Timeline Generation with Social Attention

Wayne Xin Zhao[1], Yanwei Guo[1], Rui Yan[1], Yulan He[2] and Xiaoming Li[1]
[1]School of Electronic Engineering and Computer Science,Peking University, China
[2]School of Engineering & Applied Science, Aston University, UK
batmanfly@gmail.com, pkuguoyw@gmail.com, rui.yan.peking@gmail.com, y.he@cantab.net, lxm@pku.edu.cn

ABSTRACT

Timeline generation is an important research task which can help users to have a quick understanding of the overall evolution of any given topic. It thus attracts much attention from research communities in recent years. Nevertheless, existing work on timeline generation often ignores an important factor, the attention attracted to topics of interest (hereafter termed "social attention"). Without taking into consideration social attention, the generated timelines may not reflect users' collective interests. In this paper, we study how to incorporate social attention in the generation of timeline summaries. In particular, for a given topic, we capture social attention by learning users' collective interests in the form of word distributions from Twitter, which are subsequently incorporated into a unified framework for timeline summary generation. We construct four evaluation sets over six diverse topics. We demonstrate that our proposed approach is able to generate both informative and interesting timelines. Our work sheds light on the feasibility of incorporating social attention into traditional text mining tasks.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Performance, Evaluation

Keywords

Timeline, social media attention, user interest

1. INTRODUCTION

Timelines [3, 1, 5] provide temporal summaries of the evolution of news stories related to a topic, which are often desirable for users who do not closely follow news and want to quickly gain an overall picture of the major events related to a topic. Typically, sentences which describe major events are extracted in chronological order to form timeline summaries. We show an example timeline presentation of "Nobel Prize" in Table 1 created by professional editors[1].

Recently, Yan et al. proposed a task of automatically generating evolutionary timeline summaries [5, 4]. They formally formulated the task as an optimization problem via iterative sentence substitution in order to maximize the objective function with four factors: relevance, coverage, coherence and cross-date diversity [5]. In [4], Yan et al. further extended the work by modeling inter-date and intra-date dependencies between timestamped sentences, and incorporating these two kinds of dependencies into a sentence ranking function.

The timeline summaries generated in [5, 4] are mainly based on semantic information and do not necessarily reflect the public attention an event has attracted. In Table 1, we can observe that not all the events described by the summary sentences receive equal social attention, as evidenced by the number of related tweets listed in the "Hotness" column. Since the timeline summaries are usually kept succinct, it is desirable to present sentences which are likely to attract the attention of the majority of users. With today's social web, it is possible to obtain the social attention signals from social media content such as Twitter feeds.

In this paper, we study how to capture social attention and incorporate it into the generation of timeline summaries. The "Hotness" column in Table 1 reveals that there is a very skewed social attention distribution over the events described by the summary sentences in the example timeline. Millions of users engage in a diverse range of activities on the social web such as posting status messages and interacting with items generated by others, for example, forwarding messages. These activities are often interest driven. Thus learning from online social media could be a good way to capture social attention. In our proposed method, users' collective interests are learned from the Twitter data, and are represented as pseudo sentences. We run a modified graph-based method to propagate collective interest biased scores, which represent a trade-off between informativeness and interestingness, i.e., users' collective interests.

To the best of our knowledge, it is the first study which utilizes social attention to improve both the informativeness and interestingness of timeline summaries. We evaluate our method on four datasets constructed from Twitter and compare it with a number of state-of-the-art timeline generation methods. Our experimental results show that by considering social attention, the proposed method for timeline summary generation gives better performance in terms of both informativeness and interestingness.

2. PROBLEM FORMULATION

For our timeline summarization problem, we assume the following input data are available.

Time span: A time span $I = (t_s, t_e)$ is defined by a start time t_s and an end time t_e.

Query: A user issues a query $\mathcal{Q} = \{q_1, q_2, \ldots, q_{|\mathcal{Q}|}\}$ within I to the timeline summarization system, where q_i denotes a query word in the vocabulary $\mathcal{V}$.

News articles: We assume that news articles related to $\mathcal{Q}$ within I have been retrieved. We represent these relevant news articles as a set of sentences $\mathcal{C}$. Each sentence in $\mathcal{C}$ has a timestamp (e.g.

[1]http://timelinesdb.com

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

Table 1: Timeline summaries between July and December 2009 on “Nobel Prize" generated by professional editors.

