NTCIR Experiments Using the OASIS System

Vitaliy KLUEV
The Core and Information Technology Center
The University of Aizu
Tsuruga, Ikki-machi, Aizu-Wakamatsu city, Fukushima, 965-8580, Japan
vkluev@u-aizu.ac.jp
Mikhail BESSONOV
Wilhelm–Schickard Institut
Technische Informatik, Universitaet Tuebingen
Sand 13, D-72076, Tuebingen, Germany
Bessonov@informatik.uni-tuebingen.de
Vladimir DOBRYNIN
Department of Programming Technology
Saint Petersburg State University
Bibliotechnaya pl., 2, Petrodvoretz, St. Petersburg, 198904, Russia
vdobr@oasis.apmath.spbu.ru

Abstract

This paper reports the results of NTCIR experiments on Japanese and English text retrieval carried out with the OASIS System. The OASIS system is designed to support multiple languages. The main aim of these experiments is to test particular methods to support Japanese. Results showed that some improvements in the search engine need to be considered. More experiments are needed to test the effect of automatically query expansion using top relevant ranking documents.
Keywords: *OASIS, search engine, phrasal indexing, vector space model, full text searching.*

1 Introduction

Presently there are a number of search engines designed to support multiple languages, including some that support Japanese [1]. The OASIS system is one such system. Our aim in participating in the NTCIR Workshop is to test some methods to see how well OASIS supports Japanese. The basic idea of our approach is to convert Japanese text into English encoding and to apply methods used in English text retrieval that are designed in the OASIS system. Some ideas of indexing methods for documents in Japanese applied here are taken from citations [6, 3, 7, 8]. Classical methods and information processing techniques concerning Japanese texts presented in [5] were also very useful in our work. The paper is organized as follows. The short description of the OASIS system is presented in section 2. Methods used in our tests are described in section 3. Retrieval results are discussed in section 4. Final remarks can be found in section 5.

2 Description of the OASIS System

OASIS (Open Architecture Server for Information Search and Delivery) was developed by an international consortium in the framework of the INCO Copernicus program of the Commission for the European Communities [1] in 1997 - 1999. The OASIS service presents a distributed system of Internet search engines. The system provides search services for plain text and HTML documents stored on publicly accessible HTTP and FTP servers on the Internet.

Every OASIS server keeps a local index of relatively small portion of documents available on the Internet. As most users queries are subject oriented, topic specific indexes are required for scalable distributed query processing. OASIS servers are not required to be mutually exclusive in terms of the topic areas they cover. It is possible to have more than one server that cover or overlap the same topic area.

A user can contact any OASIS server with a search query. This query is then processed locally by the OASIS server or automatically propagated to other servers in the system. If the query is propagated, the OASIS

[1] OASIS, Project PL 1116-96, INCO Copernicus, Framework IV.

server that received the user query acts as a client of other OASIS servers. Before returning a result data set to the search initiator, the client OASIS server eliminates duplicate records and sorts the results by their relevance score. The OASIS system is optimized for processing poorly specified search criteria. It plays a special attention to user ranking of results and the use of this relevance feedback for the improvement of search result accuracy.

The OASIS system is designed to support multiple languages.

The tight cooperation of academic and industrial partners from four countries was a key factor to make this project successful. The partners are: St. Petersburg State University (Russia), the University of Tuebingen (Germany), the University College Dublin (Ireland); and commercial companies: Peterlink (St.Petersburg, Russia), Valtek Ltd. (Ukraine) and DSI Ltd. (Russia). The system is an open source software. It can be obtained from http://www.oasis-europe.org.

The numbers of collection architectures were created for the OASIS system. One of the proposed variants includes the Isearch system, which is a software package for indexing and searching text documents. It supports full text searching and uses a vector space model for internal document representation and an inverted file to index documents. This software is in the public domain also, which can be downloaded from http://www.etymon.com/Isearch/index.html.

More details about the OASIS system can be found in [2, 9].

The OASIS server used in the experiments was equipped by the following hardware: a PC compatible computer, the Intel Pentium III 667 MHz processor, 196 Mb RAM , the 20 Gb hard disk, the I820 motherboard. OS Linux 6.2 was running on this computer, and it was dedicated for these experiments.

