Check for updates

Research Paper

JIS

Journal of Information Science
1–18
© The Author(s) 2018
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0165551518775304
journals.sagepub.com/home/jis
SAGE

SGSG: Semantic graph-based storyline generation in Twitter

Nazanin Dehghani
Social Networks Laboratory, School of Electrical & Computer Engineering, University of Tehran, Iran

Masoud Asadpour
Social Networks Laboratory, School of Electrical & Computer Engineering, University of Tehran, Iran

Abstract
Twitter is a popular microblogging service that has become a great medium for exploring emerging events and breaking news. Unfortunately, the explosive rate of information entering Twitter makes the users experience information overload. Since a great deal of tweets revolve around news events, summarising the storyline of these events can be advantageous to users, allowing them to conveniently access relevant and key information scattered over numerous tweets and, consequently, draw concise conclusions. A storyline shows the evolution of a story through time and sketches the correlations among its significant events. In this article, we propose a novel framework for generating a storyline of news events from a social point of view. Utilising powerful concepts from graph theory, we identify the significant events, summarise them and generate a coherent storyline of their evolution with reasonable computational cost for large datasets. Our approach models a storyline as a directed tree of socially salient events evolving over time in which nodes represent main events and edges capture the semantic relations between related events. We evaluate our proposed method against human-generated storylines, as well as the previous state-of-the-art storyline generation algorithm, on two large-scale datasets, one consisting of English tweets and the other one consisting of Persian tweets. We find that the results of our method are superior to the previous best algorithm and can be comparable with human-generated storylines.

Keywords
Multi-tweet summarisation; storyline generation; Twitter

1. Introduction

Twitter is currently the most popular microblogging service provider with more than 313 million monthly active users generating more than 500 million tweets a day.[1] Twitter users have been generating a massive amount of short texts at an unprecedented rate which are rapidly diffused through the Twitter follow graph. The Twitter follow graph exhibits structural characteristics of both information and social networks [1]. Unlike official news agencies and journalists who are late to react to unfolding events, Twitter users can quickly respond and report what is happening in front of their eyes. This makes them live reporters of the events happening around the globe. However, the unprecedented volume and velocity of incoming information on Twitter leads to information overload which in turn causes users to feel overwhelmed and confused.

Multi-tweet summarisation (MTS) techniques help a lot to generate a description for a specific event, but their output has no structure, that is, 'zero-dimensional'. When information is abundant on an evolving topic, MTS techniques are inadequate for providing a comprehensive and picturesque summary of the topic and fail to utilise trans-temporal characteristics among isolated events [2]. The appearance of Timeline [3, 4] brings about a 'one-dimensional' visual progress for browsing the event evolution in chronological order. Most of the existing tweet summarisation systems with structured output have focused on timeline generation [5]. Timeline summarisation is only adequate for simple stories which are inherently linear. However, complex stories usually branch into storylines, intertwining narratives and side stories.

Corresponding author:
Nazanin Dehghani, Social Networks Laboratory, School of Electrical & Computer Engineering, University of Tehran, Tehran 14395-515, Iran.
Email: ndehghany@ut.ac.ir

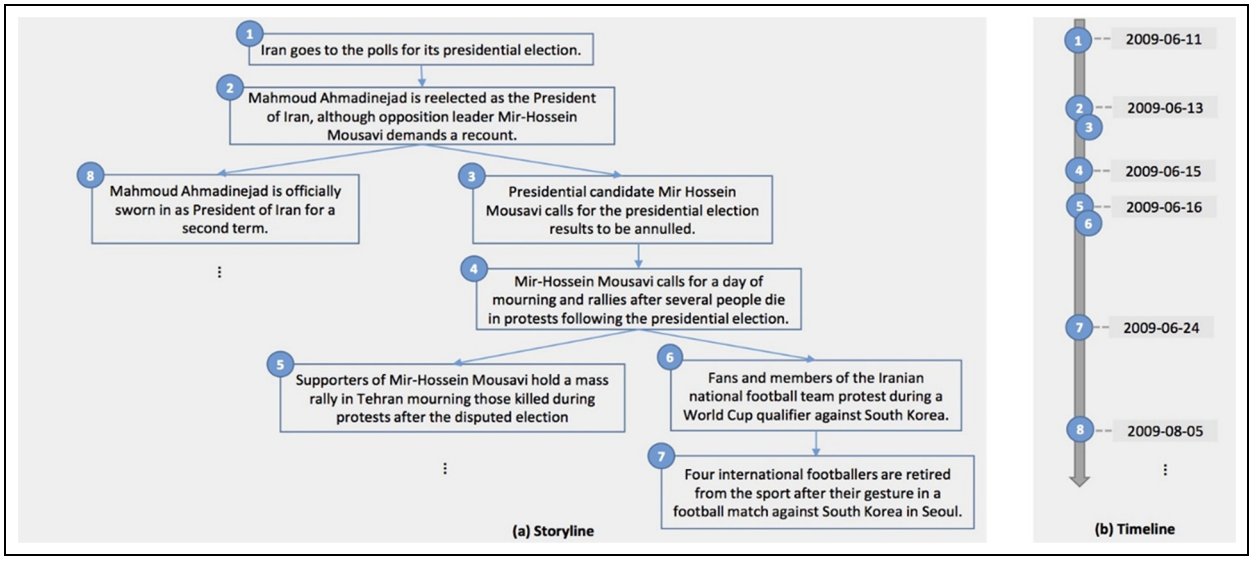

Figure 1. (a) An example of the storyline of the 2009 Iranian presidential election that shows how this story evolves over time through capturing the semantic relations between events. (b) The corresponding timeline of 2009 Iranian presidential election which only shows the temporal order of events.

Therefore, researchers have turned their attention to study how the structure of stories evolves on Twitter [6–8]. This has opened a new research area on a 'two-dimensional' storyline generation. A storyline is a chain of events that characterise a certain aspect of a news topic [6].

Storylines can summarise evolving stories as a series of individual but correlated events and offer an option to understand the big picture of this evolution. A storyline is similar to a timeline in terms of showing the temporal order of relevant events, but additionally represents the semantic relations among them. Figure 1 shows an example of storyline representation versus timeline representation for part of the story of 2009 Iranian presidential election.

Although storyline generation is now a hot and trending topic and there are numerous works which generate storylines from news articles [9–11], storyline generation from social media like Twitter is more complex and is still in infancy [6]. There are many challenging issues when it comes to generating storylines on Twitter. In contrast with well-written and edited news releases, tweets have the limitation of 140 characters in their length[2] and variability of writing styles. Therefore, there are large amounts of convoluted tweets with informal and abbreviated words coupled with spelling and grammatical errors. In addition, we have large amounts of unrelated posts about personal conversations [12] and high redundancy in the Twitter stream. Such data sparseness, lack of context and diversity of vocabulary make the storyline generation techniques for news articles less suitable for tweets. Furthermore, proposed solutions should be designed efficiently in order to scale to large-scale Twitter datasets. Our goal is to generate storylines from tweets in a way that can overcome the aforementioned challenges. In this article, we propose a semantic graph-based storyline generation (SGSG) method which identifies the main events of the story and arranges them in a tree to represent its evolution.

Our proposed method addresses the aforementioned challenges as follows: First, to cope with high redundancy in the Twitter stream, our approach filters out duplicate tweets and just keep essential information in super-tweets. Second, to consider the novel language use and writing standards within tweets, we calculate semantic similarity between tweets using word mover's distance (WMD) [13], which is based on word vectors pre-trained on large Twitter datasets[3] in which, for instance, vectors of 'ppl' and 'people' are significantly close. WMD also mitigates the shortage of textual content in tweets by incorporating the external word correlations knowledge provided by word embeddings. Third, to filter out personal tweets, irrelevant outside of one's social circle, and keep tweets about news events, we utilise the k-shell decomposition method [14] to keep super-tweets that attract more social attention and also take the author's social influence into account to generate the most general summary for each event. Finally, we employ locality-sensitive hashing (LSH) [15] to find near-duplicate tweets in linear order and then apply powerful concepts in graph theory that can efficiently generate the storylines on Twitter scaling well to large datasets. The SGSG framework is hierarchical from two points of view: social granularity and structural granularity. The social granularity can be tuned through the value of $\theta_{k-shell}$ parameter to add more detailed events or side stories to the storyline. The structural granularity comes from hierarchical community detection that provides events in different resolution levels.

Specifically, the contributions of this article can be listed as follows:

- Introducing an abstraction concept of super-tweet which is created in a sub-linear order to avoid redundant tweets and keep distilled information in super-tweets which leads the graph operations to be performed faster.
- Proposing a novel graph-based method for identifying the main events of the story. Events with a social support greater than a predefined threshold $\theta_{k-shell}$ are considered important.
- Devising a novel graph-based technique to summarise tweets of each event which not only considers the content of tweets but also aggregates their social aspect.
- Proposing a novel graph-based approach for storytelling which connects events considering both time continuity and semantic similarity resulting in a coherent storyline.
- Evaluating on both English and non-English datasets of tweets. Since nowadays the percentage of English-speaking users in Twitter is not as high as before [16], the solution can work well on non-English tweets as well and is tested on a Persian dataset.
- Illustrating both the performance and applicability of the overall method and its modules in details through several experiments.

