Inf Retrieval (2014) 17:326–350
DOI 10.1007/s10791-014-9241-z

Identifying top news stories based on their popularity in the blogosphere

Yeha Lee · Jong-Hyeok Lee

Received: 22 May 2012 / Accepted: 29 April 2014 / Published online: 14 May 2014

Abstract A huge volume of news stories are reported by various news channels, on a daily basis. Subscribing to all the stories and keeping track of the important ones day after day is very time-consuming. This paper proposes several approaches to identify important news stories. To this end, we take advantage of the blogosphere as an information source to evaluate the importance of news stories. Blogs reflect the diverse opinions of bloggers about news stories, and the attention that these stories receive can help estimate the importance of the stories. In this paper, we define the popularity of a news story in the blogosphere as the attention it attracts from users. We measure popularity of the stories in the blogosphere from two viewpoints: content and a timeline. In terms of content, we suggest several approaches to estimate language models for a news story and blog posts, and we evaluate the importance of the story using these language models. Furthermore, we generate a temporal profile of a news story by exploring the timeline of blog posts related to the story, and evaluate its importance based on the temporal profile. We experimentally verify the effectiveness of the proposed approaches for identifying top news stories.

Keywords Blog retrieval · Blogosphere · Top news stories identification · Language model approach

1 Introduction

The volume of news stories reported in various news media has increased steadily. Keeping track of all the news stories day after day is a difficult and time-consuming job. Therefore, users increasingly need tools to help them organize and locate important news

Y. Lee (✉) · J.-H. Lee
Division of Electrical and Computer Engineering, POSTECH, Pohang, South Korea
e-mail: sion@postech.ac.kr

J.-H. Lee
e-mail: jhlee@postech.ac.kr

stories. In fact, news searches have played an increasingly important role in users' Internet activities (Del Corso et al. 2005).

The need to identify important news stories can vary according to the user's perspective. For example, one of the main tasks of a news editor is to identify top news stories which will be placed in the front page of the news website. Furthermore, news aggregators such as Yahoo! News[1] and Google News[2] receive news information from more than 4,500 news sources (Hu et al. 2008; Del Corso et al. 2005). They need to deal with a huge volume of the news information on a daily basis and evaluate the importance or newsworthiness of these stories.

From the perspective of news readers, recent events or breaking news may be most important. However, it is almost impossible for users to track all information related to a large number of events in various parts of the world. Thus, users increasingly need to automatically identify the important news stories and organize them.

This paper addresses the problem of evaluating the importance of news stories and ranking them in order of importance, in response to a given day. Solving this problem involves two types of tasks: retrospective and real-time tasks (Yang et al. 1998). The retrospective task aims to detect important news stories in a news corpus. Consider a journalist who wants to find out the biggest news stories of the year. In this case, the task can help him by automatically identifying the important ones. The real-time task focuses on evaluating the importance of news stories being reported in real-time. Compared to the retrospective task, this task can be useful to news editors who want to choose important stories for the main page of a news website. This task can also be helpful to news readers who want to follow hot news stories as they happen.

One of the key issues for identifying top news stories is that the importance of news stories can vary according to a perspective or interest of users who read the stories. As mentioned above, the purpose of identifying top news stories can vary from users' perspective. Furthermore, the importance or newsworthiness of the stories can be changed according to gender or age of news readers.

To handle this problem, in this paper, we take advantage of the blogs that is user-generated content to identify top news stories. As the popularity of the blogs is growing, many people express their opinions or thoughts on diverse topics from their daily affairs to various news issues. This implies that the blogosphere, which includes all blogs and their relationships, may capture the attention of the users about news stories. We believe that the users' attention in the blogosphere can be used as evidence to estimate the importance of the stories. In fact, the blogosphere has been widely used to retrieve or track important news events (Macdonald et al. 2010; Leskovec et al. 2009; Mishne and de Rijke 2006; Wang et al. 2008).

We define the *popularity* of a story as the amount of attention it receives from users within the blogosphere, and evaluate the popularity of news stories in terms of content and a timeline. To this end, we suggest several approaches to evaluate the similarity between the contents of the story and blog posts published on a given day. Furthermore, we estimate a temporal profile of a news story by analyzing the timeline of blog posts relevant to the story, and evaluate its importance using the temporal profile. The proposed approaches can be applied for both the real-time and retrospective tasks. The experimental results show that our approach is effective.

[1] http://news.yahoo.com.

[2] http://news.google.com.

The remainder of the paper is organized as follows. In Sect. 2, we briefly survey related work on our task. In Sect. 3, we address the framework of our system, and propose several approaches to identify the top news stories. In Sects. 4 and 5, we conduct several experiments to evaluate the performance of our approach. Finally, we conclude the paper and discuss future work in Sect. 6.

2 Related work

The top stories identification task (TSIT) was introduced as a pilot task in the TREC 2009 Blog Track (Macdonald et al. 2010). The TSIT consists of two subtasks. One aims to identify top news stories using the blogosphere, and to provide a ranked list of news stories. The other focuses on retrieving blog posts that discuss diverse aspects of the important stories. This paper deals with the first task, identifying top news stories.

There are several works for the TSIT. McCreadie et al. (2010) proposed an approach for identifying the important news stories based on the Voting Model (Macdonald and Ounis 2009) used for expert search. They also investigated historical and future temporal evidence to boost the scores of the news stories continuously mentioned in the blogosphere. Their approach is similar to our system in terms of using a temporal profile of a news story. However, as well as the temporal profile, we evaluate a newsworthiness of a news story by extracting important topics within blogs and estimating the relevance between the topics and the story. Weerkamp et al. (2010) proposed two approaches: News to Blogs and Blogs to News. They identified top news stories using an expert finding model (News to Blogs). This approach is also similar to our method using a temporal profile of a news story, but we evaluate the temporal profile based on the relevance scores between the story and blog posts. In addition, they tried to extract distinctive terms for a given day, and evaluated the importance of news stories based on the terms (Blogs to News). In our prior research published in Lee et al. (2010), we presented an approach based on the language modelling approach to IR. We proposed several approaches to estimate language models for the news stories and blog posts in response to a given day. This method is analogous to an approach proposed in this work, in the sense that both approaches try to use the contents of the story and blogs, and estimate the language models. However, in this paper, we propose a few methods to measure the importance of the topics extracted from blogs. Furthermore, the approaches proposed in this paper can be applied to both retrospective and real-time tasks. Lin et al. (2011) evaluated the importance of news stories by constructing a headline-post similarity network. To make the network, they computed the similarity between a news story and a blog post using the cosine measure.

In connection with organizing news events, Topic Detection and Tracking[3] (TDT) introduced in 1997 can be another research direction. TDT aims to retrieve and organize the broadcast news and newswire stories, and is comprised of five research tasks: Story Segmentation, Topic Tracking, Topic Detection, First Story or New Event Detection and Link Detection. The closely related New Event Detection (NED) task aims to detect whether or not a given news story is concerned with already known events to a system. For the event detection problem, many approaches have been based on clustering or classification to estimate the similarity between the events and documents (e.g. the news stories); these approaches differ in the ways by which they evaluate the similarity (Allan et al. 1998; Brants et al. 2003; Kumaran and Allan 2004; Yang et al. 1998; Zhang et al. 2002). All of

[3] http://projects.ldc.upenn.edu/TDT/.

them compare each document with existing events. If the similarity between the document and the events is lower than some predefined threshold, the document is considered to address a new event. Otherwise, the document is assigned to the event to which it is most similar.

Various features have been proposed, including timeline analysis, burstiness and named entities. Chen et al. (2003) proposed an aging theory to capture the lifecycle of a news event, and improved the performance for event detection. Chen et al. (2007) used an aging theory and sentence modeling to extract hot topics from news documents. They analyzed the timeline to identify the key terms. The burstiness of terms was used by many researchers for event detection (Chieu and Lee 2004; He et al. 2007; Kleinberg 2002; Wang et al. 2008). Kleinberg (2002) proposed an approach to identify the bursty features for the event detection from e-mail streams. They used the infinite-state automaton to model the streams. He et al. (2007) identified bursts of (a)periodic features using a Gaussian distribution, and then used them to detect (a)periodic events. Kumaran and Allan (2004) used named entities for event detection. They showed that the usefulness of named entities can change according to the situation. Zhang et al. (2007) classified terms within news stories based on named entity type and parts-of-speech tags, and assigned a different weight to each term according to the type and class of news story.

NED is different from our approach in some respects. The main goal of NED is to find a new event or a story from a stream of news stories based on clustering or classification. In contrast to NED, our approach aims to rank news stories according to their importance, and to identify the important ones. Furthermore, in contrast to the news corpus used for the NED task, we identify the top news stories using the unorganized blogosphere, not the well-defined contents of news articles.

