BIT at TREC 2009 Faceted Blog Distillation Task

Peng Jiang, Qing Yang, Chunxia Zhang, Zhendong Niu

School of Computer Science and Technology, Beijing Institute of Technology
{jp, yangqing2005, cxzhang, zniu}@bit.edu.cn

Abstract. This Paper presents the work done for the TREC 2009 faceted blog distillation task of blog track. In our approach, we use a mixture of language models based on global representation. Our model can be regarded as a combination of topic relevance model and faceted relevance model. By pseudo-relevance feedback method, we can estimate the above two models from topic relevance feedback documents and facet relevance feedback documents respectively. Experimental results on TREC blogs08 collection show the effectiveness of our proposed approach.

1 Introduction

This is the first time that Beijing Institute of Technology participates in TREC and our focus is on faceted blog distillation task of blog track. For the faceted blog distillation task, we aim to evaluate the effectiveness of combining different language models for faceted blog ranking and using global representation for feed retrieval. Experiments results on TREC blogs08 datasets show improvement by using facet-specific language models over the baseline. Moreover, the lexicons (subjective or objective lexicon) used are domain-independent. Hence our proposed approach is applicable to all retrieval tasks on any text resource containing information about topic and multiple facets such as opinionated nature, authors' trustworthiness and writing style.

The goal of the blog distillation task is to 'Find feeds that are principally devoted to topic X'. The blog distillation task is different from the other tasks in that its retrieval unit is feed which contains a number of web documents. This year, blog distillation task involves a number of facets, such as opinionated, personal and in-depth. It goes beyond the topic relevance and thus the facet nature must be considered.

The rest of the paper is organized as follows. In Section 2, a system overview is presented briefly. The query generation is described in Section 3. Section 4 introduces the dataset and index information. The whole approach is described in Section 5. The experimental results are presented in Section 6. Finally we conclude the paper and discuss the future work in Section 7.

2 System Overview

Our system consists of four main parts: the query generator, the blog retrieval component, the language model building component and the rank component. The

architecture is shown in Fig. 1.

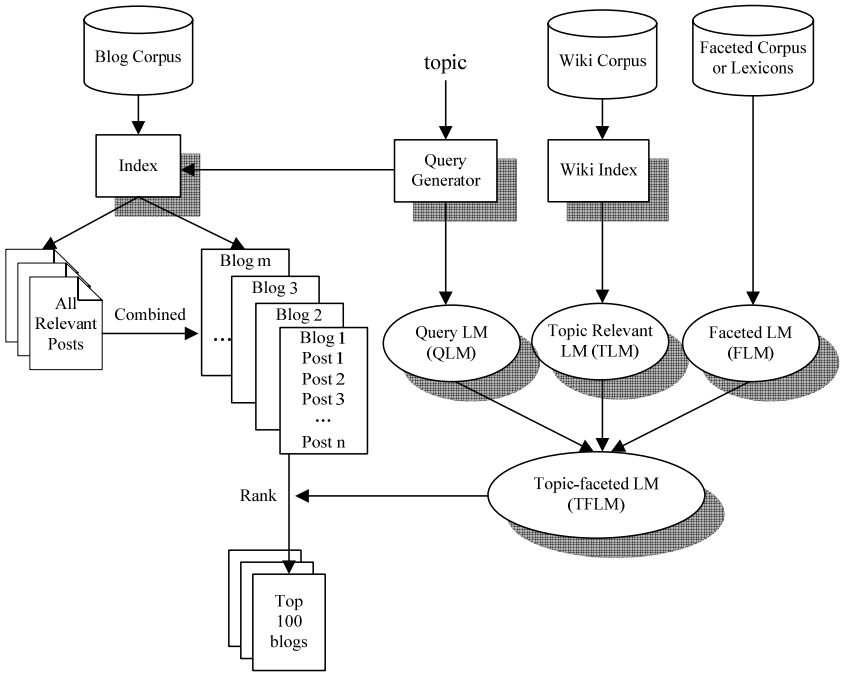


Fig. 1. The System Architecture.

