RIAO 2004, pp. 514-523

Chinese Character Expansion for Retrieving Japanese Paraphrases

Tokunaga Takenobu, Tezuka Yoshiki, Tanaka Hozumi
Department of Computer Science, Tokyo Institute of Technology
Tokyo Meguro Ôokayama 2-12-1, 152-8552 Japan
take@cl.cs.titech.ac.jp

Abstract

This paper proposes two methods of query expansion for retrieving paraphrase candidates indexed by *Kanzi* (Chinese) characters. The idea is to calculate similarity between *Kanzi* characters based on an ordinary thesaurus defining relations between words. The local analysis method calculates similarity of *Kanzi* characters based on a semantic class to which words in a query belong, in contrast, the global analysis method calculates similarity based on whole semantic classes in the thesaurus. The methods were evaluated by using the EDR concept dictionary which defines about 410,000 concepts. In the experiments, both headwords and concept descriptions of dictionary entries were indexed by *Kanzi* characters, and concept descriptions were retrieved by giving a headword as a query. The experiments showed that *Kanzi* character expansion is significantly effective, and the global analysis method is better than the local analysis method in recall.

1 Introduction

We can use various linguistic expressions to denote a concept by virtue of richness of natural language. However this richness becomes a crucial obstacle when processing natural language by computer. For example, mismatches of index terms cause failure of retrieving relevant documents in information retrieval systems, in which documents are retrieved on the basis of surface string matching. To remedy this problem, the current information retrieval system adopts query expansion techniques which replace a query term with a set of its synonyms (Baeza-Yates and Riberto-Neto, 1999). The query expansion works well for single-word index terms, but more sophisticated techniques are necessary for larger index units, such as phrases. The effectiveness of phrasal indexing has recently drawn researchers' attention (Lewis, 1992; Mitra et al., 1997; Tokunaga et al., 2002). However, query expansion of phrasal index terms has not been fully investigated yet (Jacquemin et al., 1997).

To deal with variations of linguistic expressions, paraphrasing has recently been studied for various applications of natural language processing, such as machine translation (Mitamura, 2001; Shimohata and Sumita, 2002), dialog systems (Ebert et al., 2001), QA systems (Katz, 1997) and information extraction (Shinyama et al., 2002). Paraphrasing is defined as a process of transforming an expression into another while keeping its meaning intact. However, it is difficult to define what "keeping its meaning intact" means, although it is the core of the definition. On what basis could we consider different linguistic expressions denoting the same meaning? This becomes a crucial question when finding paraphrases automatically.

In past research, various types of clues have been used to find paraphrases. For example, Shinyama et al. tried to find paraphrases assuming that two sentences sharing many Named Entities and a similar structure are likely to be paraphrases of each other (Shinyama et al., 2002). Barzilay and McKeown assumed that two translations from the same original text contain paraphrases (Barzilay and McKeown,

2001). Torisawa used subcategorization information of verbs to paraphrase Japanese noun phrase construction "NP_1 *no* NP_2" into a noun phrase with a relative clause (Torisawa, 2001). Most of previous work on paraphrasing took corpus-based approach with notable exceptions of Jacquemin (Jacquemin et al., 1997; Jacquemin, 1999) and Katz (Katz, 1997). In particular, text alignment technique is generally used to find sentence level paraphrases (Shimohata and Sumita, 2002; Barzilay and Lee, 2002).

Against this background, we have proposed a method to find paraphrases of a Japanese noun phrase in a large corpus using information retrieval techniques together with natural language processing techniques (Tokunaga et al., 2003). The significant feature of our method is use of character-based indexing. Japanese uses four types of writing; *Kanzi* (Chinese characters), *Hiragana*, *Katakana*, and Roman alphabet. Among these, *Hiragana* and *Katakana* are phonographic and unique to Japanese, and *Kanzi* is an ideographic writing. Each *Kanzi* character itself has a certain meaning and provides a basis for rich word formation ability for Japanese. We use *Kanzi* characters as index terms to retrieve paraphrase candidates, assuming that noun phrases sharing the same *Kanzi* characters could be paraphrases of each other. For example, character-based indexing enables us to retrieve a paraphrase "通学する子供 (a commuting child)" for "学校に通う子供 (a child going to school)". Note that their head is the same, "子供 (child)", and their modifiers are different but sharing common characters "通 (commute)" and "学 (study)". As shown in this example, the paraphrases generated based on Japanese word formation rule cannot be classified in terms of the past paraphrase classification (Jacquemin et al., 1997).

