A Retrieval Method Based on Language Model Considering Neighboring Contents

Koya Tamura[1], Kenji Hatano[2], Hiroshi Yadohisa[2]
[1]UX Department, mixi, Inc., 2-34-17 Jingumae
Sumitomo Fudosan Harajuku Building 17F
Shibuya, Tokyo 150-0001, Japan
koya.tamura@mixi.co.jp
[2]Faculty of Culture and Information Science
Doshisha University, 1-3 Tatara-Miyakodani
Kyotanabe, Kyoto 610-0394, Japan
{khatano, hyadohis}@mail.doshisha.ac.jp

JDIM
Journal of Digital Information Management

ABSTRACT: *The World Wide Web (WWW) has a massive number of Web pages, so that it is difficult for users to get useful information. In recent years, however, it is said that the probabilistic language model can help to improve retrieval accuracies of some kinds of search engines. The probabilistic language model has statistical background and can adapt previous text information retrieval model. However, we cannot directly adapt the probabilistic language model to Web search engines, because data on the Web has a hyperlink environment while the probabilistic language model can only text information on it. In this paper, we propose a novel approach for searching Web pages considering a Web page as well as its neighboring oneson the hyperlink environment.*

Categories and Subject Descriptors
F.1.2 [Modes of Computation]; Probabilistic computation: **H.3.5 [Online Information Services**]; Web-based services: **H.3.3 [Information Search and Retrieval]**

General Terms: Web retrieval, Probabilistic language models, Neighborhood relations

Keywords: Web Search, Language Model, Hyperlink Analysis

Received: 11 September 2011, Revised 17 October 2011, Accepted 23 October 2011

1. Introduction

In today's advanced information society, the World Wide Web (WWW) has a massive number of Web pages. Therefore, Web search engines are necessary to find useful information. Never the less, it is difficult to find useful information exactly, because the data on the Web has been increasing over the last ten years. Thus, improving retrieval accuracy of Web search engines is one of the important tasks in the research field of Web information retrieval.

In order to improve the retrieval accuracy of the Web search engines, many retrieval models had proposed in past researches, e.g. the Boolean model, the vector space model, and so on [8]. Especially in the research field of information retrieval, the probabilistic language model has attracted much attention in recent years. It is mainly used in the machine translation and the speech recognition [1]; however, Ponte and Croft have adopted it for searching documents [9]. This model, which is called the query likelihood model in their research, is defined using term appearance, so that it can be said that it has statistical background and can adapt previous information retrieval model easily. Therefore, it is natural that the query likelihood model can be used for searching valuable Web pages.

Incidentally, the query likelihood model usually handles only text information in each document for searching documents. Thus, we have to extend the query likelihood model to handle text information as well as hyperlink structure among Web pages. The hyperlink structure is also useful information for searching valuable Web pages, because some previous researches have pronounced that the hyperlink structure helps to improve the retrieval accuracies of Web search engines. For example, the PageRank and HITS algorithms, which are well-known Web search techniques [5, 6], utilize the hyperlink structure to evaluate which Web page is valuable or not. However, they consider not Web page contents but the hyperlink structure; therefore, it can be said that it is far from handling text information in Web pages.

To solve these problems of the previous researches, we propose a new retrieval model for accurately searching valuable Web pages accurately in this paper. In order to

handle both text information and the hyperlink structure among the Web pages, we firstly extend the query likelihood model to handle the hyperlink structure, and then, implement it on our Web search engine.We implement two straightforward approaches; one is to add query likelihoods of neighboring Web pages, the other is calculating keyword likelihoods of its neighboring Web pages using the hyperlink structure among Web pages. We also evaluate our approach with conventional ones to confirm effectiveness of our proposal. We believe that our approach has potential to improve retrieval accuracy of Web search engines because it employs the novel retrieval model utilizing both text information and the hyperlink structure among the Web pages.

The remainder of this paper is organized as follows: In Section 2, we describe related work and basic issue of the query likelihood model. In Section 3, we introduce our proposal considering a content of neighboring Web pages. In Section 4, we report our experimental results for evaluating our method. Finally, in Section 5, we conclude our paper and mention directions for future work.

2. Basic Issues and Related Work

In this section, we describe a basic issue of the query likelihood model and some researches related with precisely searching Web documents.

2.1 Query Likelihood Model

The query likelihood model is one of the retrieval models based on calculating query likelihoods which mean suitability between query and documents [9, 13]. In context of the query likelihood model, one document is regarded as a sample from underlying the language model [1]. The language model in the document called the document model; we have to estimate the document model for calculating the query likelihood. In order to calculate it, the unigram is generally used in past researches related with document search. The unigram model assumes that the words independently occur in each document. Thus, we can calculate their suitability as follows:

$$\hat{P}(Q \mid M_{d_i}) = \prod_{t_{ij} \in Q} \hat{P}(t_{ij} \mid M_{d_i}) \qquad (1)$$

Here, $d_i (i = 1, 2, ..., l)$ is a document, and a query Q which consists of a set of query keyword $t_{ij} (j = 1, 2, ..., m)$ is issued by a user. The query keywords are usually contained in several Web pages, so that we denote the word as t_{ij} in document d_i. At this time, m is the number of unique words contained in all Web pages. In Equation (1), $\hat{P}(Q \mid M_{d_i})$ is called a query likelihood of document d_i, so that the documents are ranked in order of their query likelihoods.

