Uniform Indexing and Retrieval Scheme for Chinese, Japanese, and Korean

Da-Wei Juang and Yuen-Hsien Tseng
Dept. of Library & Information Science,
Fu Jen Catholic University, Taipei, Taiwan, R.O.C., 242
tseng@blue.lins.fju.edu.tw

Abstract

This paper reports on our work at the third NTCIR workshop on the subtasks of Chinese, Japanese, and Korean monolingual information retrieval (IR). A Chinese IR system is applied to all document sets in these three languages. Based on the n-gram indexing model and a phrase formulation method to extract longer key terms for indexing, no language-dependent modifications were made to apply the system to Japanese and Korean IR. Our attempt is to see whether such a system originally designed for Chinese IR can still work for Japanese or Korean documents. The results turn out that it performs similarly among the document sets in these three different languages.

Keywords:
Chinese IR, Japanese IR, Korean IR, n-gram indexing, keyword extraction.

1. Introduction

Major Asian languages, especially Chinese, Japanese, and Korean, share many common properties in information retrieval. They all have a large set of characters that are encoded in two bytes, instead of one as those in English and many European languages. Words in these three languages often consist of multiple characters and have no boundaries among them in written texts, making them difficult to identify. Although the morphological and grammatical rules are different among these languages, an information retrieval system can bypass the processing of these rules without affecting the retrieval effectiveness. Thus it is interesting to show how similar will the IR systems be for these three languages.

A number of researches have shown that n-gram indexing is at least as effective as word-based indexing for retrieval of Chinese texts [1-2]. The same phenomenon is also observed in Japanese IR and Korean IR [3-4]. By participating the single language IR (SLIR) tracks of the CLIR task in NTCIR-3 [5], this work attempts to show, if no language-dependent resources and efforts are involved, how such a simple indexing method will perform if the same IR techniques are applied to the document sets in these three different languages. In fact, in this work we use the same IR system (designed for retrieving Chinese documents) to retrieve not only Chinese texts, but also Japanese and Korean texts, without any modifications. It turns out that the performance of such a system in Japanese IR and Korean IR is quite the same as that in Chinese IR.

To better describe the details of the indexing scheme used in this work, the next section introduces a keyword/key-phrase formulation method to extract repeated strings as key terms for indexing. This method is quite language independent such that it is applicable not just to Chinese, but also to Japanese and Korean with some success. Section 3 then describes the details of the indexing and the retrieval scheme used in the experiment. Retrieval effectiveness is shown in Section 4 with some failure analysis. Finally Section 5 concludes this paper.

2. Keyword Extraction

Keywords are representative terms in documents that have many applications in information retrieval. However, keywords or key-phrases have no lexical boundaries in texts, making them hard to be identified. By repeatedly merging back nearby tokens in the text based on a merging, dropping, and accepting rule, Tseng has devised an algorithm that can extract maximally repeated strings as keywords [6]. By maximally, we mean either the repeated strings are the longest ones or they occur more often than the longer strings that contain them. For example, a repeated term “public high school” may be extracted without extracting “public high” or “high school”, as they are exact substrings of the longer term. Only if “high school” occurs more often than “public high school” (in such a case we may say: “high school” subsumes “public high school”), can “high school” be possibly extracted. Figure 1 shows the algorithm. The rule for merging, dropping, and accepting terms is implicit expressed in the algorithm. Figure 2 shows a running example, in which each capital letter denotes a character.