The rest of this article is organised as follows: section 2 discusses the related works. Section 3 is devoted to presenting the proposed framework for summarised storytelling. The experiments and results are brought up in section 4. Finally, section 5 provides some concluding remarks and directions for future research.

2. Related works

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

2.1. Storyline generation

Although, the problem of storytelling has recently been focused on in Twitter, it was first formulated in Kumar et al. [17] as a generalisation of redescription mining. Given a set of objects and a collection of subsets over these objects, they relate the 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 Shahaf and Guestrin [10] and Hossain et al. [18, 19]. Hossain et al. [18] used the notions of distance threshold and clique size on a graph model to connect seemingly unrelated PubMed documents. They generalised their work in Hossain et al. [19] to construct stories in entity networks and proposed an optimisation framework.

Shahaf and Guestrin [10] 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 personalised stories. Moreover, they determined the interinfluence between documents using a bipartite graph model in which documents and words are nodes and weights of the edges are obtained by term frequency–inverse document frequency (tf-idf) weighting schema. Shahaf et al. [9, 20] proposed another notable idea in this direction by introducing the concept of metro map for creating structured summaries of information. Their method generates a concise set of documents maximising coverage of main pieces of information. Visualising the story development in metro map shows the explicit relations among documents. Formulating the problem in an optimisation 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. Recently, Song et al. [21] used an extension of Chinese restaurant process to model the relationship between clusters (i.e. topics) in time-stamped documents. They modelled these relationships in a hierarchical tree in which abstract topics are located at the low-depth levels of the tree (i.e. closer to the root), while detailed topics are near the leaves. Moreover, abstract topics cover longer time spans than detailed topics.

Overall, the aforementioned methods rely heavily on the abundance of content to make satisfactory textual reasoning, while in social media like Twitter, we face a shortage of textual content. Therefore, in this work, we incorporated additional information from the social network besides textual content to construct storylines in Twitter.

A few studies have dealt with storyline generation in Twitter. It is important to note that story detection [8, 22], which is concerned with finding tweets pertaining to a story as it evolves over time, is different from the task of *storyline generation* which is concerned with generating a structured summary of how the story evolves over time. In Dos Santos et al. [23], a story is modelled as a graph of entities propagating through spatial regions in a temporal sequences. Hence, using spatiotemporal 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. [7]. The authors first model tweets in a multi-view graph preserving both temporal and similarity relations among tweets. Then, they select the most representative summary using the dominating set (DS) approximation and apply the Steiner tree (ST) which spans all DS members to generate a storyline.

Unlike these approaches, our method generates a tree over main events by considering social, textual and temporal aspects of tweets. Another distinguished feature of our system is its hierarchical capability of adding more details or tuning it at an abstract level in terms of social support. Moreover, the efficient memory and computational costs of our method make it quite practical in large datasets of tweets for online applications.

2.2. MTS

Despite many years of studies and research conducted on document summarisation [24, 25], tweet summarisation is still in its infancy. On one hand, the high volume of tweets with redundant and/or irrelevant information in users' timeline and search results uncovered the need for tweet summarisation. On the other hand, unique characteristics of tweets, from limited character length and informal language to special markup language using hashtags or mentions, makes direct use of document summarisation techniques rather inefficient.

The concept of MTS was first introduced by Sharifi et al. [26] where they proposed a Phrase Reinforcement Algorithm; however, their proposed method outputs a single tweet as a summary of several tweets. Later, in Inouye and Kalita [27], they extended their work to generate multi-tweet summaries by first clustering tweets and then summarising each cluster exploiting a hybrid tf-idf weighting method.

Characterising an event by extracting the few tweets that best describe its principal sub-events has attracted much research interest in recent years. Chakrabarti and Punera introduced [28] SUMMHMM in which hidden Markov models are trained on previous events to identify sub-events based on tweet minimum activity threshold. However, their method has specifically been designed to summarise structured and recurring sport events. Summarising sport events from tweet streams also drew Kubo et al.'s attention. They produced high quality summaries by selecting tweets posted by 'good reporters' who promptly react to what is happening throughout news world [29]. Having focused on real-time summarisation of events, two papers [30, 31] have considered sudden increases in the volume of tweets with respect to the recent tweeting activity and applied different weighting methods to select representative tweets for sub-events. For real-time monitoring of news in Twitter, Lee and Chien [32] proposed a density-based clustering approach to detect event topics and then summarise the information of high-priority events via topic ranking. In Fang et al. [33], authors summarised bunches of tweets by detecting hot topics among them using multi-view clustering which integrates semantic relation, social tag relation and temporal relation among tweets.

There have also been studies on generating visual summaries of tweets. Tweetmotif [34] groups tweets by frequent terms and characterises a topic by a tag cloud as its label and a set of tweets that contain the label. Marcus et al. [35] provide a visual summarisation of tweets about user-specific events in which relevant tweets along with their geo-location are displayed on a map and the positive or negative sentiment of selected topics are coloured blue or red, respectively. Similar to Tweetmotif, their summaries are labelled by tag clouds.

Twitter, in addition to being an information network, is also a social network and incorporating social aspects can make up for the information shortage in tweets. However, most of the research on tweet summarisation mainly deals with a tweet's text and rarely pays attention to their social aspect.

Liu et al. [36] concentrate on utilising social features to better identify main tweets from a given set of tweets. They incorporate the social concepts of a tweet, such as the number of retweets it has received, number of followers its author has and the readability of tweet's text. They also add user diversity to their method. However, through experimentation, we concluded that the number of retweets alone is not good enough to represent the social attention a tweet has received, since a tweet can be retweeted by its author or a community of authors multiple times. Thus, the number of distinct users that retweet a tweet can better represent its social salience. Moreover, 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 have no active relationship with most of them in a given context.

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

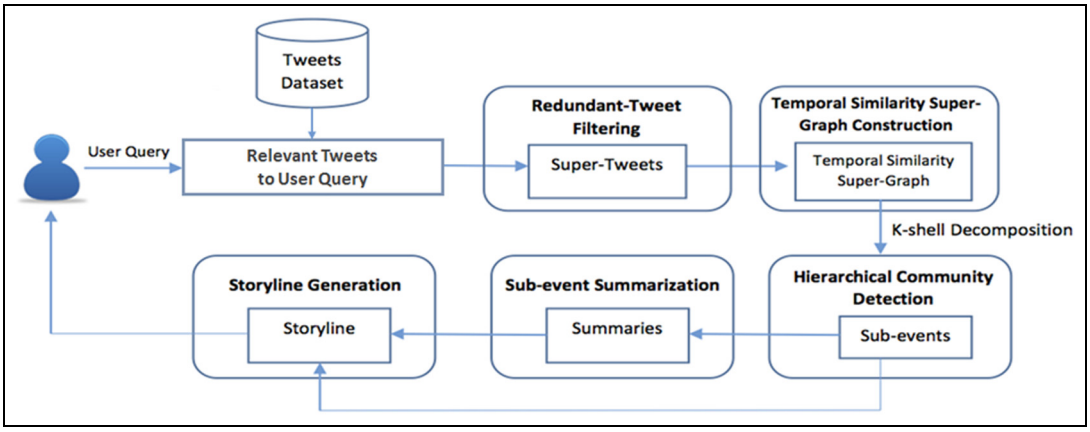

Figure 2. A summarised schematic of our storyline generation framework.

3. Our proposed framework

Our method takes a query about a story and a set of related and relevant tweets to that story as input, and our goal is storytelling as a summary of the main events of the story, as well as the relationships between these events. The generated storyline has a tree structure where each node is labelled by a summary of an individual event of the story and edges indicate semantic relationship between neighbouring nodes.

Our proposed framework named SGSG is made up of five main modules including

1. Redundant-tweet filtering
2. Temporal-semantic super-graph construction
3. Sub-event detection
4. Sub-event summarisation
5. Storyline generation

Figure 2 shows the overall picture of SGSG framework and details of its modules are presented in the following subsections.

3.1. Redundant-tweet filtering module

In the course of storyline generation, the first obstacle is the enormous amount of redundant tweets due to direct or indirect retweets and near-duplicate tweets, which from now on will be referred to as duplicate tweets. Duplicate tweets impose computational costs in terms of both time and space. In our methodology, we select a single tweet to represent the whole set of tweets and duplicated tweets. To avoid losing information, we also keep the earliest posting time of the tweet and the list of unique users who tweeted/retweeted/copied it. As a result, we maintain distilled information in the super-tweet, defined as follows.

