Generating Event Storylines from Microblogs

Chen Lin
Xiamen University
Shanghai Key Laboratory of Intelligent Information Processing
Xiamen, China
chenlin@xmu.edu.cn

Chun Lin
Xiamen University
Xiamen, China
chunlin@stu.xmu.edu.cn

Jingxuan Li
Florida International University
Miami, FL, U.S.A.
jli003@cs.fiu.edu

Dingding Wang
Florida International University
Miami, FL, U.S.A.
dwang003@cs.fiu.edu

Yang Chen
Duke University
Durham NC, U.S.A
ychen@cs.duke.edu

Tao Li
Florida International University
Miami, FL, U.S.A.
taoli@cs.fiu.edu

ABSTRACT

Microblogging service has emerged to be a dominant web medium for billions of individuals sharing and spreading instant news and information, therefore monitoring the event evolution on microblog sphere is crucial for providing both better user experience and deeper understanding on realtime events. In this paper we explore the problem of generating storylines from microblogs for user input queries. This problem is challenging due to the sparse, dynamic and social nature of microblogs. Given a query of an ongoing event, we propose to sketch the real-time storyline of the event by a two-level solution. We first propose a language model with dynamic pseudo relevance feedback to obtain relevant tweets, and then generate storylines via graph optimization. Comprehensive experiments on Twitter data sets demonstrate the effectiveness of the proposed methods in each level and the overall framework.

Categories and Subject Descriptors

H.2.8 [**Information Systems**]: Database applications—*Data mining*; H.3.3 [**Information Search and Retrieval**]: Information filtering

General Terms

Algorithms, Design, Experimentation

Keywords

Social media, microblog, language model, dynamic pseudo relevance feedback, storyline

1. INTRODUCTION

Microblogging service has rapidly increased its popularity in recent years. People are attracted to microblogging sites, such as Twitter, for instant first-hand reports on real-life events. In the meantime, instead of using web search engines, users are more willing to propose event queries on Twitter to obtain information about an ongoing event [33]. Systems that deliver realtime event notification on Twitter are also available [28].

It would be helpful for industry, academia, and end-users, if a skeleton of an event by request is automatically generated from the huge volume of tweets. We refer this problem as *Generating Event Storyline from Microblogs (GESM)*. For example, Figure 1 presents the storyline based on an event query of "Egypt Revolution". The vertical location of each frame indicates the time-stamp of the corresponding phase. The hierarchical structure depicts how major progress happens in adjacent phases. The branches partition simultaneously happened tweets into different semantic groups. Auto-generated storylines facilitates easy navigation in microbloghspere and also supports a wide range of mining systems on collective intelligence.

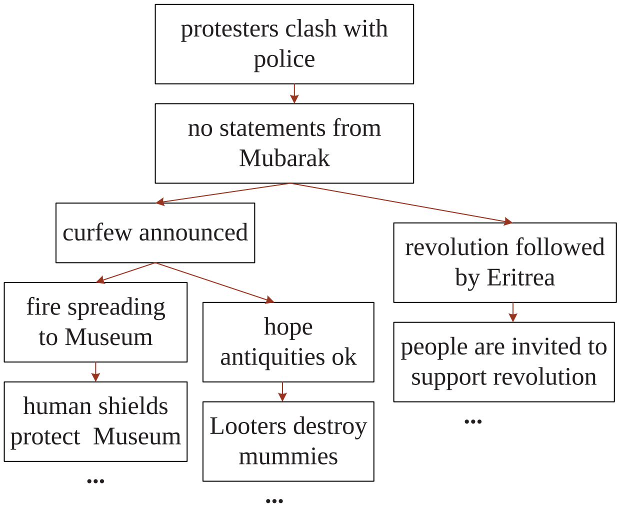

protesters clash with police
no statements from Mubarak
curfew announced
revolution followed by Eritrea
fire spreading to Museum
hope antiquities ok
people are invited to support revolution
human shields protect Museum
Looters destroy mummies
...
...
...

Figure 1: A sample storyline for event query "Egypt Revolution"

GESM is a challenging problem. There exist studies in generating storylines from news articles [21, 36, 13]. Also,

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...$15.00.

there are fruitful research efforts on the area of topic detection and tracking (TDT) [1]. However, differences between GESM and prior studies are remarkable. 1) All of previous studies are designed for a collection of well edited facts, e.g., news articles and high quality web pages, which lack effective mechanisms for handling extremely short noisy text streams such as microblogs. 2) To facilitate users information preferences, GESM requires user identified queries as inputs. Compared with TDT, GESM provides personalized service in massive microblog data. 3) Since GESM is tailored for user input queries, a two-level framework is necessary: at the low level, finding all relevant tweets through the time-line of the event by a retrieve model; and at the high level, summarizing relevant tweets and the latent structure to produce a storyline.

Challenges of microblog storyline generation arise from the following aspects. First of all, the dynamic and sparse nature of microblogs remains a large obstacle to improve performance of microblog retrieval system for event queries. Event queries usually only contain basic descriptive terms, e.g., the location of the event, the person involved, and the main theme, etc. Microblogs are streams of very short texts reporting recent brief updates. How to match the underlying event expressed by the vague event query to potential relevant tweets which possibly not contain any query terms is a severe problem. State-of-art IR models including a number of variants of pseudo relevance feedback have been proposed as attempts at this problem. However, they expand the original query based on frequencies, therefore temporal relevant keywords during certain periods may be misjudged if they are not significant in frequency throughout the time-line of event. As we shall see in later sections, this framework should be modified to better capture the dynamic of microblogs. Secondly, the social nature of tweets increases the difficulty of integrating semantic similarity with chronological order in generating a storyline. Information sharing in microblog sphere yields numerous duplicate tweets and direct and undirect re-tweets. Duplicate tweets and re-tweets are created after the right time point, and they will trigger confusion in partitioning the event time-line. Thus a naive method, which employs traditional text summarization strategy in each time segment, is not applicable.

In this work, we focus on resolving the above challenges. Major contributions of this work are: (1) A novel problem of generating event storylines from microblogs is proposed and a two-level solution to this problem is provided. (2) A *dynamic pseudo relevance feedback (DPRF)* language model is presented to retrieve relevant tweets given an event query. By making an assumption that the prior probability of pseudo relevance feedback is dependent on the burst periods of given event, DPRF expands the original query with representative keywords in active phases of the event. Thus the accuracy of retrieval is enhanced. (3) The problem of storyline generation on the retrieved microblogs is formulated as a graph-based optimization problem and is solved by approximation algorithms of minimum-weight dominating set and directed Steiner tree. The generated storylines ensure both temporal continuity and content coherence.

