POSTECH at TREC 2010 Blog Track: Top Stories Identification

Yeha Lee, Woosang Song, Hun-young Jung, Vinh Tao Thanh, and Jong-Hyeok Lee

Division of Electrical and Computer Engineering
Pohang University of Science and Technology
San 31, Hyoja-Dong, Nam-Gu, Pohang, 790–784, Republic of Korea
{sion,woosang,blesshy,vinhtt,jhlee}@postech.ac.kr

Abstract. This paper describes our participation in the TREC 2010 Blog Track. For the Top Stories Identification Task, we explore the relationship among news events, news stories and blog posts. We first extract important news events from the TRC2 corpus using a probabilistic mixture model. Then, we propose a probabilistic approach to identify top news stories. Furthermore, we use an additional feature that can be useful in identifying top news stories. For the News Blog Post Ranking Task, we apply the Maximal Marginal Relevance method (MMR) to make the aspects of the blog posts more diverse.

1 Introduction

Blog Track explores information seeking behavior in the blogosphere. In TREC 2010, the Blog Track has two main tasks: Faceted Blog Distillation Task and Top Stories Identification Task. We only participate in the Top Stories Identification Task.

The Top Stories Identification Task was first introduced in the TREC 2009 Blog Track, and consists of two subtasks: Story Ranking Task and News Blog Post Ranking Task. The tasks explore the usefulness of blogosphere in identifying top news stories.

The Story Ranking Task aims to find the most important news stories for a given date query. The TREC 2010 Blog Track task has some properties distinguishing it from the last year. First, the task was treated as online event detection. Therefore, we can use only blog posts which were published at or before a query date. Second, for the task, the TRC2 newswire corpus was provided by Thomson-Reuters. The TRC2 corpus contains news contents as well as news headlines. Finally, we need to provide ranked lists according to each of five categories: world, us, sport, scitech and business.

The News Blog Post Ranking Task retrieves blog posts relevant to a given news story. The retrieved blog posts should cover diverse aspects of the news story.

For the Top Stories Identification Task, our system consists of three components: Preprocessing, News Stories Ranking and News Blog Post Ranking. In Preprocessing, we remove HTML tags and non-relevant contents such as site descriptions and menus. For the Stories Ranking Task, we extract news events from a set of news stories and propose a probabilistic approach to identify the important news stories. For the News Blog Post Ranking Task, we use the Maximal Marginal Relevance [1] to cover diverse aspects related to a given news story.

2 Preprocessing Step

The TREC Blogs08 collection contains permalinks, feed files and blog homepages. We only used the permalink pages for the Top Stories Identification Task. The permalinks are encoded by HTML, and there are many different styles of permalinks. Besides the relevant textual parts, the permalinks contain many non-topical or non-relevant contents such as HTML tags, advertisements, site descriptions, and menus.

The non-relevant contents consist of many different blog templates which may be provided by commercial blog service venders. We used the DiffPost algorithm [2, 3] to deal with the non-relevant contents.

To preprocess the Blogs08 corpus, we firstly discarded all HTML tags and applied the DiffPost algorithm to remove non-relevant contents. DiffPost segments each document into lines using the carriage return as a separator. DiffPost tries to compare sets of lines and then regards the intersection of sets as non-content information.

For example, let P_i and P_j be blog posts within the same blog feed. Let S_i and S_j be the sets of lines corresponding to P_i and P_j, respectively.

$$NoisyInformation(P_i, P_j) = S_i \cap S_j \tag{1}$$

We discarded non-relevant contents through the set difference between a document and noisy-information. Then, we performed additional preprocessing by stemming using the Porter stemmer and eliminating stopwords.

3 Story Ranking Task

The Story Ranking Task aims to find important news stories for a given day (i.e. query). For this task, we explore the relationship among news events, news stories and blog posts. We assumed that news events happen, and then news stories related to them are reported. Subsequently, blog users keep up with the events from the news stories.

Let B_q and N_q be a set of blog posts and a set of news stories published on a given day q (i.e. query), and $I(n_i, q)$ be the importance of a news story n_i on the date. We evaluate the importance of the news story n_i as follows:

$$I(n_i, q) \propto P(n_i | B_q, N_q) \tag{2}$$

The news story set N_q can capture the events that happened at a query date q. Then, by our assumption, we can rewrite the probability as follows:

$$P(n_i | B_q, N_q) \propto P(B_q | n_i) P(n_i | N_q) \tag{3}$$

We call these two probabilities Blog Post Likelihood (BPL) and News Story Likelihood (NSL).

In the following sections, we address how to estimate each probability: BPL and NSL. In addition, we use an additional feature which can be useful in identifying top news stories.