Definition 3.1. A *super-tweet st* is defined as a quadruple (*TEXT, rt, A, date*), where *TEXT* is the text vector of the earliest occurrence of the tweet, *rt* is the number of duplicate tweets it represents, *A* is the list of unique authors who have posted (or reposted) this tweet and *date* is the posting time of the earliest tweet with granularity of day.

In order to identify duplicate tweets, the simplest method is scanning the entire corpus to discover duplicates of the tweet, but it is time consuming and inefficient. So, to reduce the computational cost, in this work, we utilise LSH [15]. LSH significantly reduces the running time of finding similar items in high-dimensional spaces by mapping them to the same buckets with high probability. This method uses a family of locality-sensitive hash functions to hash objects to buckets, such that objects close to each other in their high-dimensional space have a higher probability to be hashed to the same bucket. Given the theoretical proofs and experiments reported in Shrivastava and Li [37], we adopted MinHash [38] as LSH hashing method. In our case, each tweet's text is represented as a set of 4-gram character shingles. The LSH method has two steps: indexing and finding. In the indexing step, LSH first selects k different hash functions

$\{h_1, h_2, \ldots, h_k\}$ that map the members of the 4-gram set of tweets to distinct integers. Then, it represents each tweet S as a hash code by MinHash mapping [38]

$$H_{min}(S) = [h_{min_1}(S), h_{min_2}(S), \ldots, h_{min_k}(S)] : h_{min_i}(S) = \text{the minimal member of } S \text{ with respect to } h_i \tag{1}$$

All tweets that are projected to the same k values are members of the same (k-dimensional) bucket. The process of hashing the dataset to k-dimensional buckets is repeated L times with k independently chosen random hash functions to increase the chance of collision for similar items in at least one of them. Indexing time is $O(nLkt)$, where n is the number of tweets and t is the time to perform a hash function on an input tweet.

In the finding step, buckets that have multiple tweets in them (we call these tweets Candidates) are checked and the ones whose 4-gram character shingles have a high degree of Jaccard similarity are considered as duplicate tweets. This method dramatically reduces the number of comparisons needed to find a tweet's duplicates from $n - 1$ to the number of tweets that collide with it in LSH (i.e.$|Candidates|$). In practice, it can provide constant or sublinear search time [39]. It is also easy to parallelise the method using MapReduce [40].

In fact, our method keeps distilled information in super-tweets to reduce both storage and computation cost by discarding duplicate tweets. This step can be performed in a language-independent manner, since no special pre-processing is used. The data structure of super-tweets can be updated incrementally by entering new relevant tweets. Whenever a new relevant tweet arrives, it will be hashed to a bucket and checked to determine whether it is a duplicate or not by measuring the similarity among the candidates in that bucket.

3.2. Temporal-semantic super-graph construction module

This module is dedicated to the construction of a temporal-semantic super-graph over super-tweets for the purpose of storing semantic and temporal information among them. Given a set of super-tweets, we define a node-weighted temporal-semantic super-graph as follows.

Definition 3.2. A *temporal-semantic super-graph* $G = (V, E, w)$, where V is a set of nodes which represent super-tweets, E is a set of directed edges which represent the semantic similarity between tweets and $w : V \to \mathbb{N}$ is a weight function that sets the weight of each node to the rt of the corresponding super-tweet, that is, $w(st) = st.rt$. The direction of the edge is from the earlier super-tweet to the later one according to their date.

There exists an edge from super-tweet st_i to st_j if and only if the first tweet in st_i has been posted earlier than the first tweet of st_j and their semantic similarity is greater than $(1 - \theta_{semantic})$ where $\theta_{semantic}$ is a predefined distance threshold such that a pair of tweets with WMD less than $\theta_{semantic}$ are considered semantically similar. The adjacency matrix (M) is calculated as follows

$$\begin{aligned} &M = T \times S \\ &T = [t_{ij}], \begin{cases} t_{ij} = 1 & \text{if } 0 \le st_j.date - st_i.date < \theta_{temporal} \\ t_{ij} = 0 & \text{otherwise} \end{cases} \\ &S = [s_{ij}], \begin{cases} s_{ij} = 1 & \text{if } WMD_{\theta_{semantic}}(st_i.TEXT, st_j.TEXT) < \theta_{semantic} \\ s_{ij} = 0 & \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

where T captures both temporal order and closeness and S indicates the semantic similarity between super-tweets. In order to find semantic similarity between tweets, we introduce $WMD_{\theta_{semantic}}$ as a distance measure between tweets which is the bounded version of WMD [13].

WMD is based on word embeddings as a semantic vector space model that represents each word with a real-valued vector. High quality of pre-trained word embeddings that are trained on billions of tweets leads to capturing knowledge about them and reveals semantic similarity. The WMD metric extends word similarity to document similarity.

For example, consider two tweets '*Obama speaks to the media in Illinois*' and '*The President greets the press in Chicago*'. Although these tweets have no words in common and will have a cosine distance of one, they are talking about the same concept.

The WMD solves this problem by incorporating the external word correlation knowledge provided by word embeddings and considers the closeness of the word pairs: (Obama, President), (speaks, greets), (media, press) and (Illinois, Chicago) in word embedding space.

The WMD is a measure of dissimilarity between two tweets and is defined as the minimal cost required for the embedded words of one tweet to '*move*' to the embedded words of another tweet. Kusner et al. [13] casted it to the following optimisation problem to find the minimum cumulative cost of moving tweet t to t'

$$\begin{aligned} WMD(t,t') = & \underset{T \geq 0}{min} \sum_{i,j=1}^{n} T_{ij} c(i,j) \\ \text{Subject to} : & \sum_{j=1}^{n} T_{ij} = t_i \forall i \in \{1, \ldots, n\} \\ & \sum_{i=1}^{n} T_{ij} = t'_j \forall j \in \{1, \ldots, n\} \end{aligned} \tag{3}$$

where $c(i,j) = \left\| x_i - x_j \right\|_2$ is the Euclidean distance between embedded vectors of word i in t and word j in t' and T_{ij} determines how much of word i in t moves to word j in t'. Also, $t_i = f_i / \sum_{j=1}^{n} f_j$ if word i appears f_i times in tweet t and hence the constraints ensure that the outflow from word i equals t_i and the inflow to the word j equals t'_j.

In practice, WMD performs best on finding short documents[4] that are similar to each other and therefore works well in finding similar tweets. Although WMD captures semantic similarity with high accuracy, it is slow for large datasets as its time complexity scales $O(p^3 log_p)$ [13], where p is the maximum number of unique words in tweets. In our case of constructing the temporal-semantic super-graph, we just keep close distances and do not need the exact distance values for all pairs. Therefore, we modified the WMD to the bounded WMD with a threshold of $\theta_{semantic}$ as follows

$$WMD_{\theta_{semantic}}(st.TEXT, st'.TEXT) = \min(WMD(st.TEXT, st'.TEXT), \theta_{semantic}) \tag{4}$$

which allows fast filtering of large distances ($\gg \theta_{semantic}$). Moreover, since the optimisation in equation (3) is a special case of earth mover's distance (EMD) [13], we can apply the fast algorithm proposed in Pele and Werman [41] for bounded EMD to the bounded WMD. Bounded WMD runs an order of magnitude faster than the original WMD using a fast algorithm proposed in Pele and Werman [41].

The construction of temporal-semantic super-graphs naturally depends on word embeddings and can be applied to all languages that have pre-trained word embeddings. Also, word embeddings can be learned in an unsupervised manner [42], and therefore, the approach can be extended to any language.

3.3. Sub-event detection module

Once the temporal-semantic super-graph is constructed, the SGSG method identifies different sub-events of the story. 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 duplicate tweets have been discarded by now, there are still some irrelevant and personal tweets which are not related to the news events of interest. These kinds of tweets are neither similar to many other tweets nor retweeted many times. We consider these tweets as noisy tweets and applying community detection on the noisy temporal-semantic super-graph may produce inaccurate results.

Among many measures calculating the importance of nodes in a graph, degree centrality and clustering coefficient [43] can only characterise local information about nodes. The computational complexity of betweenness centrality [44] is high due to the need to calculate the shortest paths. In 2010, Stanley et al. [14] introduced the k-shell decomposition method pointing out that nodes with large k-shell values are important nodes that constitute the core of the network. A k-shell is a maximal connected sub-graph in which every node's degree is at least k [14]. The k-shell value of a node is k if and only if the node belongs to a k-shell but not to any $(k + 1)$-shell [14]. 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 [14]. We applied the k-shell decomposition method for pruning nodes with k-shell value less than a threshold $\theta_{k-shell}$ in order to keep only the important nodes located in the core of the graph.

Since we are looking for important super-tweets, we can keep nodes with large k-shell values located in the core and discard the others. However, since each node in the super-graph is a super-tweet which represents a set of tweets, we propose a variant of the k-shell decomposition method for weighted graphs to be applied to our temporal-semantic 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 retweeted more than ordinary tweets. However, they are more probably reported in different narrations while preserving the same connotations. The k-shell structure serves the purpose of identifying significant sub-events with $\theta_{k-shell}$ serving as a social support threshold.