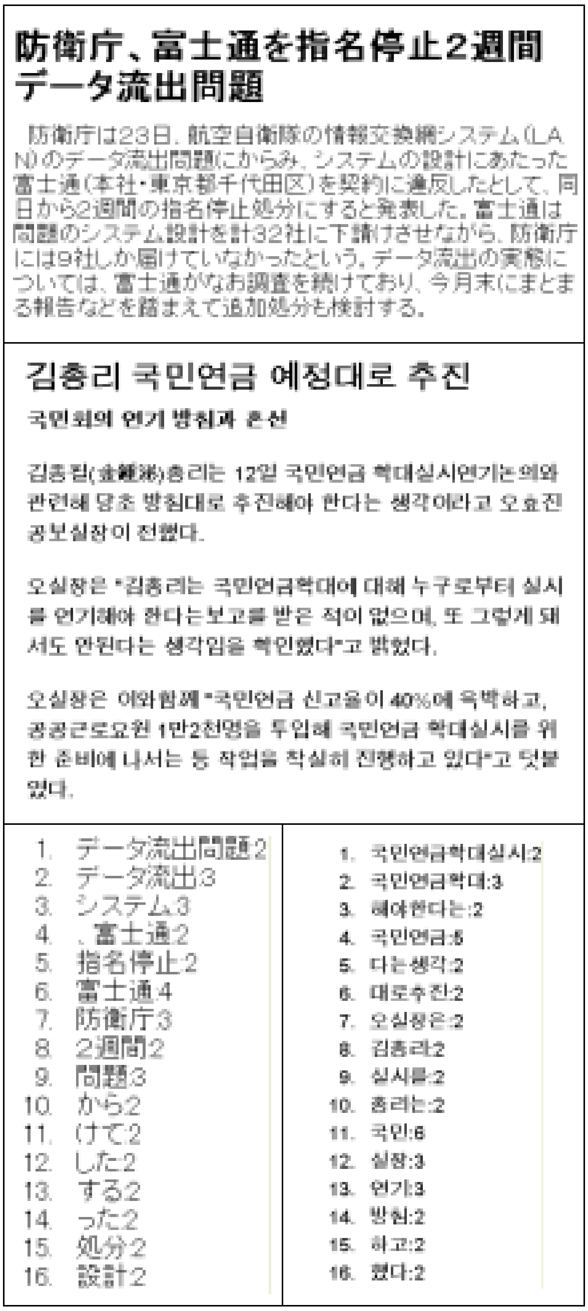
```
1. Convert the input into a LIST.
2. Do Loop
2.1    Set MergeList to empty.
2.2    Put a separator to the end of LIST as a sentinel and
         set the occurring frequency of the separator to 0.
2.3    For I from 1 to NumOf(LIST) – 1 step 1, do
         If LIST[ I ] is the separator, Go to Label 2.3.
         If Freq(LIST[ I ]) > threshold and
                Freq(LIST[ I+1]) > threshold, then
              Merge LIST[ I ] and LIST[ I +1] into Z.
              Put Z to the end of MergeList.
         Else
              If Freq(LIST[ I ]) > threshold and LIST[ I ]
                     did not merge with LIST[ I - 1], then
                   Save LIST[ I ] in FinalList.
              If the last element of MergeList is not the
                     separator, then
                   Put the separator to the end of MergeList.
       End of For loop
2.4    Set LIST to MergeList.
  Until NumOf(LIST) < 2.
```

Figure 1. The keyword extraction algorithm.

```
Example: Given an input string: BACDBCDABACD.
            Let threshold=1, separator=x.
Step 1: Create a list of single tokens:
    LIST = (B:3, A:3, C:3, D:3, B:3, C:3, D:3, A:3, B:3,
              A:3, C:3, D:3)
Step 2:
  After 1st iteration :
     MergeList = (BA:2, AC:2, CD:3, DB:1, BC:1, CD:3,
                   DA:1, AB:1, BA:2, AC:2, CD:3)
     FinalList = ( )
  After 2nd iteration :
     MergeList = (BAC:2, ACD:2, x, BAC:2, ACD:2)
     FinalList = (CD:3)
  After 3rd iteration :
     MergeList = (BACD:2, x, BACD:2)
     FinalList = (CD:3)
  After 4th iteration :
     MergeList = (x)
     FinalList = (CD:3, BACD:2)
```

Figure 2. A running example of the algorithm, where the number following a semicolon denotes the occurring frequency of the associated token.