3 News story ranking model

To identify top news stories, we rank them according to their importance or newsworthiness on a specific day. The newsworthiness of a news story can be determined using several criteria (MediaCollege 2009) as follows:

- **Timing** News stories that are happening now are often more newsworthy than those that happened a week ago.
- **Significance** The number of people involved in a news story is important.
- **Proximity** News stories that occur near us are more important than distant ones.
- **Prominence** News stories about famous people are more newsworthy than stories about ordinary people.
- **Human-Interest** Human-interest stories are generally soft news. They appeal to emotions.

However, evaluating these criteria requires deep-level semantic analysis. Besides, the influence of each criterion of newsworthiness can change according to gender, age or the interest of the news reader. In this paper, we evaluate the importance of news stories using the blogosphere, instead of carrying out the direct evaluation of these criteria.

A blog, web log, is a special type of website in which users (individuals or groups) express their opinions or thoughts on several subjects, and the blogosphere includes all blogs and their relationships. According to the "State of the Blogosphere"[4] reported by

[4] http://technorati.com/state-of-the-blogosphere/, last accessed on Dec. 22, 2013.

Technorati,[5] the blogosphere has grown exponentially every year from various angles including the number of blog users and the posting volume.

With the popularity of the blogs increasing, the importance of the blogs as a source of information is also growing. Several commercial search engines such as Google and Technorati have already provided search services dedicated to the blogosphere. Users' information needs for blog searches differ from those for general web searches. A large portion of the query logs from blog search engines are news-related queries (Macdonald et al. 2010; Mishne and de Rijke 2006). In other words, many users find information about news stories in the blogosphere. This implies that the blogosphere may be helpful when locating news stories. In fact, many previous works have shown that blogs can be a good indicator of important news stories (McCreadie et al. 2010; Tsagkias et al. 2009, 2011; Becker et al. 2010; Sun et al. 2008).

The blogosphere has a few advantages in terms of identifying top news stories. In general, blogs reflect news events almost instantly (Leskovec et al. 2009; Thelwall 2006). Leskovec et al. (2009) showed that there is a time gap of 2.5 hours between posting volume in blogs and news media. Thelwall (2006) also demonstrated that bloggers reacted promptly to the news of the London attacks. Furthermore, compared with traditional newspapers, blogs, which contain user-generated contents, cover a wide range of demographics (McCreadie et al. 2010). This means that the blogs can capture the diversity of users' thoughts or opinions in an unbiased way.

Based on these characteristics of the blogs, we take advantage of the blogosphere as evidence to evaluate the importance of news stories on a given day. We assume that a news story mentioned in more blogs is more important on a specific day, because a top news story satisfying the above criteria may receive attention from many blog users and the users' attention will be quickly reflected in the blogosphere. We call the users' attention the popularity of news stories in the blogosphere. In the following sections, we show how to estimate popularity in the blogosphere for identifying the important news stories.

3.1 Framework

Let n be a news story and let q and B_q be a given query day[6] and a set of blog posts published on the query day q, respectively. Motivated by our assumption, we evaluate the importance of a news story based on its popularity within the blogosphere on a given day q. To achieve this, we define the importance of a news story n using a probability that the story will be published (or generated), given a query day q and blog posts B_q, as follows:

$$Importance(n, q) \propto P(n|B_q, q) \tag{1}$$

where $Importance(n, q)$ indicates the importance of a news story n for a given day q. The joint conditional probability $P(n|B_q, q)$ in general is approximated by a linear interpolation of each conditional probability (Bendersky and Croft 2008).

$$P(n|B_q, q) = \alpha \underbrace{P(n|B_q)}_{\substack{\text{NewsStory} \\ \text{Likelihood}}} + (1-\alpha) \underbrace{P(n|q)}_{\substack{\text{NewsStory} \\ \text{TemporalProfile}}} \tag{2}$$

where α is a weighting parameter, ranging from 0 to 1.

[5] http://technorati.com/, last accessed on Dec. 22, 2013.

[6] A query is given as a date, NOT a text.

Finally, we estimate the importance of a news story n using two components: $P(n|B_q)$ and $P(n|q)$. The component $P(n|B_q)$ is the probability that a set of blog posts B_q will generate a news story n. The important news story will be mentioned by many users in the blogosphere, as a result, most of blog posts B_q will be similar to the story. We estimate $P(n|B_q)$ based on the similarity between contents of the news story and the set of blog posts B_q. In other words, we evaluate $P(n|B_q)$ by measuring the similarity between language models of a news story n and blog posts B_q. We call this component $P(n|B_q)$ the news story likelihood (NSL).

The component $P(n|q)$ captures the probability that a news story n will be reported on a query day q. To evaluate $P(n|q)$, we first search blog posts relevant to the news story, and then generate a temporal profile of the story by analyzing the distribution of the blog posts over days. We name this component $P(n|q)$ the news story temporal profile (NSTP).

For a given query q, the NSL and the NSTP capture the popularity of the news story on the blogosphere, in terms of the content and the timeline, respectively.

3.2 News story likelihood

The NSL represents a probability that a news story will be generated from a set of the blog posts published at a query day. We use the language model framework widely used for various information retrieval tasks to evaluate the NSL.

To this end, we should estimate two language models for a news story n and blog posts B_q. However, the blog posts may discuss various topics from individual daily affairs to important events that have happened recently. If we model the blog posts using a single language model, the language model cannot correctly capture the topics buried in the blog posts.

To handle this problem, we first extract the topics from the blog posts, and then use them to estimate the NSL. Let $T = \{t_1, t_2, \ldots, t_K\}$ be a set of the topics within the blog posts B_q. We rewrite the probability $P(n|B_q)$ using the topics.

$$P(n|B_q) = \sum_{k=1}^{K} P(n|t_k)P(t_k|B_q) \tag{3}$$

Finally, we impose one constraint to evaluate $P(n|B_q)$. In general, a news story is dedicated to only one news event or topic. Therefore, we select one topic most relevant to a news story, and use the topic to evaluate the NSL. In Sect. 3.2.3, we address this issue in more detail.

In the following sections, we show how to extract the topics from the blog posts and estimate the NSL.

3.2.1 Topic modeling

To extract the topics, we divide the blog posts into K clusters. We assume that each cluster can accurately reflect one of the various topics mixed in the blog posts. For this purpose, we cluster the blog posts using the K-means algorithm.

We represent each document (blog post) d using a term vector.

$$\mathbf{d} = \{f_1, f_2, \ldots, f_{|V|}\}$$

where $|V|$ is the size of the vocabulary and f_i is a weight for a term w_i.

We use the incremental tf-idf Model (Brants et al. 2003) to calculate the weight of the term. In the tf-idf approach, the document frequency of a term w_i has a static value, does

not change over a time t. However, in the incremental tf-idf model, the document frequency is updated according to a time t. The incremental approach has some benefits for our system.

First, we want our system to support the real-time task. In this regard, some statistics such as the document frequency should vary with the time t. Next, in general, news stories related to new events are likely to be important. New terms frequently occur with the new events, and the incremental tf-idf approach can provide fine weights for such new terms (Brants et al. 2003).

We define a weight for a term w_i within a document d as follows:

$$f_i = tf(w_i) \times \log \frac{N_t}{df_t(w_i)}$$

where $tf(w_i)$ is the number of times a term w_i occurred in a document d. N_t and $df_t(w_i)$ mean the total number of documents and the document frequency of a term w_i at time t, respectively.

We use the cosine similarity as the distance function between two documents.

$$Similarity(\mathbf{d_i}, \mathbf{d_j}) = \frac{\mathbf{d_i} \cdot \mathbf{d_j}}{|\mathbf{d_i}| \times |\mathbf{d_j}|} \tag{4}$$

where $|\mathbf{d_i}|$ and $|\mathbf{d_j}|$ indicate the length of $\mathbf{d_i}$ and $\mathbf{d_j}$, respectively.

After clustering, we can generate a topic language model (TLM) from each cluster. Let $D_k = \left\{d_{k_1}, d_{k_2}, \ldots, d_{k_{|D_k|}}\right\}$ be a set of documents for the kth cluster, and θ^k_{TLM} be the kth TLM. The document set D_k contains information relevant to a topic of the document set, but also contains background information. Therefore, to estimate the TLM, we assume that the documents are generated by a mixture model of θ^k_{TLM} and the collection language model θ_C that reflects the background information.

$$P(D_k) = \prod_i \prod_w \left\{(1-\lambda)P(w|\theta^k_{TLM}) + \lambda P(w|\theta_C)\right\}^{c(w;d_{k_i})} \tag{5}$$

where $c(w; d_{k_i})$ is the number of times the term w occurred in a document $d_{k_i}, P(w|\theta_C) = \frac{ctf_w}{|C|}$: ctf_w is the number of times the term w occurred in the entire collection, $|C|$ is the length of the collection, and λ is a mixing parameter.

