NTCIR-4 WEB Experiments at Osaka Kyoiku University

- Static/Dynamic Scoring Using Link Structure Analysis and Web Page Grouping -

Hitoshi Nakakubo Peng Zhang Takashi Sato
Osaka Kyoiku University
4-698-1 Asahigaoka. Kashiwara, Osaka 582-8582 Japan
{nakaku, zp}@ss.osaka-kyoiku.ac.jp sato@cc.osaka-kyoiku.ac.jp

Abstract

We did gram-based indexing and the retrieval with NTCIR-4 WEB task. The time required to make indices are 34.7 hours. The size of indices is 30.2Gbyte. The median of retrieval time par word is 26msec. The ranking algorithm of retrieval results is based on a traditional probabilistic model. We report on the result of gram-based indexing and the retrieval, and propose a scoring method based on link structure analysis.

Keywords: *NTCIR, Web Retrieval, Link Structure Analysis, Gram-based Index*

1. Introduction

Nowadays, it is the most general method to use search engines to retrieve aimed WEB pages. WEB pages including key words are expected to be extracted by inputting them to search engines when we retrieve WEB pages. This can be achieved by using full-text retrieval systems.

We report on construction of a full-text retrieval system using gram-based indices, and propose scoring method based on link structure analysis in this paper. We describe gram-based indexing and retrieval in sections 2 through 5. And, we propose scoring method based on link structure analysis in sections 6 through 8, and describe conclusions in the last section.

2. Gram-based index

For full-text retrieval systems, index lists are effective and systems based on gram are effective also. We can retrieve all compound word including key words by using gram-based indices. No morphological analyses are necessary for gram-based indexing. Then, gram-based indices are suitable for index technique of WEB contents in Japanese, which include 2 bytes character code.

The size of corpus used with NTCIR-4 WEB task is 100Gbyte. Since this is very huge, we cannot store whole data in main memory of a general computer. Therefore, first we made batch indices from subset of corpus, which fit in main memory, then we merge them.

We used the computer with the following specifications to experiment.

✓ OS:	FreeBSD
✓ CPU:	Pentium4 2.4GHz
✓ MEM:	1Gbyte

3. Indexing

We computed gram value of each strings in data files, which are converted from 'cooked' data of NTCIR-4 WEB task by normalizing katakana notation[1]. We made an index as an inverted file of grams, and sorted them in order of their value.

Because the index is huge, we cannot put it in main memory. Then, we put sparse index in main memory by dividing it into five. Other part was stored in external storage area.

Table 1 shows the size of corpus, 'cooked' data, 'converted' data, and the index. Table 2 shows the time required data conversion and indexing. Part # in the table shows the number of the divided indices.

As a result of this processing, 4.4 of 2 bytes characters (grams) in average are coded in 6 bytes.

Table 1. Size of corpus, cooked data, converted data and index

	data size [Gbyte]					
corpus	100.0					
cooked	44.0					
converted	37.8					
part#	0	1	2	3	4	total
index	5.9	6.6	6.2	5.9	5.6	30.2

Table 2. Time to make index

	time [hour]					
convert	20.0					
part#	0	1	2	3	4	total
index	3.0	3.2	3.1	2.8	2.6	14.7

4. Query making

We made the retrieval key words from the topics, which has been offered from NTCIR-4 WEB. Compound key words are segmented in words, and then possible combinations of words are made of a compound word.

We extracted 678 retrieval key words from 153 topics by this processing. We applied the table, which are used in document data conversion for katakana notation, to the extracted retrieval key words also.

5. Index Searching and Document Ranking

We first find the data file which stores the gram of the retrieval key word from a sparse index, then scan the data file.

The indexing method we used does not maintain positions where the gram appears inside the document. Therefore, the results of the compound word retrieval are product sets of the document number sets where each retrieval key word that divides the compound word is included.

We ranked of each document by using the $tf \cdot idf$ method. Table 3 shows the retrieval time required including the ranking processing.

6. Link Structure Analysis

The retrieval result of our submitting to NTCIR-4 WEB task did ranking by using the $tf \cdot idf$ method. This is a method of ranking by the appearance frequency of the retrieval key word. Therefore, the quality of each document data is not considered. We think that the link structure on the WWW space is analyzed, and that each document data requests the level to be referred. It seems that the ranking by which the quality of each document data is considered becomes possible by giving high score to the document with high level to be referred.

Table 3. Retrieval time

corpus / query	100.0Gbyte / 153query
retrieval words	678x5
time to search all words	244sec
search time par word (average, median)	(72msec, 26msec)
retrieval time par query	1596msec

PageRank algorithm[2] has already been proposed as a method of scoring based on link structure of WWW space. The PageRank algorithm does scoring that considers “Number to be linked” and "Quality of the document data of the link origin" of all the document data. This score does not change its value depending on retrieval key words but it always becomes the level to be referred for each document in the entire WWW space.

We propose the static scoring algorithm with grouped WEB pages, and the dynamic scoring algorithm, as a target of applying the PageRank algorithm.

7. Static Scoring Algorithm with Grouped Web Pages

As for the PageRank algorithm, the quality of the document that is not adjacent in link structure is hardly reflected in the score. There is a possibility that the quality of document is not reflected between documents included in the same contents but it is not adjacent in link structure, when documents are regarded as a set with one meaning such as contents unit.

To solve this problem, we propose to apply the PageRank algorithm to grouped documents. As a result, the score, which indicates the level to be referred of each group will be obtained. As for the algorithm for grouping, the method of making subtrees of the tree structure a group is considered.

8. Dynamic Scoring Algorithm

We propose to apply PageRank algorithm not to the entire WWW space but to the retrieval result set. As a result, it is possible that the ranking by which the quality of each document including the retrieval key word is considered. The obtained score becomes a dynamic score that changes according to retrieval key words because it applies the PageRank algorithm to the retrieval result set.

We propose the method of the ranking of using $tf \cdot idf$ and the dynamic PageRank method are merged. Then the quality of document and the ranking by retrieval key words are considered. We expect that the accuracy of retrieval be improved.

9. Conclusions

We experimented on the search engine using gram-based index. We made a plural index without using morphological analysis. The time required to make indices is 34.7 hours. And, the size of these indices is 30.2Gbyte. The median at time of the retrieval a word is 26msec.

Moreover, we proposed two applications of PageRank algorithm in order to improve accuracy of retrieval. We hope that we obtain an improved scoring result by the merging two proposed techniques with an existing ranking method . We will experiment on the proposed PageRank algorithm application method in the future.

References

[1] T. Sato, T. Satomoto, and K. Han, NTCIR-3 PAT Experiments at Osaka Kyoiku University, In *Working Notes of the 3rd NTCIR Workshop Meeting Part III: Patent Retrieval Task*, pages 21-24, Tokyo, Japan, 2002.

[2] S. Brin and L. Page, The Anatomy of a Large-Scale Hypertextual Web Search Engine, In *Proceedings of the 7th International World Wide Web Conference (WWW7)*, pages 107-117, 1998.

[3] L. Page, The PageRank Citation Ranking: Bringing Order to the Web. (http://google.stanford.edu/~backrub/pageranksub.ps)