3.1 Blog Post Likelihood

Blog Post Likelihood (BPL) is the probability that a news story will generate a set of blog posts B_q. If a news story is important or newsworthy, the story will attract a lot of attention from many blog users. Then, the users express their opinions or thoughts on their blogs. Therefore, the popularity of the news story in the blogosphere can be used to evaluate its importance. We used BPL to capture the popularity of the news stories in the blogosphere.

To estimate BPL, we adopted a method proposed in [5]. The authors first estimate Blog Language Model (BLM) and News Story Language Model[1] (NSLM), and evaluate BPL based on the language model framework.

We gather blog posts published on a query date (i.e. B_q), and estimate BLM based on the posts. However, the posts may cover diverse topics including sports, economy and science. If you try to capture these topics using a single language model, the language model cannot correctly capture the diverse topics. To mitigate this problem, we divided the blog posts B_q into K_B clusters using the K-means algorithm and estimate the K_B number of BLMs from the clusters. We set the number of clusters $K_B = 300$.

For estimating NSLM, because the contents of a news story are available unlikely the last year's task, we used the contents of a news story instead of blog posts relevant to the news story. Let θ_n be a language model of a news story n. We estimated θ_n using the maximum likelihood estimate of the contents of n and the Dirichlet smoothing [6].

$$P(w|\theta_n) = \frac{c(w;n) + \mu P(w|\theta_C)}{|n| + \mu} \tag{4}$$

where $|n|$ is the length of n, μ is a smoothing parameter which was set to 1000, and $P(w|\theta_C)$ is a collection language model of Blogs08 corpus.

Finally, we estimated the query likelihood using the maximum value among scores, which are evaluated by the KL-divergence language model [7], between the BLMs and the NSLM as follows:

$$P(B_q|n) \propto \max_k \left\{ \sum_w P(w|\theta_{BLM_k}) \log P(w|\theta_n) \right\} \tag{5}$$

3.2 News Story Likelihood

News Story Likelihood (NSL) is the probability that a set of news stories N_q will generate a news story n_i. For evaluating NSL, we extracted news events that happened at the query day from the TRC2 corpus and used the events to estimate the importance of the news story.

Let $E = \left\{e_1, e_2, \ldots, e_{|K_E|}\right\}$ be a set of events that happened at a query day. We can rewrite NSL as follows:

$$P(n_i|N_q) = \sum_{j=1}^{|K_E|} P(n_i, e_j|N_q) = \sum_{j=1}^{|K_E|} P(n_i|e_j) P(e_j|N_q) \tag{6}$$

[1] Blog Language Model and News Story Language Model correspond to Query Language Model and News Headline Language Model, in [5]

where $|K_E|$ indicates the number of news events that happened at a query day. We set $|K_E| = 100$.

For extracting news events, we used a probabilistic mixture model [4] which was used for extracting salient themes (topics) from a stream of text. The main idea of this approach is to assume that each word in the document is a sample from a mixture model with multiple language models, each representing a theme. We regarded the events as the salient topics in N_q.

We used two kinds of language models as components of a mixture model. One is the TRC2 corpus background model to capture common words used in news articles. The other is a set of news event models representing each event that occurred at a query day. Then a news story n_i can be modeled as a sample of words drawn from the following mixture model:

$$P(w;n_i) = \lambda_{TRC2} P(w|\theta_{TRC2}) + (1-\lambda_{TRC2}) \sum_{j=1}^{K_E} \{\pi_{i,j} P(w|e_j)\} \tag{7}$$

where w is a word in a news story n_i, λ_{TRC2} is the mixing weight for θ_{TRC2}, θ_{TRC2} is background model and $\pi_{i,j}$ is a mixing parameter that control a weight for a news story n_i to select a news event e_j such that $\sum_{j=1}^{K_E} \pi_{i,j} = 1$. We set the parameter λ_{TRC2} to 0.8.

We estimate the background probability $P(w|\theta_{TRC2})$ using the TRC2 corpus as follows:

$$P(w|\theta_{TRC2}) = \frac{\sum_{n \in TRC2} c(w;n)}{\sum_{w \in V} \sum_{n \in TRC2} c(w;n)} \tag{8}$$

where $c(w;n)$ is the count of word w in a news story n.

We used the EM algorithm to estimate the parameters, $\pi_{i,j}$ and $P(w|e_j)$, by maximizing the log-likelihood of the news stories N_q according to the mixture model.

The remaining issues are how to evaluate NSL, that is, two probabilities, $P(n_i|e_j)$ and $P(e_j|N_q)$. First, $P(n_i|e_j)$ implies the probability that a news event will generate a news story. We calculated the probability using $\pi_{i,j}$ which was already estimated from the EM algorithm. As mentioned in the event extraction step, $\pi_{i,j}$ means a weight that a news story n_i will choose a news event e_j. We can view $\pi_{i,j}$ as the conditional probability $P(e_j|n_i)$. Thus, we estimated the probability $P(n_i|e_j)$ using Bayes' rule as follows:

$$P(n_i|e_j) = \frac{\pi_{i,j}}{\sum_{j=1}^{K_E} \pi_{i,j}} \tag{9}$$

We assume that $P(n_i)$ for all the news stories has an uniform distribution.