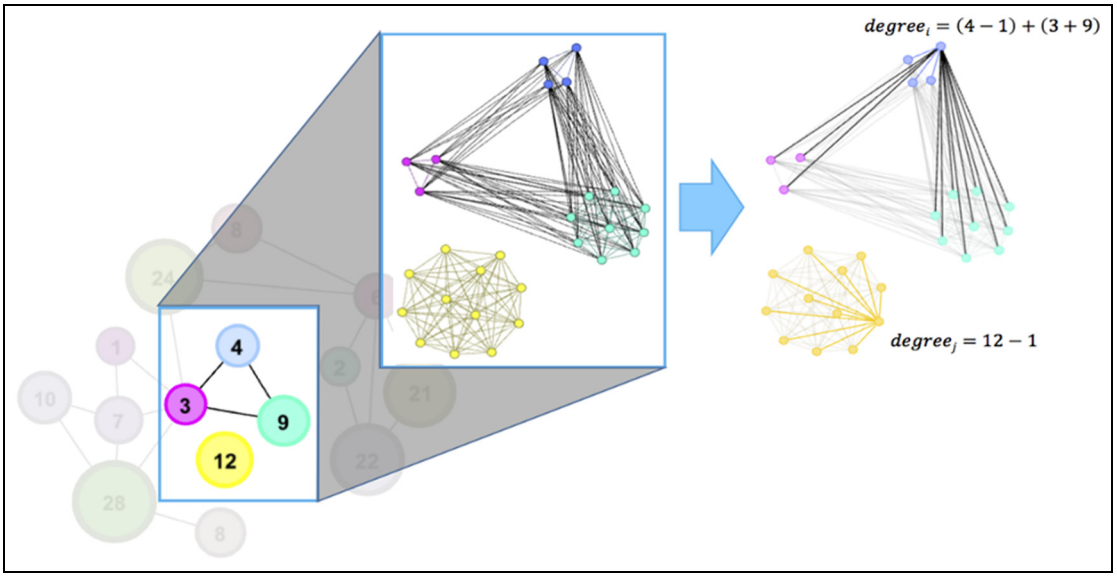

Figure 3. Virtual degree calculation according to super-tweet's weight and degree in super-graph. In this graph, each node is a super-tweet $\boldsymbol{st_i}$ and its label indicates its weight $\boldsymbol{w}(\boldsymbol{st_i})$. Each node $\boldsymbol{st_i}$ can be represented as a clique of size $\boldsymbol{w}(\boldsymbol{st_i})$. The isolated node with label 12 can be represented as a clique of size 12 in which the degree of all nodes are 11, and therefore, its virtual degree is 11. The node with label 4 has the virtual degree of 15 where (4–1) of them are intra-clique degree and (3 + 9) are inter-clique degree with its neighbouring cliques.

From another standpoint, the k-shell structure empowers us to provide a hierarchical storyline in which events are arranged in order of their social support rank. This means that by starting from the maximum k-shell of the graph and gradually decreasing the value of $\theta_{k-shell}$, we can add more detailed events or side stories to the storyline step-by-step. Applying the k-shell method can also accelerate the procedure of community detection in our graph. Considering that, for those news stories that last for a long time, similarity graphs might have millions of super-tweets, performing community detection to extract their sub-events may take several hours. However, using the SGSG method, the performance of the community detection procedure on the remaining k-shells is not only fast but also more accurate.

To apply the k-shell decomposition method on our node-weighted super-graph, one can imagine each super-tweet node st_i as a representative of a clique of size rt_i, because there should be an edge between a tweet and its duplicates in the similarity graph. Therefore, we set a virtual degree to each super node according to the following equation

$$Virtual\ degree(st_i) = \begin{cases} w(st_i) - 1 + \sum\limits_{st_j \in nb(st_i)} w(st_j) & degree(st_i) > 0 \\ w(st_i) - 1 & degree(st_i) = 0 \end{cases} \quad (5)$$

where $w(st_i)$ stands for the weight of ith super node indicating the number of duplicate tweets represented by super-tweet st_i, $nb(st_i)$ is the set of neighbours of super-tweet st_i and $degree(st_i)$ is the degree of st_i in the temporal-semantic super-graph. Figure 3 provides an example to show the logic behind equation (5). By substituting virtual degree for the degree in k-shell decomposition proposed in Stanley et al. [14], the k-shell values of super-tweets are obtained. Afterward, we filter super nodes with k-shell values lower than a threshold $\theta_{k-shell}$. This technique trims the graph by deleting noisy super-tweets with less social attention and hence grants salience to important phases of the story. By changing $\theta_{k-shell}$, we can control the intricacy of storyline.

Then, a community detection method is employed on the remaining k-shells to aggregate the semantically related super-tweets into the same event. Communities are groups of nodes which share common properties and play similar roles within the graph [45]. Community structure [46, 47] reveals the organisation of groups of nodes that have comparatively large density of internal edges with respect to few edges connecting nodes of different groups. In our temporal-semantic super-graph, events are represented as communities. Community detection methods [48, 49] are designed to identify strongly intra-connected groups. We chose to apply the Infomap multilevel partitioning algorithm of Rosvall and Bergstrom [50], since it is reported as the most accurate method among several considered in Lancichinetti and Fortunato

[51]. Infomap is specifically designed to detect communities in large networks. Moreover, Infomap gives us a hierarchical partitioning that enables us to zoom in on sub-events and get more details on them.

3.4. Event summarisation module

After mapping detected communities to sub-events, it is time to provide a summary for each community and use this summary as a label for the corresponding node in the storyline tree. The event summarisation task is to extract a subset of super-tweets that best covers the important aspects of the event. From a social viewpoint, a tweet is important and socially salient if it has been posted or retweeted by many important users. This encourages us to pick socially salient tweets covering the content in addition to having easy-to-understand writing style. In order to generate summaries that not only cover the content but are also selected from socially salient tweets, we extended our super-graph to a graph of super-tweets and users.

Definition 3.3. A *tweet-user graph is defined as* $G = (V, E, F)$, where the vertices set $V = V_{ST} \cup V_U$ is the union of the sets of super-tweets (V_{ST}) and users (V_U), 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, a tweet which is retweeted by many different users is socially more important than a tweet retweeted many times by a single user. This phenomenon is modelled in the tweet-user graph such that if a single user reposted the same tweet, there will exist only one corresponding edge.

It is important to note that using the number of followers or global PageRank [52] score 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 context, it is important to rank users in that context instead of ranking them globally to estimate users' influence (or a kind of expertise) in that context. Therefore, in the next step, we perform the Hyperlink-Induced Topic Search (HITS; Kleinberg's HITS [53]; also known as Hubs and Authorities) algorithm on the tweet-user graph to assign weight to super-tweets. HITS is a link analysis algorithm designed to rank web pages and, unlike PageRank, it makes a distinction between 'authorities' and 'hubs'. The idea behind Hubs and Authorities stemmed from the fact that certain web pages, known as hubs, were not actually authoritative in their information content, but link to other authoritative pages. In our case, super-tweets play the role of authorities and users play the role of hubs in the tweet-user graph. Using social network of users, the user hub score estimates the value of its links, while the tweet authority score estimates the social importance of a tweet. Therefore, a good hub is a user that posted many tweets, and a good authority is a tweet that was tweeted or retweeted by many different users. For each event, we consider its sub-graph in the temporal-semantic super-graph as an undirected graph. Then, we find the minimum-weighted DS on this graph to have the most social salient super-tweet(s) covering the whole content of the sub-event.

3.4.1. Minimum-weight DS problem. A subset of vertices of a graph is called a DS if every vertex in the graph is contained in the DS or has a neighbour in it. The minimal DS problem is to find a DS of the smallest size. If every vertex of the input graph is associated with a weight, the minimum-weight dominating set (MWDS) problem computes the DS of minimum weight. Although the MWDS problem is known to be non-deterministic polynomial (NP)-hard [54], the greedy approximation provided in Shen and Li [55] gives a $log\Delta$ approximation, where Δ is the maximum degree of the graph.

Algorithm 1 illustrates the steps of summarising sub-events. This algorithm takes an input graph $G = (V, E, F, H, w)$ where the vertices $V = VST \cup VU$ correspond to users and super-tweets with high k-shell values from the sub-event detection module, 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 that user and H is a set of undirected edges, representing the semantic similarities between members of VST. w is the weight function that maps each super-tweet st in VST to the normalised authority score of st (lines 3–5). The algorithm also takes a set of detected sub-events with their members and returns the summary of each sub-event as a set of dominant super-tweets. The while loop (lines 10–17) shows the approximation of the MWDS. In each iteration, $s(v)$ is computed as the number of remaining nodes that v can span. Then, the node with maximum number of uncovered neighbours proportional to its weight is added to the summary set and its neighbours are added to the temp set. The summary of each subevent is used as a label for its corresponding node in the generated storyline tree.