Then, we can estimate θ^k_{TLM} using the EM algorithm (Dempster et al. 1977). The EM updates for $p(w|\theta^k_{TLM})$ are as follows:

$$t^n_w = \frac{(1-\lambda)P^n(w|\theta^k_{TLM})}{(1-\lambda)P^n(w|\theta^k_{TLM}) + \lambda P^n(w|\theta_C)} \tag{6}$$

$$P^{n+1}(w|\theta^k_{TLM}) = \frac{\sum_{i=1}^{n} c(w; d_{k_i}) t^n_w}{\sum_{w'} \sum_{i=1}^{|D_k|} c(w'; d_{k_i}) t^n_{w'}} \tag{7}$$

3.2.2 News story language model

In this section, we use two approaches to estimate a news story language model for each of the news stories reported on a query day. The news story language model is estimated from

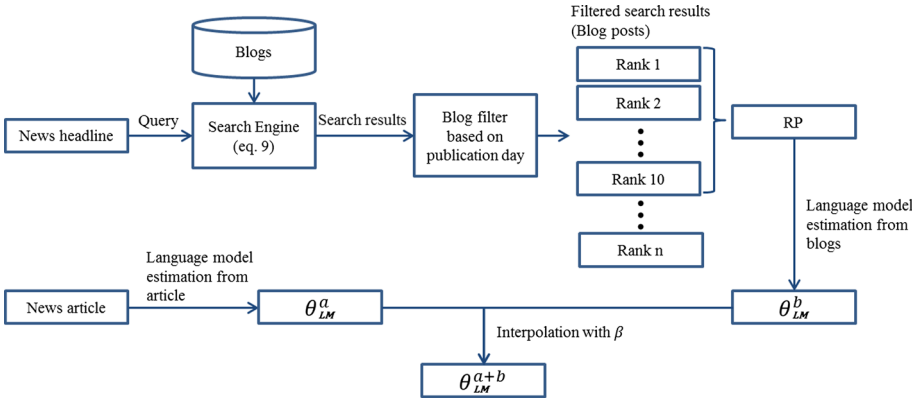

Fig. 1 The process for selecting a set of blog posts *RP* and estimating a news story language model

an article[7] of the news story and blog posts relevant to the story. First, we propose an approach using an article of a news story. This is a natural way to estimate the language model for the story. Let $\theta^a_{LM_n}$ be a language model of a news story n, estimated using its article. We estimated $\theta^a_{LM_n}$ using the maximum likelihood estimate of the article and the Dirichlet smoothing (Zhai and Lafferty 2004).

$$P(w|\theta^a_{LM_n}) = \frac{c(w;n) + \mu_a P(w|\theta_C)}{|n| + \mu_a} \tag{8}$$

where $|n|$ is the length of n and μ_a is a smoothing parameter.

Another way to estimate the language model is to use blog posts relevant to a news story n. Compared to $\theta^a_{LM_n}$, which models the language model from a news story itself, this approach uses user-generated contents to estimate the news story language model. Therefore, the language model can reflect users' points of view about the news story. We retrieve blog posts relevant to a news story using the headline as a query. The headline is used for the query because it usually consists of a few keywords that can be representative of the story. We evaluate the relevance between a news story n and a blog post d using the language model approach, one of the state-of-the-art information retrieval models (Song and Croft 1999).

$$Score(n,d) \stackrel{def}{=} \sum_{w \in h} \log P(w|d) \tag{9}$$

where $Score(n,d)$ is a relevance score between a news story n and a blog post d, and h indicates the headline of the story. The language model of $d, P(w|d)$, is estimated the maximum likelihood estimates of d with the Dirichlet smoothing (Zhai and Lafferty 2004). In our experiments, the smoothing parameter is set to 1,000 for blog post retrieval.

Then, we use only the blog posts whose issued date is within a certain period from the query day, because a large time gap between the issued day of a blog post and the query day often means that the blog post is mentioning an event different from those that happened on that day (Yang et al. 1998). In other words, the blog post is likely to be relevant to a topically similar, but different news story.

[7] We use the term "article" to refer to both the headline and contents of a news story.

Previous work in TDT shows that most events vanish within 2 months (Yang et al. 1998). In this work, we set the period between −28 and +28 days from a query day for the retrospective task. For the real-time task, the period is set between −28 and +0 days from the query day. Then, we gather only the blog posts published in the period, and choose 10 blog posts that can provide information relevant to the news story. The blog posts are selected according to the relevance score of each blog post obtained from Eq. 9. This procedure is similar to that of the pseudo-relevance feedback (PRFB) for the document retrieval. For the document retrieval, the PRFB uses top ranked documents in the first search to update query model. Similar to the PRFB, we uses top ranked blog posts retrieved from Eq. 9 to update the news story language model.

Let RP be a set of the selected blog posts and $\theta^b_{LM_n}$ be a language model of a news story n, estimated using RP. We estimate the language model using the maximum likelihood estimate of RP and the Dirichlet smoothing (Zhai and Lafferty 2004).

$$P(w|\theta^b_{LM_n}) = \frac{c(w;RP) + \mu_b P(w|\theta_C)}{|RP| + \mu_b} \tag{10}$$

where $|RP|$ is the length of RP and μ_b is a smoothing parameter. Figure 1 shows the process selecting the RP using a news headline and estimating a news story language model.

Finally, we estimate the language model using two language models, $P\left(w|\theta^a_{LM_n}\right)$ and $P\left(w|\theta^b_{LM_n}\right)$.

$$P\left(w|\theta^{a+b}_{LM_n}\right) = \beta P\left(w|\theta^b_{LM_n}\right) + (1-\beta)P\left(w|\theta^a_{LM_n}\right) \tag{11}$$

where $P(w|\theta^{a+b}_{LM_n})$ is a final language model for a news story and β is a weighting parameter that controls the influence of two language models.

3.2.3 Score function for news story likelihood

We introduced the equation for evaluating the NSL in Eq. 3. The equation consists of two probabilities, $P(n|t_k)$ and $P(t_k|B_q)$.

First, for calculating $P(n|t_k)$, we already estimated a topic language model (3.2.1) and a news story language model (3.2.2) for a topic t_k and a news story n, respectively. We evaluate $P(n|t_k)$ using the *negative* KL-Divergence Language model approach of Zhai and Lafferty (2001) as follows,

$$-\sum_w P(w|\theta_{LM_n}) \log \frac{P(w|\theta_{LM_n})}{P\left(w|\theta^k_{TLM}\right)}$$

where $P(w|\theta_{LM_n})$ can be one of $P\left(w|\theta^a_{LM_n}\right), P\left(w|\theta^b_{LM_n}\right)$ or $P\left(w|\theta^{a+b}_{LM_n}\right)$.

Next, $P(t_k|B_q)$ is the probability that a topic t_k will be selected from the blog posts published on a query day. We evaluate $P(t_k|B_q)$ using two approaches in this work. One regards $P(t_k|B_q)$ as the uniform distribution.

$$P_{unif}(t_k|B_q) = \frac{1}{|T|} \tag{12}$$

where $|T|$ is the total number of topics (i.e. $|T| = K$).

The other approach assigns the probability proportional to the number of documents that belong to a topic t_k (i.e. cluster D_k). We assume that a topic with a lot of blog posts relevant to it is likely to be important.

$$P_{len}(t_k|B_q) = \frac{1}{Z}\ln(N(t_k)+1) \tag{13}$$

where Z is a normalization constant ($Z = \sum_k \ln(N(t_k)+1)$), and $N(t_k)$ is the number of the documents that belong to a topic t_k.

Finally, we impose one constraint on our NSL model. In Eq. 3, the NSL is calculated by averaging over all the topics. However, a news story is usually dedicated to only one news event or topic. Therefore, we choose one topic that is most relevant to a news story as follows,

$$\widehat{t_k} = \underset{t_k \in T}{\operatorname{argmax}} P(n|t_k)P(t_k|B_q) \tag{14}$$

Then, we evaluate the NSL using the topic $\widehat{t_k}$.

$$S_{NSL}(n,q) = P(n|\widehat{t_k})P(\widehat{t_k}|B_q) \tag{15}$$

where $S_{NSL}(n,q)$ indicates the popularity of a news story n, estimated using the NSL, for a given query q.

3.3 News story temporal profile

The NSTP implies a probability that a news story will be issued on a query day. We estimate the NSTP using the temporal profile of a news story. We assume that if a news story is important for a query day, large numbers of blog posts will be relevant to the story. Motivated by this assumption, we estimate the temporal profile of a news story by analyzing the distribution of blog posts relevant to the story over days.

First, we rewrite the probability $P(n|q)$ in Eq. 2 with Bayes' Theorem.

$$P(n|q) = \frac{P(q|n)P(n)}{P(q)} \tag{16}$$

where we assume that $P(n)$ and $P(q)$ are the uniform distribution, then $P(n|q) \propto P(q|n)$.