3 Description of Methods Used

3.1 Japanese text retrieval task

Our approach includes the following steps:

- generate tokens from the Japanese texts
- build a vocabulary from them
- segment documents and search topic using the vocabulary constructed
- convert segmented parts to a 7 bit encoding (it was necessary to do because our search engine does not support 8 bit texts)
- index documents and build an inverted index
- run search

Table 1. Distribution of keywords in the *ntc1-j1.mod* collection

Length in bytes	Number keywords	Length in bytes	Number keywords
2	769	22	10675
4	10385	24	7439
6	24113	26	5110
8	63509	28	3477
10	50727	30	2145
12	49517	32	1357
14	36840	34	952
16	29394	36	610
18	21139	38	392
20	15498	40	251

Most researchers have reported that using bi-gram indexing for Japanese texts usually produces relatively better results of the search, when compared to other approaches [4]. We carried out our first test to compare a bi-gram indexing method and a word segmentation method. The ntc1-j1.mod test collection was used for this purpose. To do word–segmentation we were required to build a vocabulary. Fields “KYWD”, “AUPK”, “CONF” and “SOCN” from each document were utilized to construct it. Components from these fields with lengths of more than 3 bytes and less than 15 bytes were selected as words or multiple–word phrases. (Components with lengths of 18 bytes were used in the official run *OASIS4* with the *ntc2-j0k* collection.) In other words, each component consisted of $2-7$ Japanese characters. Non Japanese characters were employed as word boundaries. If the component was larger that seven characters, then only the first 14 bytes of length were taken into account. We tried to find domain specific terminology using this kind of the operation. Statistics concerning the distribution of keywords in the ntc1-j1.mod test collection are presented in Table 1.

In our case, a text is sequentially scanned to match a word dictionary. We applied word–based segmentation based upon byte length as follows. First, a string consisting of 14 bytes is matched against the dictionary entries; then, a string with 12 bytes of length and so on. If a word is found, then the beginning of the string is shifted to the end of this word. In the other case, the beginning of the string is moved in 2 byte segments. The system generates an index for “TITL”, “AUPK”, “CONF”, “CONFD”, “ABST”, “KYWD”, “KYWE” and “SOCN” fields in documents for both bi-gram and word segmentation methods. The constructed vocabulary was used as an indexing base for word-segmentation method. Before indexing, segmented parts (bi-grams for the first method) of the Japanese texts were converted to a 7 bit encoding as

Table 2. Number of keywords in the *ntc1-j1.mod* collection

Description	Number keywords
Total number keywords	1341957
Different keywords	376971
Keywords with lengths of less than 41 characters	334301
Keywords with lengths of less than 15 characters	296992

follows. Each Japanese character (two 8 bit bytes) was transformed into a sequence of three 7 bit, printable bytes. Queries were automatically generated from the search topics. For this purpose only the "description" field was employed. The "OR" operator was put between words in queries.

Because word–based segmentation method produced slightly better results than bi-gram indexing, it was used in the official runs.

Some improvements were made after sending the official runs.

- First, we changed the length of the aforementioned components to include them in the dictionary. The vocabulary was built on the base components with a length of less than 41 bytes. English terms from Japanese strings were also put into the vocabulary and employed as indexing elements. (This kind of change was used in the official run *OASIS4* with the *ntc2-j0k* collection.) A total number of different kind of keywords in the *ntc1-j1.mod* collection can be found in Table 2.

- Second, the experiments showed that word-based segmentation with the overlap match produced better results when compared to the method used in the official runs. This kind of matching indicates the following. Tokens generated from the text can overlap each other across matching boundaries. Many Japanese words were indexed several times as separate words and as components of multiple-word phrases.

- Third, the improved results were produced based upon accurate text segmentation: Hiragana characters were used as boundaries of words. We applied an approach similar to the one presented in [3].

- Fourth, more improvements were obtained by applying automatically query expansion. The simplest method was used. As in the official run, the automatically generated query was submitted to the system. The first returned document (from the top of the ranking list) was considered as a new query. All words from the indexed fields were included in this query. Words from an original query had a higher score (two times higher). The system response on this query was taken into account as a final result.

Table 3. Vocabularies from collections

Collection	Number of words
ntc1-j1.mod	396161
ntc2-j0g	
ntc2-j0k	386192
All together	789277

One more experiment was then carried out. Vocabularies of all Japanese collections were merged into one vocabulary. Numbers of words are presented in Table 3; however, we gained nothing regarding retrieval improvement.