In contrast to existing approaches, such as LexRank [56] which only considers lexical centrality as salience, our approach takes both social and semantic centrality into account through HITS and MWDS, respectively. In addition, the generated summary has minimal redundancy since the set is of minimum size.

Algorithm 1. Sub-events summarisation.

Input: $G = (V,E,F,H)$ where the vertices $V = V_{ST} \cup V_U$ correspond to super-tweets and users, and w is the weight function w: $V_{ST} \rightarrow \mathbb{R}$. 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: *Summaries*, which is the list containing summary of sub-events.

$G' = G.subgraph(V,E,F)$
$\langle hub,authority \rangle = G'.HITS()$
for *each node st in* V_{ST} **do**
    $w(st) = 1 - \frac{authority(st)}{\sum_{i \in V_{ST}} authority(i)}$
end
$Summaries = []$
for *each sub − event SE in SubeventsSet* ***do***
    $G''(V'',H'',w) = G.subgraph(V_{st},H,w)$ // corresponds to the subgraph of super-tweets in *SE* and their similarity edges
    $Summary_{SE} = \varnothing; T = \varnothing;$
    while $Summary_{SE} \cup T \neq V''$ ***do***
        for $v \in V'' - \text{Summary}_{SE}$ ***do*** // greedy approximation of MWDS
            $s(v) = \|\{v' | (v',v) \in H''\} \setminus T\|$
        end
        $v* = argmax_v \frac{s(v)}{w(v)}$
        $Summary_{SE} = Summary_{SE} \cup \{v*\}$
        $T = T \cup \{v'' | (v'',v*) \in H''\}$
    end
    $Summaries.add(Summary_{SE})$
end
return *Summaries*

3.5. Storyline generation module

Given a set of sub-events with their labels, the goal of this module is to organise them in a tree structure that tells the story conveyed by the sub-events. Given the tree structure of the storyline, the earliest event is located at the root and the story evolves over different branches of the tree. To generate the storyline over the sub-events, first we construct a weighted directed sub-event graph, where nodes represent sub-events and edge weights represent the semantic distance between sub-events. The direction of the edges is from the earlier sub-event to the later one.

In constructing the sub-event graph over the temporal-semantic super-graph, we add one node per community (i.e. sub-event). For nodes C and C' in the sub-event graph, the direction of the edge between them is determined by the ratio of edges from community C to community C' and vice versa in the super-graph. We calculate the distance between sub-events C and C' by calculating the average similarity between any pair of super-tweets

$$distance(C, C') = 1 - \frac{\sum_{st_i \in C} \sum_{st_j \in C'} sim(st_i, st_j)}{|C| \times |C'|} \tag{6}$$

Then, we apply Edmonds' algorithm [57] 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 continuum and semantic similarity and results in a coherent storyline which reveals the temporal structure of a story.

4. Experiments and results

In this section, we first introduce the datasets we have used for experiments. Then, we evaluate the performance of the overall methodology through comparative analysis. We also evaluate the performance of each module individually based on its goal.

Table 1. Statistics of IranElection, IranNuclearProgram and USPresidentialDebates datasets.

	IranElection	IranNuclearProgram	USPresidentialDebates
Language	English	Persian	English
Number of (original + retweeted) tweets	1,868,261 + 372,048	27,598 + 51,476	3,246,763 + 7,673,170
Number of users	6,691	2,102	1,015,128
Average tweet length	16.42	17.29	14.31
Average tweet length after stop word removal	9.81	10.75	8.87

4.1. Dataset

Our experiments are conducted on three datasets. The first one consists of English tweets sent during the Iranian Presidential Election 2009 using #IranElection. This tag was among the top 10 tags on Twitter in the year 2009 and it is historically important as it was the first time that a social network was being used to broadcast and coordinate a social movement [58]. We refer to this dataset as the *IranElection* dataset. The second dataset, referred to as the *IranNuclearProgram* dataset, consists of Persian tweets about negotiations on Iran's nuclear program in 2015, posted under the Persian hashtag #مذاکراتهستهای. We have crawled these datasets using Twitter API in our Social Network Lab.[5] These two datasets are ideal for the task of storyline generation since they are on developing stories containing many sub-events and side stories well covered on Twitter. Moreover, there is currently no gold-standard dataset for the task of storyline generation available, and hence, we had to manually annotate a dataset for the automatic evaluation of our task. Therefore, we have used these datasets because they pertain to events we are significantly familiar with, allowing us to use human evaluation for the subjective aspects of our experimental results as will be described in sub-sections 4.3 and 4.4.3.

The third dataset, referred to as the *USPresidentialDebates* dataset, consists of English tweets about the series of debates held for the 2016 United States presidential election. Harvard Dataverse [59] published ~13 million tweet ids from tweets of the first, second and third presidential debates and the vice presidential debate. Among them ~11 million tweets were available at the time we retrieved the complete tweets using the Twitter API.[6] The experiments which do not need human evaluation (i.e. redundant tweet filtering and sub-event detection modules) have been tested on the *USPresidentialDebates* dataset as well (for more details, see sections 4.4.1 and 4.4.2). More details about the datasets are shown in Table 1.

4.2. Hyper-parameters

When setting the parameter values, we treat $\theta_{temporal}$ and $\theta_{semantic}$ as systematic parameter whose values are experimentally determined using the development dataset.[7] $\theta_{k-shell}$ determines the number of events that the final storyline would have. For example, to have 50 events in the final storyline, we set $\theta_{k-shell}$ to 98 for the *IranElection* dataset and set $\theta_{k-shell}$ to 2318 for the *USPresidentialDebates* dataset. There is no intrinsic metric to define the optimal number of events and the parameter value is tuned based on whether the user wants to have more detailed events or more general ones. $\theta_{temporal}$ depends on the life time of the topic we want to generate the storyline for, and in our experiments, it is set to 1 week ($\theta_{temporal} = 7$) for all datasets. $\theta_{semantic}$ is an independent variable determining the threshold for the semantic distance of two similar tweets. In this article, we examined different values from the range of 0.1–0.5 with 0.05 step size and empirically found that setting $\theta_{semantic}$ to 0.2 obtained the best result.

4.3. Evaluation of the overall system

To the best of our knowledge, the study by Lin et al. [7] is the state-of-the-art research on storyline generation from microblogs [6]. In their system, dynamic pseudo relevance feedback (DPRF) is used to retrieve the most relevant tweets given a story query. Then, the tweets are modelled in a multi-view graph and the most representative summary is selected using the DS approximation. Finally, the ST is applied which spans all DS members to generate a storyline. We implemented their method called DS + ST as a baseline system with the parameter values mentioned in their paper and compared the result with our system.

The primary difference to our framework are threefold. First, they construct the multi-view graph over all relevant tweets regardless of whether they are retweets or duplicates, whereas we remove duplicates. Second, they only use the textual content of tweets and ignore their associated social aspects, whereas we utilise social features like number of

Table 2. Comparative analysis of the overall system in terms of Acc_{edge}, P_{path}, R_{path} and $F1_{path}$.

	IranElection				IranNuclearProgram			
	Acc_{edge}	P_{path}	R_{path}	$F1_{path}$	Acc_{edge}	P_{path}	R_{path}	$F1_{path}$
SGSG	0.73	0.79	0.75	0.77	0.67	0.75	0.73	0.74
DS + ST	0.63	0.67	0.71	0.69	0.59	0.65	0.67	0.66

SGSG: semantic graph-based storyline generation; DS: dominating set; ST: Steiner tree.

retweets and follower–followee network of users to improve the performance. Third, they used the cosine measure to calculate similarity between tweets, which can only capture the similarity at the lexical level, whereas we use WMD to capture the semantic similarity in addition to the lexical similarity between tweets.

Also, as claimed in Lin et al. [7], there is no standard evaluation platform and criteria for automatic storytelling. Therefore, we developed a crowdsourcing system which recruited human candidates, 21 graduate students with a self-declared interest in and following the news, as annotators to evaluate the output storylines of SGSG and DS + ST. In the case of our system, we tune the minimum k-shell threshold ($\theta_{k-shell}$) such that we retrieve the top 50 important events for both *IranElection* and *IranNuclearProgram* datasets and generate their storyline. Then, for each pair of events, we asked annotators to decide whether there is a relation between them or not, or, in other words, whether there should be an edge from the earlier event to the later one in the storyline tree. For each edge, a majority vote of two out of three annotations is used to decide whether the edge is valid or not. The same process is applied to the events extracted via the DS + ST approach. We used the transitive reduction of human-generated storyline trees as gold-standard storylines.