To generate the temporal profile of a news story, we use the temporal profiling approach proposed by (Jones and Diaz 2007). The temporal profile of a news story n is defined as follows:

$$P(q|n) = \sum_{d \in R} P(q|d)\frac{Score(n,d)}{\sum_{d' \in R} Score(n,d')} \tag{17}$$

where $Score(n,d)$ indicates the relevance score between a news story n and a document d obtained by Eq. 9. R is a set that consists of 500 blog posts selected in order of their relevance score. Similar to the NSL, the blog posts for R are selected if they are published between -28 and $+28$ days from a query day for the retrospective task. For the real-time task, we use the blog posts published between -28 and 0 days from the query day. We define $P(q|d)$ as follows,

$$P(q|d) = \begin{cases} 1 & \text{if } q \text{ is equal to the document date} \\ 0 & \text{otherwise} \end{cases}$$

Then, the temporal profile of a news story n (i.e. $P(q|n)$) is related to a peakedness of the story in response to a given query day q.

This temporal profile is defined on each single day. However, if a news story is important for a query day q, the blog posts relevant to it may be published over a certain period from the day due to the bursty nature (Kleinberg 2002). Therefore, we smooth the temporal profile model with the model for adjacent days.

Let $S_{NSTP}(n, q)$ be the popularity of a news story n, estimated using the NSTP, for a given query q.

$$S_{NSTP}(n, q) = \frac{1}{|\phi|} \sum_{i \in \phi} P(q + i|n) \tag{18}$$

where ϕ indicates a period from the query day q.

3.4 Integration of two components

We proposed two approaches for evaluating the importance of the news stories: the NSL and the NSTP in Sects. 3.2 and 3.3. They capture the different characteristics of important news stories. For the NSL, we analyze the contents of the blog posts, and extract the dominant topics buried in them. Then, we estimate the importance of the news stories using the probability that each story is generated by the topics. For the NSTP, we investigate the distribution of the number of blog posts relevant to a news story, in terms of a timeline, for evaluating its importance.

To identify the top news stories, we combine the NSL and NSTP via score fusion. To achieve this, we first adjust each score from 0 to 1.

$$S_i^{'}(n, q) = \frac{S_i(n, q) - min_i}{max_i - min_i} \tag{19}$$

$$min_i = \min_{n'} S_i(n, q^{'}) \quad \text{and} \quad max_i = \max_{n'} S_i(n, q^{'})$$

where $S_i(n, q)$ indicates a score of $S_{NSL}(n, q)$ or $S_{NSTP}(n, q)$.

Finally, we define the ranking function as follows:

$$S(n, q) = \alpha S_{NSL}(n, q) + (1 - \alpha) S_{NSTP}(n, q) \tag{20}$$

where α is the weighting parameter that adjusts the importance between the NSL and the NSTP, and the parameter α is the same as that used in Eq. 2.

4 Experiment setup

4.1 Data set

Our experiments were conducted within the context of the Blog Track at TREC 2009 (Macdonald et al. 2010) and 2010[8]. The Blog Track introduced the TSIT that aims to identify important news stories by taking advantage of the blogosphere, in response to a given day (query). In the TSIT, the Blogs08 corpus (Macdonald et al. 2010) was used for the sample of the blogosphere. The Blogs08 corpus was created by monitoring 1 million

[8] http://ir.dcs.gla.ac.uk/wiki/TREC-BLOG.

blogs from January 14, 2008 to February 10, 2009, and consists 808GB of feeds, 1445GB of permalink documents and 56GB of homepages.

Furthermore, for the TSIT, two news corpora were provided for the sample of the news collection. One is the news corpus from the New York Times (NYT) used at the TREC 2009 Blog Track. The NYT corpus does not include the contents of news stories, only the headlines of the stories. The other corpus is the TRC2 news corpus from Thomson-Reuters, which contains the contents of the news stories as well as the news headlines, and was used at the TREC 2010 Blog Track. Both the NYT and TRC2 corpora consist of news stories published during the interval covered by the Blogs08 corpus.

Our experiments were performed using only the Blog08 and the two news corpora without resorting to any other resources. For the evaluation, we used the 55 queries and relevance judgments from the TREC 2009 TSIT, and the 50 queries and relevance judgments from TREC 2010 TSIT.

We only used the permalinks (blog post) for the experiments. We discarded the HTML tags of each blog post, and applied the DiffPost algorithm (Nam et al. 2009) to remove the non-relevant contents[9] of each blog post. Each blog post was also processed by stemming using the Porter stemmer and eliminating stopwords using the INQUERY words stoplist (Allan et al. 2000).

Table 1 shows the statistics of the Blogs08 and two news corpora (NYT and TRC2).

4.2 Evaluation method

Similar to the experimental setting of TSIT'09, for each query, we considered only the news stories corresponding to $q \pm 1$ days as ranking candidates, because of the time discrepancy between the day on which the news story was reported was and the day q on which the news story actually happened (Macdonald et al. 2010). Then, for each query, we retrieved 100 news stories according to their importance on that day. We used the mean average precision (MAP) and the precision at rank 5 and 10 (P@5 and P@10) as the evaluation measures.

The experimental setting of TSIT'10 differs from that of TSIT'09. For the TSIT'10, we needed to provide a ranked list of news stories with five categories (WORLD, U.S., SPORT, TECHNOLOGY+SCIENCE and BUSINESS), instead of an overall ranking. That is, for each query, we retrieved 100 news stories in order of their importance in response to each of five categories. For this purpose, we first ranked all the news stories issued on a query day q regardless of their category, and then classified them into five categories using an SVM classifier[10]. To train the classifier, we used the categories of the New York Times: WORLD, U.S., SPORT, TECHNOLOGY+SCIENCE and BUSINESS. Among the news stories published throughout the whole timespan of the Blogs08 corpus, we randomly selected 2,000 news stories from each category. We trained the classifier using the linear kernel and a binary feature of unique terms. We used the statMAP, statMNDCG@10 and statMPC@10 as the evaluation measures[11] (Aslam and Pavlu 2008).

[9] In (Nam et al. 2009), the non-relevant contents of a blog post refer to useless contents for the blog search, such as menu, banner and site description.

[10] LIBSVM: http://www.csie.ntu.edu.tw/cjlin/libsvm.

[11] In our experiments, statMAP, statMNDCG@10 and statMPC@10 indicate the average of statMAP, statMNDCG@10 and statMPC@10 for each category, respectively.

Table 1 Statistics for the Blogs08 and two news story (NYT and TRC2) corpora

Type	Corpus	Docs	Timespan
Blog	Blogs08	28,488,766	14/01/08–10/02/09
News story	NYT	102,853	01/01/08–28/02/09
	TRC2	1,613,707	

'Docs' indicates the number of blog posts for the Blogs08 corpus, and news stories for the NYT and TRC2 corpora

5 Results and discussion

In this section, we performed several experiments to evaluate our system for the TSIT. We measured the performance of the NSL and the NSTP, respectively. We also explored the influence of the combination of two components on the performance of the TSIT, with a varying weight parameter α. In the experiments, we evaluated the performance of the proposed approaches for each of the real-time and retrospective tasks.

5.1 News story likelihood

First, we evaluate the performance of the NSL to identify important news stories. The aim of the experiments is to investigate (1) how correctly our method extracts various topics buried in blog posts published on the query day; and (2) how the proposed approaches for estimating the news story language model (i.e. $P(w|\theta_{LM_n})$) and the topic selection probability (i.e. $P(t_k|B_q)$) affect the performance of the news story ranking.

To investigate the effect of topic modeling, we evaluate the performance of the NSL according to various K values.

$$K : 1, 30, 50, 100, 300, 500, 1000, 1500$$

To estimate the topic language model θ_{TLM}, we need to set a parameter λ in Eq. 5. λ controls the influence of the background model θ_C. In general, a large λ decreases a probability that stop words will be generated in topic language models (Mei et al. 2006), and this may give high probability to discriminative words for each topic. Mei et al. (2006) mentioned that a value between 0.9 and 0.95 is suitable for λ in the case of blog documents. Although their experimental setting is different from ours, we set λ to 0.90, which they recommend.

Furthermore, for the experiments, we estimate the news story language model using blog posts (Eq. 10). We trained the parameter μ_b in Eq. 10 using the TSIT'10 topics to evaluate the performance of the TSIT'09 topics, and vice-versa. The search space for the parameter μ_b is given by: {500, 800, 1000, 1500, 2000, 2500, 3000, 4000}. We selected the parameter resulting in the best MAP or statMAP score for each task. For both TSIT'09 and '10, we obtained the best results when μ_b was set to 1500. In addition, we set the probability $P(t_k|B_q)$ that a topic will be chosen from B_q as the uniform distribution.

Figure 2 shows the MAP and statMAP scores according to varying K values, for two types of tasks. While the real-time task uses blog posts published at $q - 28 \leq b_t \leq q$, the retrospective task uses them with $q - 28 \leq b_t \leq q + 28$; b_t indicates the day where a blog post b is published. We use $K = 1$ as the baseline score. This means that the topic modeling is not applied.