The rest of the paper is organized as follows. Section 2 introduces related work on information retrieval, multi-document summarization and microblog mining. Overview of the framework is presented in section 3. Language model based query expansion via dynamic pseudo relevance feedback is described in Section 4. Section 5 presents the framework for storyline generation. In Section 6, experimental results are analyzed and a user study is conducted. We conclude our work in Section 7.

2. RELATED WORK

Several research directions are related to our work, including microblog mining, information retrieval, and multi-document summarization.

2.1 Microblog Mining

The emergence of Twitter motivates recent research works on mining microblogs, including microblog search [8], identifying emerging topics on Twitter [23], and summarizing tweets in a certain period [32]. A few research works have been devoted to event detection [28, 29], but they focus on the detection of novel events without a global view.

To achieve a better performance, several research methods have been proposed to deal with the unique characteristics of microblogs, e.g. expanding tweets by hashtags [7], utilizing social relations for identifying influential tweets [10], incorporating sentiment categorization [3], promoting most recent tweet [9], employing transfer latent topic models for overcoming abbreviated texts [42], and expanding queries by recently frequently co-occurred terms [22].

The dynamic and social nature of microblogs is not fully explored by previous research efforts. By adding a temporal dimension in the event storyline generation system, our work sheds light on the understanding and mining of microblogosphere.

2.2 Language Model for IR

Pseudo relevance feedback (PRF) has been proved to be helpful in the IR community. The state-of-art PRF by language model approaches are surveyed in [41]. In addition, to reinforce model-based feedback, a classifier is adopted before query expansion to determine the "goodness" of expansion candidates in [4]. Adaptive query expansion is implemented in [38] to select query expansion candidates from different sources.

A few works study text stream retrieval. For example, time-based model in [19] assigns documents with prior associated with the "freshness" of documents. The temporal factor can be introduced into query expansion [9, 22]. But all of the above mentioned approaches favor only recent documents, therefore are not applicable to cover the lifespan of the whole event. In [11], a temporal profile is constructed for each query throughout its lifespan to categorize the query and predict query performance, however there is no mechanism to incorporate the temporal profile into query expansion.

Most PRF methods expand the original query in a static manner. On the contrary our method selects event specific expansion terms, which are temporally correlated with query terms in the pseudo relevant documents. Our empirical study shows that our DPRF method is superior than previous researches in the scenario of event query retrieval.

2.3 Text Summarization and TDT

Multi-document summarization conveys the main and most important meaning of several documents. One type of summarization systems select representative sentences, e.g. with significant frequency [40], or structural centroid in sentence graph [18, 13]. Another type is based on matrix decomposition [17, 35, 31]. Some prior researches focus on clustering query-induced results [37].

Unlike multi-document summarization, research in the area of topic detection and tracking (TDT) [1] aims to thread streams of texts. TDT includes five main tasks: story segmentation, topic tracking, topic detection, first story detection and link detection. Among the five tasks, topic tracking and link detection are similar to components of the GES problem addressed in this paper. Most researches in story segmentation and link detection are devoted to clustering and classifying similar texts, without considering the timesstamps of articles, e.g. relevance model in [16] adopts symmetric similarity comparison. Others consider the influence among articles to be unidirectional and directly dependent, e.g. topic structure is identified in [25] by forgetting out-of-date statistics, the bursty structure is recognized in [12] by estimating state transition probability in an infinite state automation.

To conclude, although these methods have been successfully applied in several domains, they are not applicable to the event storyline generation problem. The quality of generated storyline is determined by the quality of summary in each phase, and the quality of phase segmentation, and previous summarization and TDT methods lack the ability to generate a complete and coherent storyline. On the contrary, our graph optimization based method has a built-in mechanism to simultaneously generate the summary for each virtual phase and naturally integrate the generated summaries to form the storyline.

2.4 Timeline and Storyline Construction

Not until recently, a limited number of studies devote to summarizing documents with time stamps, mostly news articles. For example, in [24], an HMM style model is presented to discover evolutionary theme patterns (term distributions). BlogScope [2] discovers hot trend and temporal keyword correlations. Similarly, a burstness-aware search framework is presented in [15]. A finite mixture model is presented in [25] for tracking dynamics of topic trends. ETS [36] returns the evolution skeleton along the timeline by extracting representative and discriminative sentences at each phase. In [39] representative sentences are chosen based on relevance, coverage, coherence and cross-date diversity. In [32] summarization consists of median tweets in each time segment. These timeline generation methods can hardly be applied to our storyline generation problem because the asynchronism of information propagation in the microblogosphere makes it difficult to partition the timeline of an event into different phases.

There also exists few studies on storyline generation. [21] proposes a storyline generation framework to identify events using self-organizing maps and to extract the main storyline by assigning weights to different events based on the similarity between the events and given topics. The system then generates storyline-based summaries using term weighting schemes, but does not take the temporal information into consideration. Very recently a pictorial storyline generation method has been proposed [34] that combines image and text analysis to obtain a storyline containing textual, pictorial, and structural information to provide a sketch of the topic evolution. This method first constructs a multi-view graph, in which each node is a picture and its text description. Then representative nodes are selected by finding a minimum dominant set on the graph. Finally, a Steiner tree approximation is employed to connect the dominating nodes to form a pictorial storyline. In this paper, we take the storyline generation procedures in [34]. However, our work differs from [34] as follows. Unlike the traditional text and image analysis, our work focuses on microblogs which have their unique characteristics. For example, the length of a microblog entry is limited, the content may be noisy, and tags and links appear frequently. These characteristics make storyline generation from microblogs very challenging. Dealing with the dynamic and sparse nature of microblogs is one of the most important contributions in this work.

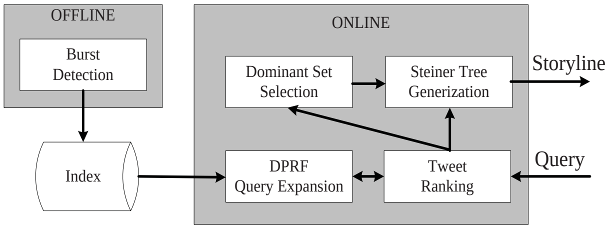


Figure 2: The System Architecture for Event Storyline Generation

3. THE FRAMEWORK OVERVIEW

Given an event query Q, which is a set of user defined keywords or phrases describing an ongoing event in real life, our goal is to mine the storyline from the relevant tweets. The generated storyline should be a graph structure, where each node is labeled by a summary of an individual phase of the event, and each edge represents causal relationship between two phases. Consequently, the proposed framework consists of two models for retrieving relevant tweets and generating storyline respectively. Figure 2 shows the framework of our proposed storyline generation system. In the off-line layer, each tweet is indexed, and the temporal information of each term is stored and pre-processed in the burst detection module. The retrieving module and storyline generation module are implemented in cascade online layers, details of which will be introduced in section 4 and 5.