When retrieving paraphrase candidates, it is important to gain recall as much as possible to obtain novel paraphrases. To gain recall, an input query is expanded as usually done in information retrieval systems. In our case, since index terms are *Kanzi* characters, each *Kanzi* character in a query is expanded to a set of *Kanzi* characters sharing the same meaning. However, a *Kanzi* thesaurus which defines similarity between *Kanzi* characters is rarely available. We need to construct this knowledge from already existing resources. In this paper, we focus on the stage of retrieving paraphrase candidates, and propose two methods of expanding *Kanzi* indexed queries.

The structure of the paper is as follows. Section 2 provides an overview of the system. In Section 3, two methods of *Kanzi* character expansion are described. Section 4 describes experiments to evaluate the proposed methods using an electronic dictionary. Finally, Section 5 concludes the paper and looks at the future work.

2 Overview of the System

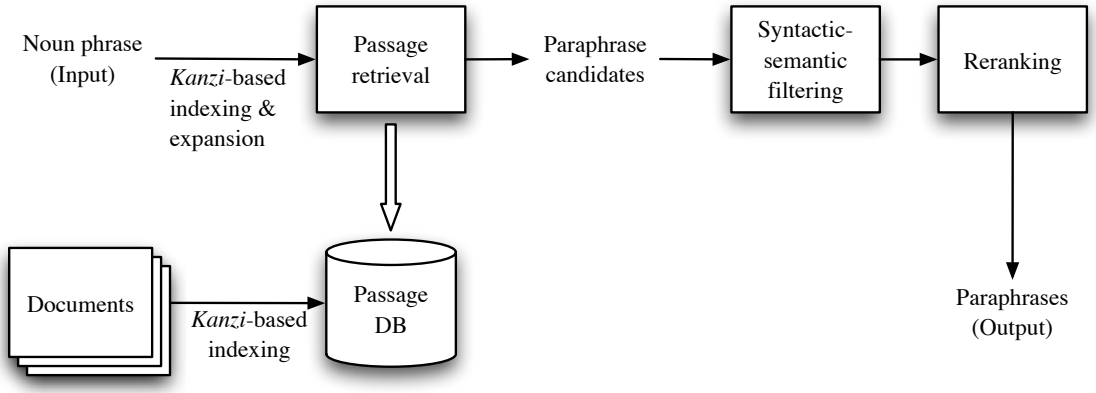

Figure 1: Overview of the system

Figure 1 shows an overview of the system which returns a ranked list of paraphrases for an input

Japanese noun phrase.

The system consists of the following three modules:

1. A passage retrieval module which retrieves paraphrase candidates from a collection of passages.
2. A filtering module which filters out irrelevant retrieved candidates based on syntactic and semantic appropriateness.
3. A reranking module which reranks the resultant candidates by considering the structure and context of the candidates.

In the following subsections, we briefly describe these three modules. Further details are found in (Tokunaga et al., 2003).

2.1 Passage Retrieval

Since our aim is to retrieve paraphrases of an input noun phrase, documents are too large as a target to retrieve. Therefore documents are segmented into a set of passages based on surface clues such as punctuation symbols. These passages are retrieved by a given query, a noun phrase.

The input noun phrase and the passages are segmented into words and they are assigned part of speech tags by a morphological analyzer. Among these tagged words, content words (nouns, verbs, adjectives, adverbs) and unknown words are selected. *Kanzi* characters contained in these words are extracted as index terms. *Kanzi* characters are expanded to a set of characters sharing the similar meaning. The *Kanzi* character expansion is the main topic of this paper and will be described in Section 3.

An index term is usually assigned a certain weight according to its importance in a query and documents. There are many proposals of term weighting, and most of them are based on term frequency in a query and documents (Baeza-Yates and Riberto-Neto, 1999). Term frequency-based weighting resides on Luhn's assumption (Luhn, 1957) that a repeatedly mentioned expression denotes an important concept. However it is obvious that this assumption does not hold when retrieving paraphrase candidates from a set of passages. Term weighting method will be described in detail together with *Kanzi* character expansion in Section 3.