In the query likelihood model, we have to estimate probabilities of occurring individual query keywords. They depend on not context of a document but the document model, so that we can calculate the query likelihoods using the maximum likelihood estimate of individual word t_{ij} as follows:

$$P_{mle}(t_{ij} \mid M_{d_i}) = \frac{tf_{t_{ij}}^{d_i}}{N_{d_i}} \qquad (2)$$

where M_{d_i} is the document model which is the language model in document d_i, $tf_{t_{ij}}^{d_i}$ is occurrence of word t_{ij} in document d_i, and N_{d_i} is the length of document d_i .

However, the query likelihood model has a problem called "zero-probability problem." When a word does not appear in a document, probability of occurring word would be zero. As a result, the query likelihood of the document would also be zero even if many query keywords exist in a query. To cope with this problem, the probability of the word would not be zero using some smoothing techniques to combine them in a document. The most famous smoothing technique, which is called Jelinek-Mercer [4], is defined as following equation:

$$\hat{P}\left(t_{ij} \mid M_{d_i}\right) = \omega P_{mle}\left(t_{ij} \mid M_{d_i}\right) + (1-\omega) P_{mle}\left(t_{ij} \mid M_c\right) \qquad (3)$$

where ω is a weighting parameter $0 \leq \omega \leq 1$, and M_c is the corpus model which is based on probabilities of occurring words in all documents. Using the corpus model, we can avoid the zero-probability problem even if $P_{mle}(t_{ij} \mid M_{d_i})$.The query likelihoods based on the corpus model M_c can also be defined as the following equation:

$$P_{mle}\left(t_{ij} \mid M_c\right) = \frac{\sum_{d_i \in c} tf_{t_{ij}}^{d_i}}{\sum_{d_i \in c} N_{d_i}} \qquad (4)$$

In the later of this paper, we regard this query likelihood model as baseline method.

2.2 Related Work

In order to develop an effective Web search engine, the first thing we have to consider is handling the contents of Web pages precisely. Analyzing the hyperlink structure on the Web is also well known approach, and calculates existing importance of each Web page based on graph-theoretical analysis. [5, 6] reported that they were able to search Web pages more accurately than the past approaches based on the traditional retrieval models.

However, Sugiyama et al. had pointed out the following two problems to the approaches:

- The importance of a Web page is simply defined. In short, contents of the Web page are not considered.
- The relatively of content between hyperlinked Web pages are not taken into consideration for calculating existing importance of each Web page.

To cope with these problems, we should treat contents of not only a Web page but also its neighboring ones as its

contents. In fact, some researches related with developing Web search engines have achieved a satisfactory level of performance in their research fields [10, 14]. These researchers consider the neighboring Web pages based on the TF-IDF term weighting scheme [11] on the vector space model [12]. In recent years, however, it is reported that the query likelihood model described in Section 2.1 can understand the contents of Web pages more accurately than the vector space model.

We think that the retrieval model considering the contents of Web pages as well as neighboring ones based on the query likelihood model should be proposed; however, there has not been any novel approach yet. As far as we know, one extension of the query likelihood model, which is called the cluster language model [7], has proposed and is similar to the query likelihood model in many respects. This method divides all documents into K clusters, and likelihoods of each query keyword related with a cluster and corpus are used for estimating the query likelihood of a document. However, Tao et al. had pointed out that using a document cluster is not suitable for the query likelihood model [17]. A cluster of documents may contain row similarity documents possibly; therefore, they had proposed the document expansion model to extract similar documents using neighboring ones for calculating query likelihood. In these studies, a clustering algorithm is used to obtain similar documents; however, it took a lot of time to get K cluster.

3. Our proposal

As we described in Section 2.2, we assume that considering neighboring Web pages is useful for searching Web pages accurately related with user's information needs. In this section, therefore, we propose two novel approaches for considering contents of neighboring Web pages. Our approach is to reflect the contents of the Web page in addition to its neighboring ones to its query likelihood. Hereinafter, we represent a Web page with query likelihood $P(Q \mid M_{d_i})$ as target page d_i (i =1, 2, ..., l).

Among Web pages on the Web, there are some hyperlinks; that is, we believe that two Web pages with a hyperlink have some kind of relativity. Therefore, we can assume that a target Web page is satisfied with a query Q if its neighboring ones have high query likelihoods. This is because the neighboring Web pages have contents suitable for the query , so that the target Web page also has the same contents related with the query As a result, we can consider contents of the Web page accurately. At this time, the first problem is how to reflect the query likelihoods of neighboring Web pages to target one. In the following sections, we explain each method in more details.

3.1 Method ST

One of our approaches is to recalculate a query likelihood of a Web page using its neighboring ones [15]. In order to recalculate the query likelihood of each Web page, the following two steps are processed in our method.

1. Calculate query likelihoods of all Web pages based on the query likelihood model.

2. Recalculate them using the query likelihoods of neighboring pages.

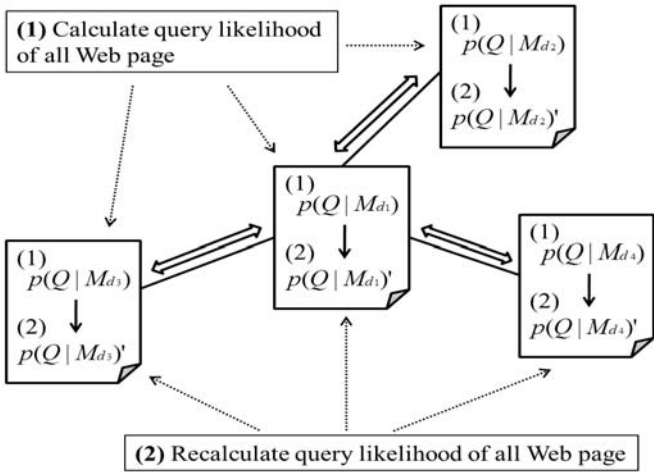

Figure 1. Outline of Method ST

Figure 1 is schematized above two steps. Here, we explain the processes in Step 2 because this step is our contribution of to the research.