Next, the probability $P(e_j|N_q)$ can be viewed as the prior about the importance of the news event. There can be a number of approaches to estimate the prior. We evaluate the prior using blog data, because we want to investigate the usefulness of the blogosphere in identifying top news stories. For this purpose, similar to BPL, we use the probability that a news event will generate blog posts as the prior.

We first smooth the news event language model, $P(w|e_j)$, estimated in the Event Extraction step using Jelinek-Mercer smoothing [6].

$$\widetilde{P}(w|e_j) = (1-\lambda) P(w|e_j) + \lambda P(w|\theta_C) \tag{10}$$

where λ is a smoothing parameter, we set λ to 0.8. Then, we can estimate the probability $P(e_j|N_q)$ as follows:

$$P(e_j|N_q) \propto \max_k \left\{ \sum_w P(w|\theta_{BLM_k}) \log \widetilde{P}(w|e_j) \right\} \tag{11}$$

Now, using the results of Eq. 9 and 11, we can evaluate NSL from Eq. 6. However, a news story is usually dedicated to only one news event. We reformulate Eq. 6 as follows:

$$P(n_i|N_q) = \sum_{j=1}^{|K_E|} \delta(e_j, n_i) P(n_i|e_j) P(e_j|N_q) \tag{12}$$

where

$$\delta(e_j, n_i) = \begin{cases} 1 \text{ if } e_j \text{ is same to a news event of } n_i \\ 0 \text{ otherwise} \end{cases}$$

We assume $\delta(e_\gamma, n_i) = 1$ when $\gamma = \operatorname{argmax}_j \pi_{i,j}$.

3.3 Additional Feature

Similar to last year's approach, we used an additional feature for this task, Temporal Profile.

Temporal Profile uses the temporal information of blog posts relevant to a news story n. To achieve this, we first retrieved blog posts using a news story as a query. Similar to event extraction in section 3.1, we assumed that the news story n is generated by a mixture model consisting of a query model (news story model) θ_n and a background model θ_{TRC2}. We use the EM algorithm to estimate the query model, and evaluate the relevance score between the news story n and a blog post d using the KL-divergence language model [7].

We evaluated the temporal profile of n using an approach proposed in [5]. We select 100 blog posts between -14 and +0 days from a query date in order of relevance score. The smoothing parameter α was set to 0.5 and the cosine kernel parameter σ was set to 50. The period ϕ was set between -7 and +0 days from the query day.

3.4 Final Ranking Function

Let $S_I(n_i, q)$ and $S_F(n_i, q)$ be scores estimated using the importance of a news story and the temporal profile, respectively.

$S_I(n_i, q)$ can be calculated using Eq. 2 as follows:

$$\begin{aligned} I(n_i, q) &\propto P(B_q|n_i)P(n_i|N_q) = P(B_q|n_i)P(n_i|e_\gamma)P(e_\gamma|N_q) \\ S_I(n_i, q) &= \log P(B_q|n_i)P(n_i|e_\gamma)P(e_\gamma|N_q) \end{aligned} \tag{13}$$

To integrate two scores, we first adjust each score from 0 to 1.

$$\widetilde{S}(n, q) = \frac{S(n, q) - min_{n,q}}{max_{n,q} - min_{n,q}} \tag{14}$$

where $min_{n,q} = \min_n S(n,q)$ and $max_{n,q} = \max_n S(n,q)$. $S(n,q)$ indicates one score of $S_I(n,q)$ and $S_F(n,q)$.

Finally, we defined the ranking function as follows:

$$Score(n,q) = (1-\beta)\widetilde{S_I}(n,q) + \beta\widetilde{S_F}(n,q) \tag{15}$$

where β is the weighting parameter. We set β to 0.8.

3.5 News Stories Classification

In the TREC2010 Blog Track, we need to provide a ranked list with five categories (world, us, sport, scitech and business), instead of an overall ranking. After ranking all the news stories using the approaches described in the above sections, we classified them into five categories using the SVM classifier[2].

To train the classifier, we used the categories of the New York Times[3]: WORLD, U.S., SPORTS, TCHNOLOGY+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.

4 News Blog Post Ranking Task

For the News Blog Post Ranking Task, we viewed a given news story as a query and retrieved blog posts relevant to the news story. Then, to select blog posts that provide diverse aspects of the news story, we used the Maximal Marginal Relevance [1] method.

$$\hat{d} = \underset{d_i \in R \setminus S}{\operatorname{argmax}} \left\{ \lambda Sim_1(d_i, n) - (1-\lambda) \max_{d_j \in S} Sim_2(d_i, d_j) \right\} \tag{16}$$

where R is the ranked list of blog posts, S is the subset of blog posts in R already selected. $R \setminus S$ is the set difference, i.e, the set of as yet unselected documents in R, Sim_1 is the similarity metric used in document retrieval. Finally, Sim_2 is a similarity metric between d_i and d_j. We defined Sim_2 using the cosine measure.

