_i - 1 + \sum_{j \in \text{nb}(i)} w_j & \text{degree}(i) > 0 \\ w_i - 1 & \text{degree}(i) = 0 \end{cases} \quad (1)

Figure 3 provides an example to show the logic behind the (1). By substituting virtual degree for degree in k-shell decomposition proposed in [Stanley and Makse 2010], k-shell value of super-tweets are obtained. Afterward, we first filter super-nodes with k-shell score lower than a threshold K. This technique cleans the graph by deleting noisy super-tweets with less social attention and hence leads important phases of the event come into sight.

Then, a community detection method is employed on the remained k-shells to aggregate the related news into the same story. Among several community detection methods, we applied the Infomap multi-level partitioning algorithm of Rosvall and Bergstrom [Rosvall and Bergstrom 2007]. In [Lancichinetti and Fortunato 2009], several methods are evaluated and Infomap is reported as the most accurate method. Infomap is specifically proposed for detecting communities in large networks. Moreover, Infomap gives us a hierarchical partitioning that enable us zoom in sub-events and have more details on a sub-event.

3.4. Event Summarization Module

After mapping detected communities to sub-events, its time to provide summaries of each sub-event and use this summary as a label for nodes in the storyline tree. The event summarization task is to extract a subset of super-tweets that best cover important aspects of the event. From social viewpoint, a tweet is important if it has been posted / re-tweeted by many important users. This encourages us to pick social salient tweets covering the content. In order to generate summaries that not only cover more content but also select among social salient tweets, we modeled our super-tweets as a bipartite graph of super-tweets and users:

Definition 3.3. A tweet-user bipartite graph $G = (V, E, F)$, where the vertices $V = V_{ST} \cup V_U$ correspond to super-tweets and users, and $E$ is a set of directed edges from users to their posted super-tweets, and $F$ is a set of directed edges from followers of a user to the user.

Needless to say that, a tweet which re-tweeted by many different users is socially more important than a tweet re-tweeted repeatedly many times by a unique user. In the next step, we perform the Hyperlink-Induced Topic Search (HITS) (also known as Hubs and authorities) algorithm on the tweet-user bipartite graph to assign weight to super-tweets. Unlike Pagerank, HITS makes a distinction between “authorities” and “hubs” where in our case, super-tweets play the role of authorities and users play the role of hubs in the tweet-user bipartite graph. In fact, using social network of users, user’s hub value estimates the value of its links while tweet’s authority estimates the social importance of a tweet.

Then, we apply a minimum weighted dominating set on super-tweets similarity graph of each event to have the most social salient super-tweets that cover the whole content of the sub-event.

3.4.1. Minimum-Weight Dominating Set Problem. A subset of vertices of a graph is called a dominating set if every vertex in the graph is contained in the dominating set or has a neighbor in it. Every vertex in dominating set, dominates itself and all its neighbors. The minimum dominating set problem is to find a dominating set of smallest size. If every vertex of the input graph is associated with a weight, the minimum-weight dominating set problem computes the dominating set of minimum weight. Although the minimum-weight dominating set problem is known to
ALGORITHM 1: Sub-events Summarization

**Input:** $G = (V, W, E, F, H)$ where the vertices $V = V_{ST} \cup V_U$ correspond to super-tweets and users, and $W$ is the weights of $V_{ST}$. $E$ is a set of directed edges from users to their posted super-tweets, $F$ is a set of directed edges from followers of a user to the user, and $H$ is a set of undirected edges, which represents the similarities between $V_{ST}$.

SubeventsSet which represents a set of sub-events where each sub-event has a list of its super-tweets.

**Output:** The summary of each sub-event ($Summary_{SE}$) gets assigned by dominant super-tweets.

$$W = \emptyset;$$
$$G' = G.subgraph(V, E, F);$$
$$\langle hub, authority \rangle = G'.HITS();$$

for each node $st$ in $V_{ST}$ do

$$w(st) = 1 - \sum_{i \in V_{ST}} \text{authority}(i);$$

end

for each sub-event $SE$ in SubeventsSet do

$$G''(V'', W'', H'') = G.subgraph(\text{V}_{st}, W, H) //corresponds to the subgraph of super-tweets in SE and their similarity edges ;$$

$Summary_{SE} = \emptyset ; T = \emptyset ;$

while $|Summary_{SE}| < W''$ and $Summary_{SE} \neq V''$ do

for $v \in V'' - Summary_{SE}$ do

$$s(v) = ||\{v' | (v', v) \in H''\}|T||;$$

$$v^* = \arg \min_{v' \in V''} \frac{w'(v)}{s(v')};$$

$$Summary_{SE} = Summary_{SE} \cup \{v^*\};$$

$$T = T \cup \{(v'', v^*) | (v'', v^*) \in H''\};$$

end

end

end

be NP-hard [Raz and Safra 1997], a greedy algorithm for an approximate solution is provided in [Shen and Li 2010].