Basic idea of Step 2 is that a Web page whose neighboring ones are suitable for user's information need should be searched by Web search engines even if its query likelihood is small. In an opposite manner, a Web page whose neighboring ones are not related with user's query should not be returned as a search result even if its query likelihood is large. This is because a content related with user's information need tends to be divided into some Web pages; as a result, a Web page as well as its neighboring ones is important for developing Web search engines. In short, we have to control the ranking of Web pages using the query likelihoods of their neighboring ones.

Here, we have to consider how to reflect query likelihoods of the neighboring Web pages to calculate that of the target one. In the research field of information retrieval, summation and multiplication are straightforward techniques often used to combine several factors. Particularly in previous work [11], product is good to combine some factors; therefore, we recalculate the query likelihood of a target Web pages to multiply by those of neighboring ones.

In this approach, we recalculate a query likelihood of Web page emphasizing its original query likelihood. This is because a query likelihood of target Web page will be small if those of its neighboring ones are small. For example, we assume that Web page d_i in Figure 2 has neighboring pages with small query likelihoods. In this situation, query likelihood of Web page d_i would significantly be small if we emphasize query likelihoods of neighboring pages. As a result, the query likelihood of Web page d_i is recalculated as low value even if Web page d_i is relevant Webpage.

To solve this problem, we recalculate the query likelihood of a Web page emphasizing its original one. The advantage

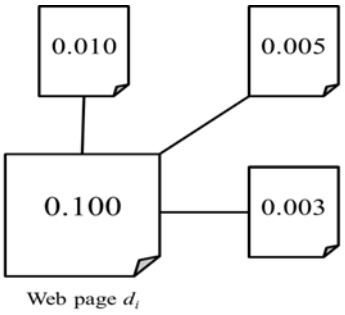

Figure 2. Query Likelihoods of Target Web Page and its Neighboring Pages

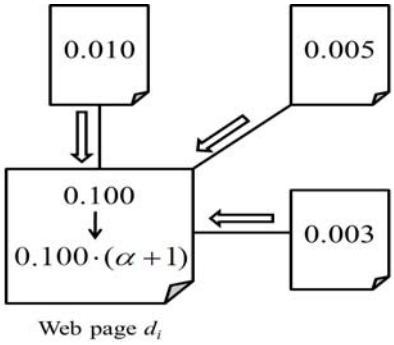

Figure 3. Emphasizing Query Likelihood of Target Web Page

of this approach is that we can reflect query likelihoods of neighboring pages to that of the target one retaining the original query likelihood of each Web page, if desired. Therefore, a recalculated query likelihood of Web page is unaffected by those of neighboring ones.We call our method the summation of query likelihood of neighboring Web pages (method ST, for short), and a query likelihood of a Web page can be calculated as follows:

$$P_{ST}\left(Q \mid M_{d_i}\right) = P\left(Q \mid M_{d_i}\right).(1+\sum_{d_i^k \in L_{d_i}} P(Q \mid M_{d_i^k})) \quad (5)$$

For instance, query likelihood of target Web page d_i is 0.1 and those of its neighboring ones are 0.01, 0.005, and 0.003, respectively in Figure 3. Moreover, α is recalculating factor that is summation of neighboring pages' query likelihood. As a result, recalculated query likelihood of target Web page *da* is

$P_{ST}(Q \mid M_{d_i}) = 0.1 \,.\, (1+(0.01+0.05+0.003)) = 0.1018$

3.2 Link-Based Language Model

As we described in previous section, method ST is to reflect the query likelihood of the Web page in addition to its neighboring ones to its query likelihood. Although, the query that the user submits is made up of many query keywords, we can calculate probabilities of each query keyword. Figure 4 shows how to calculate the query likelihood of Web page d_1 using its neighboring pages d_2 and d_3. Usually, $P(Q \mid M_{d_1})$ is calculated from $P(Q \mid M_{d_2})$ and $P(Q \mid M_{d_3})$ which are query likelihoods of d_2 and d_3. These query likelihoods are calculated based on issuing a query; however, we have to handle the likelihoods of query keywords in order to calculate $P(Q \mid M_{d_1})$ precisely. In short, $P(Q \mid M_{d_1})$ should be calculated based on $P(k_1 \mid M_{d_2})$, $P(k_2 \mid M_{d_2})$, $P(k_1 \mid M_{d_3})$, and $P(k_2 \mid M_{d_3})$ if the query is made up of k_1 and k_2 [15].