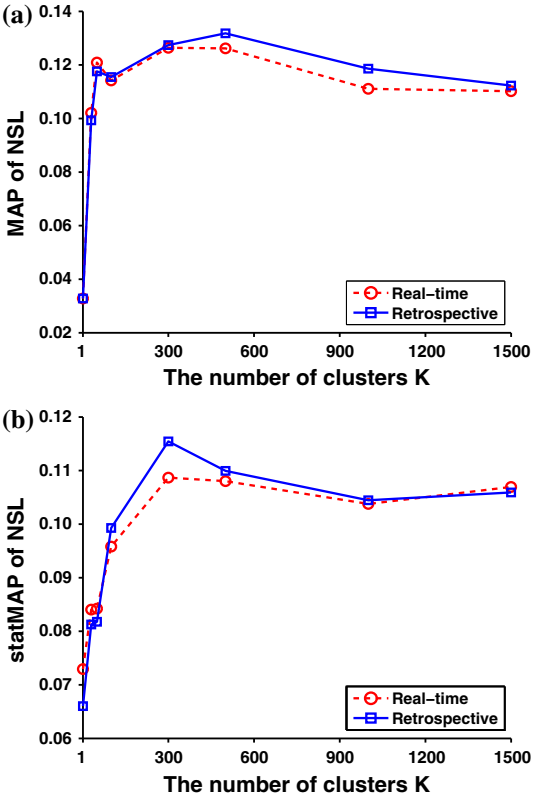

Fig. 2 The performance of the news story likelihood according to varying the number of clusters K. **a** For TSIT'09, the MAP scores of the real-time and retrospective tasks. **b** For TSIT'10, the statMAP scores of the real-time and retrospective tasks

Compared with the baseline, the performances for all $K > 1$ were significantly improved, and the best performance for TSIT'09 and '10 were obtained when using $K = 500$ and 300, respectively. In the following experiments, we set K to 500 and 300 for TSIT'09 and '10, respectively. Although the K values are similar for TSIT'09 and '10 setting, the values were set at the best possible cases, which may not be practical. From these results, we can confirm that a single language model cannot correctly capture the contents of the blog posts, because of the topical diversity of the blog posts. This weakness reduced the ability of our system to identify the top news stories.

As the number of clusters K increased to 500, the respective topics buried in the blog post were captured by the K clusters. The topic language model (TLM) estimated using each cluster led to improved performance of the NSL. When $K > 500$, the clusters had been overfitted, and did not provide enough information relevant to each topic. As a result, the performance decreased.

Next, we investigate the influence of the proposed approaches for the news story language model $P(w|\theta_{LM_n})$ on the performance of the NSL. Let NSL_A be the NSL using the language model estimated from the news article (Eq. 8). Let NSL_B^{real} and NSL_B^{ret} be the NSL using the language model estimated from the blog posts (Eq. 10) in response to the real-time task and retrospective task, respectively.

The experiments using NSL_A were conducted only for TSIT'10. Because the NYT corpus used for TSIT'09 does not include the body of the news stories, we cannot estimate

Table 2 The performance of the news story likelihood according to methods for estimating the news story language model

TSIT'09

Model	MAP	P@5	P@10
NSL_B^{real}	0.1262	0.2400	0.2182
NSL_B^{ret}	0.1318	0.2327	0.2345

TSIT'10

Model	statMAP	statMNDCG@10	statMPC@10
NSL_B^{real}	0.1087	0.0795	0.2286
NSL_B^{ret}	0.1154	0.0847	0.2413
NSL_A	0.2139	0.1628	0.4984
NSL_{A+B}^{real}	0.2258	0.1675	0.5009
NSL_{A+B}^{ret}	0.2298	0.1721	0.5129

Model	Mean statMAP	statMAP by category				
		Business	Sci-Tech	Sport	U.S.	World
NSL_{A+B}^{real}	0.2258	$0.1734^{\ddagger}$	$0.1780^{\ddagger,\S}$	$0.2025^{\ddagger}$	$0.2542^{\ddagger,\P}$	$0.3208^{\ddagger}$
NSL_{A+B}^{ret}	0.2298	$0.1786^{\ddagger,\S}$	$0.1780^{\ddagger,\S}$	$0.2064^{\dagger,\S}$	$0.2589^{\ddagger,\P}$	$0.3268^{\ddagger}$

The NSL_B^{real} and NSL_B^{ret}, which correspond to real-time and retrospective tasks, use the news story language model estimated from the blog posts. The NSL_A uses that estimated from a news article. The NSL_{A+B}^{real} and NSL_{A+B}^{ret} use the news story language model estimated from both blog posts and a news article. For TSIT'10, the statistical significance at the 0.05 and 0.01 level is indicated by † and ‡ for improvement from NSL_B, respectively, § and ¶ for improvement from NSL_A, respectively

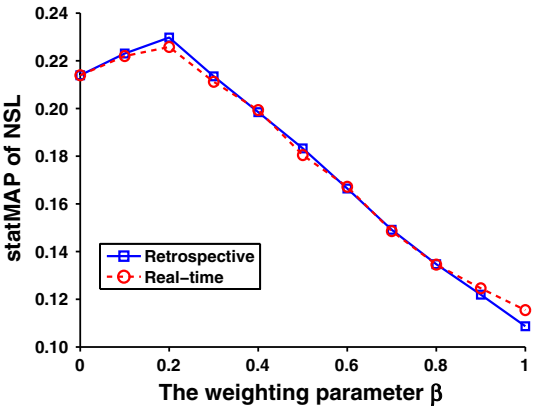

Fig. 3 For TSIT'10, the statMAP scores of the news story likelihood according to varying the parameter β that controls the weight between $P(w|\theta^a_{LM_n})$ and $P(w|\theta^b_{LM_n})$ for estimating the news story language model. $\beta = 0$ and $\beta = 1$ indicate NSL_A ($P(w|\theta^a_{LM_n})$) and NSL_B ($P(w|\theta^b_{LM_n})$), respectively

Table 3 The performance of the news story likelihood when $P(t_k|B_q)$ is set to $P_{len}(t_k|B_q)$

TSIT'09

Model	MAP	P@5	P@10
len- NSL_B^{real}	$0.1512^{\ddagger}$	0.2836	$0.2800^{\ddagger}$
len- NSL_B^{ret}	$0.1516^{\dagger}$	0.3091	0.2636

TSIT'10

Model	statMAP	statMNDCG@10	statMPC@10
len- NSL_{A+B}^{real}	0.2201	0.1610	0.4945
len- NSL_{A+B}^{ret}	0.2208	0.1623	0.4859

Model	Mean	statMAP by category				
	statMAP	Business	Sci-Tech	Sport	U.S.	World
len- NSL_{A+B}^{real}	0.2201	$0.1595^{\ddagger}$	0.1547	$0.1974^{\ddagger}$	$0.2875^{\ddagger,\P}$	$0.3013^{\ddagger}$
len- NSL_{A+B}^{ret}	0.2208	$0.1535^{\ddagger}$	$0.1547^{\dagger}$	$0.1999^{\ddagger}$	$0.2914^{\ddagger,\P}$	$0.3044^{\ddagger}$

For TSIT'09 and TSIT'10, the statistical significance at the 0.05 and 0.01 level is indicated by † and ‡ for improvement from NSL_B, respectively, § and ¶ for improvement from NSL_A, respectively

the language model from the news article for TSIT'09 topics. For the experiment, we set μ_a to the same value as μ_b without any further optimization.

Table 2 shows the performances of our system according to various methods for estimating a language model of a news story. For TIST'10, we also performed the Wilconxon signed rank test to see whether the improvement of the performance of the combining approach NSL_{A+B} over those of NSL_A and NSL_B was statistically significant. As mentioned above, in TSIT'10 setting, we provide five ranked lists of news stories for each five categories. Therefore, for TSIT'10, we conduct the rank test according to each category.[12]

Compared with NSL_B^{real}, NSL_B^{ret} resulted in better performance for both TSIT'09 and '10. In the case of NSL_B, we choose 10 blog posts relevant to a news story to estimate the language model. The only difference between two models is the period for gathering blog posts. NSL_B^{real} uses blog posts with $q - 28 \leq b_t \leq q$, while NSL_B^{ret} exploits blog posts published at $q - 28 \leq b_t \leq q + 28$. The important news stories can be discussed after the day when they are issued. For example, news stories related to the London bombings were continuously reported after the event occurred, and many blog posts about the event were published for a few weeks (Thelwall 2006). NSL_B^{ret} can use this future evidence to estimate the language model, and this leads to the improved performance.

NSL_A, which estimates the language model of the story from a news article, outperforms NSL_B, which uses blog posts. These results may be inevitable because a news article can

[12] The rank test was performed between models of the same task (e.g. NSL_B^{real} and NSL_{A+B}^{real}). For NSL_A, the rank test was conducted with both NSL_{A+B}^{real} and NSL_{A+B}^{ret} because the model can be applied both real and retrospective tasks.