3.2 English text retrieval task

The system indexed all of the fields in English documents. Stop words were not eliminated from documents and queries because some queries used significant words which are typical stop-words (see for example ftp://ftp.cs.cornell.edu/pub/smart). Since queries were generated automatically, only the "description" fields were taken into account. As in the case of Japanese text retrieval task, the "OR" operator was put between words in queries. We could not test this approach because the English version of training topics were not available.

4 Retrieval Results

Official runs were conducted for all test collections: ntc-j1.mod, ntc2-j0g, ntc2-j0k (Japanese) and ntc1-e1.mod, ntc2-e0g, ntc2-e0k (English). Results for Japanese and English monolingual retrieval tasks are presented in Table 4 and in Table 5 respectively. These results were calculated using the rel2_ntc2-j2_0101-0149 relevant file (level 2). It means that S-, A- and B-judgments are treated as "relevant". Values which were put in these tables differ a bit from official results because relevant judgments for each collections were extracted from the aforementioned file. These selected judgments were given as the input of the *trec_eval* program. It should be noted, our method for producing queries generated two empty queries 25 and 27 from training set of topics and one empty query 0136 in the official *OASIS4* run for the *ntc2-j0k* collection. This

Table 4. Evaluation results for the Japanese monolingual retrieval task

Run	OASIS4	OASIS1	OASIS3	OASIS9	OASIS7	OASIS8
Collection	ntc2-j0k	ntc1-j1.mod	ntc2-j0g	ntc2-j0k	ntc1-j1.mod	ntc2-j0g
Average precision at						
0.0	0.4894	0.3405	0.3839	0.5625	0.4612	0.5136
0.1	0.3405	0.2511	0.2665	0.4194	0.3668	0.3490
0.2	0.2395	0.1714	0.1155	0.3374	0.2767	0.3012
0.3	0.1530	0.1410	0.1163	0.2542	0.2454	0.2193
0.4	0.1072	0.0836	0.0888	0.1912	0.1805	0.1874
0.5	0.0548	0.0730	0.0609	0.1628	0.1516	0.1446
0.6	0.0373	0.0454	0.0555	0.1319	0.1276	0.1310
0.7	0.0167	0.0326	0.0307	0.0764	0.1031	0.1073
0.8	0.0165	0.0216	0.0084	0.0475	0.0954	0.0757
0.9	0.0155	0.0113	0.0066	0.0270	0.0807	0.0415
1.0	0.0127	0.0113	0.0066	0.0101	0.0788	0.0395
Average precision	**0.1145**	**0.0899**	**0.0943**	**0.1826**	**0.1759**	**0.1745**
5 docs :	0.3021	0.1488	0.1684	0.3957	0.2326	0.2579
10 docs:	0.2617	0.1256	0.1105	0.3447	0.1930	0.2105
15 docs:	0.2284	0.1054	0.0930	0.3078	0.1535	0.1877
20 docs:	0.2128	0.0930	0.0829	0.2766	0.1349	0.1684
30 docs:	0.1773	0.0822	0.0605	0.2433	0.1178	0.1412
100 docs:	0.1053	0.0391	0.0316	0.1551	0.0602	0.0742
200 docs:	0.0644	0.0252	0.0220	0.1082	0.0431	0.0482
500 docs:	0.0344	0.0134	0.0117	0.0615	0.0245	0.0254
1000 docs:	0.0200	0.0077	0.0075	0.0380	0.0146	0.0150
R-Precision Exact:	**0.1419**	**0.1216**	**0.1079**	**0.2036**	**0.1805**	**0.1772**