Unlike LexRank [Erkan and Radev 2004] that only considers lexical centrality as salience, applying HITS in tweet-user bipartite graph takes social centrality into consideration and minimum weighted dominating set covers the lexical content. The generated summary is with minimal redundancy since the set is of minimum size.

It is important to note that using number of followers or global PageRank of tweet’s author as user’s rank may lead to unfair results. In fact, when we generate the storyline of a specific query in a specific topic, it is important to rank users in that context not globally. This rank can be helpful in ranking tweets’ authors which will estimate the user influence (or a kind of expertness) in that context.

Algorithm 1 demonstrates the steps of summarizing sub-events. This algorithm takes an input graph $G = (V, W, E, F, H)$ where the vertices $V = V_{ST} \cup V_U$ correspond to super-tweets of high $K-shell$ values from Sub-event Detection Module and users, and $W$ is the weights of $V_{ST}$ that is assigned by one minus normalized authority score of super-tweets. $E$ is a set of directed edges from users to their posted super-tweets, $F$ is a set of directed edges from followers of a user to the user, and $H$ is a set of undirected edges, which represents the similarities between $V_{ST}$. It also takes the detected sub-events with their members. The algorithm returns the summary of each sub-event as a set of dominant super-tweets.
3.5. Storyline Generation Module

Given a set of sub-events, we organize them in a structure that tells the story among them. We chose the tree structure so the first event is located at the root and the story evolves in different levels of the tree. To generate the storyline among sub-events, we first construct a weighted directed sub-event graph, where nodes represent sub-events and edges' weight represent the semantic distance between sub-events. The direction of the edges is from the earlier sub-event to the later one. In construction of the sub-event graph over the temporal similarity super-graph, we add one node per community. Then, according to the ratio of edges from community \( C \) to community \( C' \) and vise versa, we determine the direction of the edge between the nodes that represents \( C \) and \( C' \). We calculate the distance between sub-events \( C \) and \( C' \) by calculating the average similarity between any pair of super-tweets:

\[
\text{distance} (C, C') = 1 - \frac{\sum_{st_i \in C} \sum_{st_j \in C'} \text{sim} (st_i, st_j)}{|C| \times |C'|}.
\]

(2)

Then, we apply the Edmonds’ algorithm [Edmonds 1967] for finding a spanning arborescence of minimum weight in the sub-events graph which is analogous to the minimum spanning tree in directed graphs. This algorithm can be performed in \( O(n^2) \) where \( n \) is the number of sub-events. The obtained spanning tree, connects sub-events considering both time continuity and semantic similarity and result in a coherent storyline that reveals the temporal structure of the stories as they evolve.

4. EXPERIMENTS AND RESULTS

We evaluate the performance of our proposed framework through several experiments. In the following, we first introduce the data set we have used and then present the experiments and evaluations on phase detection, storyline generation and summarization.

4.1. Dataset

Our experiments are conducted on a dataset that we have crawled using Twitter API in our lab from tweets sent during the Iranian Presidential Election (

\( \# \)IranElection). This tag was among the 10 top tags in Twitter in year 2009 and it was historically important as it was the first time that a social network was being used for broadcast and coordination of a social movement. The corpus is comprised of more than one year tweets from 1st January 2009 until 8th August 2010. The corpus is Bilingual, including English and Persian and so on. More details of the collection are illustrated in Table I. For our experiments, non-English tweets and English Tweets containing less than two words are filtered. The first 200,000 English tweets are chosen for development and the remainder for testing. We treat \( \alpha, \beta \) and minimum \( K - Shell \) as systematic parameters, whose value are experimentally determined on the development dataset.

| Table I. Statistics of \#IranElection Dataset |
|-----------------------------------------------|
| Number of tweets: 4398692                   |
| Number of English tweets: 1868281           |
| Number of users: 6691                      |
| Median English Tweet Length: 17            |
| Average English Tweet Length: 16.42        |
| Median Pre-processed English Tweet Length: 10|
| Average Pre-processed English Tweet Length: 9.81|

ACM Transactions on Knowledge Discovery from Data, Vol. V, No. N, Article A, Publication date: January YYYY.
This dataset is ideal for the task of storyline summarization since it is on a developing event containing many sub-events and side stories well reflected in Twitter. Moreover, there is currently no gold-standard dataset for the task of storyline construction and summarization available and hence, we should manually annotate a dataset for automatic evaluation of our task. Therefore, we have used this dataset that we are well aware of its events and their relations so that we can evaluate the subjective parts of our experiments through human evaluation.

