NTCIR-4 PATENT Experiments at Osaka Kyoiku University

—Gram-based Passage Index and Essential Words—

Takashi Sato Nao Hatta
Osaka Kyoiku University
4-698-1 Asahigaoka, Kashiwara, Osaka, Japan
sato@cc.osaka-kyoiku.ac.jp

Abstract

Long gram-based indices are experimented at NTCIR-4 patent task. No morphological analyses are required to make gram-based indices. The ABJ and DEJ tag fields are extracted and indexed from NTCIR-4 patent corpus. Passages are extracted and indexed also. The total index size is 240Gbyte and time to make indices is about 86 hours. By merging the result of passage retrieval with the result of document retrieval by ABJ and DEJ field, we aimed at improvement in accuracy. Ranking algorithm used is based on a traditional probabilistic model. We also tried to set essential words in a query.

Keywords: *gram based index, passage index, essential word, NTCIR*

1 Introduction

Patent retrieval using computers is commonplace today in patent applications and examinations. Nowadays, patent documents are accessible via Internet, the chances ordinal people retrieve patents will increase more and more. Specialists sometimes say that they can find words in mind well using full text retrieval systems than keyword retrieval systems. Among full text retrieval systems, systems whose indices are based on suffix array[1-5] or grams[6-10] are effective, since every character sequences including words, compound word, etc. are retrievable. In making indices, they need no dictionary and no morphological analyses.

In order to make suffix array efficiently, we have to put corpus on computer main memory. So one problem of suffix array is that we cannot make indices for big corpus.

The size of corpus for NTCIR-4 PATENT is more than 100Gbyte, which is far bigger than that of former NTCIR tasks. Since this size exceeds main memory capacity of most computers, it is impossible to make suffix array indices practically.

2 Indexing

Before indexing, we extract ABJ and DEJ fields from every document. We made indices of ABJ and DEJ tag field for every 2-year patent documents. Index structure of these indices is inverted file of gram values[11-13]. We also made PASSAGE indices from all passages in patent documents of each year. Figure 1 shows indexing process. Size and time for indexing are shown in table 1, 2 and 3. We made ABJ and DEJ indices by Pentium4 2.4GHz with 512MB memory computer, and made PASSAGE indices by Pentium4 2.6GHz with 1GB memory computer.

Table 1. Size of ABJ and DEJ indices

year	ABJ	DEJ
1993-1994	709MB	10.1GB
1995-1996	706MB	10.6GB
1997-1998	704MB	11.4GB
1999-2000	741MB	12.8GB
2001-2002	760MB	14.2GB
in total	3.62GB	59.1GB

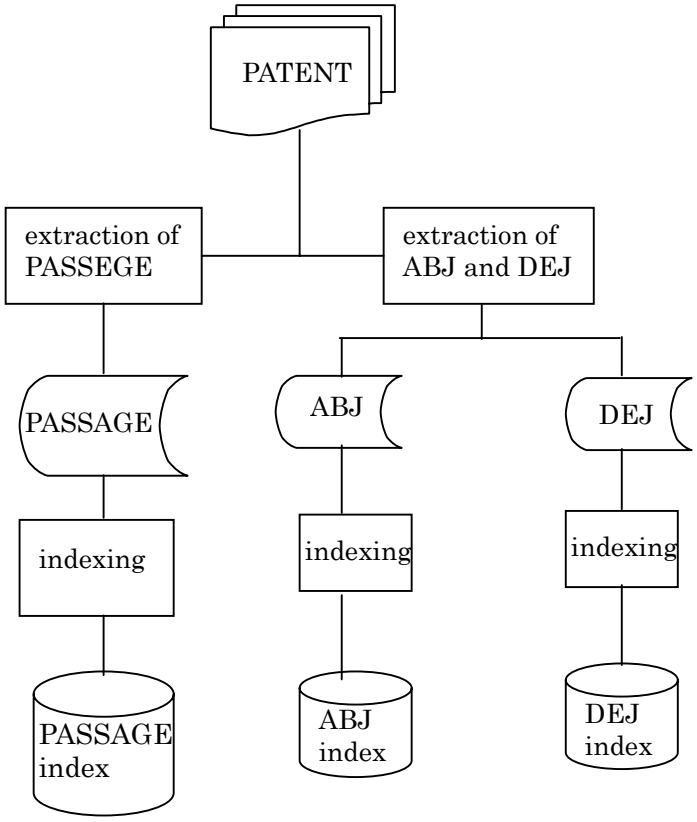

Figure 1. Indexing process

Table 2. Size of PASSAGE indices

year	PASSAGE
1993	13.6GB
1994	15.2GB
1995	15.7GB
1996	17.1GB
1997	16.9GB
1998	18.4GB
1999	19,7GB
2000	21.0GB
2001	22.2GB
2002	24.2GB
in total	184GB

Table 3. Indexing time

index	time
ABJ	10.9hour
DEJ	37.6hour
PASSAGE	37.4hour
in total	85.9hour

3 Retrieval Queries

We extracted words from each patent for search topic by morphological analysis using chasen[1]. These words were filtered by stop words. Technical terms were often analyzed as unknown words. We made phase queries from adjacent words in topics by manual.

In this task, we set essential words also. Plural words can be essential in OR, considering we can set synonyms. But we restrict them to have single meaning for simplicity. From top results file, we removed documents which do not include any essential words.

4 Retrieval and Passage Ranking

We retrieved ABJ index by queries made from ABJ field of each topic. We retrieved DEJ index by queries made from DEJ field of each topic also. PASSAGE index was retrieved by words extracted from all fields of topics.