The above algorithm is based on the assumption that a document concentrating on a topic is likely to mention a set of strings a number of times. Many natural language documents have this property, including melody strings in music [6]. We found that a longest repeated string often is a correct word (or phrase), since its repetition provides evidence for decision on its left and right boundaries as a word. Similarly, a repeated string that subsumes the others may also be likely to be a legal term. The sources of errors in Chinese keyword extraction mainly come from some single-character functional words that



防衛庁、富士通を指名停止2週間
データ流出問題

防衛庁は23日、航空自衛隊の情報交換網システム（LAN）のデータ流出問題にからみ、システムの設計にあたった富士通（本社・東京都千代田区）を契約に違反したとして、同日から2週間の指名停止処分にすると発表した。富士通は問題のシステム設計を計32社に下請けさせながら、防衛庁には9社しか届けていなかったという。データ流出の実態については、富士通がなお調査を続けており、今月末にまとまる報告などを踏まえて追加処分も検討する。

김총리 국민연금 예정대로 추진

국민회의 연기 방침과 혼선

김종필(金鍾泌)총리는 12일 국민연금 확대실시연기논의와 관련해 당초 방침대로 추진해야 한다는 생각이라고 오효진 공보실장이 전했다.

오실장은 "김총리는 국민연금확대에 대해 누구로부터 실시를 연기해야 한다는보고를 받은 적이 없으며, 또 그렇게 돼서도 안된다는 생각임을 확인했다"고 밝혔다.

오실장은 이와함께 "국민연금 신고율이 40%에 육박하고, 공공근로요원 1만2천명을 투입해 국민연금 확대실시를 위한 준비에 나서는 등 작업을 착실히 진행하고 있다"고 덧붙였다.

1. データ流出問題2
2. データ流出3
3. システム3
4. 、富士通2
5. 指名停止2
6. 富士通4
7. 防衛庁3
8. 2週間2
9. 問題3
10. から2
11. けて2
12. した2
13. する2
14. った2
15. 処分2
16. 設計2

1. 국민연금확대실시:2
2. 국민연금확대:3
3. 해야한다는:2
4. 국민연금:5
5. 다는생각:2
6. 대로추진:2
7. 오실장은:2
8. 김총리:2
9. 실시를:2
10. 총리는:2
11. 국민:6
12. 실장:3
13. 연기:3
14. 방침:2
15. 하고:2
16. 했다:2

Figure 3. A Japanese article from http://www.asahi.com/national/update/0823/015.html (first row), a Korean article from http://www.korealink.co.kr/11_home/199903/h101143.htm (middle row), extracted keywords from the Japanese article (left column), and extracted keywords from the Korean article (right column).

occur together with other words. The algorithm can extract key-phrases without any resources (such as corpora, lexicons, or dictionaries) with a precision level of 86% for Chinese news articles [7]. As to other languages, Figure 3 shows as examples two articles in Japanese and in Korean respectively and their extracted keywords. As can be seen, the algorithm is not optimized for Japanese and Korean such that some commas and functional words are not filtered in the extracted terms.

3. Indexing and Retrieval Scheme

An n-gram is a string of n consecutive characters in a text document. To select a set of terms for representing a document for indexing, n-grams, although maybe meaningless, are a good choice in addition to document words themselves. These n-grams can overlap with each other and can have variable lengths instead of a fixed length. Because no language-dependant knowledge is required, n-gram indexing is often used in multilingual or OCR degraded text retrieval [8]. In monolingual text retrieval, n-gram indexing also shows good performance, despite that it leads to larger index size and slower query response.

However, the selection of a suitable n for determining the length of the n-gram requires some analysis of the text collections and the language of the texts. Our previous studies on Chinese text retrieval in a collection of 8438 OCR documents showed that a combination of unigrams and overlapping bigrams for indexing outperforms overlapping bigram indexing alone and unigrams indexing as well [9]. This may be due to the fact that longer n-grams provide better discriminant power among documents while shorter n-grams help match desired documents in case of vocabulary mismatch so that a combination of them yields better results than using each alone.

