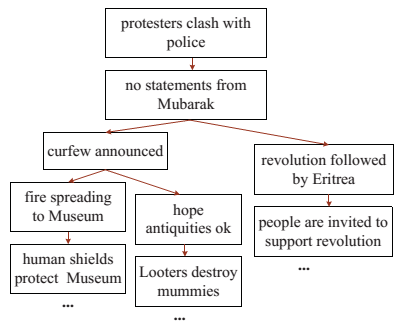
**Generating Event Storylines from Microblogs(GESM)**

Chen Lin , Chun Lin , Jingxuan Li , Dingding Wang , Yang Chen , Tao Li

Summary submitted by *AnunayaSrivastava*

The goal is to generate a storyline, which is chronologically ordered chain of related events, from microblogs(Tweets) for the given user query. The user query is a set of user defined keywords or phrases describing an ongoing event in real life. The generated storyline is a graph structure where each node is labelled by a summaryof an individual phase of the event and each edge represents causal relationship between two phases.

The author has used microblogs for storyline generation as users prefer to fire event queries on Twitter to obtain information about an ongoing event. It would be helpful if we have a framework that can automatically generate a skeleton of an event requested by a user. An example of storyline for the query ‘Egypt Revolution’ is as follows.



GESM is different from Topic detection and tracking(TDT) and previous algorithms used for storytelling. TDT is event-based organization of news to detect emergence of new events and monitor their development with time. The difference is that the previous works were designed for well edited texts like news articles, and not for mining microblogs which contain short and noisy data. Mining microblogs is challenging because microblogs are dynamic and sparse. Secondly, event queries contain basic description terms like location of event, person name etc. while a microblog about the query may not contain any query related term. Thirdly, there can be numerous duplicate tweets and retweets. Hence, it is not possible to apply traditional text summary algorithms to mine microblogs. The quality of a storyline is determined by the quality of summary in each phase and the quality of phase segmentation. Author claims that his method is able to produce a good quality summary for each phase.

The main contributions of this work are – (1) A novel problem of generating storyline from microblogs is proposed. (2) A dynamic pseudo relevance feedback (DPRF)languagemodel is presented to retrieve relevant tweets given an eventquery.DPRF will be explained latter. (3) Storyline generation problem is formulated as a graph-based optimization problem and is solved using MWDS and Steiner tree approximation algorithm which will be discussed later.

The approach used can be described as follows –

1. **Relevant tweet retrieval using DPRF**

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 θF, we can set the new query to be a linear combination of original query Q and relevant model θF



Relevance model method is followed to infer θF



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.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 centredaround each burst period.Dynamic Pseudo Relevance Feedback(DPRF) is dynamic i.e. prior probability of relevant document is given by a probability distribution. The author has suggested 3 different probability distributions for prior probability which I am not mentioning here for the sake of simplicity.

1. **Summarization**

The author has compared 10 summarization approached including Minimum Weight Dominant Set(MWDS) algorithm which he has used in this paper. He claims that MWDS gives the best results.

MWDS summarization approach can be shown as –

**Definition: MULTI-VIEW TWEET GRAPH**

*A multi-view graphis a quadruple 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, τ1,τ2, τ1<τ2. There is edge from vi to vj iff tweet similarity is greater than α and   
τ1<= tj – ti<= τ2. ti and tj are the timestamps of vi and vj .Similarity between user query Q and a vertex vi is given by score(Q,vi) which is calculated using cosine similarity. Vertex weight, w(vi) = 1 – score(vi).

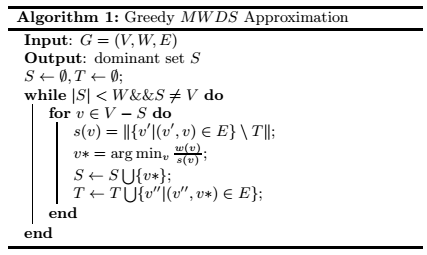
**MWDS Algorithm**

Input: Graph G=(V,W,E)

Output: Minimum Weight Dominant Set

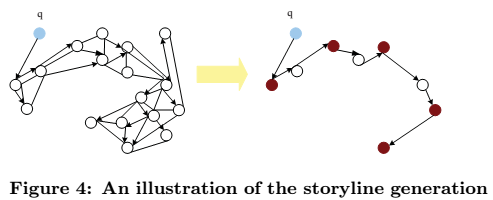
MWDS is a greedy approximation algorithm to find a minimum weight dominant set. A dominant set(DS) of a graph G is a set of vertices such that every vertex either belongs to DS or is adjacent to a vertex in DS. This algorithm states that the weight of a newly added vertex is shared among its newly covered neighbours and selects the node which minimizes this load for each round of iteration.

I am giving the algorithm below your reference. No further explanation is being given as it will discussed in detail in another review document.



1. **Storyline Generation**

Once the most representative summary for each phase is selected using MWDS the next step is to form a storyline using capturing the temporal and structural information of event-related tweets.



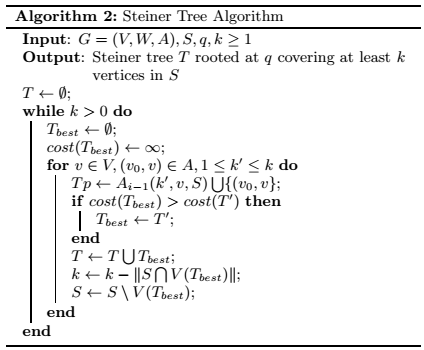
**Definition: STEINER TREE**

Given a directed graph G=(V,W,A), a set S of vertices(terminals), and a root q ∈ S from which every vertex of S is reachable in G, Steiner tree is the sub-tree of G rooted at q containing S with the smallest total vertex weight.

**Steiner Tree Algorithm, Ai(G,k,q,S)**

Input: G=(V,W,A),S(MWDS generated in algorithm 1),q,k≥1

Output: Steiner tree Trooted atqcovering at leastkvertices inS



q is set to the vertex with the earliest timestamp. k is set to the size of S. k is a level parameter i = 1 is a default case where straightforward algorithm selects k vertices closest to root and returns the union of the shortest paths. The output of the tree is interpreted as the storyline transitioning from the root vertex to all the vertices in S.

The detailed working of the algorithm is mentioned in ‘Moses Charikarand Chandra Chekuri, 1999, Approximation Algorithms for DirectedSteiner Problems’.