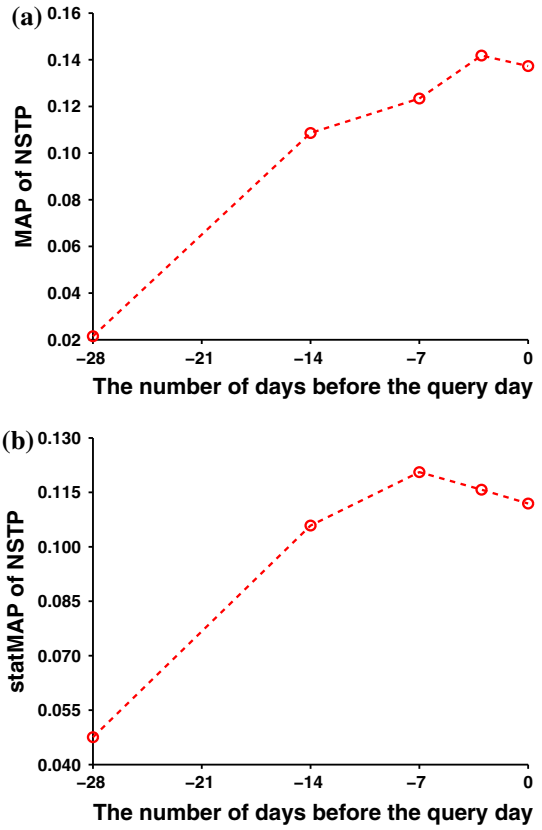

Fig. 4 For the real-time task, the performance of the news story temporal profile according to varying the period ϕ. The best scores were obtained when ϕ was set to -3 and -7 for TSIT'09 and '10, respectively. **a** The MAP scores for TSIT'09. **b** The statMAP scores for TSIT'10

provide information that is more focused on and relevant to the news story than blog posts retrieved using the headline of the news story as a query.

To explore the influence of two language models for estimating the news story language model (Eq. 11), we measured the statMAP scores according to varying the weighting parameter β[13], in Fig. 3. Let NSL^{real}_{A+B} and NSL^{ret}_{A+B}, which correspond to real-time and retrospective tasks, be the NSL using the language model estimated by $P(w|\theta^{a+b}_{LM_n})$.

For both tasks, the best statMAP scores were obtained when $\beta = 0.2$ (NSL^{real}_{A+B} : 0.2258, NSL^{ret}_{A+B} : 0.2298). NSL^{real}_{A+B} achieved 11.71 and 1.19 % further increases in statMAP over NSL^{real}_{B} and NSL_A, respectively. NSL^{ret}_{A+B} also improved the statMAP scores by 11.44 % and 1.59 % over NSL^{ret}_{B} and NSL_A, respectively. From these results, we can verify that the blogs can be useful in identifying the important news stories, and blog posts and news stories should be considered together to improve the performance. These two types of evidence capture different characteristics. For a news event, the blog posts can capture diverse aspects of the event from users' viewpoint, while a news story can reflect news editors' point of view.

[13] We can estimate the NSL_A for only TSIT'10 corpus, because the TSIT'09 corpus do not have the news articles. For this reason, we set the parameter β using TSIT'10 and used it for evaluating TSIT'10 topics. That is, we set the parameter β to an optimal values, which may not be practical.

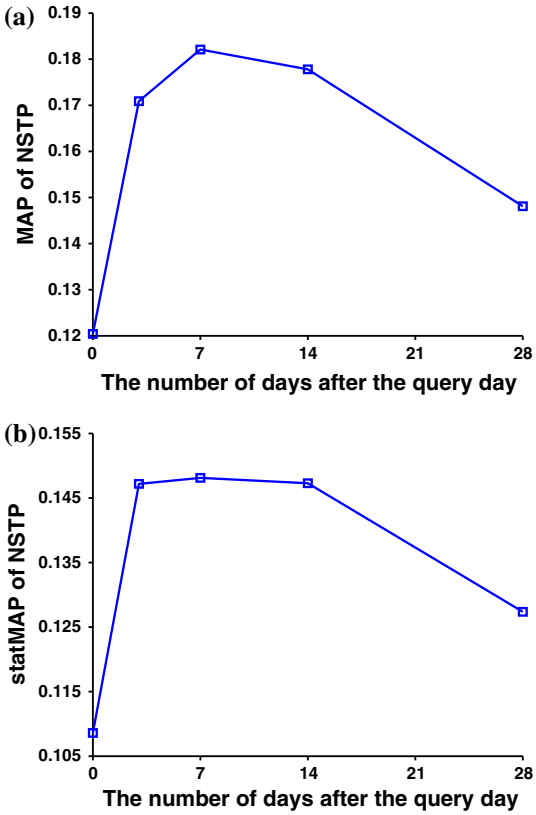

Fig. 5 For the retrospective task, the performance of the news story temporal profile according to varying the period ϕ. The best scores were obtained when ϕ was set to $+7$ for both TSIT'09 and '10 **a** The MAP scores for TSIT'09. **b** The statMAP scores for TSIT'10

Until now, we assume the probability $P(t_k|B_q)$ that a topic t_k will be chosen from B_q as a uniform distribution. We test the performance of the NSL when $P(t_k|B_q)$ is set to $P_{len}(t_k|B_q)$. For these experiments, we used models resulting in the best performance when using $P_{unif}(t_k|B_q)$ for each task. To indicate approaches using $P_{len}(t_k|B_q)$, we add the prefix "*len-*" to each model.

Table 3 shows the experimental results for TSIT'09 and '10 when using $P_{len}(t_k|B_q)$ as $P(t_k|B_q)$. We also performed the Wilconxon signed rank test to see whether the improvement of the performance over results of Table 2 was statistically significant.

For TSIT'09, the usage of the cluster length for $P(t_k|B_q)$ significantly improved the performance of the NSL for both the real-time or retrospective tasks. These results support our assumption about $P_{len}(t_k|B_q)$. If a topic t_k is important on a given day, many bloggers pay attention to the topic. As a result, the number of blog posts within a cluster corresponding to the topic will be greater than that of other clusters.

In contrast to TSIT'09, for TSIT'10 topics, the usage of the cluster length failed to improve the performance of the NSL. One possible reason can be found in the relevance judgments of the TSIT'10. For TSIT'10 topics, the importance of each news story can be judged differently according to each of five categories: WORLD, U.S., SPORT, TECHNOLOGY+SCIENCE and BUSINESS. However, we divided blog posts B_q into the K number of clusters without considering their category. These clusters can decrease the effectiveness of the NSL. This issue needs to be investigated more carefully.

Table 4 The performance of systems integrating the news story likelihood and the news story temporal profile

TSIT'09

Type	Model	MAP	P@5	P@10
Real	len-NSL_B^{real}	0.1512	0.2836	0.2800
	$NSTP_0^{-3}$	0.1418	0.2655	0.2709
	$Final_{09}^{real}$	**0.1751**‡,¶	**0.3236**†	**0.3127**§
Ret	len-NSL_B^{ret}	0.1516	0.3091	0.2636
	$NSTP_{+7}^{-3}$	0.1821	0.3455	0.3218
	$Final_{09}^{ret}$	**0.2002**‡,¶	**0.3636**	**0.3527**‡,§

TSIT'10

Type	Model	statMAP	statMNDCG@10	statMPC@10
Real	NSL_{A+B}^{real}	0.2258	**0.1675**	**0.5009**
	$NSTP_0^{-7}$	0.1206	0.0911	0.2626
	$Final_{10}^{real}$	**0.2306**	0.1661	0.4739
Ret	NSL_{A+B}^{ret}	0.2298	**0.1721**	**0.5129**
	$NSTP_{+7}^{-7}$	0.1481	0.1172	0.3122
	$Final_{10}^{ret}$	**0.2347**	0.1696	0.5005

Model	Mean	statMAP by category				
	statMAP	Business	Sci-Tech	Sport	U.S.	World
$Final_{10}^{real}$	0.2306	0.1751¶	0.1801¶	0.2116	0.2625¶	0.3236¶
$Final_{10}^{ret}$	0.2347	0.1770¶	0.1829¶	0.2133†	0.2688†,¶	0.3318¶

'real' and 'ret' indicate the real-time and retrospective tasks, respectively. The best performance is shown in bold. The statistical significance at the 0.05 and 0.01 level is indicated by † and ‡ for improvement from the news story likelihood, respectively, § and ¶ for improvement from the news story temporal profile, respectively

5.2 News story temporal profile

The NSTP evaluates the popularity of a news story in the blogosphere in terms of the timeline. The NSTP considers a certain period ϕ from a query day to gather evidence for estimating the importance of news stories.

The blog posts and news stories related to hot events that happened on a query day are published on that day or in the following days due to the bursty nature. Furthermore, they can be published on preceding days for various reasons. For instance, the events such as elections and Olympics are predetermined, and can be discussed on the blogosphere before they occur.