Similarity between an input noun phrase and a passage is calculated by summing up the weights of terms which are shared by them. Note that since we do not use term weights of passages, we do not introduce normalization by passage length.

2.2 Filtering Candidates

Our method utilizes *Kanzi* characters as index terms. In general, using smaller index units such as characters increases exhaustivity to gain recall, but, at the same time, it decreases specificity to degrade precision (Sparck Jones, 1972). We aim to gain recall by using smaller units as index terms at the cost of precision. Even though *Kanzi* characters are ideograms and have more specificity than phonograms, they are still less specific than words. Therefore there would be many irrelevant passages retrieved due to coincidentally matched *Kanzi* characters. The filtering module filters out irrelevant retrieved passages based on both semantic and syntactic appropriateness.

Semantic appropriateness In the indexing phase, we have decomposed an input noun phrase and passages into a set of *Kanzi* characters for retrieval. In the filtering phase, on the basis of these characters, we verify if concepts mentioned in the input noun phrase are also included in the retrieved passages.

To achieve this, a retrieved passage is syntactically analyzed and dependencies between *bunsetu* (word phrase) are identified. Then, the correspondence between words of the input noun phrase and *bunsetu* of the passage is verified. This matching is done on the basis of sharing the same *Kanzi* characters. Passages missing any of the concepts mentioned in the input noun phrase are discarded in this phase.

Syntactic appropriateness Since passages are generated on the basis of surface clue such as punctuation symbols, each passage is not guaranteed to have a syntactically proper structure. In addition, a part of a passage rather than a whole passage tends to be a paraphrase of the input noun phrase. In such cases, it is necessary to extract a corresponding part from the retrieved passage and transform it into a proper syntactic structure.

When checking the semantic appropriateness, we have already identified a set of *bunsetu* covering the concepts mentioned in the input noun phrase. We extract a minimum dependency structure which covers all the identified *bunsetu*.

Finally the extracted structure is transformed into a proper phrase or clause by changing the ending of the head (the right most element) and deleting unnecessary elements such as punctuation symbols, particles and so on.

2.3 Reranking

Retrieved passages are ranked according to the similarity with an input noun phrase as described in 2.1. However this ranking is not necessarily suitable from a viewpoint of paraphrasing. Some of the retrieved passages are discarded and others are transformed through processes described above. Therefore remaining passages are reranked according to their appropriateness as paraphrases of the input noun phrase. We take into account the following three factors for reranking.

- Similarity score of passage retrieval
- Distance between words in paraphrase candidates
- Contextual information in which paraphrase candidates appear

3 Expanding *Kanzi* Characters

As mentioned in Section 1, the query expansion technique is often used in information retrieval to solve the surface notational difference between queries and documents. We also introduce query expansion in order to gain recall of retrieval. Since we use *Kanzi* characters as index terms, we need linguistic knowledge defining groups of similar *Kanzi* characters for query expansion. However this kind of knowledge is not available at hand. We obtain similarity of *Kanzi* characters from an ordinary thesaurus which defines relations of words. In this paper, we consider two approaches to calculate *Kanzi* similarity, one is local analysis and other is global analysis.

3.1 Local Analysis

The local analysis was used in our previous work (Tokunaga et al., 2003). In this paper, we modified its term weighting method to improve the performance.

Given a *Kanzi* word t composing an input noun phrase, it is expanded to a set of *Kanzi* characters $E(t)$ which is defined by (1), where C_t is a semantic class to which word t belongs, K_C is a set of

Kanzi characters used in words of semantic class C, $fr(k, C)$ is a frequency of a *Kanzi* character k used in words of semantic class C, and K_t is a set of *Kanzi* characters in word t.

$$E(t) = \{k | k \in K_{C_t}, k' = \arg\max_{l \in K_t} fr(l, C_t), fr(k, C_t) > fr(k', C_t)\} \cup K_t. \quad (1)$$

$E(t)$ consists of *Kanzi* characters which is used in words of semantic class C_t more frequently, than the most frequent *Kanzi* character in word t.