[1] http://chasen.aist-nara.ac.jp/

From ABJ and DEJ index retrieval, we got document numbers, which contain query words. We scored retrieved documents by using probabilistic model. From passage index retrieval, we get passage numbers, which contain passage's query words. We scored retrieved passages by using probabilistic model also.

Then we sorted retrieved documents by document number, and retrieved passages by passage number. As we can lookup a document number, which contains each passage number, we merged these document and passage score. Again we re-sorted the group, which consists of merged score, passage number and document number, in descendent order of merged score. Figure 2 shows this processes.

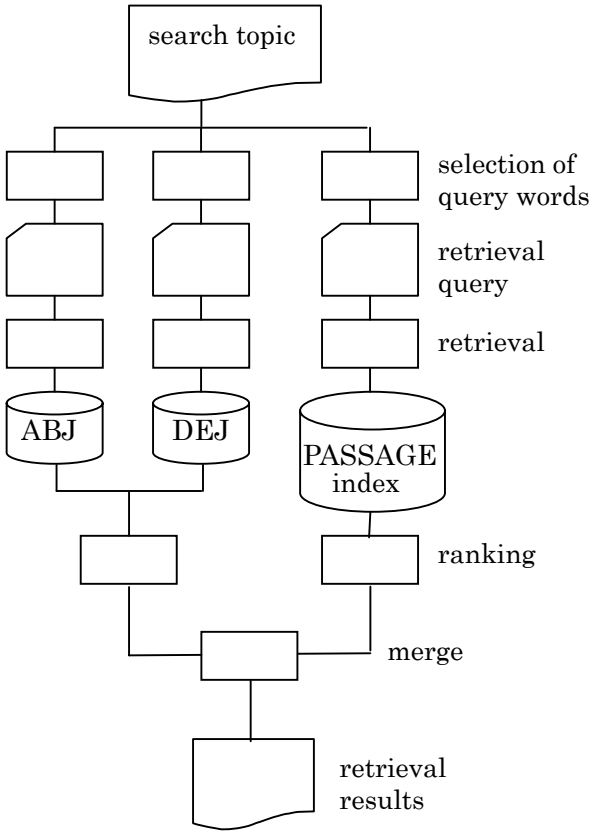

Figure 2. Retrieval and ranking process

5 Discussions

Since most passages are short, it is difficult to attain exact retrieval by passage retrieval only. By merging the result of passage retrieval with the result of document retrieval by ABJ and DEJ field, we aimed at improvement in accuracy. Since the topics themselves are patents, and they are long, it is also important subject to make suitable questions.

In this experiments, although we made the retrieval word the phrase, which connected the contiguity word centering on the technical terms, we cannot necessarily say that it is appropriate. Rather, the way combined with the general word relevant to the technical term can expect an effect.

6 Conclusions

We experimented our long gram based indices at NTCIR-4. The ABJ and DEJ tag fields are extracted from NTCIR-4 patent corpus. PASSAGE indices are prepared also. The total index size is 240Gbyte and time to make indices is about 86 hours. By merging the result of passage retrieval with the result of document retrieval by ABJ and DEJ field, we aimed at improvement in accuracy. Ranking algorithm used is based

on a traditional probabilistic model. We also tried to set essential words in a query.

References

[1] Gonnet, G., Baeza-Yates, R. and Snider, T., New Indices for Text: Pat Trees, in *Information Retrieval: Data Structure & Algorithms* chapter 5, Frakes, W. and Baeza-Yates, R. Ed., pp. 66-82 (1992).

[2] Shang, H. and Merrett T., Trees for approximate string matching, *IEEE Trans. Knowledge and Data Eng.*, Vol. 8, No. 4, pp. 540-547 (1996).

[3] Itoh, M., An Efficient Method for Constructing Suffix Arrays of Large Texts, *IPS Japan SIG Notes*, 99-NL-129-5 (1999).

[4] Yamashita, T., Fujio M. and Matsumoto Y., Language Independent Tools for Natural Language, *Proc. 18th ICCPOL*, pp.237-240 (1999).

[5] Ferragina, P. and Grossi, R., Fast string searching in secondary storage: Theoretical developments and experimental results, *Proc. ACM-SIAM Symposia on Discrete Algorithms*, Vol. 7, pp. 373-382 (1996).

[6] Ogawa, Y. and Iwasaki, M., A new character-based indexing method using frequency data for Japanese documents, *In Proc. 18th ACM SIGIR Conf.*, pp. 121-129 (1995).

[7] Sugaya, N. *et al.*, A full-text search system for large Japanese text bases using n-gram indexing method, *Proc. 53th Annual Convention IPS Japan*, 5T-2,3 (1996).

[8] Akamine, S. and Fukushima, T., Flexible string inversion method for high-speed full-text search, *Proc. Advanced Database Symposium '96* (1996).

[9] Matsui K., Namba, I. and Igata, N., Full-text searching engine for large-scale data, *Proc. 1997 IEICE General Conference*, D-4-6 (1997).

[10] Kikuchi, C., A fast full-text search method for Japanese test database, *Trans. IEICE*, Vol. J75-D-1, No. 9, pp. 836-846 (1992).

[11] Sato, T., Fast full test search with free word using TS-file, *Proc. 19th ACM SIGIR Conf.*, p.342 (1996).

[12] Sato, T., Fast full test retrieval using gram based tree structure, *Proc. ICCPOL '97*, Vol.~2, pp. 572--577 (1997).

[13] Sato, T. *et al.*, Gram based full test search system and its application, IPSJ SIG Notes, 98-DBS-114-2 (1998).