The query generator is responsible for parsing topic and generates query strings. The Blog retrieval component indexes blog corpus, and finds all blog posts corresponding to each query string. All posts are then combined to blogs by their feedno. After this, the component finds all posts which belong to a given blog by the feedno. The Language model building component is responsible for generating topic-faceted language model, which is a combination of query language model, topic relevant language model and facet language model. This component generates the above language models respectively and then combines them. The Rank component is in charge of ranking all blogs based on Kullback–Leibler divergence language model.

3 Query Generation

All of the three fields (title, description, narrative) are used for generating query. First, we filter out unnecessary punctuation marks in the text of the above three fields from each topic. All verbs are replaced by their infinitives and all nouns by their singular forms. Then, we extract the keywords to build three bags of words to three fields of topic, after which WordNet is used to expand these three bags of words. The word and its synonyms in WordNet are built into an indri query unit, e.g. the indri query unit of “subsidy”:

#or(subsidy grant subvention)

The query units of all words in the bag form an indri query. For example, a bag of words (w_1, w_2, w_3, …, w_n) can form into an indri query:

#combine(u_1, u_1, …, u_i, …, u_n)

u_i is the indri query unit such as #or(w_i, s_{i1}, s_{i2}, …, s_{im}), where s_{im} is m-th synonym of w_1. The three fields of topic are assigned different weights, so the final indri query for a given topic is:

#weight(0.6 $field_{title}$ 0.2 $field_{desc}$ 0.2 $field_{narr}$)

$field_{title}$, $field_{desc}$ and $field_{narr}$ are the indri query strings generated by title, description and narrative field of topic.

4 Dataset and Index

TREC Blog09 collection contains permalinks, feed pages and homepages. We use the permalinks and homepages for the faceted blog distillation task. Feed pages collection is not used because feed pages usually contain a few sentences of each post and therefore cannot reflect the topic or opinion well.

The permalinks and homepages are encoded by HTML. We use Indri to index them respectively. We specify some fields and metadata for the index so that we can search and combine posts to blogs flexibly. These fields and metadata are shown in Table 1. Krovetz stemmer and a list with 450 stop words (e.g. a, about, above or many other common but useless words) are used.

Table 1. Fields and Metadata of Permalinks and Homepages Index

Name	Type	Description
TITLE	Field	The content in title tag
DATE_XML	Field	Data information of a permalink document
FEEDNO	Metadata for permalinks	The ID of a feed document
FEEDURL	Metadata for permalinks	The URL of a feed document
BLOGHPURL	Metadata for permalinks	The URL of a blog
PERMALINK	Metadata for permalinks	The URL of a permalink document
HPNO	Metadata for homepages	The ID of a homepage document
URL	Metadata for homepages	The URL of a homepage document

5 Our Approach

We choose Global Representation model[1] to represent feed. This model treats a blog as a virtual document which is comprised of all posts of the blog. Thus, this model can factually reflect the recurring interest in a given topic over the time span of the feed. In addition, since we use language model based approach to rank feed, Global Representation model, which combines many posts into a large document, can avoid the problem of sparsity of words as far as possible.

In order to combine all posts into one large virtual document, the first step is to find all relevant posts for a given topic, which are then combined into some feeds by

their feedno fields. Then, metadata search is used to get all posts of a given feed by the feed's feedno. Finally, we obtain all relevant feeds which contain not only relevant posts but also irrelevant posts.

We rank feeds by the Kullback-Leibler Divergence of a feed language model and a topic-facet language model. In this solution, two different language models are defined: one for a topic with facet value (θ_{TF}); and another for the virtual document of a feed (θ_D). That is, we assume that θ_{TF} represents the topic and facet information need, while θ_D represents a feed. The KL-divergence of these two models is able to measure how close they are to each other. Thus, we can rank feeds by following formula:

$$\begin{aligned} score_{TF}(D) &= -D(\theta_{TF} \,||\, \theta_D) \\ &= -\sum_w p(w|\theta_{TF}) \log \frac{p(w|\theta_{TF})}{p(w|\theta_D)} \\ &= \sum_w p(w|\theta_{TF}) \log p(w|\theta_D) + cons(\theta_{TF}) \end{aligned} \quad (1)$$

Because the constant $cons(\theta_{TF})$ (the entropy of θ_{TF}) does not affect the results of ranking feeds, we do not compute it in our system. θ_D can be estimated by query likelihood retrieval model[2]. Thus, the main task is to estimate θ_{TF}.