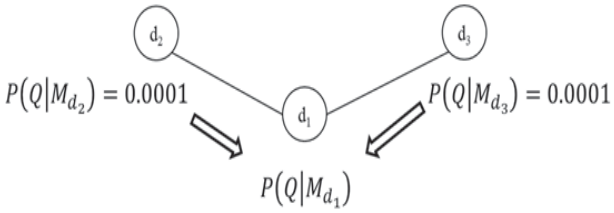

Figure 4. Considering whole query likelihood of neighboring pages

Figure 5 shows an example of this solution. In this situation, query Q is comprised of three query keywords 'obama', '*family*', and '*tree*'; that is, the user wants information related with the family tree of President Obama. In order to consider the contents of the neighboring pages, we should reflect the query likelihoods of neighboring pages to that of Web page. Hereinafter, we call likelihoods of each query keyword as keyword likelihood. In Figure 5, we calculate likelihoods of each query keyword of a Web page d_2 as $P(\text{'}obama\text{'} \mid M_{d_2}) = 0.01$, $P(\text{'}family\text{'} \mid M_{d_2}) = 0.1$, and $P(\text{'}tree\text{'} \mid M_{d_2}) = 0.1$ and query likelihood as $P(Q \mid M_{d_2}) = 0.00\,01$ by Equation (1). Hence, *'family'* and *'tree'* have high probabilities of occurring in a, so that has contents related with *'family'* and *'tree'*. In contrast, we calculate keyword likelihood of a Web page d_3 as $P(\text{'}obama\text{'} \mid M_{d_3}) = 0.1$, $P(\text{'}family\text{'} \mid M_{d_3}) = 0.01$ and $P(\text{'}tree\text{'} \mid M_{d_3}) = 0.1$, and query likelihoods as $P(Q \mid M_{d_3}) = 0.0001$. Consequently, d_3 has information about '*obama*' and '*tree*'.

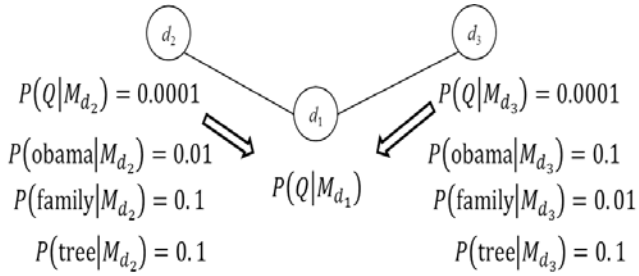

Figure 5. Existing query keyword likelihood in each neighboring page

The problem here that both the query likelihoods $P(Q \mid M_{d_2})$ and $P(Q|M_{d_2})$ are the same values. Consequently, if we apply method ST, we regard different kinds of Web pages as the same ones. For this reason, as shown Figure 5, we must reflect the keyword likelihood of neighboring pages to that of d_1.

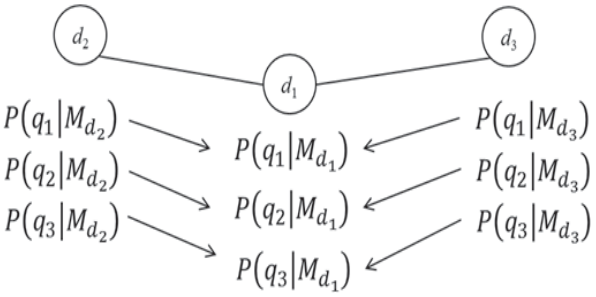

Figure 6. Considering each query keyword likelihood of neighboring pages

As mentioned in the above concept, we propose a Link-Based Language Model (LBLM) which considers the keyword likelihoods of neighboring pages connected by hyperlinks. In this proposal, we calculate the keyword likelihood in a set of neighboring pages called the link model. Moreover, we combine the document model, the corpus model described in Section 2.1, and the link model to calculate keyword likelihood.

In Figure 6 is schematized above mentioned processes. In order to feature the combination of three models, we proposed two methods. One of the methods is to combine the query likelihood model (Equation (3)) and the link model as follows;

$$P(t_{ij} \mid M_{d_i}) = \alpha\{\beta P(t_{ij} \mid M_{d_i}) + (1-\beta)P(t_{ij} \mid M_c)\} + (1-\alpha)P(t_{ij} \mid M_{L_{d_i}}) \quad (6)$$

Here α and β are weighting parameters where $0 \le \alpha \le 1.0$, $0 \le \beta \le 1.0$ Therefore, there are 121 combinations of parameters.

On the other hands, we combine three models independently as follows:

$$P\left(t_{ij} \mid M_{d_i}\right) = \lambda_1 P\left(t_{ij} \mid M_{d_i}\right) + \lambda_2 P\left(t_{ij} \mid M_{L_{d_i}}\right) + \lambda_3 P\left(t_{ij} \mid M_c\right) \quad (7)$$

Here, $\lambda_1, \lambda_2, \lambda_3$ are weighting parameters where $\lambda_1+\lambda_2+\lambda_3=1$, $0 \le \lambda_1, \lambda_2, \lambda_3 \le 1.0$. Therefore, there are 66 combinations of parameters. Moreover, $M_{L_{d_i}}$ is the link model. $P(t_{ij} \mid M_{L_{d_i}})$ is calculated by the likelihood of term t_{ij} under the Link Model M_L as follows:

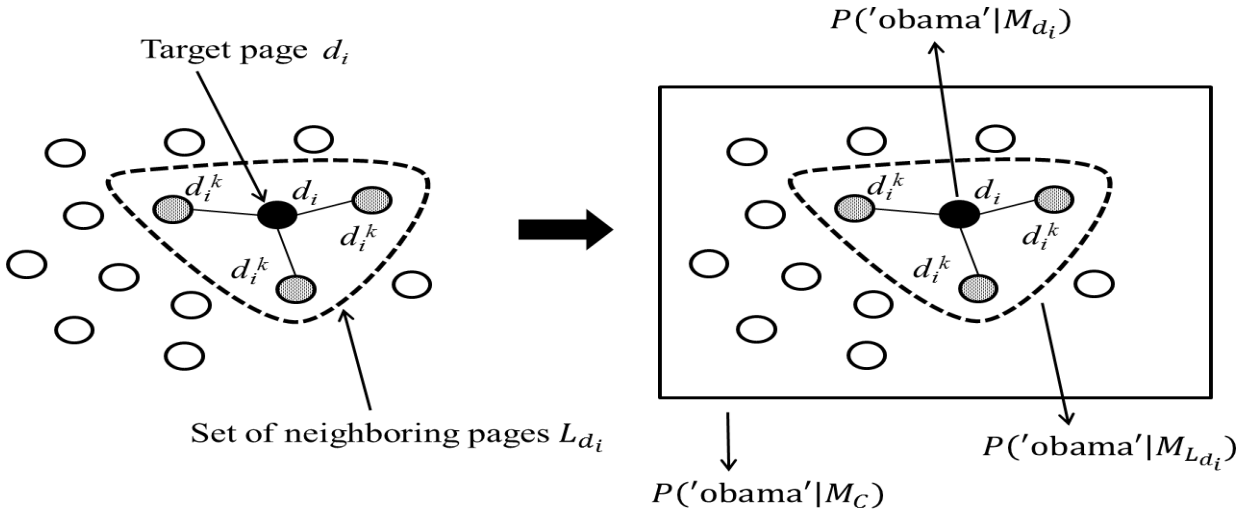

Figure 7. Link-Based Language Model

$$P(t_{ij} \mid M_{L_{d_i}}) = \frac{\sum_{d_i^k \in L_{d_i}} tf_{t_{ij}}^{d_i^k}}{\sum_{d_i^k \in L_{d_i}} N_{d_i^k}} \quad (8)$$

4. Experiments

To evaluate the effectiveness of our proposal, we used a TREC test collection. In preliminary experiments in Section 4.2 and 4.3, we have to compare the retrieval accuracies of our methods using different the hyperlink types and combination of parameter settings for LBLM. We can also compare our best method with a baseline method not considering the query likelihoods of neighboring Web pages mentioned in Section 2.

4.1 Test Collection

Our experiment uses ClueWeb09 Dataset Category B [2] provided by TREC[1](Text REtrival Conference). TREC is a workshop focusing on information retrieval (IR) research areas, and co-sponsored by US Department of Defense and the National Institute of Standards and Technology (NIST). This dataset consists of 50 million English Web pages (Unique URLs: 428,136,613, Total out-links: 454,075,638) collected in 2009, 50 topics, and their sets of answers. Eventually, to make our rank list of documents in the top 1,000, we eliminate stop words from all documents using Salton's stop word list[2] and do a stemming processing based on the Porter Stemmer[3].

To evaluate the effectiveness of our proposal, we use precision (Prec.) and the number of retrieved relevant documents (Rel. Retr.). We calculate the precision as follows:

$$\text{Precision} = \frac{\text{the number of retrieved relevant web pages}}{\text{the number of retrieved web pages}} \quad (9)$$

In particularly about precision, we also use top 10 (P@10), at the 11 point of the number of retrieved document (0.0-1.0), and mean average precision(MAP).

[1]http://trec.nist.gov/
[2]ftp://ftp.cs.cornell.edu/pub/smart/english.stop
[3]http://www.tartrus.org/%7Emartin/PorterStemmer/

4.2 Considering Different Types of Hyperlink

There are two types of hyperlinks in Web pages, in-link and out-link. Out-link which is put in place by the author goes to other Web pages. In contrast, in-link which is created independently of the author comes from the other Web pages. In this situation, we must decide which type of hyperlink we choose to achieve the best result in our experiments. Thus, we must compare the results of LBLM retrieved by using each of the following: out-link only, in-link only, and the combination of both of them. Table 1 shows MAP and Rel.Retr. in each experiment. These results are averaged by 66 combinations of parameters.

Averages of MAP and Rel.Retr. using out-link only is the best in three experiments, and their standard deviation (SD) using out-link only are also the smallest. From these reason, we believe choosing out-link only is stable in our experiment.

4.3 Parameter Setting

As we described in Section 4.2, we obtain a set of neighboring pages using the out-link only to perform our preliminary experiment. LBLM requires the settings of α, β or $\lambda_1 - \lambda_2$ in Equation (6), (7). Therefore, in this section, we discuss how to set all of them.

α and β are related to the query likelihood model and the link model, respectively, and fulfill the conditions $0 \leq \alpha \leq 1.0$ and $0 \leq \beta \leq 1.0$; therefore, there are 121 combinations of parameters. In contrast, λ_1 ,λ_2 and λ_3 are related to likelihoods in the language model , the link model $P(t_{ij} \mid M_{L_{d_i}})$, and the corpus model , respectively, and fulfill the conditions $\lambda_1 + \lambda_2 + \lambda_3 = 1$ and $0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 1.0$ therefore, there are 66 combinations of parameters. Eventually, we can calculate 121 pairs and 66 pairs of MAP and Rel.Retr. shown in Table 2 and 3.

As shown in Table 2 and 3, $(\lambda_1, \lambda_2, \lambda_3) = (0.4, 0.1, 0.5)$ is the best result in MAP and Rel.Retl.. Moreover, when we set $\lambda_2 \leq 0.5$, retrieval accuracy is relatively-good. Hence, it is said that we can improve retrieval accuracy to small parameter related with the link model.

In-link		
	MAP	Rel.Retr.
Ave.	0.2174	4,727
SD	0.03404	754.8
Min.	0.154	3,213
Max.	0.2593	5,557

Out-link		
	MAP	Rel.Retr.
Ave.	0.2518	5,469
SD	0.02054	152.6
Min.	0.1825	4,980
Max.	0.2638	5,580

In-link and Out-link		
	MAP	Rel.Retr.
Ave.	0.2576	4,735
SD	0.03399	751.9
Min.	0.1541	3,228
Max.	0.2594	5,562

Table 1. Results with hyperlinks type in LBLM

4.4 Experimental Results

As we described in Section 4.3, we set $(\lambda_1, \lambda_2, \lambda_3) = (0.4, 0.1, 0.5)$ as parameters in Equation (7). Using these parameters, we can get the results shown in Table 2. These results are given by above mentioned criteria. In order to compare our approach (ST and LBLM) with a baseline (BL), we conduct another experiment using the hypothesis testing. In this experiment, we use *wilcoxon* signed rank test [3] another experiment for testing the difference. "*" in Table 4 represents significance difference. Experimental results indicate as follows;

• In comparison of BL and ST, MAP and Rel.Retr. of ST improve by 2.60% and 4.21%. Especially, we can confirm the significant difference of Rel.Retr..