Date	Summaries	Hotness*
Jul. 19	Frank McCourt, former NYC teacher and Irish-born author,died of cancer.	512
Aug. 18	Former South Korean Pres. Kim Dae-jung (85) died.	199
Oct. 6	Hilary Mantel won the 2009 Man Booker Prize for her historical novel “Wolf Hall."	450
Oct. 8	Herta Mueller won the Nobel Prize in literature.	362
Oct. 9	The Nobel Peace Prize was awarded to US President Barack Obama.	21760
Dec. 10	In Oslo(Norway), President Barack Obama accepted the Nobel Peace Prize	2448

*Here “Hotness" is measured by # of related tweets (retrieved by manually generated query keywords) in our Twitter dataset and not part of the timeline.

the publish date) and $\mathcal{C}_t$ denotes a collection of sentences at the tth time point.

Tweets: We also assume that we can retrieve a set of tweets published within I and relevant to $\mathcal{Q}$. We use $\mathcal{C}'$ to represent these tweets. Similarly, each tweet in $\mathcal{C}'$ also has a timestamp.

Given the input specified above, the system is expected to generate the following output: for each time point $t \in I$, the system generates a summary $\mathcal{R}_t$, which consists of a set of *news sentences* from $\mathcal{C}_t$. All the generated summaries are expected to capture the most important information and meanwhile attract considerable social attention about the topic within I. It is worth noting that we do not select sentences from tweets since our aim here is to generate a timeline summary which has a good coverage of the most prominent events happened related to a specific topic. We only use tweets to derive social attention signals.

3. OUR PROPOSED ALGORITHM

We propose a novel approach to capture social attention from tweets through learning users' collective interests by a generative mixture model. We then show how to incorporate the learned collective interests into a state-of-the-art timeline generation algorithm.

3.1 Learning Users' Collective Interests

Existing timeline generation methods only consider news streams. Hence, the timeline summaries generated do not reflect users' collective interests in the real world. We propose to learn users' collective interests from the Twitter. We assume that we can obtain a set of relevant tweets relating to users' queries at each time point in a time span I. It is worth mentioning that we do not attempt to discover individual user's interests, instead, we aim to identify the most prominent collective interests of the majority.

Formally, let $\mathcal{C}'_t$ denote all the *tweets* at time point t, our aim is to learn a collective interest model $\theta_{U,t}$ from $\mathcal{C}'_t$.[2] We notice that tweets are often noisy and contain many background words (*home, good, time, lol, love, etc.*) which are not relating to users' topical interest with respect to a given query . Thus, words in $\mathcal{C}'_t$ can be clustered into two groups, words closely related to a specific query topic and general background words. We assume that words in $\mathcal{C}'_t$ are generated either from a model θ_U which represents users' collective topical interest or from a general background model θ_B. With this model, we can reduce the effects of background words and learn a model which better captures words concentrating around users' collective interests. We further assume that both θ_U and θ_B are represented as multinomial distributions over the vocabulary. This kind of two-mixture model has been shown to be effective in pseudo relevance feedback for information retrieval [6]. With the two-mixture model, we have

$$p(\mathcal{C}'_t|\theta_U,\theta_B) = \prod_{i=1}^{|\mathcal{C}'_t|}\Big((1-\eta)p(w_i|\theta_U) + \eta p(w_i|\theta_B)\Big), \qquad (1)$$

[2]To simplify notation, we represent $\theta_{U,t}$ as θ_U by dropping the time index.

where η is the probability (weight) that a word w_i in $\mathcal{C}'_t$ is generated from the general background model. A large η tends to make the interest-related language model more discriminative because more general words are generated from the background model. In practice, we can set η empirically and fix it during the learning process. Let $c(w, \mathcal{C}'_t)$ be the count of word w in $\mathcal{C}'_t$ and $\mathcal{V}$ be the vocabulary of the corpus, we then have

$$\log p(\mathcal{C}'_t|\theta_U,\theta_B) = \sum_{w\in\mathcal{V}} c(w,\mathcal{C}'_t)\log\Big((1-\eta)p(w|\theta_U) + \eta p(w|\theta_B)\Big).$$