Thus we use unigrams and overlapping bigrams in this experiment for indexing. In addition, because the collection sizes in NTCIR-3 are quite large, a total of 381,681 documents in Chinese, 249,387 documents in Japanese, and 66,146 in Korean, for n-grams to have higher discriminant power longer n-grams than bigrams are considered for indexing. However, due to the large sizes of characters in CJK (Chinese, Japanese, and Korean) languages, n-grams where n longer than 2 yield tremendous index terms that may severely degrade the efficiency of a retrieval system. To overcome this problem, we add longer keywords instead of all longer n-grams in the index to reduce the index size. The keywords are those extracted by the algorithm mentioned above. Since they are maximally repeated strings, they are in some sense the representatives of the documents in which they occur and they are far less in number than those arbitrary longer n-grams.

These index terms are then used to compute the similarity between a query and each of the documents by the following formula:

$$Sim(d_i, q_j) = \frac{\sum_{k=1}^{T} d_{i,k} q_{j,k}}{(bytesize_{d_i})^{0.375} \sqrt{\sum_{k=1}^{T} q_{j,k}^2}}$$

where the byte size denotes the number of bytes of a document. The use of byte size as the normalization factor is first introduced in the work of Singhal, et al [10] for OCR text retrieval, where the authors found that the commonly used cosine normalization factor has negative effects on retrieval when documents contain erroneous terms, such as those garbled by OCR errors or those mistyped or misspelled terms. Singhal et al also applied this byte size normalization to large collections of TREC documents. They found that it also leads to better effectiveness than cosine normalization for ordinary documents. Besides, the byte size normalization is easier to compute than the cosine normalization. Thus we use this formula in our retrieval experiment.

The document term weight $d_{i,k}$ in the above is calculated by the term frequency (tf) and the inverse document frequency (df), i.e., log(1+tf) x log(N/df), where N is the collection size. The query term weight $q_{j,k}$ is calculated by the term frequency in the query and the length of the term, i.e., tf x (3 w – 1), where w is the number of characters in the term.

The above indexing and retrieval techniques were applied to each of the Chinese, Japanese, and Korean collections for monolingual retrieval, without making any modifications.

4. Experiment Results

The CLIR task is concerned with retrieval of documents in one language by queries in another language. NTCIR-3 provides three document sets in three Asian languages, namely, Chinese, Japanese, and Korean. Monolingual retrieval or single language information retrieval (SLIR), i.e., retrieving documents with queries in the same language is provided as a subtask (also called track) in the CLIR task. Thus there are 3 SLIR tracks, denoted as C-C, J-J, and K-K, representing the tracks of Chinese to Chinese IR, Japanese to Japanese IR, and Korean to Korean IR, respectively. Due to limited manpower and resources, we only participate the 3 SLIR tracks in the CLIR task this year.

The query topics provided by NTCIR-3 consist of *title*, *description*, *narrative*, and *concept* fields. We submit two runs for each of the three SLIR tracks for evaluation. One run uses the texts in the description field as the query string (denoted as D in the run name), the other run uses the concept terms in the concept field (denoted as C in the run name). Overall, there are six runs submitted, where the run names are C-C-D, C-C-C, J-J-D, J-J-C, K-K-D, and K-K-C, all of them are prefixed with our group name: FJUIR. Each submitted run is evaluated in two criteria, one is *relax*, meaning that the relevance judgment is done in a less strict way, the other is *rigid*, meaning that the relevance is judged is in a more rigid sense.

4.1 C-C track

The C-C track consists of 42 topics. There are 14 groups submitting a total of 33 runs in the C-C track this year. Among these 33 runs, 14 runs used the description field as the sources of queries and 4 runs used the concept field as the source of the queries. Table 1 shows the average precision of our runs and the maximum, average, and minimum values of the average precisions of all runs using the same field.

Table 1: Average precisions of the C-C track.