• In comparison of BL and LBLM, MAP and Rel.Retr. of LBLM improve by 3.72% and 5.78% respectively compared with those of BL.

• In comparison of ST and LBLM, Map and Rel.Retr. of LBLM also improve by 1.10% and 1.51% respectively compared with those of ST. Especially, we can confirm the significant difference of both MAP and Rel.Retr.

As we described above, ST and LBLM is more suitable for searching Web pages based on the language model. In other words, considering contents of neighboring Web pages helps to improve the retrieval accuracies of our Web search engine. Moreover, above results are represented effectiveness of LBLM to ST, we believe that considering a feature of query keyword likelihood in neighboring Web pages is more effective than considering whole query likelihood.

4.5 Discussion of Time Complexity

When we search the Web page using Web search engine, amount of time required to obtain a search result is considerable factor. In this section, we discuss about time complexity ties of the cluster based language model (CBLM) [7] and LBLM. Especially, we focus on process of obtaining neighboring pages. The reason of comparing their time complexity is that CBLM is the most relate to LBLM and applying clustering to whole Web pages is visionary.

To extract a set of neighboring pages in CBLM, K-means

α, β	MAP	Rel.Retr.
0.0, 0.0	0.02455	178
0.0, 0.1	0.02480	183
0.0, 0.2	0.02504	183
0.0, 0.3	0.02519	184
0.0, 0.4	0.02543	184
0.0, 0.5	0.02641	184
0.0, 0.6	0.02547	184
0.0, 0.7	0.02484	183
0.0, 0.8	0.02370	182
0.0, 0.9	0.02302	181
0.0, 1.0	0.00297	119
0.1, 0.0	0.0246	178
0.1, 0.1	0.04144	1,474
0.1, 0.2	0.05976	1,810
0.1, 0.3	0.07523	2,249
0.1, 0.4	0.09113	2,785
0.1, 0.5	0.1104	3,404
0.1, 0.6	0.1344	4,004
0.1, 0.7	0.1698	4,509
0.1, 0.8	0.2073	4,813
0.1, 0.9	0.2411	4,994
0.1, 1.0	0.2466	5,086
0.2, 0.0	0.02455	178
0.2, 0.1	0.05781	1,772
0.2, 0.2	0.08084	2,454
0.2, 0.3	0.1030	3,211
0.2, 0.4	0.1287	3,911
0.2, 0.5	0.1606	4,475
0.2, 0.6	0.1918	4,843
0.2, 0.7	0.2268	5,076
0.2, 0.8	0.2463	5,171
0.2, 0.9	0.2533	5,240
0.2, 1.0	0.2540	5,264
0.3, 0.0	0.02455	178
0.3, 0.1	0.06811	2,046
0.3, 0.2	0.09738	2,996
0.3, 0.3	0.1272	3,880
0.3, 0.4	0.1614	4,500
0.3, 0.5	0.1935	4,885
0.3, 0.6	0.2252	5,093
0.3, 0.7	0.2441	5,212
0.3, 0.8	0.2548	5,281
0.3, 0.9	0.2571	5,323
0.3, 1.0	0.2578	5,350
0.4, 0.0	0.02455	178
0.4, 0.1	0.07767	2,308
0.4, 0.2	0.1118	3,503
0.4, 0.3	0.1500	4,300
0.4, 0.4	0.1851	4,815
0.4, 0.5	0.2166	5,069
0.4, 0.6	0.2415	5,210
0.4, 0.7	0.2533	5,305
0.4, 0.8	0.2576	5,348
0.4, 0.9	0.2589	5,388
0.4, 1.0	0.2597	5,423
0.5, 0.0	0.02455	178
0.5, 0.1	0.08513	2,553
0.5, 0.2	0.1259	3,875
0.5, 0.3	0.1678	4,615
0.5, 0.4	0.2033	4,974
0.5, 0.5	0.2350	5,174
0.5, 0.6	0.2496	5,275
0.5, 0.7	0.2569	5,338
0.5, 0.8	0.2596	5,394
0.5, 0.9	0.2603	5,434
0.5, 1.0	0.2607	5,454
0.6, 0.0	0.02455	178
0.6, 0.1	0.09275	2,795
0.6, 0.2	0.1390	4,127
0.6, 0.3	0.1834	4,777
0.6, 0.4	0.2180	5,088
0.6, 0.5	0.2428	5,244
0.6, 0.6	0.2543	5,328
0.6, 0.7	0.2590	5,390
0.6, 0.8	0.2604	5,422
0.6, 0.9	0.2610	5,456
0.6, 1.0	0.2614	5,477
0.7, 0.0	0.02455	178
0.7, 0.1	0.09942	3,031
0.7, 0.2	0.1528	4,360
0.7, 0.3	0.1953	4,895
0.7, 0.4	0.2317	5,165
0.7, 0.5	0.2488	5,284
0.7, 0.6	0.2577	5,368
0.7, 0.7	0.2600	5,406
0.7, 0.8	0.2609	5,438
0.7, 0.9	0.2615	5,470
0.7, 1.0	0.2616	5,480
0.8, 0.0	0.02455	178
0.8, 0.1	0.1065	3,265
0.8, 0.2	0.1636	4,559
0.8, 0.3	0.2063	4,993
0.8, 0.4	0.2391	5,203
0.8, 0.5	0.2536	5,312
0.8, 0.6	0.2592	5,381
0.8, 0.7	0.2609	5,424
0.8, 0.8	0.2617	5,463
0.8, 0.9	0.2619	5,474
0.8, 1.0	0.2622	5,489
0.9, 0.0	0.02455	178
0.9, 0.1	0.1126	3,437
0.9, 0.2	0.1723	4,641
0.9, 0.3	0.2150	5,064
0.9, 0.4	0.2431	5,224
0.9, 0.5	0.2551	5,332
0.9, 0.6	0.2591	5,378
0.9, 0.7	0.2607	5,425
0.9, 0.8	0.2608	5,444
0.9, 0.9	0.2613	5,461
0.9, 1.0	0.2615	5,465
1.0, 0.0	0.02455	178
1.0, 0.1	0.1197	3,592
1.0, 0.2	0.1802	4,690
1.0, 0.3	0.2233	5,031
1.0, 0.4	0.2467	5,158
1.0, 0.5	0.2548	5,237
1.0, 0.6	0.2582	5,281
1.0, 0.7	0.2592	5,321
1.0, 0.8	0.2595	5,338
1.0, 0.9	0.2598	5,342
1.0, 1.0	0.2600	5,353