Let us see an expansion example of word "温泉 (hot spring)". Here we have $t =$ "温泉" to expand, and we have two *Kanzi* characters composing the word, i.e. $K_t =$ {"温", "泉"}. Suppose "温泉" belongs to a semantic class C_t in which we find a set of words { 温泉郷 (hot spring place), ぬるま湯 (lukewarm water), 温水 (warm water), ... }. From this word set, we extract *Kanzi* characters and count their occurrence to obtain $K_{C_t} =$ {"湯 (hot water)" (35), "村 (village)" (22), "泉 (spring)" (20), "温 (warm)" (8),... }, where the number in parentheses denotes the frequency of a character in the semantic class C_t. Since the most frequent character of K_t in K_{C_t} is "泉" in this case, more frequently used character "湯" and "村" are added to $E(t)$.

In our previous work, a weight of a term k in word t was calculated by (2).

$$w(k) = \frac{\log fr(k, C_t)}{\sum_{l \in E(t)} \log fr(l, C_t)}. \quad (2)$$

A *Kanzi* character is assigned a weight according to its frequency in the semantic class C_t, where k is used in word t.

In the previous example of "温泉", we have obtained an expanded term set {"湯", "温", "村", "泉"}. Among this set, these four characters are assigned weight according to its frequency in the class. For example, "湯" is assigned weight $\frac{\log 35}{\log 35 + \log 22 + \log 20 + \log 8} = 0.303$.

As we can see from the above definition, this weighting method tends to emphasize more frequent *Kanzi* characters in the semantic class even though they are not used in the input noun phrase. We found that this tendency does not always work well. In this paper, the term weighting method has been changed to (3).

$$w(k) = \begin{cases} \dfrac{\log fr(k, C_t)}{\sum_{l \in K_t} \log fr(l, C_t)} & \text{if } k \in K_t \\ \dfrac{w(k_{min}) \log fr(k, C_t)}{\sum_{l \in E(t) - K_t} \log fr(l, C_t)} & \text{if } k \notin K_t \\ \text{where } k_{min} = \arg\min_{m \in K_t} w(m). & \end{cases} \quad (3)$$

In this new term weighting method, the minimum weight of term $k_{min} \in K_t$ is distributed to expanded terms according to their relative frequency among the expanded *Kanzi* characters. Consequently, terms in the input noun phrase gain more weight than the expanded terms. In this new method, *Kanzi* character "湯" is assigned weight $\frac{\log 8}{\log 20 + \log 8} \cdot \frac{\log 35}{\log 35 + \log 22} = 0.222$. As shown in this example, the new method assigns less weight on the expanded terms.

3.2 Global Analysis

The local analysis considers only *Kanzi* characters appearing in a semantic class C_t to which the input word belongs. On the other hand, the global analysis calculates similarity between *Kanzi* characters based on all semantic classes defined in a thesaurus.

Firstly, each *Kanzi* character is represented in terms of a weight vector of semantic classes in the thesaurus. In other words, a *Kanzi* character is represented as a set of semantic classes to which words including the *Kanzi* character belongs. The weight of a semantic class is calculated by the ordinary *tf· idf* weighting method.

Given a *Kanzi* character k_i used in the input noun phrase, its similarity to other *Kanzi* character k_e is calculated by the product of their cosine and the ratio of k_i's number of semantic classes which are shared with k_e as shown in (4).

$$sim(k_i, k_e) = \frac{\vec{k_i} \cdot \vec{k_e}}{\|\vec{k_i}\|\|\vec{k_e}\|} \times \frac{cf(k_i \cap k_e)}{cf(k_i)}, \tag{4}$$

where $\vec{k_i}$ denotes a vector representation of k_i, and $cf(k_i)$ denotes the number of semantic classes in which k_i appears. Note that this similarity measure is asymmetric because of the second factor in (4).