Table 5. Evaluation results for the English monolingual retrieval task

Run	OASIS6	OASIS5	OASIS2
Collection	ntc2-e0g	ntc2-e0k	ntc1-e1.mod
Average precision at			
0.0	0.4698	0.4792	0.4302
0.1	0.3533	0.3597	0.3191
0.2	0.2849	0.2597	0.2211
0.3	0.2318	0.1901	0.1693
0.4	0.1868	0.1587	0.1258
0.5	0.1411	0.1503	0.1181
0.6	0.0834	0.0922	0.0968
0.7	0.0561	0.0675	0.0788
0.8	0.0450	0.0606	0.0628
0.9	0.0329	0.0528	0.0439
1.0	0.0322	0.0493	0.0439
Average precision	**0.1565**	**0.1566**	**0.1424**
5 docs :	0.2121	0.2273	0.2150
10 docs:	0.1485	0.1659	0.1750
15 docs:	0.1414	0.1379	0.1433
20 docs:	0.1348	0.1227	0.1275
30 docs:	0.1051	0.0992	0.1067
100 docs:	0.0579	0.0439	0.0495
200 docs:	0.0368	0.0277	0.0309
500 docs:	0.0190	0.0144	0.0161
1000 docs:	0.0109	0.0083	0.0093
R-Precision Exact:	**0.1841**	**0.1778**	**0.1667**

happened because these queries consisted of words, which did not occur in the vocabulary. As it was noted earlier the *OASIS4* run differs from *OASIS1* and *OASIS3* runs in two ways: Substrings with lengths of less than 19 bytes were used to build a vocabulary, and English terms from Japanese strings were employed as indexing elements. For this reason results of the *OASIS4* run are much better than results of *OASIS1* and *OASIS3* runs.

Columns *OASIS9, OASIS7* and *OASIS8* in Table 4 show results for the same collections after first two corrections from the aforementioned set. We obtained improvements in average precision for the largest collection ntc2-j0k by 59%. For other collections improvements were also significant.

Results of the official runs are as disappointing for Japanese as they are for English collections. Additionally, the ranking styles are similar. These results mandate additional improvements to our search engine.

5 Conclusions

Results of participating in the NTCIR Workshop demonstrated the need to make some improvements to our search engine since it produced relatively poor results for both English and Japanese collections. Our improvements for the Japanese retrieval task generate much better results. The most important improvements are:

- word–based segmentation with the overlap match
- hiragana characters as word boundaries
- automatic query expansion using the document from the top of the ranked list of returned documents

More experiments are needed as well as careful observation on the effect of automatic query expansion from relevant documents (relevance feedback). We plan to carry out experiments in which words with 2 byte lengths will be taken into account. Since our system consists of distributed search engines in the Internet, our aim is to obtain the most relevant documents in response to the query inside the list of the first 10 documents.

6 Acknowledgments

Results of the *OASIS7*, *OASIS8* and *OASIS9* runs were submitted after the deadline. we would like to thank the organizers of the NTCIR Workshop for the evaluation of them.

References

[1] Japanese search engines. http://www.atrium.com/search/search.html.

[2] A. Patel, L. Petrosjan and W. Rosenstiel, editors. *OASIS: Distributed Search System in the Internet.* St. Petersburg State University Published Press, St. Petersburg, Russia, 1999. (ISBN: 5-7997-0138-0).

[3] Aitao Chen, Fredric C. Gey, Kazuaki Kishida, Hailing Jiang and Qun Liang. Comparing multiple methods for japanese and japanese–english text retrieval. In *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, Tokyo, Japan, 1999. (ISBN: 4-92-4600-77-6).

[4] Fujii H, and Croft W. B. A conparison of indexing techniques for japanese text retrieval. In *SIGIR 93*, pages 237–246, 1993.

[5] K. Lund. *CJKV Information Processing.* O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472, 1999. ISBN: 1-56592-224-7.

[6] OGAWA Yasushi. Pseudo–frequency method: an efficient document ranking retrieval method for n-gram indexing. In *SIGIR 2000*, pages 321–323, Athens, Greece, 2000.

[7] Sumio FUJITA. Notes on phrasal indexing jscb evaluation experiments at ntcir ad hoc. In *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, Tokyo, Japan, 1999. (ISBN: 4-92-4600-77-6).

[8] Toshikazu Fukushima and Susumi Akamine. A character–based indexing and word–based ranking method for japanese text retrieval. In *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, Tokyo, Japan, 1999. (ISBN: 4-92-4600-77-6).

[9] V. Kluev, V. Dobrynin and S. Garnaev. Intelligent construction of thematic collections. In N. Mastorakis, editor, *Recent Advances in Applied and Theoretical Mathematics*, pages 103–106. World Scientific and Engineering Society Press, Athens, Greece, 2000. (ISBN: 960-8052-21-1).