4.2. Redundant-tweet Detection

In order to construct the data structure of super-tweets, we need to identify re-tweets/duplicated tweets. In our dataset, out of over one million tweets, only 0.28\% of duplicated tweets are explicit re-tweets. Calculating the cosine similarity between tweets and considering those with similarity greater than $\alpha$ as duplicated tweets, the following Precision and Recall are obtained and illustrated in Figure 4.

While larger value of Precision leads us to a more accurate phase detection, large value of Recall results in higher compression ratio and reduces the computational time and space costs.

It is important to note that in our task, high Precision is more important than high Recall. Since, considering some tweets mistakenly as duplicated tweets leads to regard their super-tweet as an important sub-event while it is not so important and hence, the precision of the whole method will be ruined. However, if we miss some duplicated tweets and assign them to different super-tweets, in the similarity super-tweet graph, there will be an edge between their super-tweets and their k-shell should be high enough to be considered as an important sub-event. Setting the value of $\alpha$ to 0.9 resulted in the best performance on the development dataset and hence it is used for the remaining experiments. This leads to 387173 pair of duplicated tweets, encapsulating them resulted in 194892 super-tweets which reduces the size of dataset to 18\% percent. As noted earlier, since our focus is on news scope, tweets received no re-tweet or no duplication are discarded.
4.3. Graph Construction and Evaluation of Event Detection

Afterward, we construct temporal-similarity super-graph with different $\beta$ thresholds for similarity between super-tweets. There is a trade off to identify the best value of $\beta$. Smaller values of $\beta$ leads to have more edges between communities and hence detecting sub-events are less accurate, while, larger values of similarity threshold leads to miss the relations between sub-events. We also examine well-known retrieval metrics as distance measure besides cosine similarity. Experimental results show that although cosine similarity works well in identifying duplicate tweets, retrieval methods are better in detecting super-tweets talking about same topic. Therefor, we construct an index of super-tweets in Lucene\(^2\) and then search for similar super-tweets with “MoreLikeThis” similarity queries. The temporal-similarity super-graph is then constructed by drawing an edge between similar super-tweets with proportional Lucene score greater than $\beta$. The value of $\beta$ is experimentally set to 0.5, which yields the best performance on the development dataset.

As explained earlier, we set the number of duplicated tweets of a super-tweet as its weight. However, as shown in Figure 5, the number of duplications is an inaccurate overestimation of the number of unique users who share the super-tweets. Therefore, we set the size of a super-tweet’s authors list ($\text{Authors}_{i}.\text{size}()$) as its weight.

We apply our variation of k-shell decomposition method on the constructed graph to prune the graph and remove noise from it. Figure 6 demonstrates the number of nodes and edges remained after discarding nodes (and their relative edges) with k-shells less than $K$ threshold. Note that keeping those super-tweets with high k-shell values resulted in sub-events which attracts many social attention. Furthermore, k-shell of nodes follows the Power-law distribution with $\alpha = 2.78(3)$ and thats why we can clearly identify important sub-events.

4.4. Sub-Event Detection

Then, we apply Infomap method to reveal community structure in similarity graph and consider each community as a sub-event. The structure of communities in our dataset is shown on Figure 7. This figure clearly illustrates that nodes with large

\(^2\)http://lucene.apache.org

Fig. 5. Number of re-tweets is not a good estimator of unique users posting the tweet.
Fig. 6. Number of nodes and edges remained after discarding nodes with k-shell values less than a threshold.

Fig. 7. The community structure of sub-events of Iranelection in Twitter.
Fig. 8. Super-tweets talking about different aspects of a sub-event are grouped together in a community. k-shell values in dense communities are well located in the core of the graph. Furthermore, community detection method groups super-tweets talking about different aspects of a sub-event pretty well as shown in Figure 8. The graph visualizations are based on Gephi toolkit 3.