Edge accuracy ($\mathrm{Acc}_{\mathrm{edge}}$) as evaluation criterion only assess edges individually to make the point-wise evaluation and loses sight on the structure as a whole. Therefore, additional measures are required in order to judge the fitness of the generated structure as a whole. We used a relaxed path accuracy defined in Wang et al. [60] as the overlap between the path from one node to the root in ground-truth versus in the generated path to the root for the same node and used path precision P_{path} and recall R_{path} as

$$\begin{aligned} P_{path} &= \frac{\sum_{i=1}^{n-1} (|path_{\bar{Y}}(i) \cap path_{Y'}(i)|/|path_{Y'}(i)|)}{n-1} \\ R_{path} &= \frac{\sum_{i=1}^{n-1} (|path_{\bar{Y}}(i) \cap path_{Y'}(i)|/|path_{\bar{Y}}(i)|)}{n-1} \end{aligned} \tag{7}$$

where $path_{\underline{Y}}(i)$ and $path_{Y'}(i)$ are the set of nodes in the path from node i to the root in ground-truth $\bar{Y}$ and in our system output Y', for example, $path_{\bar{Y}}(5)$ in Figure 1(a) is $\{5, 4, 3, 2, 1\}$. According to this metric, mistakes in the edges closer to the root would be penalised more than mistakes in the lower levels of the tree.

Table 2 shows the performance comparison of the SGSG and DS + ST systems in terms of $Accuracy_{edge}$, P_{path} and R_{path} for both datasets. We used the gold-standard storyline as a ground-truth in these metrics.

It is worth to note that the lack of manual annotation for *USPresidentialDebates* dataset and hence human-generated storylines as gold-standard storylines is the reason that prevents us from evaluating this dataset and consequently its exclusion from Table 2.

Results reveal that our system is capable of producing coherent summaries and evolving structure of main events. Our proposed method outperforms DS + ST because instead of using cosine similarity, we computed the semantic similarity between tweets using WMD, which incorporates the external word correlations knowledge provided by word embedding. This leads to more accurate edges between nodes in the temporal-semantic super-graph. Therefore, community detection outperforms to identify sub-events. Furthermore, the semantic similarity versus lexical similarity leads to better estimates of distances between communities, which improves the coherence of the final storyline.

4.4. Evaluation of modules

As mentioned earlier, the overall system is evaluated via crowdsourcing. In addition, each module of the overall system can be isolated and evaluated automatically.

Table 3. Performance evaluation of redundant-tweet filtering module.

	Precision (%)	Recall (%)	No. of comparisons (TP + FP)	No. of super-tweets
IranElection	59	93	2,910,472	156,826
IranNuclearProgram	44	91	91,165	2096
USPresidentialDebates	31	87	7,663,029	268,343

TP: true positives; FP: false positives.

4.4.1. Redundant-tweet filtering module. In order to construct the data structure of super-tweets, we need to identify retweets/duplicate tweets. In our datasets, only 28% of duplicate tweets are explicit retweets. For the LSH, we set k to be 5 and L to be 15 such that the probability (δ) of missing a similar tweet within the Jaccard similarity (s) of 0.8 is less than 2.5% (i.e. $\delta = (1 - s^k)^L$). Then, we need to set the value of the upper bound of the distance (R) between near-duplicate tweets. As suggested in Datar et al. [61], by first guessing the value of R and doing a binary search in the range of 0.1–0.4 with step size 0.05, we chose a distance of 0.2, and hence, a similarity of 0.8 as reasonable estimates of the similarity threshold. Setting k to 5 leads to a good balance between time spent for distance computations and computing hash functions. We used the OpenLSH[8] package, which is an open-source platform that implements LSH.

Table 3 shows the performance of redundant-tweet filtering module. Note that false positives (FP) are the cases in which dissimilar tweets are mapped into the same bucket and false negatives (FN) are the cases in which similar items are not located in the same bucket in none of the hash tables. This module performed well in finding duplicate tweets for both English and Persian datasets with sub-linear order with regard to quadratic order of comparing all tweet pairs. For example, it found 93% of similar pairs in the *IranElection* dataset on the reduced search space from $\binom{1,868,261}{2} = 1.7452e12$ to 2,910,472 pairs. Furthermore, removing duplicate tweets resulted in 156,826, 2096 and 268,343 super-tweets in *IranElection, IranNuclearProgram* and *USPresidentialDebates* datasets, respectively. As noted earlier, since our focus is on news scope, tweets that received no retweet and no duplication are discarded. For instance, in the *IranElection* dataset, after removing duplicate tweets, the temporal-semantic super-graph has 156,826 nodes and 84,159 edges. However, if we wanted to construct a temporal-semantic graph over tweets rather than super-tweets, the graph would have 2,240,309 nodes and 11,559,932 edges.

4.4.2. Sub-event detection module. For the English datasets, we applied the bounded WMD based on Glove [42] pre-trained word vectors for Twitter, and therefore, we used their script for pre-processing Twitter data. This script detects the hashtags, mentions, URLs, smileys, numbers and elongated words using regular expressions and replaces them with special tags. The Glove word vectors have vectors for these special tags, such as $<\text{USER}>$, $<\text{SMILE}>$, $<\text{HASHTAG}>$, $<\text{ELONG}>$ and so on. As suggested in the WMD paper [13], all words in the SMART [62] stop word list are removed as they do not contribute a lot to the information content of the tweets. For the Persian dataset, we customised the pre-processing script for the Persian language and we trained word vectors using the Glove learning algorithm on 50 million Persian tweets.[9]

Then, we applied our variation of the k-shell decomposition method to the constructed graph to prune the graph and remove noise. Note that keeping those super-tweets with high k-shell values results in sub-events which attracted a lot of social attention. Furthermore, the k-shell of nodes follows the Power-law distribution with $\alpha = 2.78(3)$ and that is what allows us to explicitly identify important sub-events.

For evaluating the performance of sub-event detection, we used the WikiTimes [63] dataset which is a Resource Description Framework (RDF)-based knowledge base of news events and is available for free. The WikiTimes dataset includes all news stories from the year 2000. Each STORY object contains a chronologically ordered list of EVENTs that constitute the story timeline. Each EVENT object consists of a list of entities involved in the event in the form of an ENTITY object plus the date of the event and the URL of the Wikipedia page of the event.

The corresponding stories to our Twitter *IranElection* dataset in WikiTimes are 'Iranian presidential election 2009', '2009 Iranian presidential election protests', 'Quds Day' and 'Ashura protests'. These stories contain 63 events. We used the events of these stories as a gold-standard to evaluate our sub-event detection procedure. We set the value of $\theta_{k-shell}$ to 56 to have the top 63 important events. Our system detects 48 of these events, and so, the accuracy of our system is 76%. Results reveals that in some cases, people use the power of social networks to spread unrelated, but socially relevant messages at considerable speed among as many people as possible. Therefore, tweets like '*hospital source: Severe*

Table 4. Performance evaluation of our sub-event detection compared with the baseline method for both *IranElection* and *IranNuclearProgram* datasets.

		SGSG				DS + ST		
	No. of gold events	Accuracy	P@100	R@100	F_1	*P*	*R*	F_1
IranElection	63	76%	54%	86%	66%	30%	81%	44%
IranNuclearProgram	29	48%	22%	76%	34%	17%	65%	27%
USPresidentialDebates	36	64%	30%	83%	44%	14%	78%	24%

SGSG: semantic graph-based storyline generation; DS: dominating set; ST: Steiner tree.

Table 5. Performance of our event summarisation module compared with four other summarisation methods. The bold values show that our summarisation approach, which utilises both social and textual information, outperforms the others.

	ROUGE 1	ROUGE 2
LexRank	0.4501	0.3821
LSA	0.5122	0.4614
Dominating set	0.5654	0.4793
DS + ST	0.5801	0.4985
Our event summarisation method	**0.6239**	**0.5476**

LSA: latent semantic analysis; DS: dominating set; ST: Steiner tree.

shortage of blood supplies across ALL Iran hospitals, plz donate blood' that needed to be retweeted many times received high social attention although they have not been reported in the WikiTimes dataset. When we tune the $\theta_{k-shell}$ to have the top 100 important events, 54 out of 63 of sub-events are detected leads to P@100 to be 54% while recall is 86%.

The corresponding stories to *IranNuclearProgram* dataset are 'Nuclear program of Iran' and 'Negotiation of a comprehensive nuclear agreement with Iran'. These stories contain 29 events of which our system detects 14 (We used the English to Persian human-translated version of these events' description). Unfortunately, WikiTimes only has abstract events and does not cover many other important events related to the nuclear program of Iran that are important for Iranians and well reflected in Twitter by Persian users.

For the *USPresidentialDebates* dataset, we used the list of 36 topics addressed in debates from Wikipedia[10] as a gold-standard for our sub-event detection module evaluation. We set the value of $\theta_{k-shell}$ to 2380 to have the top 36 important events. Our system detects 23 of these topics which leads to accuracy of 64%. When we tune the value of $\theta_{k-shell}$ to have the top 100 important events, 30 out of 36 of topics are detected.

