An Article Language Model for BBS Search

Jingfang Xu, Yangbo Zhu, and Xing Li

Department of Electronic Engineering, Tsinghua University
Beijing 100084, P.R. China
{xjf02,zhuyangbo99}@mails.tsinghua.edu.cn, xing@cernet.edu.cn

Abstract. Bulletin Board Systems (BBS), similar to blogs, newsgroups, online forums, etc., are online broadcasting spaces where people can exchange ideas and make announcements. As BBS are becoming valuable repositories of knowledge and information, effective BBS search engines are required to make the information universally accessible and useful. However, the techniques that have been proven successful for web search are not suitable for searching BBS articles due to the nature of BBS. In this paper, we propose a novel article language model (LM) to build an effective BBS search engine. We investigate the differences between BBS articles and web pages, then extend the traditional LM to author LM and category LM. The article LM is powerful in the sense that it can combine the three LMs into a single framework. Experimental results shows that our article LM substantially outperforms both INQUERY algorithm and the traditional LM.

1 Introduction

Bulletin Board System(BBS) are online broadcasting spaces where people can exchange ideas and make announcements. Unlike web site, where users only browse web pages, BBS are virtual places where, besides browsing, people carry on discussion with others. Some users post articles on BBS to ask questions, answer questions or share information with others, while others browse the articles in BBS for information they need. As BBS are valuable repositories of knowledge and information, there is a tremendous need for BBS search techniques.

BBS search can be simply defined as the search engine specific to BBS. Unfortunately, the techniques that have been proven successful for web search are not suitable to BBS search due to the nature of BBS. In this paper, BBS articles are defined as messages in BBS which contain documents, authors and categories. We investigate the differences between BBS articles and web pages to discover the nature of BBS which affects search performance. Comparing BBS articles and web pages, there are three primary differences. First, BBS articles consist of documents, authors who wrote it, and categories which them belong to. While web pages usually only contain documents; the authors and the categories of them are not available. Second, as users sometimes post articles to ask questions or answer questions, the articles are often shorter than web pages. Finally, there are no links between BBS articles while web pages are closely connected by hypertext links. Consequently, on the one hand many techniques

D. Lowe and M. Gaedke (Eds.): ICWE 2005, LNCS 3579, pp. 152–160, 2005.

applied in web search, e.g., link analysis and anchor text, are not suitable for BBS search; one the other hand, the authors and categories of BBS articles may be helpful for ranking retrieved articles. However, to our best knowledge, there is no effort devoted to building special models for BBS search.

In this paper, we explore the problem of building an effective BBS search with the article language model(LM). According to the nature of BBS, we apply the traditional document LM to the documents belong to an author or a category, and propose the author LM and the category LM. Combining the document, author and category LMs, the article LM is used to rank the retrieved articles according to the probability that the article LM produces the query. To improve the retrieval performance, we propose a smoothing method to address the problem that BBS articles are usually short, which affects the precision of sampling. Experimental results show that the article LM achieves significantly better performance than both INQUERY ranking algorithm and the traditional LM.

The rest of the paper is organized as follows. First we briefly review the related work in section 2. Then the traditional LM, the article LM, and a smoothing method for BBS search are described in section 3. Section 4 presents the experiments and results, which shows the effectiveness of the article LM empirically. Finally we conclude with future work in section 5.

2 Related Work

As BBS becomes more and more popular, considerable effort has been devoted to investigate the special phenomenon in BBS, such as its complex network model[1], its aliasing users[2], and its relationship of interests [3]. Many information retrieval(IR) techniques such as PageRank[4], HITS[5] and anchor text[6] are designed for web search, but they are not suitable for other applications. Therefore, many research have been done to improve IR performance in other context, e.g., web site[7], newsgroup[8], workplace[9], etc. In this paper we focus on improving the retrieval performance in BBS.

In the mean time, LMs, successfully used for speech recognition, have been applied to various IR systems, e.g., web search[10], resource selection[11], etc. Generally, IR systems can be classified by the underlying conceptual models, such as Boolean model, vector-space model, probabilistic model and LM. Ponte and Croft originally proposed LM for IR [10], then Song put emphasis on data smoothing techniques in LM[12]. Recently, many variations of traditional LM have been developed to improve IR performance, such as relevance-based language model[13], time-based language model[14] and title language model[15]. In this paper, we extend the traditional document LM to the author LM and the category LM according to the nature of BBS articles.