Usually the background model θ_B can be estimated directly from the entire tweet collection using maximum likelihood estimation. After deriving θ_B and fixing η, we can use the expectation maximization (EM) algorithm to estimate the collective interest language model. The updating formulae of the E-step and the M-step are shown below:

$$\text{E-step: } \alpha^{(n+1)}(w) = \frac{(1-\eta)p^{(n)}(w|\theta_U)}{(1-\eta)p^{(n)}(w|\theta_U) + \eta p^{(n)}(w|\theta_B)},$$

$$\text{M-step: } p^{(n+1)}(w|\theta_U) = \frac{c(w,\mathcal{C}'_t)\alpha^{(n+1)}(w)}{\sum_{w'\in\mathcal{V}} c(w',\mathcal{C}'_t)\alpha^{(n+1)}(w')}.$$

After obtaining the interest-related model θ_U, we can easily transform it into a vector by simply setting the weight of each word dimension to the word's corresponding probability in θ_U. Such a vector can be treated as a pseudo sentence which describes users' collective interests at a time point t.[3]

3.2 Timeline Summary Generation

We will first give a brief introduction of a timeline generation algorithm, denoted as ETTS, which was proposed in [4]. Then we will present how to incorporate the learnt collective interests into ETTS. It is noteworthy that our proposed method can be easily incorporated into any other graph based summarization algorithms.

Given a news collection $\mathcal{C}$ partitioned according to the start/end time specified in a time span I, i.e. $\mathcal{C} = \{\mathcal{C}_t\}_{t\in I}$, we split each news article into sentences for each $\mathcal{C}_t$. At a time point t, we refer to sentences with timestamp t as local sentences and sentences with other timestamps as global sentences. ETTS models the inter-date and intra-date dependencies between the timestamped sentences and incorporate these two types of correlations into a sentence ranking function.

Specially, ETTS constructs two probability transition matrices. One is for modeling global affinity and the other is for modeling local affinity. For global affinity, it means that summary sentences should be correlative with sentences from neighboring dates to capture the overall topic evolution patterns, i.e., inter-date dependency. To allow the modeling of global affinity, global sentences are temporally projected onto time point t such that links between local and global sentences can be built. For local affinity, it means intra-date dependency, i.e., the timeline summary at time point t should be informative within $\mathcal{C}_t$, and links are built between local sentences. With these two transition matrices, both the global and local affinity propagation can be run using the standard LexRank algorithm. At time point t, a set of global ranking $\mathcal{R}_g = \{r_i^{(g)}\}$ and local ranking $\mathcal{R}_l = \{r_i^{(l)}\}$ of sentences in $\mathcal{C}_t$ are obtained. ETTS then uses an optimization algorithm to combine both the global and the local rankings, and the final ranking of sentence $s_i \in \mathcal{C}_t$ is a weighted combination between its global ranking and local ranking

[3]Note that depending on how we set the time point, there might be very few tweets relating to a query topic at some time points, e.g., fewer than 100 tweets. In this case, we do not learn users' collective interests from tweets.

$$r_i = \frac{\alpha}{\alpha+\beta} r_i^{(g)} + \frac{\beta}{\alpha+\beta} r_i^{(l)}, \tag{2}$$

where $0 \leq \alpha$, $0 \leq \beta$. α and β can be tuned on different date sets to make a tradeoff between global scores and local scores. See [4] for a detailed description of ETTS.

Now we study how to incorporate users' collective interests into ETTS. Note that regardless of the modeling of either global affinity or local affinity, we can formulate both problems in a standard LexRank form

$$\boldsymbol{\lambda} = \mu \cdot \boldsymbol{\lambda} \cdot \mathbf{M} + (1-\mu) \cdot \boldsymbol{y}, \tag{3}$$

where $\boldsymbol{\lambda}$ is the saliency score vector of sentences, $\mathbf{M}$ is the transition probability matrix and $\boldsymbol{y}$ is the restart probability vector usually set to be uniform. The main idea is that instead of using a uniform restart distribution $\boldsymbol{y}$, we use an interest biased restart distribution. Recall that at each time point, we have modeled users' collective interests as a pseudo sentence. We can add a new vertex which represents such a pseudo sentence at that time point. It can be naturally incorporated into the above LexRank algorithm. We build the similarity links between the interest-related pseudo sentence and *all the local sentences* using the cosine similarity measurement. At the beginning of each iteration of LexRank, this pseudo sentence has a large restart probability to be visited. During the learning process, it gradually propagates the interest-related score to other similar vertices. Letting v_0 to denote the vertex representing users' collective interest, each entry of $\boldsymbol{y'} = [y_0, \boldsymbol{y}]$ can be modified as follows