For example, let us calculate similarity of *Kanzi* character "泉 (spring)" against "湯 (warm water)". Suppose "泉" is used in words of the semantic classes {0505 (15), 0748 (7), 2446 (4), 0749 (4), 0462 (4), 0889 (3), 0466 (3), 0423 (3), ...}, and "湯 (warm water)" appears in { 0749 (31), 1611 (15), 0905 (8), 0911 (7), 0906 (6), 1998 (6), 0857 (5), 0902 (5), 0742 (4), 0748 (4), 0851 (4), 2419 (4),...}, where four digit codes denote semantic classes defined in a thesaurus, and the numbers in parentheses are their frequency. Weight of each semantic class is calculated based on *tf· idf*, and cosine of these two vector is calculated to provide the first factor in (4). Suppose the semantic classes shared by these two *Kanzi* characters are {0466, 0505, 0748, 0749, 2419, 2446}, then the second factor would be $\frac{6}{17}$, where 17 is the total number of semantic classes in which "泉" appears. The similarity of "泉" and "湯" is calculated by the product of these two factors.

By setting a threshold, we can obtain a set of similar *Kanzi* characters to an input *Kanzi* character based on this similarity. Table 1 shows an actual ranking of similar *Kanzi* characters to "泉 (spring)".

Table 1: Similar *Kanzi* to "泉"

rank	*Kanzi*	meaning	similarity
1	泉	(spring)	1.000
2	水	(water)	0.302
3	湯	(warm water)	0.097
4	噴	(blowing)	0.074
5	涌	(spring)	0.064
6	源	(source)	0.061
7	湧	(spring)	0.059
8	歇	(pause)	0.054
9	温	(warm)	0.044
10	清	(clean)	0.039

4 Experiments

4.1 Data and Preprocessing

In this paper, we focus on evaluating the effectiveness of *Kanzi* character expansion rather than evaluating the total system performance. For this purpose, we used the EDR concept dictionary (Japan Electronic Dictionary Research Institute, LTD., 1993) as a test data. The EDR concept dictionary defines about 410,000 concepts together with the relations between them. We conducted experiments of

retrieving concept descriptions of the dictionary entries by giving a headword as a query. Since the relations between headwords and their concept description are defined by the dictionary, this information enables us to avoid creating relevance judgment from scratch, which usually costs very much.

Among the EDR dictionary, a collection of dictionary entries is extracted based on the following criteria.

- A headword is a content word. In particular, we used only nouns and verbs.
- Both a headword and its concept description have two or more *Kanzi* characters in type. It is less likely to retrieve proper concept descriptions with only one index term, since we adopt character-based indexing.
- A headword does not appear in its concept description. Since our aim is to retrieve paraphrase candidates, in other words, we are searching for a different linguistic expression denoting the same meaning, the entries which contain the headword as is in their description are useless as paraphrase candidates.

As a result of selecting entries based on these criteria, we obtained 134,758 concept descriptions corresponding to 94,155 headwords, 86,810 nouns and 7,345 verbs. The distribution of the number of *Kanzi* characters in these headword are shown in Table 2.

No. of *Kanzi*	No. of headwords
2	61,856
3	19,127
4	9,935
5	1,933
6∼	1,304

Table 2: Distribution of No. of *Kanzi* characters

Most of concept descriptions consist of one or two sentences, therefore we use a concept description as a passage to retrieve. These passages are indexed based on *Kanzi* characters, then stored in the GETA retrieval engine (IPA, 2003). The average number of index terms of a passage was 5.4.

Since we have relevance judgment for all queries (headwords), we can use all queries for evaluation. However, due to a limitation of computational resources, we randomly selected about 1,000 headwords for each of groups which were made based on the number of *Kanzi* characters in the headword. In this experiments, we made three groups that is, two *Kanzi* group, three *Kanzi* group and four and more *Kanzi* group. When submitting these headwords as queries, if the headword of a retrieved passage (a concept description) is the same as the input headword, the retrieval is judged correct.

As a thesaurus for *Kanzi* character expansion, we used *Nihongo Goi Taikei* (Ikehara et al., 1997) which classifies a total of about 260,000 words into 2,710 semantic classes. Out of the extracted 94,155 headwords of the EDR dictionary, 36,132 words (38.4%) are also found in the thesaurus. In contrast, 4,555 *Kanzi* characters appear in the 94,155 EDR headwords and 4,150 of them (91.1%) also appear in the thesaurus. This difference suggests that we can expect better recall with *Kanzi* indexing.