3 Language Model

3.1 Document Language Model

As described by Ponte and Croft[10], in document LM each document is viewed as a language sample and a query is treated as a generation process, sampled

from the language model. Then the retrieved documents are ranked based on the probabilities of producing the query from the corresponding language models of them. Given the language model M_d of document d, the maximum likelihood estimate of the probability of term t produced by the corresponding LM is computed by Equation 1:

$$p_{(t|M_d)} = \frac{tf_{(t,d)}}{N_d} \tag{1}$$

where $tf_{t,d}$ is the raw term frequency of term t in document d, and N_d is the total number of tokens in document d. We treat a query as a sequence of terms. Then each term is viewed as an independent event, and the query is viewed as the joined event[12]. Consequently, the query probability is computed by multiplying the individual term probabilities, as shown in Equation 2:

$$p_{(Q|M_d)} = \prod_{t \in Q} p_{(t|M_d)} \tag{2}$$

where t is the term in the query sequence Q. Notice that, in this paper, first we use Boolean model to get the documents that contain the whole query, then apply LMs to rank them. So the zero probability problem does not exist in our research.

3.2 Author Language Model and Category Language Model

As mentioned above, each BBS article contains an author and a category, which can be used to improve the performance of BBS search. The traditional document LM infers a LM for each document and views the document as a sample of the model. Similarly, we infer an author LM for each author and a category LM for each category. That is to say, all the articles of an author are viewed as a huge document, a sample from the corresponding LM. Like probability in the document LM, we compute the probability of term t produced by the author LM M_a. We also get the category LM M_c, corresponding to all the articles belonging to a category, and compute the probability of producing the query term, as shown in Equantion 3, 4:

$$p_{(t|M_a)} = \frac{tf_{(t,a)}}{N_a} \tag{3}$$

$$p_{(t|M_c)} = \frac{tf_{(t,c)}}{N_c} \tag{4}$$

where $tf_{t,a}$ and N_a are the raw term frequency of term t and the total number of tokens in all articles written by author a, and $tf_{t,c}$ and N_c are the raw term frequency of term t and total number of tokens in all articles belong to category c.

3.3 Article Language Model

In order to improve the BBS search performance, we combine three LMs: document, author, and category LMs into the article LM. Each BBS article is viewed

as a trigram (d, a, c), where d is the document text, a is the author and c is the category. Then the retrieved articles are ranked according to the probability $p_{(Q|article)}$ that the query is sampled form the article LM.

$$p_{(Q|article)} = p_{(Q|d,a,c)} = \prod_{t \in Q} p_{(t|d,a,c)} \tag{5}$$

Assuming that the three LMs are independent, $p_{(t|d,a,c)}$ is calculated as follows:

$$\begin{aligned} p_{(t|d,a,c)} &= \frac{p_{(d,a,c|t)}p_{(t)}}{p_{(d,a,c)}} \\ &= \frac{p_{(d|t)}p_{(a|t)}p_{(c|t)}p_{(t)}}{p_{(d,a,c)}} \\ &= \frac{\frac{p_{(t|d)}p_{(d)}}{p_{(t)}} \frac{p_{(t|a)}p_{(a)}}{p_{(t)}} \frac{p_{(t|c)}p_{(c)}}{p_{(t)}} p_{(t)}}{p_{(d)}p_{(a)}p_{(c)}} \\ &= \frac{p_{(t|d)}p_{(t|a)}p_{(t|c)}}{p^2_{(t)}} \end{aligned} \tag{6}$$

where $p_{(t|d)}$ is the probability that the document LM produces the query term, $p_{(t|a)}$ is the probability of the author LM and $p_{(t|c)}$ is the probability of the category model. Therefore, $p_{(Q|article)}$ is the product of the probabilities in three LMs, as shown in Equation 7:

$$p_{(Q|article)} = \prod_{t \in Q} \frac{p_{(t|d)}p_{(t|a)}p_{(t|c)}}{p^2_{(t)}} = \prod_{t \in Q} \frac{p_{(t|M_d)}p_{(t|M_a)}p_{(t|M_c)}}{p^2_{(t)}} \tag{7}$$

where $p_{(t|M_d)}$, $p_{(t|M_a)}$ and $p_{(t|M_c)}$ are computed according to Equation 2, 3, 4 and $p_{(t)}$ can be computed according to Equation 8:

$$p_{(t)} = \frac{tf_{(t,corpus)}}{N_{corpus}} \tag{8}$$

where $tf(t, corpus)$ is the term frequency of term t and N_{corpus} is the total number of tokens in the corpus.

3.4 Data Smoothing

Some articles in BBS are too short, e.g., less than 15 words, as users post them only to ask or answer questions. The small length affects the precision of maximum likelihood estimate as the sparse sampling data can not reflect the underlying LM exactly. Moreover, the probability is biased towards short articles. To address this problem, we smooth the length by adding the average sample length of the LM to the original length. That is to say, in document LM N_d is replaced by $N_d + aveg_{N_d}$, where $aveg_{N_d}$ is the average length of all the documents. Similarly, some authors post few articles and some categories contain few articles, which lead to sparse sampling data in both the author LM and the category LM. We apply similar smoothing methods on the author LM M_a and the category LM M_c, as presented in Equation 9:

$$
\begin{aligned}
p_{(t|M_d)} &= \frac{tf_{(t,d)}}{N_d + aveg_{N_d}} \\
p_{(t|M_a)} &= \frac{tf_{(t,a)}}{N_a + aveg_{N_a}} \\
p_{(t|M_c)} &= \frac{tf_{(t,c)}}{N_c + aveg_{N_c}}
\end{aligned} \tag{9}
$$

4 Experimental Results

4.1 Data

To demonstrate the effectiveness of the article LM, we conducted experiments on a Chinese BBS site, *Tsinghua University BBS (SMTH)*[1], which is the most famous and most large BBS site in China. The BBS search engine that we build for SMTH is accessible on line[2]. As table 1 lists, in SMTH there are 1,954,689 articles, totally 11 gigabytes information, belongs to 85,062 authors and 424 categories. And the average lengthes of document, author and category are 150, 3,447 and 691,517.

Table 1. Statistics of SMTH BBS Data

Data Size	11G
#Articles	1,954,689
#Author	85,062
#Category	424
Average length of documents	150
Average length of author	3,447
Average length of category	691,517

4.2 Implementation

Three different ranking algorithms are used in our experiments. The article LM is compared with INQUERY ranking formula and the traditional document LM. INQUERY ranking formula, which uses Robertson's *tf* score and a standard *idf* score, is lists as follows:

$$
p_{t,d} = 0.4 + 0.6 \times \frac{tf_{t,d}}{tf_{t,d} + 0.5 + 1.5\frac{N_d}{arvgN_d}} \times \frac{log(\frac{N+0.5}{df_t})}{log(N+1)} \tag{10}
$$

where $p_{t,d}$ is that probability that document d is relevant to term t, N is number of documents in collection, and df_t is the document frequency in collection. In INQUERY system, documents are ranked by the probability $p_{t,d}$.

Traditional LM is described as the document LM above, and the retrieved documents are ranked based on the probability $p_{(t|M_d)}$. In the article LM, retrieved articles are ranked based on the probability $p_{(Q|article)}$.

[1] http://www.smth.org
[2] http://bbs.compass.edu.cn

4.3 Evaluation Metrics

The recall/precision and minimizing Kendall's τ distance [16] are chosen to measure the performance of various models. Recall measures how many relevant articles are retrieved, while precision measures how highly the retrieved articles are relevant to the query. In recall/precision metric, relevant articles are unordered, which is opposite to the fact that users review the result lists orderly. To evaluate the order of result lists, minimizing Kendall's τ distance is adopted to measure the similarity between the ground truth and the top-k lists produced by the models.