We perform several experiments according to varying the period ϕ. We first evaluate the performance using only the past evidence (i.e. the real-time task). The period is set between $\{-28, -14, -7, -3, 0\}$ and 0 days from a query day. Figure 4 shows the

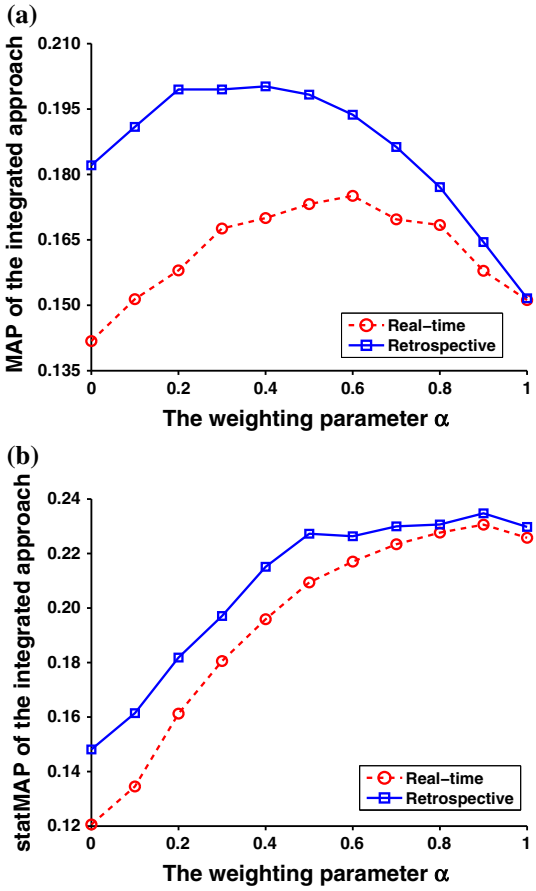

Fig. 6 The performance of the integrated approach according to varying the parameter α that controls the weight between the NSL ($\alpha = 1$) and the NSTP ($\alpha = 0$). **a** For TSIT'09, the MAP scores of the real-time and retrospective tasks. **b** For TSIT'10, the statMAP scores of the real-time and retrospective tasks

performance of the NSTP for the real-time task, according to varying the period ϕ. The best results are obtained when using -3 and -7 days for TSIT'09 and '10, respectively. These results verify that past evidence can help identify the important news stories. However, the evidence should be treated cautiously, because blog posts issued too far prior to the query day may provide noisy information, and this reduces the performance of the system.

For the retrospective task, we explore the influence of future evidence on the performance of the NSTP. For these experiments, we used the model resulting in the best performance for the real-time task as the baseline. That is, we set the days used for the past evidence to -3 and -7 days for TSIT'09 and '10, respectively, and carry out the experiments according to varying the days for the future evidence. For TSIT'09 and '10, we define the period ϕ as follows,

- TSIT'09: ϕ is set between -3 and $\{0, 3, 7, 14, 28\}$ days from the query day.
- TSIT'10: ϕ is set between -7 and $\{0, 3, 7, 14, 28\}$ days from the query day.

Figure 5 shows the performance of the NSTP for the retrospective task, according to varying the period ϕ. For both TSIT'09 and '10, we obtained the best results when the period is set to $+7$ days for the future evidence. Similar to the real-time task, the usage of

the future evidence is helpful to improve the performance for our system. However, the blog posts published on too subsequent to the query day decreases the performance of the system. Here, even if the period ϕ was set to similar values for TSIT'09 and 10, we are looking at the best cases, which may not be practical.

Overall, from the experimental results, we can confirm our assumption that if a news story is important for a given day, blog posts relevant to it will be posted for several days. As a result, the temporal profile of a news story, estimated from the blog posts, can be a good indicator for identifying the important news stories, in response to a given day.

5.3 Integration of NSL and NSTP

We measured the performance of each component: the NSL and the NSTP. In this section, we investigate the influence of integrating the NSL with the NSTP on the performance of the system. For this purpose, we used the best performing models for each component. For TSIT'09, we combine two models as follows,

- $Final_{09}^{real}$: *len-* NSL_B^{real} and $NSTP_0^{-3}$ for the NSL and the NSTP, respectively.
- $Final_{09}^{ret}$: *len-* NSL_B^{ret} and $NSTP_{+7}^{-3}$ for the NSL and the NSTP.

For TSIT'10, we integrate two approaches as follows,

- $Final_{10}^{real}$: NSL_{A+B}^{real} and $NSTP_0^{-7}$ for the NSL and the NSTP.
- $Final_{10}^{ret}$: NSL_{A+B}^{ret} and $NSTP_{+7}^{-7}$ for the NSL and the NSTP.

where $NSTP_y^x$ means the NSTP where the period ϕ is set between x and y days from the query day.

Table 4 shows the performance of the integrated approaches, $Final_{09}^{real}$, $Final_{09}^{ret}$, $Final_{10}^{real}$ and $Final_{10}^{ret}$. For comparison, we also reported the best performance of the NSL and the NSTP. We performed the Wilconxon signed rank test to examine whether the improvement of the performance over that of each component was statistically significant.

For both TSIT'09 and '10, and both the real-time and retrospective tasks, integrating two components outperformed the best performing models for each component. In particular, the performance improvement is notable for TSIT'09 topics. In terms of the real-time (and retrospective) task, the combining approach gives 2.39 % (4.41 %) and 3.33 % (1.81 %) absolute improvements in MAP, compared to the NSL and the NSTP, respectively.

These results imply that the performance can be enhanced by merging the NSL and the NSTP. They evaluate the popularity of a news story in the blogosphere in different ways. While the NSL identifies the important news stories by modeling the dominant topics in blog posts and evaluates the relevance between each topic and a news story, the NSTP explores the temporal profile of a news story by taking advantage of the timeline of blog posts relevant to the story for evaluating its importance.

Figure 6 shows the performances of the integrated approaches for TSIT'09 and 10, according to varying the parameter α. The weighting parameter α controls the weight between the NSL and the NSTP. From these results, we can again verify that the performance can be improved when integrating the two components, the NSL and the NSTP. In this paper, we set the weighting parameter α to an optimal values[14], which may not be

[14] Search space for $\alpha = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.

Table 5 Comparison our best approach for the retrospective task with the best performing runs of TSIT'09

Run	MAP	P@5	P@10
uogTrTStimes	0.1862	0.3236	0.3127
POSTECH_KLE	0.1605	0.2836	0.2964
I1psTSExP	0.1354	0.2655	0.2745
$Final_{09}^{ret}$	**0.2002**‡,§	**0.3636**‡,§	**0.3527**‡,§

The statistical significance at the 0.05 level is indicated by †,‡, and § for improvement from 'uogTrTStimes', 'POSTECH_KLE', and 'I1psTSExP', respectively. The best performance is shown in bold

Table 6 Comparison our best approach for the real-time task with the best performing runs of TSIT'10

Run	Mean statMAP	statMAP by category				
		Business	Sci-Tech	Sport	U.S.	World
KLERUN1	0.2206	0.1851	0.1821	0.1916	0.2458	0.2986
ikm100jing	0.2151	0.1144	0.2483	0.1725	0.3897	0.1504
ICTNETTSRun2	0.2138	0.0969	0.1898	0.2405	0.3025	0.2396
$Final_{10}^{real}$	**0.2306**	0.1751‡,§	0.1801	0.2116†,‡	0.2625	0.3236‡

The statistical significance at the 0.05 level is indicated by †,‡, and § for improvement from 'KLERUN1', 'ikm100jing', and 'ICTNETTSRun2', respectively. The best performance is shown in bold

practical. In other words, to evaluate the performance of the TSIT'09 (TSIT'10), we set the parameter α using the TSIT'09 (TSIT'10) dataset as training set.

Finally, we compare our approaches with the best performing runs of the TSIT'09 and '10 (Macdonald et al. 2010; McCreadie et al. 2011). Tables 5 and 6 show the results of each approach.[15] The TSIT'09 is the retrospective task, while the TSIT'10 is the real-time task. Therefore, we used $Final_{09}^{ret}$ and $Final_{10}^{real}$ as our best runs for TSIT'09 and '10, respectively. Our approaches consistently outperformed the best runs for both TSIT'09 and '10. For TSIT'09, $Final_{09}^{ret}$ achieved 1.4, 4 and 4 % further increases in MAP, P@5 and P@10 over the best performance of TSIT'09 runs. In addition, for TSIT'10, $Final_{10}^{real}$ increased the statMAP score by 1%. We conducted the Wilconxon signed rank test to see whether the improvement of the performance over that of each run was statistically significant.

6 Conclusion and future work

In this study, we have presented several approaches for identifying top news stories, in response to a query day. We evaluate the importance of news stories by investigating the popularity of the stories in the blogosphere. We proposed two components: the NSL and the NSTP. The NSL measures the popularity of a news story based on the similarity between the contents of the story and blog posts published at a given day. For this purpose, we divided the blog posts into K number of clusters so that each cluster can accurately

[15] We are just reusing the results from past TRECs, and that we did not operate under the same conditions as the actual TREC participants who worked within a tight schedule.

contain one of the various topics buried in the blog posts, and then we estimated a language model for each topic. We also introduced two approaches to estimate a language model for the news story. One estimates the news story language model using an article of the story. The other retrieves blog posts relevant to the story, and utilizes them to estimate the language model. Then, we measure the popularity of the news story by evaluating the similarity between the news story language model and the topic language models. The NSTP evaluates the popularity of news stories using a temporal profile of the stories. To estimate the temporal profile of the story, we explore the distribution of the number of blog posts relevant to the story in terms of the timeline. We evaluated the popularity of the news story using its temporal profile.