4. THE RETRIEVAL MODEL

4.1 Preliminaries

In modern information retrieval, language model approaches estimate probability distribution θ_d over the vocabulary W for each document d in the corpus C. By modeling each query Q as θ_Q, relevant documents can be ranked according to query likelihood. However, the original query is usually short and vague, and can not fully cover the underlying information need. To enhance the query expressibility, query expansion is adopted to replace the original query Q by a new high quality query Q'. In a pseudo relevance manner, suppose the few top ranked documents $d+$ by the initial query Q builds a relevant model θ_F, we can set the new query to be a linear combination of original query Q and relevant model θ_F [41]:

$$p(w|\theta_{Q'}) = (1-\alpha)p(w|\theta_Q) + \alpha p(w|\theta_F), \tag{1}$$

where α controls the degree of coherence of the new query to pseudo relevance.

In this paper, we follow relevance model method to infer θ_F. The relevance model method approximates θ_F as the query model, and each pseudo relevant document is a sample from the query model. Therefore relevance model method defines term distribution in θ_F as the likelihood of generating terms from pseudo relevance:

$$p(w|\theta_F) \propto \sum_{d+\in F} p(d+)p(w|d+)p(Q|d+), \tag{2}$$

where $p(Q|d+) = \prod_{q\in Q} p(q|d+)$.

4.2 Dynamic Pseudo Relevance Feedback

In traditional pseudo relevance feedback (PRF), the prior $p(d+)$ is usually set to be uniform. However, this assumption doesn't hold in an instant broadcast medium like Twitter. Consider the event "Egypt Revolution" in Figure 1, there are several distinct phases (e.g. 2011-01-24 to 2011-01-26, 2011-02-01 to 2011-02-03) during which the event encounters major progress and discussion bursts out.

Intuitively, in the initial search results of an event query of "Egypt Revolution", a top tweet published on 2011-01-25 is more likely to be a truly relevant tweet than a tweet published on 2011-01-01 on a near position in the ranking list. Suppose that the event is detected to have K burst periods (detection detail is introduced in the next subsection), the prior distribution of relevant tweets should be centered around each burst period.

We first assume that the prior probability of relevant document $d+$ is dependent on the distance of t_{d+} to the centroid of burst periods, denoted as $\Phi = \{\phi_1 \cdots \phi_K\}$. We define the following three probability functions, each of which is controlled by scale parameter σ. As shown in Figure 3, these probability functions have various mechanisms to model the effective range of burst period, decay coefficient and skewness.

1. **Mixture Gaussian Distribution** assumes that the prior probability is normally distributed, with multiple peaks located at the centroid of each burst period:

$$p(d+) = \Sigma_{i=1}^{K} \frac{1}{K} \times \frac{\exp -\frac{(t_{d+}-\phi_i)^2}{2\sigma^2}}{\sqrt{2\sigma^2\pi}}, \tag{3}$$

   where σ implies the effective range of each peak.

2. **Local Power Distribution** assumes that the prior probability is restricted to the neighborhood around the nearest burst period. This is the major difference from mixture Gaussian distribution, in which the information diffusion is cumulative from each burst period. Suppose that the probability decreases as the distance $r(d+, \phi_k)$ between $d+$ and the nearest centroid ϕ_k increases in a polynomial function $(1+r(d+, \phi_k))^{-\sigma}$, the effective range of each burst period is $\mathcal{R}$, a subset of $[\frac{\phi_k+\phi_{k-1}}{2}, \frac{\phi_k+\phi_{k+1}}{2}]$, which means that $p(d+) = 0$ outside $\mathcal{R}$. The distribution can be written as

$$p(d+) = \frac{1}{N} \times \frac{(1+r(d+,\phi_k))^{-\sigma}}{C(\Phi)}, \tag{4}$$

   where $C(\Phi)$ is a constant for ϕ_k to guarantee $\int p(d+) = 1$, It is a function of all boundaries R for burst periods. If we set the boundary to be in middle of every two adjacent period, then $C(\Phi) = (\sigma - 1)\Sigma_{k=1}K(2 - (1+\frac{\phi_k-\phi_{k-1}}{2})^{1-\sigma} - (1+\frac{\phi_{k+1}-\phi_k}{2})^{1-\sigma})$. Otherwise for fixed $\mathcal{R} = [\phi_k - R_1, \phi_k + R_2]$,$C(\Phi) = K(\sigma-1)(2-(1+R_1)^{1-\sigma} - (1+R_2)^{1-\sigma})$. $\sigma > 1$ is the scale parameter controlling the degree of probability decay.

3. **Skewed Linear Distribution** assumes that the prior probability is positive skew, with a longer tail after the burst period. It is difficult to derive a probability distribution with a defined peak. Therefore we use a linear density function to approximate the skewed disribution:

$$p(d+) \propto \sum_{k=1}^{K} C - r(d+, \phi_k), \tag{5}$$

   where $r(d+, \phi_k)$ has different definition before and after the burst period.

$$r(d+,\phi_k) = \begin{cases} 0 & \text{if } \phi_k - R \le t_{d+} \le \phi_k + R \\ \sigma(t_{d+} - \phi_k - R) & \text{if } t_{d+} > \phi_k + R \\ C & \text{else} \end{cases}, \tag{6}$$

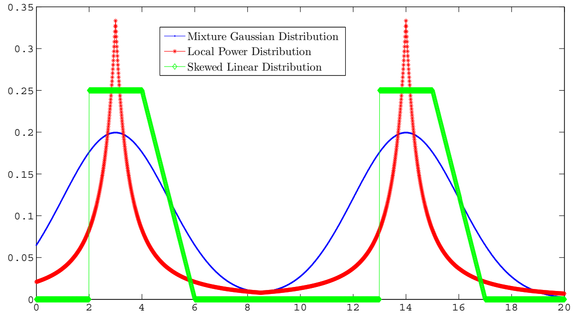


Figure 3: An illustration of prior probabilities of pseudo relevant tweets, with two burst periods centered at time points 3 and 14.

4.3 Burst Period Detection

Intuitively, at time point ϕ_k, each query term should 1) appear more frequently than usual 2) be continuously frequent around the time point. Following these intuitions, we propose to detect burst periods of the event by 1) for each query term, finding the time intervals with arbitrary length in which the query term appears constantly frequent; 2) picking the time points within these intervals with the largest sum of frequencies over all query terms.