The minimizing Kendall's τ distance is defined as follows. Given two top k lists τ_1 and τ_2 with result domains $D_{\tau 1}$ and $D_{\tau 2}$, we define $P(\tau 1, \tau 2)$ to be the set of all unordered pairs of distinct elements in $D_{\tau 1} \cup D_{\tau 2}$. The minimizing Kendall's τ distance is calculated according to Equation 11:

$$K_{min}(\tau_1, \tau_2) = \frac{\sum_{\{i,j\} \in P(\tau_1, \tau_2)} K^{min}_{\{i,j\}}(\tau_1, \tau_2)}{k \times (k-1)/2} \tag{11}$$

Let $r_{1,i}$ denote the rank of result i in list τ_1, and result i is ahead of result j in τ_1 if $r_{1,i} < r_{1,j}$. Then $K^{min}_{\{i,j\}}$ is calculated according to Equation 12:

$$K^{min}_{\{i,j\}} = sign\{(r_{1,i} - r_{1,j})(r_{2,i} - r_{2,j})\} \tag{12}$$

4.4 Results and Discussions

Several volunteers are required to pose 50 queries, which are the top frequent queries in the log of BBS search engine that we build, to three systems in our experiment. For each query, volunteers evaluate the relevance of the top 50 articles returned by each system. Then they list the top 20 relevant articles based on the relevance, which referred to as the ground truth.

Only considering the articles in the ground truth as relevant results, we calculate the eleven point recall/precison of the three systems, as shown in Table 2 and Figure 1. In table 2, column 2,3,4 compare the article LM to INQUERY, and column 5,6,7 compare the article LM to the traditional document LM. As we can see, the article LM outperforms INQUERY ranking algorithm at all levels of recall, and the article LM achieves better precision than the document LM expect at 0.9 level of recall. In addition, the eleven point average precision of three systems are 0.310, 0.356 and 0.449, and the article LM do 44.84% better than INQUERY and 26.12% better than the document LM.

The minimizing Kendall's τ distance is computed between the ground truth and the top 20 list of each system's search results. As Table 3 presents, the Kendall's τ distance of three systems and the ground truth are 0.461100, 0.446451 and 0.401906, among which the article LM has the shortest distance to the ground truth. It indicates that the order of the result list in article LM is most similar to the ground truth.

Experimental results shows that the article LM outperforms INQUERY ranking algorithm and the traditional LM. Not only the recall/precision of the article

Table 2. Comparing eleven point recall/precison of INQUERY, document LM and article LM

	Precision					
Recall	INQUERY	Article LM	%chg	Document LM	Article LM	%chg
0.0	-	-	-	-	-	-
0.1	0.576	0.740	+28%	0.610	0.740	+21%
0.2	0.473	0.707	+49%	0.555	0.707	+27%
0.3	0.463	0.666	+44%	0.525	0.666	+27%
0.4	0.442	0.613	+39%	0.484	0.613	+27%
0.5	0.376	0.552	+47%	0.419	0.552	+32%
0.6	0.318	0.507	+59%	0.348	0.507	+46%
0.7	0.233	0.394	+69%	0.282	0.394	+40%
0.8	0.180	0.240	+33%	0.218	0.280	+10%
0.9	0.040	0.063	+57%	0.111	0.063	-43%
1.0	0.003	0.003	0	0.003	0.003	0
Avg	0.310	0.449	+44.84%	0.356	0.449	+26.12%