4.2 Results and Discussion

Table 3 shows the summary of the results. In the columns, "w/o Exp." denotes the retrieval without *Kanzi* character expansion, "Local" denotes the method employed in our previous work (Tokunaga et al., 2003) and "Local+" is its modified version as described in 3.1. "Local++" denotes a expansion

No. of *Kanzi*	Measure	w/o Exp.	Local	Local+	Local++	Global		Pseudo feedback	
						($\theta = 0.1$)	($\theta = 0$)	(top 10)	(top 20)
2	Recall	75.7	81.3	81.4	87.3	83.7	98.0	89.8	93.1
	Ave. rank	1.4	2.9	2.3	6.4	5.0	27.8	7.1	8.5
	Ave. #hw.	15	1,021	553	3,235	990	5,521	6,434	11,954
	#Exp. chars.	0	2.4	2.8	77.7	4.9	3,063	34	60.4
3	Recall	84.6	86.8	87.0	90.9	89.6	99.6	94.2	95.7
	Ave. rank	1.5	2.8	2.3	6.4	5.5	27.3	6.8	8.3
	Ave. #hw.	51	857	450	2,225	1,036	4,965	4,410	7,538
	#Exp. chars.	0	2.1	2.2	92.8	7.4	3,490	36.8	66.7
4$\sim$	Recall	92.4	93.9	93.9	96.3	94.4	99.5	96.8	97.9
	Ave. rank	1.4	4.1	2.7	6.9	4.2	17.6	6.1	7.7
	Ave. #hw.	69	493	247	1,466	408	2,950	1,984	3,376
	#Exp. chars.	0	4.2	4.5	210.4	7.7	3,709	43.3	77.7

Table 3: Summary of the results

method in which frequency restriction of "Local+" is removed, that is, we use all *Kanzi* characters in a semantic class to which an input word belongs as expansion terms. Weight of the terms are assigned as the same as "Local+". "Global" denotes the global analysis method described in 3.2. We used two thresholds (θ) of similarity, 0.1 and 0. "Local++" and "Global ($\theta = 0$)" provides the upper bound of recall for "Local+" and "Global" respectively.

We conducted experiments with pseudo feedback as well. After the retrieval without *Kanzi* expansion ("w/o Exp."), the highly ranked concept descriptions were regarded as correct, and *Kanzi* characters appearing in these descriptions were added to the original query for second retrieval. Since the pseudo feedback technique does not require any linguistic resources like thesaurus, these results would be another baseline for the proposed method. We used two thresholds 10 and 20 for deciding the correct outputs. The results were shown in the last two columns in Table 3.

In the rows, we have three groups based on the number of *Kanzi* characters in headwords, each of which contains about 1,000 queries. In each of these groups, four measures are presented. "Recall" denotes the ratio of the cases in which correct headwords were retrieved. "Ave. rank" denotes the average rank of the correct headword in the retrieved ranked list, and "Ave. #hw." denotes the total number of headwords when summing the number of headwords in each rank up to the correct headword. In other words, we need to check this number of headwords on average to find a correct one. "#Exp. chars." denotes the average number of expanded *Kanzi* characters.

Table 3 shows that *Kanzi* character expansion is effective, particularly, the improvement is significant when fewer *Kanzi* characters are used in headwords.

Comparing "Local" and "Local+", the recall is almost the same, but the average rank of the correct headword is better in "Local+". Another notable thing is the difference in the average number of headwords to be checked until we find a correct one. Comparing with the "Local+" method, we need to check almost double headwords with the "Local" method.

"Global ($\theta = 0.1$)" shows higher recall than "Local" and "Local+". When the number of *Kanzi* characters in headwords is small, the improvement is significant. The number of headwords to be checked to find a correct one is comparable to that of "Local".

Superiority of "Global" over "Local+" is also shown in its upper bound, that is, 87.3% versus 98.0% in the two *Kanzi* group. Although they are extreme cases and might not be practical to use, this tendency shows that the "Global" method has better potential to gain recall by setting an appropriate threshold. Particularly, we are concerned with finding paraphrases, high recall is an important feature of retrieval.

Even though irrelevant candidates are retrieved, they could be filtered out in the later stage as described in 2.2.