Table 2. Result about each combination of two parameters

clustering algorithm is used. K-means clustering algorithm requires calculating similarities of all pair of Web pages. Generally, time complexity of CBLM is $o(n^2)$ is expressed where n is the number of Web pages. In contrast, LBLM can get neighboring pages collected by the hyperlinks in each document. In the test collection which we used, an Web page includes one hyperlink on an average. If time complexity of calculating similarity of pair of document in K-means clustering is the same as collecting the hyperlinks in a Web pages for LBLM, time complexity of LBLM is $o(n^2)$. Consequently, LBLM is more efficient than CBLM from the standpoint of time complexity.

5. Conclusions

In this paper, we have proposed two novel approaches that considering contents of neighboring Web pages. Experimental results showed that LBLM could improve

$\lambda_1,\lambda_2,\lambda_3$	MAP	Rel.Retl
0.0,0.0,1.0	0.002955	119
0.0,0.1,0.9	0.01751	192
0.0,0.2,0.8	0.01817	194
0.0,0.3,0.7	0.01815	192
0.0,0.4,0.6	0.01823	192
0.0,0.5,0.5	0.01812	192
0.0,0.6,0.4	0.01789	192
0.0,0.7,0.3	0.01772	192
0.0,0.8,0.2	0.01798	193
0.0,0.9,0.1	0.01795	193
0.0,1.0,0.0	0.01753	187
0.1,0.0,0.9	0.2418	4,999
0.1,0.1,0.8	0.2587	5,393
0.1,0.2,0.7	0.2502	5,376
0.1,0.3,0.6	0.2435	5,346
0.1,0.4,0.5	0.2341	5,292
0.1,0.5,0.4	0.2182	5,212
0.1,0.6,0.3	0.2041	5,142
0.1,0.7,0.2	0.1931	5,067
0.1,0.8,0.1	0.1825	4,980
0.1,0.9,0.0	0.1739	4,785
0.2,0.0,0.8	0.2314	5,021
0.2,0.1,0.7	0.2627	5,526
0.2,0.2,0.6	0.2622	5,525
0.2,0.3,0.5	0.2594	5,514
0.2,0.4,0.4	0.2534	5,512
0.2,0.5,0.3	0.2506	5,494
0.2,0.6,0.2	0.2465	5,446
0.2,0.7,0.1	0.2423	5,400
0.2,0.8,0.0	0.2360	5,217
0.3,0.0,0.7	0.2530	5,264
0.3,0.1,0.6	0.2633	5,561
0.3,0.2,0.5	0.2635	5,563

$\lambda_1,\lambda_2,\lambda_3$	MAP	Rel.Retr.
0.3,0.3,0.4	0.2630	5,559
0.3,0.4,0.3	0.2615	5,537
0.3,0.5,0.2	0.2574	5,533
0.3,0.6,0.1	0.2543	5,508
0.3,0.7,0.0	0.2491	5,334
0.4,0.0,0.6	0.2432	5,050
0.4,0.1,0.5	0.2636	5,580
0.4,0.2,0.4	0.2634	5,573
0.4,0.3,0.3	0.2636	5,565
0.4,0.4,0.2	0.2634	5,547
0.4,0.5,0.1	0.2629	5,540
0.4,0.6,0.0	0.2629	5,540
0.5,0.0,0.5	0.2410	5,003
0.5,0.1,0.4	0.2628	5,571
0.5,0.2,0.3	0.2631	5,572
0.5,0.3,0.2	0.2630	5,563
0.5,0.4,0.1	0.2622	5,544
0.5,0.5,0.0	0.2592	5,381
0.6,0.0,0.4	0.2401	5,012
0.6,0.1,0.3	0.2627	5,565
0.6,0.2,0.2	0.2630	5,569
0.6,0.3,0.1	0.2622	5,548
0.6,0.4,0.0	0.2598	5,395
0.7,0.0,0.3	0.2467	5,099
0.7,0.1,0.2	0.2618	5,571
0.7,0.2,0.1	0.2610	5,544
0.7,0.3,0.0	0.2589	5,399
0.8,0.0,0.2	0.2541	5,275
0.8,0.1,0.1	0.2600	5,559
0.8,0.2,0.0	0.2580	5,415
0.9,0.0,0.1	0.2458	5,068
0.9,0.1,0.0	0.2548	5,406
1.0,0.0,0.0	0.2430	5,123