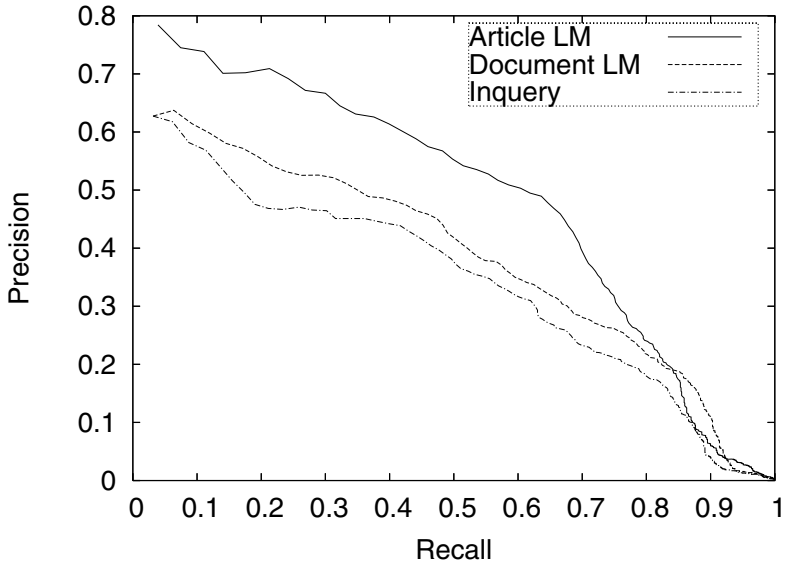

Fig. 1. Comparing eleven point recall/precison of INQUERY, document LM and article LM

LM is better than the others, but also the order of the relevant results in article LM is more close to the ground truth. The success is due to that the article LM makes use of the information from the author and the category. Each user or category has particular interests and different fields of knowledge. Obviously it will be helpful to improve IR performance that exploring differences between users or categories. Consequently, combing document, author and category LMs into the article LM would achieve significant improvement in BBS search.

Table 3. Minimizing Kendall's τ Distance Results

list τ_1	list τ_2	Kendall's τ distance
Ground Truth	INQUERY	0.461100
Ground Truth	Traditional Document LM	0.446451
Ground Truth	Article LM	0.401906

5 Conclusions and Future Work

In this paper, we propose a novel article LM for BBS search to improve the retrieval performance. According to the nature of BBS, we extend the traditional document LM to the author LM and the category LM. The article LM combines document, author, and category LMs into a single framework. Our experimental results show that the article LM outperforms significantly both INUQERY ranking algorithm and the traditional LM. It indicates that the information of the authors and the categories of the BBS articles are helpful to improve the retrieval performance in BBS search. The article LM can be easily applied in other similar systems such as blogs, newsgroups and forums.

For the future work we are planning to explore various smoothing method to improve retrieval performance. Moreover, we will explore to build an blogs search engine with our article LM. Making use of user informaion, personalized search in BBS or blogs is also planed to do.

References

1. Kou, Z., Zhang, C.: Reply networks on a bulletin board system. Physical Review E **67** (2003)
2. Novak, J., Raghavan, P., Tomkins, A.: Anti-aliasing on the web. Proceedings of the 13th international conference on World Wide Web (2004) 30–39
3. Kou, Z., , Bao, T., Zhang, C.: Discovery of relationships between intereswts from bulletin board system by dissimilarity reconstruction. Proceedings Lecture Notes in Artificial Intelligent **2843** (2003) 328–335
4. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Proceedings of the seventh international conference on World Wide Web 7 (1998) 107–117
5. Kraft, R., J.Zien: Authoritative sources in a hyperlinked environment. Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms (1998) 668–677
6. Kraft, R., J.Zien: Mining anchor text for query refinement. Proceedings of the 13th international conference on World Wide Web (2004) 666–674
7. G.Xue, H.Zeng, Chen, Z., Ma, W., Lu, C.: Log mining to improve the performance of site search. Third International Conference on Web Information Systems Engineering (2002)
8. Xi, W., Lind, J., Brill, E.: Learning effective ranking functions for newsgroup search. Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (2004) 394–401
9. Fagin, R., Kumar, R., McCurley, K.: Searching the workplace web. Proceedings of the twelfth international conference on World Wide Web (2003) 366–375

10. Ponte, J., Croft, W.: A language modeling approach to information retrieval. Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (1998) 275–281
11. Si, L., Jin, R., Callan, J., Ogilvie, P.: A language modeling framework for resource selection and results merging. Proceedings of the eleventh international conference on Information and knowledge management (2002) 391–397
12. Song, F., Croft, W.: A general language model for information retrieval. Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (1999) 279–280
13. Lavrenko, V., Croft, W.: Relevance-based language models. Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (2001) 120–127
14. Li, X., Croft, W.: Time-based language models. Proceedings of the twelfth international conference on Information and knowledge management (2003) 469–475
15. R. Jin, A.G. Hauptmann, C.Z.: Title language model for information retrieval. Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (2002) 42–48
16. Fagin, R., Kumar, R., Sivakumar, D.: Comparing top k lists. Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms (2003) 28–36