	Relax	Rigid
FJUIR-C-C-D	0.2281	0.1858
Max of C-C-D	0.4990	0.3933
Avg of C-C-D	0.2670	0.2130
Min of C-C-D	0.0443	0.0347
FJUIR-C-C-C	0.2403	0.1997
Max of C-C-C	0.2929	0.2386
Avg of C-C-C	0.2605	0.2104
Min of C-C-C	0.2403	0.1831

4.2 J-J track

The J-J track also consists of 42 topics. There are 14 groups submitting a total of 33 runs in the J-J track. Among these 33 runs, 19 runs used the description field as queries and 2 runs used the concept field. Table 2 shows the average precision of our runs and the maximum, average, and minimum values of the average precisions of all runs using the same field.

Table 2: Average precisions of the J-J track.

	Relax	Rigid
FJUIR-J-J-D	0.2240	0.1920
Max of J-J-D	0.3998	0.3457
Avg of J-J-D	0.2640	0.2199
Min of J-J-D	0.0460	0.0396
FJUIR-J-J-C	0.2674	0.2308
Max of J-J-C	0.2898	0.2448
Avg of J-J-C	0.2786	0.2378
Min of J-J-C	0.2674	0.2308

4.3 K-K track

The K-K track consists of 30 topics. Eight groups submit a total of 17 runs. Of these 17 runs, 9 runs used the description field as queries and 2 runs used the concept field. Table 3 shows the average precision of our runs and the maximum, average, and minimum values of the average precisions of all runs using the same fields.

Table 3: Average precisions of the K-K track.

	Relax	Rigid
FJUIR-K-K-D	0.1826	0.1375
Max of K-K-D	0.3602	0.2691
Avg of K-K-D	0.2501	0.1915
Min of K-K-D	0.1256	0.0936
FJUIR-K-K-C	0.2675	0.2075
Max of K-K-C	0.2938	0.2157
Avg of K-K-C	0.2807	0.2116
Min of K-K-C	0.2675	0.2075

4.4 Failure analysis

The average precisions shown above demonstrate that the simple IR techniques without language-dependant knowledge achieve similar effectiveness in three different SLIR tracks, showing the robustness of this approach. However, its performance is under the average of all runs, indicating that there is room for improvement. After inspecting a number of retrieval results, some factors that affect the performance were identified.

First, most systems participating the CLIR tasks apply pseudo relevance feedback (PRF) to improve retrieval effectiveness. This technique chooses the top n documents from the result list of an initial search and adds more terms selected from these high-similarity documents to the original query for a second-run search. Normally the second-run search result is better in performance than the initial search, especially when the initial query contains too few terms. In contrast, we did not use this technique in our experiment. Thus the effectiveness is less than most of the others in NTCIR 3. Since it is not easy to compare with those runs without using PRF from the result sets distributed by NTCIR 3, we compare to some published results from NTCIR 2. Table 4 lists our results using the Chinese collection of NTCIR 2. Compared to some C-C runs of NTCIR 2 without using PRF, the narrative run of NTHU (National Tsing Hua University, Taiwan) has an average precision of 0.5009 [11], the question run of Berkeley is 0.4758 [3], and Trans-Ez achieves an average precision of 0.3880 [12]. As can be seen from these figures, the performance of our system is not so bad as that in NTCIR 3.

Table 4: Average precisions for NTCIR 2

	Title	Question	Narrative	Concepts
Relax	0.4730	0.4880	0.5587	0.6751
Rigid	0.3208	0.3646	0.4376	0.5736

Second, the similarity measure between a query and a document is represented in only one byte in our implemented IR system (although double precision is used during the calculation). This careless decision leads to small discrepancy between retrieved documents especially when there are hundreds of thousands of documents in the collection. Sorting these retrieved documents may loose some accuracy due to the small range of the similarities.

Third, when processing the query string, only 1-grams, 2-grams, and the longest keywords found in the index are extracted as query terms. This obviously did not use all the indexed terms that may be helpful to the retrieval effectiveness. A better query term extraction method that also includes the keywords whose length is longer than 2-gram and shorter than the longest indexed terms should be explored in the future.