We first borrow the definition of a term's "bursty score" from [15]. Suppose that w is a term, which occurs for $tf(w,T)$ times in any time interval T, then the bursty score of w is defined in Eq.(7):

$$B(w,T) = \frac{tf(w,T)}{\sum_{T'} tf(w,T')} - \frac{|T|}{|\sum_{T'}|}, \tag{7}$$

where $|T|$ is the length of the time interval, and $|\sum_{T'}|$ is the length of observation time, which is experimentally set to be time span of the whole corpus. The bursty score of a term is positive if it has above average observation frequency.

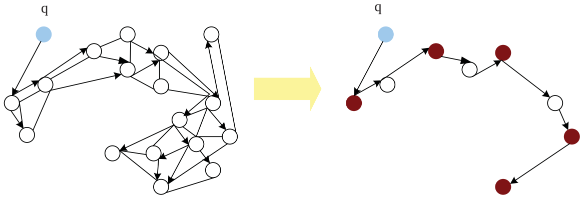


Figure 4: An illustration of the storyline generation

A linear time algorithm in [27] is proposed to find time interval $T_{w,j} =< st, et, LS, RS >$ with the maximal cumulative burst score $B(w, T_{w,j})$, where st is the starting time of $T_{w,j}$, et is the ending time of $T_{w,j}$, LS is the cumulative bursty score before $T_{w,j}$, RS is the cumulative bursty score after $T_{w,j}$.

Next, we present Eq.(8) to compute the score of any query term q at each time point:

$$H(q,t) = \begin{cases} \frac{B(q,T_j)}{|T_j|} & \text{if } \exists T_j, t \in T_j \\ 0 & \text{else} \end{cases}, \qquad (8)$$

where $B(q, T_j)$ is the bursty score of q in maximal interval T_j which is super segment of t. Then we rank each time point by $\sum_{q \in Q} H(q,t)$ and choose the largest K time point ϕ_k.

5. STORYLINE GENERATION

As discussed in Section 1, to generate the storylines from relevant tweets, obstacles are duplicated tweets and indirect retweets. Intuitively, we can pick up a good tweet to represent similar or duplicated tweets. The representative tweets provide the basic outline for each phase. Then the representative tweets are connected appropriately to depict the evolving structure of the event. In order to eliminate noisy retweets, only texts published after a certain time can be considered as subsequent phases. Finally, there may be different ways of connecting these representative tweets, and an optimistic connection should be the one that connects them most smoothly.

The storyline generation follows the processes described in [34]. Thus, the storyline generation procedure consists of three parts. In the first part, a multi-view tweet graph is constructed, in which the semantic and temporal information among relevant tweets is stored. Next, representative tweets are extracted by finding a minimum dominant set on the tweet graph. Finally, a minimum steiner tree algorithm is employed to connect the representative tweets in each phase [30].

Given an event query Q and a collection of relevant tweets by the method described in Section 4, we can construct a multi-view tweet graph.

Definition 1 (multi-view tweet graph). *A multi-view graph $G = (V, W, E, A)$, where V is a set of vertices (nodes), W is the weights of V, E is a set of undirected edges, which represents the similarities between tweets, and A is a set of directed edges (arcs), which represents the time continuity of the tweets.*

Construction of such a graph is controlled by three nonnegative real parameters α, τ_1, τ_2, $\tau_1 < \tau_2$. Each node in G represents a tweet. We use the cosine measure to calculate similarity between two tweets. To define E, we join the two nodes by an edge if and only if the text similarity between the two responding tweets is greater than α. To define A, we draw an arc from v_i to v_j if and only if $\tau_1 \le t_j - t_i \le \tau_2$, where t_i and t_j are their respective time stamps. We call $[\tau_1, \tau_2]$ the temporal window. Also, for each node v_i, its vertex weight, $w(v_i)$, is $1 - score(Q, v_i)$. In our method, we first find the dominating set on the undirected graph $G = (V, W, E)$ (i.e., without considering A in the multi-view graph), and then perform the steiner tree algorithm to connect the dominating set on the directed graph $G = (V, W, A)$ (i.e., without considering E in the multi-view graph) which takes the time continuity into consideration and leads to a coherent storyline.

A subset S of the vertex set of an undirected graph is a dominating set if for each vertex u, either u is in S or is adjacent to a vertex in S. The problem of finding a set of representative summaries can be viewed as the minimum-weight dominating set problem on the undirected graph (V, W, E).

Definition 2 (MWDS). *The Minimum-Weight Dominating Set Problem (MWDS) is the problem of finding, given a vertex- weighted undirected graph G, from all dominating sets of $G = (V, W, E)$, the one whose total vertex weight is the smallest.*

We use the following straightforward greedy algorithm for obtaining an approximate solution (Algorithm 1). This algorithm views that the weight of a newly added vertex is evenly shared among its newly covered neighbors and selects the node that minimizes this share at each round of iteration. The approximation rate of this algorithm is $1 + \log(\Delta \| OPT \|)$, where Δ is the maximal degree of G and OPT is the optimal dominating set.

Algorithm 1: Greedy *MWDS* Approximation

```
Input: $G = (V, W, E)$
Output: dominant set $S$
$S \leftarrow \emptyset, T \leftarrow \emptyset$;
while $|S| < W \&\& S \neq V$ do
    for $v \in V - S$ do
        $s(v) = \|\{v' | (v', v) \in E\} \setminus T\|$;
        $v* = \arg\min_v \frac{w(v)}{s(v)}$;
        $S \leftarrow S \bigcup \{v*\}$;
        $T \leftarrow T \bigcup \{v'' | (v'', v*) \in E\}$;
    end
end
```

Once we select the most representative summary in each phase using the dominating set approximation, we need to generate a natural storyline capturing the temporal and structural information of the event-relevant tweets. To study this problem we use the concept of Steiner trees. Here a Steiner tree of a graph G with respect to a vertex subset S is the edge-induced sub-tree of G that contains all the vertices of S having the minimum total cost, where the cost is the total weight of the vertices.