θ_{TF} is the language model which reflects not only the topic information need but also the facet information need. Hence a mixture of language models is used to estimate it. In our solution, we define three language models, namely, the query language model (θ_Q), the topic relevant language model (θ_T) and the facet relevant language model (θ_F). The topic and facet language model θ_{TF} is a linear combination of the three language models θ_Q, θ_T, and θ_F:

$$\theta_{TF} = \alpha\theta_Q + \beta\theta_T + \gamma\theta_F \quad (2)$$

Where α, β and γ are given, and sum up to 1.

In equation (2), θ_Q can be computed by query likelihood retrieval model. θ_T can be regarded as a feedback topic model estimated by pseudo-relevance feedback method. In order to increase the quality of feedback document, we index the Wikipedia corpus to obtain feedback documents.

The final step is to estimate θ_F in equation (2). It reflects the users' facet information need. In this year's task, there are three facets to be considered: opinionated, personal and in-depth. Each facet has two faceted values. For opinionated facet, the values of interest are "opinionated" and "factual". A subjective lexicon and an objective lexicon are chosen. We compute the mutual information between words w_i from the above lexicons and the query Q of a given topic:

$$MI(w_i, Q) = \log \frac{p(w_i, Q)}{p(w_i)\,p(Q)} = \log \frac{Hits(\#uw15(w_i \quad Q)) \times |C|}{Hits(w_i) \times Hits(Q)} \quad (3)$$

Here $|C|$ is the total number of documents in corpus. $Hits(w_i)$ and $Hits(Q)$ are the counts of documents which contain w_i and Q respectively. $Hits(\#uw15(w_i \quad Q))$ is the count of documents which contain w_i and Q in a window of 15 terms. The reason why we use a fixed size window instead of a sentence is that: it is time-consuming and unpractical to split all text into sentences, and the problems related to inaccuracy can be ignored when large corpus is used. Finally we choose the top 100

subjective/objective words according to the mutual information value to expand query and re-retrieve from the posts which are retrieved for the first time. The subjective/objective feedback documents can be used to build opinionated/factual faceted language model by pseudo-relevance feedback method.

For personal facet, the values of interest are "personal" and "official". In order to build θ_F for these two faceted values, we seek help from icerocket blog meta-search engine and OpenSearch. Because all blogs in blogs.myspace are personal, we use icerocket to get the top 30 documents from blogs.myspace for a given topic, and all of them can be used as relevant feedback documents for personal facet value. We select some tags such as "company" and "commercial", which can distinguish official blogs from personal blogs, and add them to icerocket to search for official blog posts. The search results for official posts are then used as relevant feedback documents for official facet value. Pseudo-relevance feedback method is used again for the estimate of θ_F. For in-depth facet, we assume its language model is similar to that of opinionated facet, but with different values of parameters. For example, a factual blog with high topic relevance is always an in-depth blog, but an opinionated blog with low topic relevance is always a shallow blog.

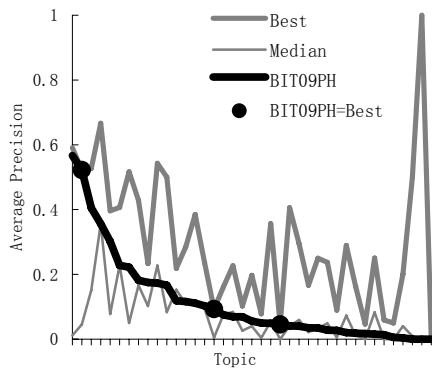
6 Experimental Results

We submit 2 runs: BIT09PH used permalinks indexed corpus, while BIT09P used homepages indexed corpus. Results are provided in Table 3. The relevance and facet judgments of the TREC 2009 faceted blog distillation task are categorized into five grades, i.e. not judged (-1), not relevant (0), relevant (1), relevant and inclined towards first facet value (2) and relevant and inclined towards second facet value (3).