$$y_i' = \begin{cases} \tau, & \text{if } i = 0, \\ \frac{1-\tau}{N}, & \text{otherwise.} \end{cases} \tag{4}$$

where τ is a positive factor and N is the number of candidate sentences. For v_0, i.e., the pseudo sentence representing users' collective interest, it has a large restart probability of τ, while any of the other vertices has a smaller restart probability of $\frac{1-\tau}{N}$. τ essentially controls the trade-off between informativeness and interestingness; the larger it is, the more emphasis we put on interestingness. We use τ_g and τ_l to denote the corresponding values of τ for global affinity ranking and local affinity ranking. In our experiments, we simply set $\tau_g = \tau_l$. In traditional LexRank, the ranking score only indicates informativeness, while our interest biased LexRank makes a trade-off between informativeness and interestingness.

4. EXPERIMENTS AND EVALUATION

4.1 Experimental Setup

Data collection. Since our summarization algorithm is query dependent, we selected six topics to cover a few important news events according to the Rule of Interpretation (ROI) category (Table 2). We then constructed corpora of news articles and tweets for each of the topics. For news articles, we submitted the topic queries into Google News[4] and crawled all the news articles from the three major sources: China Daily[5], New York Times and BBC News between July, 2009 and December, 2009. For tweets, we use the shared data set within the same time period.[6] Next we split news articles into sentences and filtered those news sentences and tweets with too many or too few word tokens. The statistics of the datasets is summarized in Table 2.

Gold standard generation. We manually constructed the gold standard for these six topics, including both the informativeness-oriented and the interestingness-oriented sets. *Informativeness- oriented set*, similar to the evaluation of timeline summaries [4] and traditional summarization [2], requires a system to return informative and relevant news sentences to cover important aspects given a query topic. We select various authoritative sources to generate the gold standard timelines, including mainstream news media (China Daily, BBC, New York Times) and the timeline Web database[7]. For each topic, we first extracted the timelines from the professional editors in at least two kinds of resources mentioned above. Then we invited a human judge to merge the information from different resources by removing redundant sentences. Finally, $25 \sim 35$ sentences were kept for each topic as gold standard. We denote this set as $\mathcal{D}$.

Table 2: Statistics of the datasets. We used the topic words in the first column as the queries to obtain relevant tweets and news sentences.

Topics	#news articles	#tweets	#Gold sentences	ROI category
Influenza A	1291	487258	35	Science, Finance
Climate Change	3006	174939	33	Science, Finance
Noble Prize	343	88816	30	Science, Politics
Flight Crash	612	231268	25	Accidents, Disasters
Earthquake	773	53118	27	Accidents, Disasters
Urumqi Riots	912	3867	25	Legal Cases, Politics

Since the informativeness-oriented set does not consider the social attention of the generated timelines, it may not reflect users' collective interests. We thus further constructed *interestingness-oriented sets* by removing less "interesting" sentences from $\mathcal{D}$. A sentence is considered to be interesting if it attracts a considerable amount of social attention. We invited 6 graduate students major in journalism for evaluation. Every volunteer was asked to remove top K least interesting sentences from $\mathcal{D}$ for each topic query. For each sentence in $\mathcal{D}$, a volunteer was required to refer to news portals (for report volume) and online social websites (for social attention) before making the judgement. We merged the results from six judges and reordered the sentences according to their total votes. To evaluate interestingness at different levels, we set K to 5, 10 and 15, for which we respectively removed the top 5, 10 and 15 sentences that the judges considered to be less interesting from $\mathcal{D}$. In such a way we ended up with another three gold standard sets, $\mathcal{D}_{-5}$, $\mathcal{D}_{-10}$ and $\mathcal{D}_{-15}$. For each topic, we computed the average pairwise agreement of six judges over the top 5/10/15 less interesting sentences, and obtained the values of 0.84/0.8/0.75 indicating good agreement.

Evaluation metrics. Following [5], we used the F scores of unigram-based ROUGE-1 (R-1), bigram-based ROUGE-2 (R-2), and the weighted longest common subsequence based ROUGE-W (R-W,W=1.2) as metrics.

Methods to compare. We used the following widely used multi-document summarization or timeline generation algorithms as the baseline systems.