Definition 3 (Steiner Tree). *Given a directed graph $G = (V, W, A)$, a set S of vertices (terminals), and a root $q \in S$ from which every vertex of S is reachable in G, find the subtree G rooted at q containing S with the smallest total vertex weight.*

Algorithm 2: Steiner Tree Algorithm

```
Input: $G = (V, W, A), S, q, k \geq 1$
Output: Steiner tree $T$ rooted at $q$ covering at least $k$
        vertices in $S$
$T \leftarrow \emptyset$;
while $k > 0$ do
    $T_{best} \leftarrow \emptyset$;
    $cost(T_{best}) \leftarrow \infty$;
    for $v \in V, (v_0, v) \in A, 1 \leq k' \leq k$ do
        $Tp \leftarrow A_{i-1}(k', v, S) \bigcup \{(v_0, v\}$;
        if $cost(T_{best}) > cost(T')$ then
            $T_{best} \leftarrow T'$;
        end
        $T \leftarrow T \bigcup T_{best}$;
        $k \leftarrow k - \|S \bigcap V(T_{best})\|$;
        $S \leftarrow S \setminus V(T_{best})$;
    end
end
```

We apply the Steiner tree approximation in [34] to generate the storyline. In this algorithm, the initial call of $A_i(k, q, S)$ with S set to the dominating set calculated by algorithm 2, q set to be event vertex assigned with the earliest time stamp, and k set to be the size of S. The algorithm takes a level parameter $i \geq 1$. $i = 1$ is the default case where the straightforward algorithm selects l vertices closest to root and returns the union of the shortest paths. The length of an arc $(u, v) \in A$ is the vertex weight of u. We will interpret the output tree as the storyline transition from the root to all the other dominating objects as illustrated in Figure 4. For a constant i, the algorithm is known to run in polynomial time and produce an $O(k^{1/i})$ approximation.

6. EXPERIMENTS

In the experiments, we evaluate the performance of the proposed framework. In particular, Section 6.2 presents the experiments on tweet retrieval, and Section 6.3 conducts the comparisons on storyline generation. We also conduct a user study to compare our system with different document understanding systems in Section 6.4.

6.1 Data Set

The data set is Tweets2011 corpus for TREC 2011 microblog track. The corpus is comprised of 2 weeks (23th January 2011 until 8th February) of sampled tweets from Twitter. Different types of tweets are presented, including replies and retweets. The corpus is multilingual, including English, Japanese and so on. More details of the collection are illustrated in Table 1.

Number of tweets	15137399
Number of English tweets	9318772
Number of retweets	1487299
Number of English retweets	1069006
Number of users	4670516
Median Tweet Length	8.66
Median English Tweet Length	10.76

Table 1: Statistics of Data set

In pre-processing, we do not remove stop-words. Instead, mentions (@somebody) are removed from the vocabulary. Non-English tweets containing less than one English word with more than 2 characters are filtered. Explicit re-tweets with HTTP code 302 are filtered. Empty tweets and forbidden tweets with HTTP code 403 and 404 are also filtered. Porter stemmer is adopted in indexing.

6.2 Tweet Retrieval

TREC 2011 microblog track provides 49 queries and relevance judgements for these queries. Each query is associated with an time stamp. Only tweets published before the time stamp are under consideration. After examination, we believe that each query is describing an ongoing real event. Therefore we use the TREC queries in this subsection. We use both highly relevant and relevant tweets annotated by TREC as the ground truth.

The dominant evaluation metric is the precision at top 30 tweets (P@30). However we also provide the mean average precision(MAP), the precision at top 100 tweets (P@100), and the R-precision (R-PREC) as supplementary measures.

6.2.1 Comparative Study

We conduct extensive comparative study to verify the enhancement of the proposed query expansion technology. The retrieve models used in the comparative study include: (1) Lucene Baseline: It searches the original query in Lucene[1]; (2) PL2: It is a language modeling baseline by Terrier [26], which scores each document as the divergence from randomness, with Poisson estimation for randomness, Laplace succession for first normalization, and Normalization 2 for term frequency normalization. The default parameter is set to 10.99; (3) KLJM: The document-query score is computed as the KL-divergence between the document language model to the query model. We adopt JM smoothing for unobserved tokens, $p_s(q|\theta_d) = \lambda p(q|d) + (1-\lambda)p(q|C)$, $\lambda = 0.5$; (4) RLM: The recency language modeling baseline in [9], which promotes most recent tweets by adding a document prior; (5) Rocchio: It is the pseudo relevance feedback (PRF) query expansion according to the Rocchio formula. The initial results are obtained by RLM. We test the performance of all PRF methods with different numbers of pseudo-relevant tweets (from 20 to 50) and different numbers of expansion terms (from 5 to 20). Due to the limit of space, we only report the best performance generated by 10 expansion terms from 30 pseudo-relevant tweets. (6) BO1: Pseudo relevance feedback by Terrier, using the scoring model PL2, query expansion model BO1; (7) KL: Pseudo relevance feedback by Terrier, using the scoring model JMKL, query expansion by maximizing the KL-divergence of pseudo relevance to the collection; (8) MG: Dynamic pseudo relevance feedback (DPRF) with mixture Gaussian distribution, with $\alpha = 0.5$,$n = 4$, most 4 burst periods, scale parameter $\sigma = 5$. Top 30 documents retrieved by RLM on the original query are selected as pseudo relevance feedback; (9) LP: Dynamic pseudo relevance feedback (DPRF) with local power distribution, with $n = 5$ burst periods, scale parameter $\sigma = 1$. We do not limit the effect range R. Other settings are the same as MG; (10) SL: Dynamic pseudo relevance feedback (DPRF) with skewed linear distribution, with $n = 2$ burst periods, scale parameter $\sigma = 7$. We set $R = 2$, by making the assumption that each burst period lasts for 4 days. Other settings are the same as MG and LP.

[1] http://lucene.apache.org/

From Table 2, we have the following observations. 1) Among the four baselines, recency language model performs best, which validates the importance of introducing a non-uniform document prior in microblog retrieval. Language modeling approaches outperform naive Lucene baseline. PL2 outperforms KLJM, because PL2 favors longer tweets, while longer tweets are usually of higher quality. 2) PRF can increase the system accuracy, compared with their original scoring functions. However, the significance of the increment depends on how well the baseline performs. 3) Dynamic pseudo relevance feedback method performs best in terms of P@30, MAP, and R-PREC, no matter which distribution is used in quantifying the pseudo relevance prior. It nearly boosts the P@30 performance of Lucene baseline for two times. It also outperforms the recency based method by nearly 20%, and the traditional PRF by nearly 40%. Significance tests show that all the DPRF methods outperforms the best comparative method RLM with a confidence level larger than 99% (denoted as ++ in Table 2), compared with the null hypothesis that DPRF is equivalent with RLM.