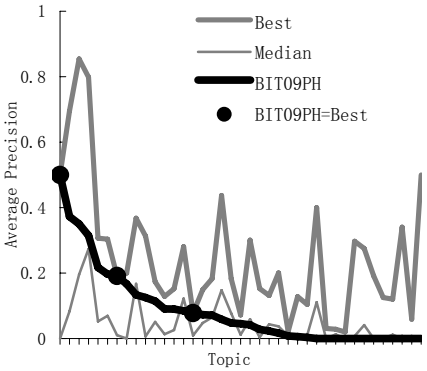
Table 2. Results on dataset excluding data of January 2008

	Run	Map	P10	R-prec	bpref
first	BIT09PH	0.1168	0.1256	0.1276	0.1028
	BIT09P	0.0895	0.1154	0.1149	0.0881
second	BIT09PH	0.0884	0.0897	0.1011	0.0758
	BIT09P	0.0673	0.0538	0.0617	0.0465
none	BIT09PH	0.1165	0.2513	0.1714	0.1347
	BIT09P	0.0708	0.2000	0.1852	0.1207

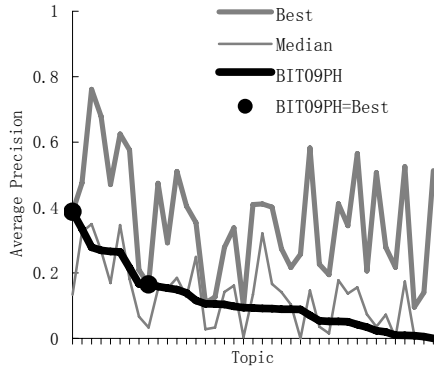
Figures 2 (a) (b) and (c) show our system's per-topic performance in terms of average precision (AP), alongside with the per-topic median and best performance, respectively. All 39 topics are sorted along the x axis in descending order of BIT09PH performance. The dots indicate the topics for which BIT09PH obtains the best performance.



(a) AP of first facet value for each topic



(b) AP of second facet value for each topic



(c) AP of none facet value for each topic

Fig. 2. Average precision of the three facet values for each topic.

The index for our submitted results does not contain the data from January 2008 (14/01/2008 - 31/01/2008) due the time constraints. Subsequently, we index this omitted data and add it to the whole index. The final results are as shown in Table 3. It can be seen that the performances has improved significantly. This is due to the large data size of January 2008: it is almost comparable to half the size of the whole dataset. Relevant blogs which are retrieved by our system from the dataset excluding January 2008 only cover 44.47% of the relevant blogs. However the coverage increases to 89.81% after the data of January 2008 is included.

Table 3. Results on dataset including data of January 2008

	Map	P10	R-prec	bpref
first	0.2228	0.2385	0.2437	0.2000
second	0.1796	0.1436	0.1565	0.1341
none	0.3009	0.4462	0.3502	0.3210

7 Conclusions

We apply a mixture of language models based on a global representation of the blogs to the faceted blog distillation task. The results have proved the effectiveness of our approach, especially after including the omitted data of January 2008. There is still a huge potential space for further research to improve the performance of blog retrieval. We will explore new models or approaches to rank blogs, and ways to fuse different models to a better model.

Acknowledgments. This work is supported by the grant from Chinese National Natural Science Foundation (No: 60705022).

References

[1]. Seo, J., Croft, W.B.: UMass at TREC 2008 Blog Distillation Task. Proceedings of TREC-08 (2008)
[2]. Zhai, C.: Statistical Language Models for Information Retrieval A Critical Review. Foundations and Trends in Information Retrieval (2008)