1) *CHIEU*: Chieu et al. [1] presented a similar timeline system with a different methodology by utilizing burstiness ranking text feature. 2) *CENTROID*: The method applies MEAD algorithm (Radev et al., 2004) to extract sentences according to the following parameters: centroid value, positional value, and first-sentence overlap. 3) *LexRank*: LexRank [2] first constructs a sentence connectivity graph based on cosine similarity and then selects important sentences based on the concept of eigenvector centrality. 4) *LexRank*$_{+i}$: Since LexRank is a graph based method, we can also incorporate users' collective interests similarly. 5) *ETTS*: ETTS proposed in [4] is an algorithm with optimized a combination of global and local biased summarization. 6) *ETTS*$_{+i}$: our proposed algorithm, which incorporates users' collective interests into ETTS.

Setup details. For fairness of comparison, we performed the same preprocessing steps and adopted Maximal Marginal Relevance to reduce redundancy for all the aforementioned baselines. We partitioned the entire sentence collection $\mathcal{C}$ into local sentence collections according to the timestamps of each sentence, i.e., $\mathcal{C} =$

[4]https://news.google.com/

[5]http://english.peopledaily.com.cn/

[6]http://an.kaist.ac.kr/traces/WWW2010.html

[7]http://timelinesdb.com/

Table 3: Performance comparison on both informativeness and interestingness oriented data sets. $\mathcal{D}_{-5}$, $\mathcal{D}_{-10}$ and $\mathcal{D}_{-15}$ are used to evaluate interestingness at different levels. Their compression rates compared with $\mathcal{D}$ are about 83%, 67% and 50% respectively, with smaller rates indicating more emphasis on interestingness.

	Informativeness			Interestingness								
	$\mathcal{D}$			$\mathcal{D}_{-5}$			$\mathcal{D}_{-10}$			$\mathcal{D}_{-15}$		
Methods	R-1	R-2	R-W	R-1	R-2	R-W	R-1	R-2	R-W	R-1	R-2	R-W
CHIEU	0.250	0.053	0.086	0.239	0.051	0.086	0.227	0.050	0.087	0.201	0.041	0.083
CENTROID	0.253	0.057	0.087	0.238	0.054	0.085	0.225	0.052	0.087	0.197	0.044	0.083
LexRank	0.266	0.060	0.090	0.251	0.057	0.090	0.234	0.054	0.089	0.206	0.046	0.084
LexRank$_{+i}$	0.277 (+4.1%)	0.062	0.093	0.265 (+5.6%)	0.062	0.094	0.251 (+7.2%)	0.059	0.095	0.221 (+7.3%)	0.048	0.089
ETTS	0.277	0.067	0.095	0.259	0.061	0.093	0.245	0.060	0.093	0.212	0.046	0.087
ETTS$_{+i}$	0.282 (+1.8%)	0.070	0.095	0.268 (+3.5%)	0.066	0.095	0.256 (+4.5%)	0.064	0.097	0.224 (+5.7%)	0.055	0.093

$\cup_{i=1}^{T}\mathcal{C}_i$. Following [1], we applied a simple mechanism to select sentences by extracting more sentences for important dates while fewer sentences for others. The allocation rate on t_i was set to $\phi_i = \frac{|\mathcal{C}_i|}{|\mathcal{C}|}$. The total number of selected sentences for each query is set to the number of sentences in the gold standard.

The parameters of all the baselines and our proposed algorithm were tuned in a way similar to cross-validation. For each query, we first found the parameters which lead to the optimal performance on this query, and then applied the model learned with these parameters on the other five queries. And finally, we averaged results over six such runs. For η in the two-mixture model (Eq. 1), we find that a value in $0.8 \sim 0.95$ usually gives good performance, which can effectively remove noisy and background words. For the restart probability of the interest-related pseudo sentence τ, we found that a value in $0.4 \sim 0.6$ usually leads to an optimal performance for both LexRank and ETTS. A large value of τ will hurt the diversity of summary sentences while a small value of τ essentially ignores the effect of social attention.

4.2 Experimental Results

We present the results of various methods on both the informativeness and interestingness oriented evaluation sets in Table 3. To better check the improvement when incorporating social attention, we also present the relative improvement on R-1 for *LexRank*$_{+i}$ over *LexRank* and *ETTS*$_{+i}$ over *ETTS*.