Figure 9 lists top 10 detected sub-events for the event of Iran Election with their social support degree.

| Sub-event                                                                 | Social Support |
|---------------------------------------------------------------------------|----------------|
| Show support for democracy in Iran, add green overlay to your Twitter avatar with 1-click http://helpiranelection.com http://iran.greenthumbnails.com | 2166           |
| Tehran is now alive with the sound of the people - Allah Akbar - Death to the Dictator. Ghazvin and Qom has gone green too with Allah-D-Akbar. Three videos of Allah o Akbar chants: http://bit.ly/13Y85x | 307            |
| Google Earth to update satellite images of Tehran http://twitition.com/c3eoa | 119            |
| Saturday Sea of Green rally with Quran in your hands- Enghelab Sq - 4pm - Mousavi Karoubi and Khatami will attend | 105            |
| United 4 Iran a global day of action July 25 2009 http://united4iran.org/ | 102            |
| Neither Shah nor Mullah but a free democratic secular Iran is what these young ppl are dying 4. We Dream of a Free Iran: Interview with a Witness to Protest http://tinyurl.com/2g2xmpy #neda | 99             |
| Protest today Tehran 4PM Half Tir Sq-Meydan 7 Tir - candle vigil sitting on floor in memory of our martyrs | 96             |
| Australian Embassy accepting injured: No 13  233rd Street Khalid Isalmubi Ave - Telephone+98 21 8872 4456 | 91             |
| hospital source - please give blood - severe shortage of blood supplies accross ALL Iran hospitals - pls donate blood | 88             |
| Security forces fired tear gas at protesters in Imam Hossein sq and in front of IR TV (IRIB). Tear gas & pepper spray first aid: http://tinyurl.com/mxar80 | 85             |

Fig. 9. Top 10 sub-events of Iran election with their social support in terms of minimum k-shell.

3https://gephi.org/toolkit

ACM Transactions on Knowledge Discovery from Data, Vol. V, No. N, Article A, Publication date: January YYYY.
For evaluating the performance of sub-event detection, the page dedicated to *Timeline of the 2009 Iranian election protests*[^4] on Wikipedia was used as a gold-standard.

We extract the sub-events discussed in the Wikipedia page and evaluate our sub-event detection procedure. To evaluate our sub-event detection procedure, once, we index Wikipedia events and search for super-tweets that report those events. This way we find out which events reported in Wikipedia, are reported by our method as important even (i.e. Recall). Figure 10 reports the number of sub-events reported by Wikipedia and our method. Results reveals that, our method missed only 3% of sub-events reported in Wikipedia while 54% of sub-events that well reflected in Twitter and received high social attention (minimum of 20 k-shell) have not been reported in the human-made wikipedia page.

4.5. Sub-Event Relations

Afterward, we construct the sub-events graph from temporal-similarity super graph and detected sub-events from Infomap community detection. Note that it is a complete graph with number of sub-events nodes. Then, we found a spanning arborescence of minimum weight in this graph. The result for the graph of sub-events with k-shell scores of higher than 50 is demonstrated in Figure 11. Sub-events of one branch is provided for example to show the general trend of developing topics. Since, currently there is no standard evaluation platform and criteria for automatic storytelling and since storylines told by different persons can be quite different, we evaluate the results of our storytelling method by human experts.

4.6. Summarization Performance

The last part is providing a summary for each sub-event. In order to evaluate the summarization approach, three annotators were involved. For each sub-event, they independently selected tweets that best described the topic from the related tweets, thus forming the gold-standard dataset. For any topic \( i \), and any pair of annotators \( X \) and \( Y \), the inter-agreement of their selected summaries are calculated using \[ \frac{|X \cap Y|}{|X \cup Y|} \] proposed in [Liu et al. 2012]. The average inter-agreement over all topics was 0.81. We applied the well-known ROUGE toolkit [Lin 2004] as an automatic metric for summarization evaluation. ROUGE-1 and ROUGE-2 are used which

[^4]: http://en.wikipedia.org/wiki/Timeline_of_the_2009_Iranian_election_protests

Fig. 10. Comparing the number of sub-events covered on Wikipedia and our proposed method with minimum of 20 k-shell value.
calculate the overlap in unigrams and bigrams between the system and the three model summaries. Table II reports a significant improvement in performance of our weighted dominating set method.

Table II. Performance of three summarization methods.

| Method               | ROUGE-1 | ROUGE-2 |
|----------------------|---------|---------|
| LexRank              | 0.4501  | 0.3821  |
| DominatingSet        | 0.5654  | 0.4793  |
| WeightedDominatingSet| 0.8239  | 0.5476  |

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel graph-based framework for storytelling in Twitter. We utilized social aspects beside textual information of tweets to generate a storyline from social point of view. We modeled significant sub-events as communities in similarity graph. Then, we summarized sub-events by solving the approximation of minimum-weight dominating set to select social salient tweets. Then, we modeled a storyline as a directed tree of sub-events evolving over time. To overcome the enormous number of redundant tweets, we kept distilled information in super-tweets. Comprehensive experiments that were conducted on a dataset of a real-world event hugely reported in Twitter showed the good performance of our method. For future we would like to work on long texts e.g. news articles. We would like also to devise methods for automatic parameter selection rather than trial and error. Our next goal is to enhance the performance of our method to support online queries.