6.2.2 Parameter Tuning

In this subsection, we compare the P@30 performance of various parameters for the three strategies used in dynamic pseudo relevance feedback. The effects of the number of detected burst periods n and the scale parameter σ are shown in Figure 5. We have the following observations. 1) Generally, dynamic pseudo relevance feedback is not extremely sensitive to parameters. The worst P@30 performance is higher than 0.405, which is better than all the baselines and PRF models. 2) An event is unlikely to have very few burst periods. Therefore $n = 1$ and $n = 2$ usually perform worst. However, since the corpus only consists of tweets published within 17 days, large n does not perform good. Appropriate value is $n = 4$ or $n = 5$. 3) For a fixed n, the best scaling parameter is between $\sigma = 3$ and $\sigma = 5$ for mixture Gaussian distribution. The performance decreases when σ is too small or too large. Note that a burst period usually lasts no more than 3 days, therefore the scaling parameter is likely to fit in a burst period. 4) For local power distribution, the bigger σ is, the more accurate the expansion terms are. Note that big σ indicates smooth decay in local power distribution. Similarly, for skewed linear distribution, the smaller σ is, the higher performance can be achieved. Therefore it is safe to claim that the information propagation process in microblogosphere is a long process.

6.3 Summarization Capability

Note that after retrieving the relevant tweets, various document summarization methods can be adapted to form the storyline by extracting the most relevant tweets. In this section, we conduct experiments to compare the summarization performance of different approaches including our proposed one, aiming to show the advantages of using the Dominant Set and the Steiner Tree to generate the storyline from the summarization aspect.

The measurement used in this subsection is mainly based on Recall-Oriented Understudy for Gisting Evaluation (ROUGE) – an evaluation toolkit for document summarization [20] which automatically determines the quality of a summary by comparing it with the human generated summaries through counting the number of their overlapping textual units (e.g., n-gram, word sequences, and etc.). In particular, F-measure scores of ROUGE-2 and ROUGE-SU4 are presented for our experiments. 49 queries provided by TREC 2011 microblog track are used in the experiments. For each query, first, DPRF is utilized to retrieve the top 1,000 tweets, then 8 students are invited to manually generate the "storyline" (50 tweets are selected) from these 1,000 tweets as the ground truth.

6.3.1 Comparison on Different Summarization Approaches

We compare our method with several well-known and recent summarization approaches including:

1. Random: randomly selects the sentence as the summary;

2. MostRelevant: picks up the sentences which are most relevant with the topic as the summary;

3. Latent Semantic Analysis (LSA): identifies semantically important sentences by conducting latent semantic analysis;

4. K-means: performs K-means over the sentences, then treats centers of all sentence clusters as the summary;

5. Non-negative Matrix Factorization (NMF) [17]: performs NMF on the sentence-term matrix and selects the high ranked sentences.

6. Symmetric Non-negative Matrix Factorization (SNMF) [35]: calculates sentence-sentence similarities by sentence level semantic analysis, clusters the sentences via symmetric non-negative matrix factorization, and extracts the sentences based on the clustering result;

7. Spectral Clustering with Normalized Cuts (NCut) [31]: performs the Spectral Clustering using Normalized Cut to cluster the sentences, and then uses centers of clusters as the summary;

8. Query-sensitive Mutual Reinforcement Chain (Qs-MRC) [37]: extends the mutual reinforcement principle between sentence and term to document-sentence-term mutual reinforcement chain, and uses query-sensitive similarity to measure the affinity between the pair of texts;

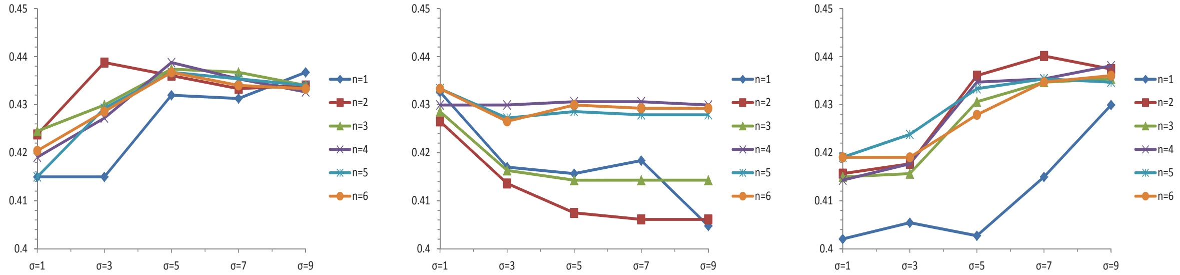
9. Multi-Document Summarization using Submodularity (MSSF) [18]: a multi-document summarization framework based on Submodularity;

10. Dominant Set (DS only): Document summarization using the Dominant Set algorithm (i.e., Algorithm 1 in Section 5).

The comparison of our proposed method (DS+ST) with other summarization methods is presented in Table 3. It can be seen from the results that our proposed DS+ST outperforms all the other summarization methods. In addition to the comparison of DS+ST against the other summarization methods, we employ the standard t-test to determine whether the performance improvement of DS+ST over the others is statistically significant. The results show that the improvements of our DS+ST on both ROUGE2 and ROUGE-SU are significant.

Type	Model	P@30	P@100	MAP	R-PREC
Baseline	Lucene	0.2306	0.1582	0.1697	0.2257
	PL2	0.3054	0.2065	0.1697	0.2245
	KLJM	0.2986	0.1900	0.2299	0.2614
	RLM	0.3837	0.1151	0.1804	0.2297
PRF	Rocchio	0.3551	0.1065	0.1727	0.2182
	Bo1	0.3265	0.2259	0.2560	0.2961
	KL	0.3041	0.2102	0.2364	0.2781
DPRF	MG	0.4388(++)	0.1890(++)	0.2547(++)	0.3055(++)
	LP	0.4333(++)	0.1900(++)	**0.2588**(++)	**0.3206**(++)
	SL	**0.4401**(++)	0.1900(++)	0.2549(++)	0.3038(++)

Table 2: Average P@30, P@100, MAP and R-PREC value of various retrieval models and query expansion methods, ++ indicates the proposed DPRF methods significantly outperforms the best comparative methods with a confidence level greater than 99%



(a) (b) (c)

Figure 5: P@30 performance of various parameters for (a) Mixture Gaussian Distribution; (b) Local Power Distribution; (b) Skewed Linear Distribution

Methods	ROUGE2	ROUGE-SU
Random	0.0425	0.0903
MostRelevant	0.0526	0.1075
LSA	0.0403	0.0857
K-means	0.0489	0.1002
NMF	0.0534	0.1043
SNMF	0.0593	0.1203
NCut	0.0635	0.1156
Qs-MRC	0.0647	0.1255
MSSF	0.0639	0.1324
DS Only	0.0731	0.1280
DS+ST	0.0895(++)	0.1363(+)

Table 3: The comparison among different summarization methods. Notice that "DS" denotes "Dominant Set", and "ST" represents "Steiner Tree". "++" and "+" indicate that DS+ST significantly outperforms the best comparative methods with a confidence level greater than 99% and 95%, respectively.