We performed several experiments to verify the effectiveness of our approaches, in the context of the Blog Track at TREC 2009 and 2010. From experimental results, we can confirm the proposed approaches are effective in identifying top news stories. Furthermore, we obtained the best performance for the task by considering two components, the NSL and the NSTP, at the same time.

Many studies remain for future work. We used K-means clustering to model various topics buried in blog posts. It would be interesting to utilize several approaches such as PLSA (Hofmann 1999) and LDA (Blei et al. 2003) to capture the topics of blog posts. To retrieve blog posts relevant to a news story, we used the headline of a news story as a query. We would like to study various approaches to find out which blog posts are relevant to a news story. For example, the usage of its contents can provide more accurate information about the story than using only the headline. Furthermore, we believe that various features such as comments or tags can be used to improve the performance when identifying top news stories.

References

Allan, J., Papka, R., & Lavrenko, V. (1998). On-line new event detection and tracking. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR'98* (pp 37–45). New York, NY, USA: ACM.

Allan, J., Connell, M. E., Croft, W. B., Feng, F., Fisher, D., & Li, X. (2000). Inquery and trec-9. In *Proceedings of TREC-9*.

Aslam, J. A., & Pavlu, V. (2008). A practical sampling strategy for efficient retrieval evaluation, Technical Report. Narth Eastern University.

Becker, H., Naaman, M., Gravano, L. (2010). Learning similarity metrics for event identification in social media. In *Proceedings of the third ACM international conference on Web search and data mining, WSDM'10* (pp. 291–300). New York, NY, USA: ACM.

Bendersky, M., & Croft, W. B. (2008). Discovering key concepts in verbose queries. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR'08* (pp. 491–498). New York, NY, USA: ACM

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, *3*, 993–1022.

Brants, T., Chen, F., & Farahat, A. (2003). A system for new event detection. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, SIGIR'03* (pp. 330–337) New York, NY, USA: ACM.

Chen, C. C., tsung Chen, Y., Sun, Y. S., & Chen, M. C. (2003). Life cycle modeling of news events using aging theory. In: T*he European conference on machine learning and principles and practice of knowledge discovery in databases* (pp 47–59).

Chen, K. Y., Luesukprasert, L., & Chou, Sc T. (2007). Hot topic extraction based on timeline analysis and multidimensional sentence modeling. *IEEE Transactions on Knowledge and Data Engineering*, *19*, 1016–1025.

Chieu, H. L., & Lee, Y. K. (2004). Query based event extraction along a timeline. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR'04* (pp. 425–432). New York, NY, USA: ACM.

Del Corso, G. M., Gullí, A., & Romani, F. (2005). Ranking a stream of news. In *Proceedings of the 14th international conference on World Wide Web, WWW'05* (pp. 97–106). New York, NY, USA: ACM.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, *39*(1), 1–38.

He, Q., Chang, K., & Lim, E. P. (2007). Analyzing feature trajectories for event detection. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR'07* (pp. 207–214). New York, NY, USA: ACM.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In: *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR'99* (pp 50–57). New York, NY, USA: ACM.

Hu, M., Sun, A., & Lim, E. P. (2008). Event detection with common user interests. In *Proceeding of the 10th ACM workshop on Web information and data management, WIDM'08* (pp. 1–8). New York, NY, USA: ACM.

Jones, R., & Diaz, F. (2007). Temporal profiles of queries. *ACM Transactions on Information Systems*. doi:10.1145/1247715.1247720.

Kleinberg, J. (2002). Bursty and hierarchical structure in streams. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'02* (pp. 91–101). New York, NY, USA: ACM.

Kumaran, G., & Allan, J. (2004). Text classification and named entities for new event detection. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR'04* (pp. 297–304). New York, NY, USA: ACM

Lee, Y., Jung, H, Song, W., & Lee, J. H. (2010). Mining the blogosphere for top news stories identification. In *Proceeding of the 33rd international ACM SIGIR conference on research and development in information retrieval, SIGIR'10* (pp. 395–402). New York, NY, USA: ACM.

Leskovec, J., Backstrom, L., & Kleinberg, J. (2009). Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'09* (pp. 497–506). New York, NY, USA: ACM.

Lin, Y. F., Wang, J. H., Lai, L. C., & Kao, H. Y. (2011). Top stories identification from blog to news in trec 2010 blog track. In *Proceedings of TREC 2010*.

Macdonald, C., & Ounis, I. (2009). Searching for expertise: Experiments with the voting model. *The Computer Journal*, *52*(7), 729–748.

Macdonald, C., Ounis, I., & Soboroff, I. (2010). Overview of trec-2009 blog track. In *Proceedings of TREC 2009*.

McCreadie, R., Macdonald, C., & Ounis, I. (2011). Crowdsourcing blog track top news judgments at trec. In *Proceedings of CSDM 2010*.

McCreadie, R. M. C., Macdonald, C., & Ounis, I. (2010). News article ranking: leveraging the wisdom of bloggers. In *Adaptivity, Personalization and Fusion of Heterogeneous Information, Le Centre De Hautes Etudes Internationales d'Informatique Documentaire, RIAO'10* (pp 40–48). Paris, France.

MediaCollege. (2009). What makes a story newsworthy? http://www.mediacollege.com/journalism/news/newsworthy.html. Last accessed on 22 December 2013.

Mei, Q., Liu, C., Su, H., & Zhai, C. (2006). A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In: *Proceedings of the 15th international conference on World Wide Web, WWW'06* (pp. 533–542). New York, NY, USA: ACM.

Mishne, G., & de Rijke, M. (2006). A study of blog search. In: Lalmas M., MacFarlane A., Rüger S., Tombros A., Tsikrika T., & Yavlinsky A. (Eds.) *Advances in information retrieval*, Lecture Notes in Computer Science, vol 3936, Chap. 26 (pp. 289–301). Berlin/Heidelberg: Springer.

Nam, S. H., Na, S. H., Lee, Y., & Lee, J. H. (2009). Diffpost: Filtering non-relevant content based on content difference between two consecutive blog posts. In: *Proceedings of the 31th European conference on IR research on advances in information retrieval, ECIR'09* (pp. 791–795). Berlin, Heidelberg: Springer.

Song, F., & Croft, W. B. (1999). A general language model for information retrieval. In *Proceedings of the eighth international conference on Information and knowledge management, CIKM'99* (pp. 316–321). New York, NY, USA: ACM.

Sun, A., Hu, M., & Lim, E. P. (2008). Searching blogs and news: a study on popular queries. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR'08* (pp. 729–730). New York, NY, USA: ACM.

Thelwall, M. (2006). Blogs during the London attacks: Top information sources and topics. In *WWE: 3rd Annual workshop on the weblogging ecosystem*.

Tsagkias, M., Weerkamp, W., & de Rijke, M. (2009). Predicting the volume of comments on online news stories. In *Proceeding of the 18th ACM conference on Information and knowledge management, CIKM'09* (pp. 1765–1768). New York, NY, USA: ACM.

Tsagkias, M., de Rijke, M., & Weerkamp, W. (2011). Linking online news and social media. In *Proceedings of the fourth ACM international conference on Web search and data mining, WSDM'11* (pp. 565–574). New York, NY, USA: ACM.

Wang, C., Zhang, M., Ru, L., & Ma, S. (2008). Automatic online news topic ranking using media focus and user attention based on aging theory. In *Proceeding of the 17th ACM conference on Information and knowledge management, CIKM'08* (pp. 1033–1042). New York, NY, USA: ACM.

Weerkamp, W., Tsagkias, M., & de Rijke, M. (2010). From blogs to news: Identifying hot topics in the blogosphere. In: *Proceedings of TREC 2009*.

Yang, Y., Pierce, T., & Carbonell, J. (1998). A study of retrospective and on-line event detection. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR'98* (pp. 28–36). New York, NY, USA: ACM.

Zhai, C., & Lafferty, J. (2001). Model-based feedback in the language modeling approach to information retrieval. In: *Proceedings of the tenth international conference on information and knowledge management, CIKM'01* (pp. 403–410). New York, NY, USA: ACM. doi:10.1145/502585.502654.

Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, *22*, 179–214.

Zhang, K., Zi, J., & Wu, L. G. (2007). New event detection based on indexing-tree and named entity. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR'07* (pp. 215–222). New York, NY, USA: ACM.

Zhang, Y., Callan, J., & Minka, T. (2002). Novelty and redundancy detection in adaptive filtering. In *Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR'02* (pp. 81–88). New York, NY, USA: ACM.