The results presented in Table 4 show that our sub-event detection method outperforms the baseline system in all datasets. Using semantic similarity versus cosine similarity added more informative edges to the graph about communities which leads to a better estimation of sub-events. Also, our method ranks sub-events based on the minimum *k*-shell value of nodes in their communities to highlight main sub-events. In contrast, the DS + ST method uses cosine similarity to find similar tweets. Moreover, this method uses all DS members in its multi-view tweet graph as sub-events.

4.4.3. Event summarisation module. In order to evaluate the summarisation approach, three annotators were enlisted and each annotator had to choose the most descriptive tweet for each sub-event based on her own personal opinion. For any event *i*, and any pair of annotated results from two annotators, denoted by *Xi* and *Yi*, the inter-annotator agreement of their selected summaries is calculated using $|X_i \cap Y_i / X_i \cup Y_i|$ [36]. The average inter-agreement over all sub-events was 0.81 which is substantial. We used those tweets for which at least two annotators agreed as the gold standard.

We applied the well-known ROUGE toolkit [64] as an automatic metric for summarisation evaluation. We used ROUGE-1 and ROUGE-2, which calculate the overlap in unigrams and bigrams between the system and the human-generated summaries. We applied our event summarisation method and some of the well-known summarisation methods including latent semantic analysis (LSA) [65], LexRank [56], DS and the baseline approach to summarise the top 50 events. The average ROUGE scores over 50 events are presented in Table 5. For those methods which summarise single documents, we concatenated the *TEXT* of super-tweets of each event while maintaining their temporal order.

The results show that our proposed event summarisation module which considers semantic similarities in graph construction and weights the super-tweets based on their social importance in terms of authority scores outperforms other summarisation methods. We also applied the *t*-test and the improvement is statistically significant ($p < 0.05$).

5. Conclusion and future work

In this article, we proposed a novel graph-based framework for generating storylines in Twitter which considers both the semantic and the social information of tweets. We modelled significant sub-events as communities in temporal and semantic similarity graph. Our experiments revealed that filtering the redundant tweets reduces the size of graph significantly in our datasets and we did it sub-linearly via LSH. One of the strengths of our model is pruning the graph of noisy tweets and keeping main events using the *k*-shell decomposition method. We quantified how much social attention a super-tweet attracts as the *k*-shell value of its corresponding node in the super-graph and our approach guarantees that detected events got an amount of social attention exceeding the threshold of $\theta_{k-shell}$. We summarised sub-events by solving an approximation of the minimum-weighted DS to select socially salient tweets. Finally, we modelled a storyline as a directed tree of sub-events evolving over time.

Comprehensive experiments conducted on English and Persian datasets of real-world events extensively reported in Twitter showed the remarkable performance of our method in comparison with human-generated storylines and the state-of-the-art storyline generation algorithm on Twitter. Our framework detected 76% and 48% of sub-events reported in WikiTimes for the *IranElection* and *IranNuclearProgram* datasets, respectively. It also detected 64% of topics addressed in the US presidential debates. Our framework detects main events based on the social hypothesis that popular events are retweeted more than ordinary tweets. However, analysing the results reveals that in some cases like 'The presidential #debates are streaming LIVE on Twitter https://t.co/fkQRq57zij' that people use the power of Twitter to spread unrelated, but socially relevant messages among as many people as possible, our system considers them as important events.

In our future work, we plan to learn the structure and relationship between events via a deep convolutional neural network. The results of this ongoing research will be reported in the near future.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship and/or publication of this article.

Notes

1. http://www.adweek.com/digital/twitter-q2-2016/
2. From September 2016, adding quotes, polls, videos or images will no longer reduce precious characters while still users get 140 characters text messages and links [66].
3. Pre-trained English word vectors are available at http://nlp.stanford.edu/data/glove.twitter.27B.zip and Persian word vectors are available at https://github.com/ndehghany/Persian-Tweets-Word-Vectors.
4. The reader is referred to empirical results reported in Xie [67].
5. Sociallab.ir
6. The Twitter API does not return tweets that have been deleted or belong to accounts that have been suspended, deleted or made private.
7. The first 200,000 tweets are chosen for development and the remainder for testing.
8. https://github.com/singhj/locality-sensitive-hashing
9. Persian word vectors are publicly available at https://github.com/ndehghany/Persian-Tweets-Word-Vectors
10. https://en.wikipedia.org/wiki/United_States_presidential_debates,_2016#Topics_addressed_and_not_addressed

References

[1] Myers SA, Sharma A, Gupta P et al. Information network or social network? The structure of the twitter follow graph. In: *Proceedings of the 23rd international conference on World Wide Web*, Seoul, Korea, 7–11 April 2014, pp. 493–498. New York: ACM.

[2] Yan R, Kong L, Huang C et al. Timeline generation through evolutionary trans-temporal summarization. In: *Proceedings of the conference on empirical methods in natural language processing*, Edinburgh, 27–31 July 2011, pp. 433–443. New York: ACM.

[3] Allan J, Gupta R and Khandelwal V. Temporal summaries of new topics. In: *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval*, *New Orleans, LA*, 9–13 September 2001, pp. 10–18. New York: ACM.

[4] Yan R, Wan X, Otterbacher J et al. Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In: *Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval*, *Beijing, China*, *24–28 July* 2011, pp. 745–754. New York: ACM.

[5] Wang Z, Shou L, Chen K et al. On summarization and timeline generation for evolutionary tweet streams. *IEEE T Knowl Data En* 2015; 27: 1301–1315.

[6] Zhou H, Yu H, Hu R et al. A survey on trends of cross-media topic evolution map. *Knowl-Based Syst* 2017; 124: 164–175.

[7] Lin C, Lin C, Li J et al. Generating event storylines from microblogs. In: *Proceedings of the 21st ACM international conference on information and knowledge management*, Maui, HI, 9 October–2 November 2012, pp. 175–184. New York: ACM.

[8] Srijith PK, Hepple M, Bontcheva K et al. Sub-story detection in Twitter with hierarchical Dirichlet processes. *Inform Process Manag* 2017; 53: 989–1003.

[9] Shahaf D, Guestrin C and Horvitz E. Trains of thought: generating information maps. In: *Proceedings of the 21st international conference on World Wide Web*. http://dl.acm.org/citation.cfm?id=2187957 (2012, accessed 18 April 2014).

[10] Shahaf D and Guestrin C. Connecting the dots between news articles. In: *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining*, *Washington, DC*, 25–28 July 2010, pp. 623–632. New York: ACM.

[11] Vossen P, Caselli T and Kontzopoulou Y. Storylines for structuring massive streams of news. In: *Proceedings of the first workshop on computing news storylines*, *Beijing, China*, 31 July 2015, pp. 40–49. Stroudsburg, PA: Association for Computational Linguistics.

[12] Hurlock J and Wilson ML. Searching Twitter: separating the Tweet from the Chaff. *In: Proceedings of the fifth international conference on weblogs and social media*, Barcelona, 17–21 July 2011, pp. 161–168. Menlo Park, CA: The AAAI Press.

[13] Kusner M, Sun Y, Kolkin N et al. From word embeddings to document distances. In: *Proceedings of the international conference on machine learning*, *Lille*, 6–11 July 2015, pp. 957–966. New York: ACM.

[14] Stanley H and Eugene Makse HA. Identification of influential spreaders in complex networks. *Nat Phys* 2010; 6(11): 888–893.

[15] Gionis A, Indyk P, Motwani R et al. Similarity search in high dimensions via hashing. In: *Proceedings of the VLDB*, Edinburgh, *7–10 September* 1999, pp. 518–529. New York: ACM.

[16] Liu Y, Kliman-Silver C and Mislove A. The tweets they are a-changin': evolution of Twitter users and behavior. In: *Proceedings of the ICWSM*, 2014, pp. 5–314, https://mislove.org/publications/Profiles-ICWSM.pdf

[17] Kumar D, Ramakrishnan N, Helm RF et al. Algorithms for storytelling. *IEEE T Knowl Data En* 2008; 20: 736–751.

[18] Hossain MS, Andrews C, Ramakrishnan N et al. Helping intelligence analysts make connections. In: *Proceedings of the 2011 AAAI workshop on scalable integration of analytics and visualization*, 2011, https://pdfs.semanticscholar.org/5d19/892003132247774e156adff00d6f86fbb7a3.pdf

[19] Hossain MS, Butler P, Boedihardjo AP et al. Storytelling in entity networks to support intelligence analysts. In: *Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining*, *Beijing, China*, *12–16 August* 2012, pp. 1375–1383. New York: ACM.

[20] Shahaf D, Guestrin C and Horvitz E. Metro maps of science. In: *Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining*, *Beijing, China*, 12–16 August 2012, pp. 1122–1130. New York: ACM.

[21] Song J, Huang Y, Qi X et al. Discovering hierarchical topic evolution in time-stamped documents. *J Assoc Inf Sci Tech* 2016; 67: 915–927.

[22] Atefeh F and Khreich W. A survey of techniques for event detection in twitter. *Comput Intell* 2015; 31: 132–164.