5 Run

5.1 Story Ranking Task

For the Story Ranking Task, we submitted 3 runs as follows:

1. **KLERUN1**: Using only Blog Post Likelihood - $P(B_q|n_i)$
2. **KLERUN2**: Using the importance of a news story - $\log P(n_i|B_q, N_q)$
3. **KLERUN3**: Interpolating the score of KLERUN2 with Temporal Profile - $0.2 \times \widetilde{S_I}(h,q) + 0.8 \times \widetilde{S_F}(h,q)$

Run	Measure	Categories				
		business	scitech	sport	us	word
KLERUN1	statMAP	0.1851	0.1821	0.1916	0.2458	0.2986
	statMPC_10	0.5314	0.2878	0.3310	0.5630	0.7501
	statMNDCG_10	0.1318	0.2227	0.0985	0.2146	0.1781
KLERUN2	statMAP	0.0976	0.1356	0.2088	0.1917	0.2597
	statMPC_10	0.2778	0.2489	0.2870	0.4407	0.6027
	statMNDCG_10	0.0719	0.1727	0.0844	0.1778	0.1525
KLERUN3	statMAP	0.1302	0.1343	0.2488	0.1683	0.2684
	statMPC_10	0.4055	0.2063	0.4363	0.3342	0.6205
	statMNDCG_10	0.1071	0.1498	0.1540	0.1293	0.1850

Table 1. Story Ranking Task

Run	Type	Measure		
		alpha-nDCG@10	P-IA@10	nERR-IA@10
KLE1	before	0.466517	0.156096	0.434547
	day	0.462569	0.164044	0.428372
	week	0.466335	0.166754	0.426638
	amean	0.465140	0.162298	0.429852
KLE2	before	0.458607	0.159821	0.422684
	day	0.457419	0.167190	0.423790
	week	0.466968	0.169818	0.429638
	amean	0.460998	0.165610	0.425371

Table 2. News Blog Post Ranking Task

Table 1 shows the performances of our runs according to each category. For all the categories except sport, KLERUN1 yields the best performances, and KLERUN2 results in the worst. These results may mean that NSL approach using news event extraction failed to improve the performance of the Story Ranking Task. We have to find a way to extract news events and take advantage of the events for the Story Ranking Task. We retain this problem for future work.

5.2 News Blog Post Ranking Task

For the News Blog Post Ranking Task, we submitted 2 runs as follows:

1. **KLE1**: $\lambda = 0.2$ in Eq. 16
2. **KLE2**: $\lambda = 0.5$ in Eq. 16

Table 2 shows the performances of our runs.

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

[3] http://www.nytimes.com

6 Conclusion

We have described our participation in the TREC 2010 Blog Track. Compared with the last year, we proposed an approach using event extraction to identify top news stories. Although the approach failed to improve the performance of the Story Rank Task, we think there is still room for refinement. We will further research in this direction. For the News Blog Post Rank Task, we used the similarity between blog posts to obtain diverse posts about the news story. We think there are several methods such as opinion mining and considering feed information that may increase the diversity of the blog posts.

7 Acknowledgement

This work is supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (MEST) (2010-0012662), in part by the BK 21 project in 2011, and in part by the POSTECH Information Research Laboratories (PIRL) project.

We appreciate providing the TRC2 newswire corpus from Thomson-Reuters.

References

1. Carbonell, J., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: SIGIR ’98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM (1998) 335–336
2. Lee, Y., Na, S.H., Kim, J., Nam, S.H., young Jung, H., Lee, J.H.: Kle at trec 2008 blog track: Blog post and feed retrieval. In: Proceedings of TREC 2008. (2008)
3. Nam, S.H., Na, S.H., Lee, Y., Lee, J.H.: Diffpost: Filtering non-relevant content based on content difference between two consecutive blog posts. In: ECIR. Volume 5478 of Lecture Notes in Computer Science., Springer (2009) 791–795
4. Mei, Q., Zhai, C.: Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In: KDD ’05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, New York, NY, USA, ACM (2005) 198–207
5. Lee, Y., Jung, H.y., Song, W., Lee, J.H.: Mining the blogosphere for top news stories identification. In: SIGIR ’10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM (2010) 395–402
6. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. **22**(2) (2004) 179–214
7. Lafferty, J., Zhai, C.: Document language models, query models, and risk minimization for information retrieval. In: Proceedings of SIGIR 2001, ACM (2001) 111–119