REFERENCES

Deepayan Chakrabarti and Kunal Punera. 2011. Event Summarization Using Tweets. ICWSM 11 (2011), 66–73.

Raimundo F. Dos Santos Jr, Sumit Shah, Feng Chen, Arnold Boedihardjo, Patrick Butler, Chang-Tien Lu, and Naren Ramakrishnan. 2013. Spatio-Temporal Storytelling on Twitter. (2013).

Jack Edmonds. 1967. Optimum branchings. Journal of Research of the National Bureau of Standards B 71, 4 (1967), 233–240.

Günes Erkan and Dragomir R Radev. 2004. LexRank: graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research (2004), 457–479.
Linton C Freeman. 1977. A set of measures of centrality based on betweenness. Sociometry (1977), 35–41.

M Shahriar Hossain, Christopher Andrews, Naren Ramakrishnan, and Chris North. 2011. Helping Intelligence Analysts Make Connections. In Scalable Integration of Analytics and Visualization.

M Shahriar Hossain, Patrick Butler, Arnold P Boedihardjo, and Naren Ramakrishnan. 2012. Storytelling in entity networks to support intelligence analysts. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 1375–1383.

David Inouye and Jugal K Kalita. 2011. Comparing twitter summarization algorithms for multiple post summaries. In Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on. IEEE, 298–306.

Mitsumasa Kubo, Ryohi Sasano, Hiroya Takamura, and Manabu Okumura. 2013. Generating live sports updates from twitter by finding good reporters. In Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on, Vol. 1. IEEE, 527–534.

Deept Kumar, Naren Ramakrishnan, Richard F Helm, and Malcolm Potts. 2008. Algorithms for storytelling. Knowledge and Data Engineering, IEEE Transactions on 20, 6 (2008), 736–751.

Andrea Lancichinetti and Santo Fortunato. 2009. Community detection algorithms: a comparative analysis. Physical review E 80, 5 (2009), 056117.

Chen Lin, Chun Lin, Jingxuan Li, Dingding Wang, Yang Chen, and Tao Li. 2012. Generating event storylines from microblogs. In Proceedings of the 21st ACM international conference on Information and knowledge management. ACM, 175–184.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Text Summarization Branches Out: Proceedings of the ACL-04 Workshop. 74–81.

Xiaohua Liu, Yitong Li, Furu Wei, and Ming Zhou. 2012. Graph-Based Multi-Tweet Summarization using Social Signals. In COLING. 1699–1714.

Adam Marcus, Michael S Bernstein, Osama Badar, David R Karger, Samuel Madden, and Robert C Miller. 2011. Twitinfo: aggregating and visualizing microblogs for event exploration. In Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 227–236.

Jeffrey Nichols, Jalal Mahmud, and Clemens Drews. 2012. Summarizing sporting events using twitter. In Proceedings of the 2012 ACM international conference on Intelligent User Interfaces. ACM, 189–198.

Brendan O’Connor, Michel Krieger, and David Ahn. 2010. TweetMotif: Exploratory Search and Topic Summarization for Twitter. In ICWSM.

Ran Raz and Shmuel Safra. 1997. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In Proceedings of the twenty-ninth annual ACM symposium on Theory of computing. ACM, 475–484.

M Rosvall and CT Bergstrom. 2007. Maps of information flow reveal community structure in complex networks. arXiv preprint physics.soc-ph/0707.0609 (2007).

Dafna Shahaf and Carlos Guestrin. 2010. Connecting the dots between news articles. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 623–632.

Dafna Shahaf, Carlos Guestrin, and Eric Horvitz. 2012a. Metro maps of science. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 1122–1130.

Dafna Shahaf, Carlos Guestrin, and Eric Horvitz. 2012b. Trains of thought: Generating information maps. In Proceedings of the 21st international conference on World Wide Web. ACM, 899–908.

Beaux Sharifi, Mark-Anthony Hutton, and Jugul Kalita. 2010. Summarizing microblogs automatically. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 685–688.

Chao Shen and Tao Li. 2010. Multi-document summarization via the minimum dominating set. In Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics, 984–992.

H Eugene Stanley and Hernán A Makse. 2010. Identification of influential spreaders in complex networks. (2010).

Stanley Wasserman. 1994. Social network analysis: Methods and applications. Vol. 8. Cambridge university press.
Arkaitz Zubiaga, Damiano Spina, Enrique Amigó, and Julio Gonzalo. 2012. Towards real-time summarization of scheduled events from twitter streams. In Proceedings of the 23rd ACM conference on Hypertext and social media. ACM, 319–320.