[23] Dos Santos RF Jr, Shah S, Chen F et al. *Spatio-temporal storytelling on twitter (computer science technical reports)*. Virginia Tech, 2015, http://vtechworks.lib.vt.edu/handle/10919/24701

[24] Ou S, Khoo CS-G and Goh DH. Design and development of a concept-based multi-document summarization system for research abstracts. *J Inf Sci* 2008; 34: 308–326.

[25] Kogilavani SV, Kanimozhiselvi CS and Malliga S. Summary generation approaches based on semantic analysis for news documents. *J Inf Sci* 2016; 42: 465–476.

[26] Sharifi B, Hutton M-A and Kalita J. Summarizing microblogs automatically. In: *Proceedings of the human language technologies: the 2010 annual conference of the North American chapter of the association for computational linguistics*, *Los Angeles, CA*, *2–4* June 2010, pp. 685–688. New York: ACM.

[27] Inouye D and Kalita JK. Comparing Twitter summarization algorithms for multiple post summaries. In: *Proceedings of the 2011 IEEE third international conference on privacy, security, risk and trust (PASSAT) and 2011 IEEE third international conference on social computing (Socialcom)*, Boston, MA, 9–11 October 2011, pp. 298–306. New York: IEEE.

[28] Chakrabarti D and Punera K. Event summarization using tweets. *ICWSM* 2011; 11: 66–73.

[29] Kubo M, Sasano R, Takamura H et al. Generating live sports updates from Twitter by finding good reporters. In: *Proceedings of the 2013 IEEE/WIC/ACM international joint conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, Atlanta, GA, 17–20 November 2013, pp. 527–534. New York: ACM.

[30] Nichols J, Mahmud J and Drews C. Summarizing sporting events using Twitter. In: *Proceedings of the 2012 ACM international conference on intelligent user interfaces*, *Lisbon*, *14–17 February* 2012, pp. 189–198. New York: IEEE.

[31] Zubiaga A, Spina D, Amigó E et al. Towards real-time summarization of scheduled events from twitter streams. In: *Proceedings of the 23rd ACM conference on hypertext and social media*, *Milwaukee, WI*, 25–28 June 2012, pp. 319–320. New York: ACM.
[32] Lee C-H and Chien T-F. Leveraging microblogging big data with a modified density-based clustering approach for event awareness and topic ranking. *J Inf Sci* 2013; 39: 523–543.
[33] Fang Y, Zhang H, Ye Y et al. Detecting hot topics from Twitter: a multiview approach. *J Inf Sci* 2014; 40: 578–593.
[34] O'Connor B, Krieger M and Ahn D. TweetMotif: exploratory search and topic summarization for Twitter. In: *Proceedings of the ICWSM*, 2010, pp. 384–385, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.478.5512&rep=rep1&type=pdf
[35]. Marcus A, Bernstein MS, Badar O et al. Twitinfo: aggregating and visualizing microblogs for event exploration. In: *Proceedings of the SIGCHI conference on human factors in computing systems*, *Vancouver, BC*, *Canada*, 7–12 May 2011, pp. 227–236. New York: ACM.
[36] Liu X, Li Y, Wei F et al. Graph-based multi-tweet summarization using social signals. In: *Proceedings of the COLING*, 2012, pp. 1699–1714, https://aclanthology.info/pdf/C/C12/C12-1104.pdf
[37] Shrivastava A and Li P. In defense of minhash over simhash. In: *Proceedings of the artificial intelligence and statistics*, 2014, pp. 886–894, http://proceedings.mlr.press/v33/shrivastava14.pdf
[38] Broder AZ, Charikar M, Frieze AM et al. Min-wise independent permutations. In: *Proceedings of the thirtieth annual ACM symposium on theory of computing*, 1998, pp. 327–336, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121. 8215&rep=rep1&type=pdf
[39] Slaney M, Lifshits Y and He J. Optimal parameters for locality-sensitive hashing. *Proc IEEE* 2012; 100: 2604–2623.
[40] Rajaraman A and Ullman J. *Mining of massive datasets*, http://books.google.com/books?hl=en&lr=&id=OefRhZyYOb0C&oi=fnd&pg=PR5&dq=Mining + of + Massive + Datasets&ots=aMyzkdCoyX&sig=OwPfpLw72jFa1ou9HaOtHk-A3Po (2012, accessed 18 April 2014).
[41] Pele O and Werman M. Fast and robust earth mover's distances. In: *Proceedings of the 2009 IEEE 12th international conference on computer vision*, Kyoto, Japan, 29 September–2 October 2009, pp. 460–467. New York: IEEE.
[42] Pennington J, Socher R and Manning C. Glove: global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543, https://www.aclweb.org/anthology/D14-1162
[43] Wasserman S and Faust K. *Social network analysis: methods and applications*. Cambridge: Cambridge University Press, 1994.
[44] Freeman LC. A set of measures of centrality based on betweenness. *Sociometry* 1977; 40: 35–41.
[45] Fortunato S. Community detection in graphs. *Phys Rep* 2010; 486: 75–174.
[46] Girvan M and Newman MEJ. Community structure in social and biological networks. *Proc Nat Acad Sci* 2002; 99: 7821–7826.
[47] Porter MA, Onnela J-P and Mucha PJ. Communities in networks. *Not Am Math Soc* 2009; 56: 1082–1097.
[48] Newman MEJ and Girvan M. Finding and evaluating community structure in networks. *Phys Rev E* 2004; 69: 26113.
[49] Papadopoulos S, Kompatsiaris Y, Vakali A et al. Community detection in social media. *Data Min Knowl Disc* 2012; 24: 515–554.
[50] Rosvall M and Bergstrom CT. Maps of information flow reveal community structure in complex networks, https://arxiv.org/pdf/0707.0609.pdf
[51] Lancichinetti A and Fortunato S. Community detection algorithms: a comparative analysis. *Phys Rev E* 2009; 80: 56117.
[52] Page L, Brin S, Motwani R et al. *The pagerank citation ranking: bringing order to the web*. Technical Report, Stanford InfoLab Publication Server, 1999.
[53] Kleinberg JM. Hubs, authorities, and communities. *ACM Comput Surv* 1999; 31: 5.
[54] Karp RM. Reducibility among combinatorial problems. In: Miller RE, Thatcher JW and Bohlinger JD (eds) *Complexity of computer computations*. Berlin: Springer, 1972, pp. 85–103.
[55] Shen C and Li T. Multi-document summarization via the minimum dominating set. In: *Proceedings of the 23rd international conference on computational linguistics*, 2010, pp. 984–992, http://www.aclweb.org/anthology/C10-1111
[56] Erkan G and Radev DR. LexRank: graph-based lexical centrality as salience in text summarization. *J Artif Intell Res* 2004; 22: 457–479.
[57] Edmonds J. Optimum branchings. *J Res Natl Bur Stand B* 1967; 71: 233–240.
[58] Tabatabaei SA and Asadpour M. Study of influential trends, communities, and websites on the post-election events of Iranian presidential election in Twitter. In: Missaoui R and Sarr I (eds) *Social network analysis-community detection and evolution*. Berlin: Springer, 2014, pp. 71–87.
[59] Littman J, Wrubel L and Kerchner D. 2016 United States presidential election tweet Ids. *Harvard Dataverse*. Epub Ahead of Print 23 November 2016. DOI: 10.7910/DVN/PDI7IN.
[60] Wang H, Wang C, Zhai C et al. Learning online discussion structures by conditional random fields. In: *Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval*, *Beijing, China*, *24–28 July* 2011, pp. 435–444. New York: ACM.
[61] Datar M, Immorlica N, Indyk P et al. Locality-sensitive hashing scheme based on p-stable distributions. In: *Proceedings of the twentieth annual symposium on computational geometry*, *Brooklyn, NY*, *8–11 June* 2004, pp. 253–262. New York: ACM.

[62] Salton G. The SMART retrieval system – experiments in automatic document processing. Upper Saddle River, NJ: Prentice-hall, 1971.
[63] Tran GB and Alrifai M. Indexing and analyzing Wikipedia's current events portal, the daily news summaries by the crowd. In: *Proceedings of the 23rd international conference on World Wide Web*, *Seoul, Korea*, 7–11 April 2014, pp. 511–516. New York: ACM.
[64] Lin C-Y. Rouge: a package for automatic evaluation of summaries. In: *Proceedings of the ACL-04 workshop text summarization branches out*, 2004, http://www.aclweb.org/anthology/W04-1013
[65] Ozsoy MG, Alpaslan FN and Cicekli I. Text summarization using latent semantic analysis. *J Inf Sci* 2011; 37: 405–417.
[66] Doing more with 140 characters, https://blog.twitter.com/developer/en_us/a/2016/doing-more-with-140-characters.html (2016, accessed 1 July 2017).
[67] Xie J. Experiment with document similarity via Matt Kusner's MWD paper, https://github.com/PragmaticLab/Word_Mover_Distance (accessed 14 June 2016).