The good results of our method benefit from the following two aspects. (1) The Dominant Set algorithm (i.e., Algorithm 1) used in our method can select tweets which are similar to both the given query and all the other tweets. Thus it is not only good at extracting the representative information from the given sentences to form a reasonable summary, but also providing an appropriate mechanism to select the "dominant" nodes to generate storylines. (2) The Steiner Tree algorithm (i.e., Algorithm 2) is capable of detecting the "outline" of all the given sentences from the dominant nodes. Thus comparing with the other traditional summarization methods, it is able to generate more natural and logical storylines/summaries.

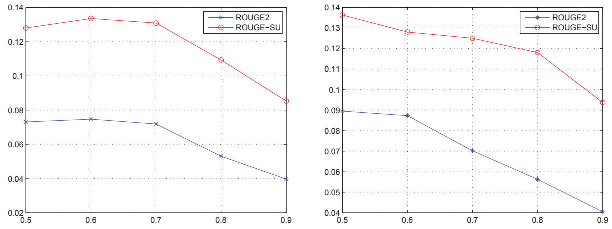
As a result, by combining the Dominant Set algorithm and the Steiner Tree algorithm, our proposed method is suitable for generating the "storyline" from the messages delivered by microblog services.

6.3.2 Parameter Tuning

In addition to the above comparison with different methods, we further study the summarization results by tuning the parameters of the Dominant Set algorithm and the Steiner Tree algorithm.

First of all, we study the Dominant Set algorithm by varying the "similarity threshold". We vary the threshold for the similarity between each tweet and the given query from 0.5 to 0.9 with a step size of 0.1 (totally 5 steps).

Secondly, one may notice that a key step before performing the Steiner Tree algorithm is to pick up a ***root*** node. A good root node could start a good story from tweets. In general, a good root should satisfy two conditions: 1) it should start as early as possible in terms of the post date of the tweet; 2) it should be similar to the given query. Usually, we choose the earliest node within the Dominant set as the root. However, we also study how the "similarity to the given query" influences the final summarization results. In other words, the earliest node may not necessarily be the root, but a later node from which every node of the Dominant set is reachable in graph G can be the root as long as it is more similar to the given query. To choose the root, we vary the similarity to the given query from 0.5 to 0.9 by a step size of 0.1. The comparison results by tuning the parameters are shown in in Figure 6(a) and 6(b).



(a) (b)

Figure 6: (a) Similarity (between a node and the given query) threshold; (b) Similarity between Root and Query

We have two observations from Figure 6(a). (1) The selection of the similarity threshold does influence the summarization performance of the Dominant Set algorithm. An inappropriate similarity threshold may weaken the Dominant Set greatly. (2) It is hard to claim that a larger similarity threshold would result in a better performance. In fact, when the similarity threshold is greater than 0.6, the summarization performance decreases as the threshold increases. The intuitive explanation is that a large similarity threshold may induce the algorithm to omit some important tweets which are not similar enough to the given query.

The observation from Figure 6(b) is that as the similarity to the given query increases, the summarization performance on both ROUGE2 and ROUGE-SU keeps going down. By analyzing it, we find that a large similarity threshold could lead to a "late" root. For example, a "late" root may exactly match the query, however, it would start the story from the middle of the whole storyline. In such a case, the tweets before the storyline's middle point are omitted, thus the evolving structure of the storyline is not well maintained.

6.4 A User Study

Since storyline generation is a subjective process, to better evaluate the retrieved tweets and the generated storylines, we conduct a user survey. The subjects of the survey are 18 students at different levels and from various majors of a research university. In this survey, we randomly sample 10 queries and 500 English tweets. Each participant is asked to read these tweets and 3 queries, and compare the results of different systems in a random order from the following point of views: relevance, coverage, coherence, and overall satisfaction. A score of 1 to 5 needs to be assigned to each system according to the user's satisfaction of the results. A rank of 5 (or 1) indicates that the result of the system is the most (or least) satisfactory. We implement the following systems for comparison. (1) Top10-Recency: presents the top 10 retrieved tweets by the recency language model RLM on the original queries. (2) Top10-DPRF: presents the top 10 retrieved tweets using the DPRF query expansion. (3) RecencySum: performs document summarization based on the retrieved tweets using the recency language model. MSSF is used as the document summarization method since it obtains the best results in Section 6.3.1. (4) DPRFSum: performs MSSF based on the retrieved tweets using the DPRF query expansion. (5) RecencyTimeline: generates timeslines [39] based on the retrieved tweets using the recency language model. (6) DPRFTimeline: generates timelines based on the retrieved tweets using DPRF query expansion. (7) RecencyStoryline: generates storylines using the methods proposed in Section 5 based on the tweets retrieved by the recency language model. (8) DPRFStoryline: generates storylines based on the tweets retrieved by DPRF query expansion.

	Relevance	Coverage	Coherence	Overall
Top10-Recency	3.06	1.67	1.50	2.06
Top10-DPRF	3.39	1.83	1.56	2.28
RecencySum	2.94	2.33	2.39	2.72
RecencyTimeline	3.06	3.06	2.83	3.33
RecencyStoryline	3.06	3.78	4.00	3.78
DPRFSum	3.22	2.50	2.44	3.05
DPRFTimeline	3.33	3.33	3.06	3.83
DPRFStoryline	**3.39**	**4.17**	**4.28**	**4.12**

Table 4: Survey Results: User ratings on different systems based on their satisfaction

Table 4 shows the user rated scores for each system. From the results, we have observations as follows. (1) The performance of tweet retrieval is critical. The proposed DPRF query expansion approach outperforms the recency language method. (2) Although the listed top 10 query results are highly relevant to the query, there also exists high redundancy among the top-ranking query results, thus the coverage and coherence of the results are poor. (3) Summarization based results achieve higher overall satisfaction than the methods of listing top query results because it can help users better understand the tweets. (4) Users prefer structured results such as timelines and storylines than pure text summaries. (5) The proposed storyline generation methods outperform the timeline generation method because the structures contained in the storylines can assist users quickly grasp the event evolution.

7. CONCLUSION

Generating storylines from microblogs can shed insight into several fields, including event detection, short text mining, and text stream mining, etc. The proposed dynamic pseudo relevance feedback model is embedded in a sophisticated theoretical framework. Experiments show that it is robust to parameters. The heuristic strategy for finding minimum weighted Steiner tree on a dominant set of relevant tweets is efficient to produce summary and evolvement structure.

This is a pioneer work on generating storylines from social media. In the future, we will further improve the approaches in the following aspects. First, the density function in Section 4 attempts to model the event prior. However since the relevance set is usually small, the assumption may not be satisfied by the observation. Second, in our work, the number of burst periods is pre-fixed in the detection. Advanced burst period detection methods can be investigated and incorporated into our current framework. Last but not least, our current storyline generation is based on multi-view tweet graph and we plan to investigate new frameworks for generating concise and coherent storylines.