5. Conclusions

An IR system designed for Chinese text retrieval is applied to Japanese and Korean IR task without modifications. The results show that it performs similarly among the documents sets in these three different languages. This is a little surprise to us since we expected that it might degrade noticeably in the Japanese and Korean IR subtasks since the IR system is optimized for Chinese only. However, our performance is under the average of all similar runs. The high performance of other systems is also a little surprise to us. Without further information on how these systems work at the time this article is written, we wonder whether it is simply the retrieval models or the retrieval strategies (such as relevance feedback) that make this difference. Anyway, we expect that our system will perform better in the near future after learning some lessons from this workshop.

This is our first year in participating NTCIR workshop. Although we have good experience in relatively small collections of OCR text retrieval [13-14], dealing with large collections in multiple languages poses another challenges to us, especially under the environment of lack of time, manpower, and hardware and software resources. However, we will benefit from participating and attending this workshop and hopefully will make more contributions to this community for the years to come.

References

[1] J. He, J. Xu, A. Chen, J. Meggs and F. C. Gey, “Berkeley Chinese Information Retrieval at TREC-5: Technical Report”, TREC-5, http://trec.nist.gov/

[2] Aitao Chen, Jianzhang He, Liangjie Xu, Fredric C. Gey, Jason Meggs, "Chinese Text Retrieval Without Using a Dictionary", Proceedings of the 20th ACM SIGIR Conference on Research and Development in Information Retrieval, 1997, pp.42-49.

[3] Aitao Chen, Fredric C. Gey and Hailing Jiang, “Berkeley at NTCIR-2: Chinese, Japanese, and English IR experiments,” Proceedings of the second NTCIR workshop, 2001, Japan.

[4] Joon Ho Lee, Hyun Yang Cho, Hyouk Ro Park, “N-gram-based Indexing for Korean Text Retrieval,” Information Processing and Management, Vol. 35, 1999, pp. 427-441.

[5] “Cross-Language Retrieval Task in NTCIR Workshop 3” http://research.nii.ac.jp/ntcir/workshop /clir/.

[6] Yuen-Hsien Tseng, "Content-Based Retrieval for Music Collections," Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval - Aug. 15-19, Berkeley, U.S.A., 1999, pp.176-182.

[7] Yuen-Hsien Tseng and Yu-I Lin "Evaluation of Fuzzy Search, Term Suggestion, and Term Relevance Feedback in an OPAC System," (in Chinese) Bulletin of the Library Association of China, No. 61, 1998, pp.103-126.

[8] Claudia Pearce and Charles Nicholas, "TELLTALE: Experiments in a Dynamic Hypertext Environment for Degraded and Multilingual Data," Journal of the American Society for Information Science, 47(4), 1996, pp.263-275.

[9] Yuen-Hsien Tseng and Douglas W. Oard, "Document Image Retrieval Techniques for Chinese" Proceedings of the Fourth Symposium on Document Image Understanding Technology, Columbia Maryland, April 23-25th, 2001, pp. 151-158.

[10] Amit Singhal, Gerard Salton, and Chris Buckley, "Length Normalization in Degraded Text Collections," Proceedings of Fifth Annual Symposium on Document Analysis and Information Retrieval, April 15-17, 1996, pp. 149-162.

[11] Jason Chang, David Yu, Ching Ting Shen, Afra Cheng, Garfield Shen, Giordano Shen and David Wong, "Nathu IR System at NTCIR-II," Proceedings of the second NTCIR workshop, 2001, Japan.

[12] Guo-Wei Bian and Chi-Ching Lin, "Trans-EZ at NTCIR-2 : Synset Co-occurrence Method for English-Chinese Cross-Lingual Information Retrieval," Proceedings of the second NTCIR workshop, 2001, Japan.

[13] Yuen-Hsien Tseng, "Error Correction in a Chinese OCR Test Collection," Proceedings of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '02, Aug. 11-15, Tampere, Finland, 2002, pp.429-430.

[14] Yuen-Hsien Tseng, "Automatic Cataloguing and Searching for Retrospective Data by Use of OCR Text", Journal of American Society for Information Science and Technology, Vol. 52, No. 5, 2001, pp. 378-390.