Table 3. Result about each Combination of three Parameters

		BL	ST	Chg(%)		LBLM	Chg(%)-BL		Chg(%)-ST	
Rel.		12,544	12,544			12,544				
Rel.Retr		5,275	5,497	4.21%	*	5,580	5.78%		1.51%	*
Prec.										
P@10		0.7896	0.7646	-3.17%	*	0.7979	1.06%	*	4.36%	
	0	0.9153	0.8932	-2.41%		0.9143	-0.11%		2.36%	
	0.1	0.3846	0.4046	5.20%	*	0.399	3.74%	*	-1.39%	
	0.2	0.2703	0.2833	4.82%	*	0.2838	4.97%	*	0.15%	*
	0.3	0.2244	0.2357	5.04%	*	0.2383	6.22%	*	1.12%	
	0.4	0.1917	0.2034	6.11%	*	0.2056	7.26%		1.08%	*
	0.5	0.1691	0.1786	5.62%	*	0.1809	6.97%		1.28%	*
	0.6	0.1513	0.1598	5.62%	*	0.1616	6.77%		1.09%	*
	0.7	0.1376	0.1442	4.76%	*	0.1455	5.75%	*	0.95%	*
	0.8	0.1265	0.1313	3.81%	*	0.1331	5.21%	*	1.35%	
	0.9	0.1163	0.1209	3.96%	*	0.1228	5.57%		1.55%	
	1	0.1082	0.1128	4.27%	*	0.1146	5.87%		1.53%	*
MAP.		0.2541	0.2607	2.60%		0.2636	3.72%	*	1.10%	

Table 4. Result about BaseLine, ST, and LBLM

retrieval accuracies more significantly than the query likelihood model.

For the future work, we have to propose a method for extracting effective hyperlinks. Our proposal is simple to extract hyperlinks, discriminating the in-link and the out-link. However, previous researches [16, 14] consider the similarity between two documents and calculate weight of link based on their similarity. Therefore we should obtain a set of neighboring Web pages using both effective the hyperlinks and not effective them, that is, we should choose effective them to obtain a set of neighboring Web pages. In order to extract effective the hyperlinks, we consider a similarity of pair of Web pages connected by hyperlink.

References

[1]Charniak, E.(1996). Statistical Language Learning. The MIT Press.

[2] Clarke, C.L.A., Craswell, N., Soboroff, I.: Overview of the TREC 2009 WebTrack, *In*: The 18thText Retrieval Conference(TREC 2009) Proceedings, NIST.

[3] Hollander, M., Wolfe, D.A.(1999). Nonparametric Statistical Methods. Wiley-Interscience.

[4] Jelinek, F., Mercer, R.L. (1980). Interpolated Estimation of Markov Source Parameters from Sparse Data. *In*: Proceeding of the Workshop on Pattern Recognition in Practice, p. 381-397.

[5] Kleinberg, J.M.(1999). Authoritative Sources in a Hyperlinked Environment. *In:* J*ournal of the ACM (JACM)*, 46 (5) 604-632. ACM.

[6] Page, L., Brin, S., Motwani, R., Winograd, T. (1999). The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab. URL http://ilpubs.stanford.edu:8090/422/

[7] Liu, X., Croft, W.B. (2004). Cluster-based Retrieval using Language Models, *In*: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004), p. 186-193.ACM.

[8] Manning, C.D., Raghavan, P., Schutze, H.(2008). Introduction to Information Retrieval. CambridgeUniversity Press.

[9] Ponte, J.M., Croft, W.B.(1998). A Language Modeling Approach to Information Retrieval. *In*: Proceedings of the 21stAnnual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98), p. 275-281. ACM.

[10] Qi, X., Davison, B.D.(2008). Classifiers without Borders: Incorporating Fielded Text from Neighboring Web Pages. *In*: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008), p. 643-650. ACM.

[11] Salton, G., Buckley, C.(1988).Term-Weighting Approaches in Automatic Text Retrieval. Information Processing and Management, 24 (5) p.513-523. Elsevier.

[12] Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. *Communications of the ACM (CACM)*,18 (11) p.613-620. ACM.

[13] Song, F., Croft, W.B.(1999). A General Language Model for Information Retrieval, *In*: Proceedings of the 8th International Conference on Information and Knowledge Management (CIKM'99), p. 316-321. ACM.

[14] Sugiyama, K., Hatano, K., Yoshikawa, M., Uemura, S.(2005). Improvement in Tf-IdfScheme for Web Pages based on the Contents of theirHyperlinkedNeighboring Pages. Systems and Computers in Japan, 36 (14) 56-68. Wiley-Interscience.

[15] Tamura, K., Hatano, K., Yadohisa, H.(2010). Characterizing Web Pages based on the Query Likelihoods of Neighboring Pages, *In*: Proceedings of the 5thInternational Conference on Digital Information Management (ICDIM 2010), p. 392-397. IEEE (2010).

[16] Tamura, K., Hatano, K., Yadohisa, H.(2011). Calculating Query Likelihoods based on Web Data Analysis. *In*: Intelligent Decision Technologies, Smart Innovation, Systems and Technologies Book Series 10, p. 707-718. Springer.

[17] Tao, T., Wang, X.,Mei, Q., Zhai, C.(2006). Language Model Information Retrieval with Document Expansion. *In:* Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL 2006), p. 407-414. ACL (2006).