Evaluation of informativeness. We first examine the results on the informativeness oriented set $\mathcal{D}$, which consists of summary sentences obtained from professional editors. We notice that the incorporation of users' collective interest yields improvement of informativeness, i.e. ETTS$_{+i}$ > ETTS and LexRank$_{+i}$ > LexRank. It indicates that users do read important sentences, and the incorporation of user interests does not hurt informativeness. A noteworthy point that users' collective interests may affect the diversity due to the fact that our method force the generated summaries to match collective users' interests.

Evaluation of interestingness. For the results on the three interestingness oriented sets, $\mathcal{D}_{-5}$, $\mathcal{D}_{-10}$ and $\mathcal{D}_{-15}$, we can see the improvement *LexRank*$_{+i}$ over *LexRank* and *ETTS*$_{+i}$ over *ETTS* becomes more significant with the decreasing number of summary sentences in the gold standards, $\mathcal{D}_{-5} < \mathcal{D}_{-10} < \mathcal{D}_{-15}$. This shows that taking into account of social attention, our proposed method is indeed able to generate timeline summaries more tailored to users' collective interests. The advantage of our method is more prominent when generating more succinct summaries. Finally, we notice that our proposed method is able to close the performance gap between LexRank and ETTS. For example, when tested on the $\mathcal{D}_{-5}$ set, ETTS outperforms LexRank by 3% ($\Delta = 0.006$) in the R-1 measure. But with our proposed method incorporated, the gap between ETTS$_{+i}$ and LexRank$_{+i}$ ($\Delta = 0.003$) is reduced to 1%.

To get an intuitive idea of why our method works, we present an example timeline generated by ETTS$_{+i}$ in Table 4. We list the top words ranked by probabilities learnt from the two-mixture user interest-related model. Clearly, these words are very meaningful and broadly reflect the social attention at that period. With them as supervision, the graph based summarization methods tend to give higher weight to sentences closely matched public interest. Compared to the results in Table 1, ETTS$_{+i}$ finds more news reflecting users' collective interest about the topic "President Obama won Nobel Peace Prize".

Table 4: Sample timeline generated by ETTS$_{+i}$ on "Nobel Prize" from July 2009 to December 2009. Due to space limit, we only present summary sentences related to the news "Obama won Nobel Peace Prize 2009". Top interest-related words are marked in Italic.

OCT. 09, 2009	*nobel peace prize obama president win*
●The Nobel Peace Prize was awarded to US President Barack Obama. ●The Nobel Peace Prize Committee has spoken out in defense of its decision to give the award to US President Barack Obama.	
NOV. 27, 2009	*nobel prize peace obama attend oslo*
●Obama will attend the start of the conference on Dec 9 before heading to Oslo to accept the Nobel Peace Prize.	
DEC. 10, 2009	*nobel prize peace obama accept speech oslo*
● US President Barack Obama, gives his Nobel speech after receiving the Nobel Peace Prize at City Hall in Oslo.	

5. CONCLUSIONS

We present a graph based approach to consider collective users' interests in timeline generation. Experiment results show that the incorporation of users' interests is helpful to improve both informativeness and interestingness. The generated summaries become more user favoring compared to traditional timeline summaries. As future work, we will explore learning user interests by dynamically adjusting the time span of a given topic instead of using a fixed time interval.

ACKNOWLEDGEMENTS. We thank Jing Jiang and Qiaoling Liu for helping to improve the work. We thank the anonymous reviewers for the constructive comments. The work was partially supported by NSFC Grant 61272340 and 60933004. Yulan He was partially supported by the EPSRC grant EP/J020427/1. Xin Zhao was supported by Microsoft Research Asia Fellowship.

6. REFERENCES

[1] H. L. Chieu and Y. K. Lee. Query based event extraction along a timeline. In *SIGIR '04*.

[2] G. Erkan and D. Radev. Lexpagerank: Prestige in multi-document text summarization. In *EMNLP*, 2004.

[3] R. Swan and J. Allan. Automatic generation of overview timelines. In *SIGIR'00*.

[4] R. Yan, L. Kong, C. Huang, X. Wan, X. Li, and Y. Zhang. Timeline generation through evolutionary trans-temporal summarization. In *EMNLP*, 2011.

[5] R. Yan, X. Wan, J. Otterbacher, L. Kong, X. Li, and Y. Zhang. Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In *SIGIR*, 2011.

[6] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM*, 2001.