Acknowledgement

Chen Lin is partially supported by Shanghai Key Laboratory of Intelligent Information Processing under grant No. IIPL-2011-004, China Natural Science Foundation under grant No.61102136,61001013, the Natural Science Foundation of Fujian Province of China under grant No.2011J05158), the Fundamental Research Funds for the Central Universities under Grant No.2011121049. Jingxuan Li and Tao Li are partially supported by US NSF grants IIS-0546280, HRD-0833093, and CNS-1126619 and DHS grants 2009-ST-062-000016 and 2010-ST-062-000039.

8. REFERENCES

[1] J. Allan Introduction to Topic Detection and Tracking. In *Topic Detection and Tracking 2002(12)*, pages 1-16.

[2] N. Bansal and N. Koudas. Blogscope: a system for online analysis of high volume text streams. In *Proceedings of VLDB 2007*, pages 1410–1413.

[3] A. Bermingham and A. F. Smeaton. Classifying sentiment in microblogs: is brevity an advantage? In *Proceedings of CIKM 2010*, pages 1833–1836.

[4] G. Cao, J.-Y. Nie, J. Gao et al. and S. Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of SIGIR 2008*, pages 243–250.

[5] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *Proceedings of WWW 2011*, pages 675–684.

[6] C. Chen, F. Li, B. C. Ooi, et al. and S. Wu. Ti: an efficient indexing mechanism for real-time search on tweets. In *Proceedings of SIGMOD 2011*, pages 649–660.

[7] M. Efron. Hashtag retrieval in a microblogging environment. In *Proceedings of SIGIR 2010*, pages 787–788.

[8] M. Efron. Information search and retrieval in microblogs. *J. Am. Soc. Inf. Sci. Technol.*, 62:996–1008, June 2011.

[9] M. Efron and G. Golovchinsky. Estimation methods for ranking recent information. In *Proceedings of SIGIR 2011*, pages 495–504.

[10] J. Hannon, M. Bennett, and B. Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of RecSys 2010*, pages 199–206.

[11] R. Jones and F. Diaz. Temporal profiles of queries. *ACM Trans. Inf. Syst.*, 25, July 2007.

[12] J. Kleinberg, Bursty and Hierarchical Structure in Streams In *Data Mining and Knowledge Discovery 2003(7)*, pages 373-397.

[13] R. Kumar, U. Mahadevan, and D. Sivakumar. A graph-theoretic approach to extract storylines from search results. In *Proceedings of SIGKDD 2004*, pages 216–225.

[14] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of WWW 2010*, pages 591–600.

[15] T. Lappas, B. Arai, M. Platakis, D. Kotsakos, and D. Gunopulos. On burstiness-aware search for document sequences. In *Proceedings of SIGKDD 2009*, pages 477–486.

[16] V. Lavrenko, J. Allan, E. DeGuzman, D. LaFlamme, V. Pollard, and S. Thomas. Relevance models for topic detection and tracking In *Proceedings of HLT 2002*, pages 115–121.

[17] D. Lee and H. Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2001.

[18] J. Li, L. Li, and T. Li. Mssf: a multi-document summarization framework based on submodularity. In *Proceedings of SIGIR 2011*, pages 1247–1248.

[19] X. Li and W. B. Croft. Time-based language models. In *Proceedings of CIKM 2003*, pages 469–475.

[20] C. Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, pages 25–26.

[21] F. ren Lin and C.-H. Liang. Storyline-based summarization for news topic retrospection. *Decision Support Systems*, 45(3):473 – 490, 2008.

[22] K. Massoudi, M. Tsagkias, M. de Rijke, and W. Weerkamp. Incorporating query expansion and quality indicators in searching microblog posts. In *Proceedings of ECIR 2011*, pages 362–367.

[23] M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the twitter stream. In *Proceedings of SIGMOD 2010*, pages 1155–1158.

[24] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *Proceedings of SIGKDD 2005*, pages 198–207.

[25] S. Morinaga and K. Yamanishi. Tracking dynamics of topic trends using a finite mixture model. In *Proceedings of SIGKDD 2004*, pages 811–816.

[26] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proceedings of ACM SIGIR-OSIR Workshop 2006.*

[27] W. Ruzzo and M. Tompa. A linear time algorithm for finding all maximal scoring subsequences. In *Proceedings of ISMB 1999*, pages 234–241.

[28] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of WWW 2010*, pages 851–860.

[29] D. A. Shamma, L. Kennedy, and E. F. Churchill. Peaks and persistence: modeling the shape of microblog conversations. In *Proceedings of CSCW '11*, pages 355–358.

[30] C. Shen and T. Li. Multi-Document Summarization via the Minimum Dominating Set. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, 2010.

[31] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[32] H. Takamura, H. Yokono, and M. Okumura. Summarizing a document stream. In *Proceedings of ECIR 2011*, pages 177–188.

[33] J. Teevan, D. Ramage, and M. R. Morris. #twittersearch: a comparison of microblog search and web search. In *Proceedings of WSDM 2011*, pages 35–44.

[34] D. Wang, T. Li, and M. Ogihara. Generating Pictorial Storylines via Minimum-Weight Connected Dominating Set Approximation in Multi-View Graphs. In *Proceddings of AAAI 2012*.

[35] D. Wang, T. Li, S. Zhu, and C. Ding. Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In *Proceedings of SIGIR 2008*, pages 307–314.

[36] D. Wang, L. Zheng, T. Li, and Y. Deng. Evolutionary document summarization for disaster management. In *Proceedings of SIGIR 2009*, pages 680–681.

[37] F. Wei, W. Li, Q. Lu, and Y. He. Query-sensitive mutual reinforcement chain and its application in query-oriented multi-document summarization. In *Proceedings of SIGIR 2008*, pages 283–290.

[38] Y. Xu, G. J. Jones, and B. Wang. Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of SIGIR 2009*, pages 59–66.

[39] R. Yan, X. Wan, J. Otterbacher, L. Kong, X. Li, and Y. Zhang. Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In *Proceedings of SIGIR 2011*, pages 745–754.

[40] W. Yih, J. Goodman, L. Vanderwende, and H. Suzuki. Multi-document summarization by maximizing informative content-words. In *Proceedings of IJCAI*, 2007.

[41] C. Zhai. Statistical language models for information retrieval a critical review. *Foundations and Trends in Information Retrieval*, 2(3):137–213, 2008.

[42] D. Zhang, Y. Liu, R. D. Lawrence, and V. Chenthamarakshan. Transfer latent semantic learning: Microblog mining with less supervision. In *Proceedings of AAAI 2011*, pages 561–566.