"Pseudo feedback" showed better results than others in recall, but this is not the case in other measures. In particular, the number of outputs to be checked were significantly larger than other methods. We can presume that many spurious *Kanzi* characters were added by expansion. We calculated the measures of the "Global" method at the threshold which gives about the same recall as the "Pseudo feedback (top 10)". The results are shown in Table 4. Compared to Table 3, we found that "Pseudo feedback" requires more than double outputs to be checked to find a correct one.

No. of *Kanzi*	Measure	Global
2	Recall	89.5
	Ave. rank	10.5
	Ave. #hw.	2,135
	#Exp. chars	15.2

Table 4: The result of "Global" method ($\theta = 0.055$)

5 Conclusions and future work

We have proposed a unique method to find paraphrases of a Japanese noun phrase from a collection of documents (Tokunaga et al., 2003). The significant feature of the method is its using *Kanzi* (ideograms) characters as index terms and retrieves paraphrase candidates from a set of passages. The retrieved candidates are then filtered out based on syntactic and semantic appropriateness.

In this paper, we focused on query expansion in the retrieval phase of this method. Unlike the ordinary information retrieval systems, we expand a *Kanzi* character to a set of *Kanzi* characters sharing the similar meaning. To achieve this, we proposed two expansion methods, local analysis and global analysis. Both methods use an ordinary thesaurus which defines relations between words in order to calculate similarity between *Kanzi* characters. The local analysis method expands a *Kanzi* character based on a semantic class to which words in a query belong. In contrast, the global analysis method calculates similarity between *Kanzi* characters based on the whole semantic classes in the thesaurus.

Experiments were conducted to evaluated the methods in which concept descriptions of the EDR dictionary were retrieved given a headword as a query. The results showed that *Kanzi* character expansion is effective, in particular when the number of *Kanzi* characters in queries is small. The global analysis method showed better performance in recall than the local analysis method, but tended to provide more candidates.

The experiments were conducted to evaluate the effectiveness of *Kanzi* character expansion *per se*. We need to evaluate performance of the total system searching for paraphrases by adopting the proposed methods.

References

R. Baeza-Yates and B. Riberto-Neto. 1999. *Modern Information Retrieval*. Addison Wesley.

R. Barzilay and L. Lee. 2002. Bootstrapping lexical choice via multiple-sequence alignment. In *Proceedings of 2002 Conference on Empirical Methods in Natural Language Processing*, pages 164–171.

R. Barzilay and K. R. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 50–57.

C. Ebert, L. Shalom, G. Howard, and N. Nicolas. 2001. Generating full paraphrases of fragments in a dialogue interpretation. In *Proceedings of the 2nd SIGdial Workshop on Discourse and Dialouge*.

S. Ikehara, M. Miyazaki, S. Shirai, A. Yokoo, H. Nakaiwa, K. Ogura, Y. Ooyama, and Y. Hayashi, editors. 1997. *Goi-Taikei — A Japanese Lexicon*. Iwanami Shoten.

IPA. 2003. GETA: Generic Engine for Transposable Association. http://geta.ex.nii.ac.jp.

C. Jacquemin, J. L. Klavans, and E. Tzoukermann. 1997. Expansion of multi-word terms for indexing and retrieval using morphology and syntax. In *Proceedings of 35th Annual Meeting of the Assosiation for Computational Linguistics*.

C. Jacquemin. 1999. Syntagmatic and paradigmatic representation of term variation. In *Proceedings of 37th Annual Meeting of the Assosiation for Computational Linguistics*, pages 341–348.

Japan Electronic Dictionary Research Institute, LTD. 1993. Edr electronic dictinary technical guide.

B. Katz. 1997. Annotating the world wide web using natural language. In *Proceedings of "Computer-assisted information searching on Internet" (RIAO '97)*, pages 136–155.

D. D. Lewis. 1992. An evaluation of phrasal and clustered representations of a text categorization task. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 37–50.

H. P. Luhn. 1957. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1(4):390–317.

T. Mitamura. 2001. Automatic rewriting for controlled language translation. In *The Sixth Natural Language Processing Pacific Rim Symposium (NLPRS2001) Post-Conference Workshop, Automatic Paraphrasing: Theories and Applications*.

M. Mitra, C. Buckley, A. Singhal, and C. Cardie. 1997. An analysis of statistical and syntactic phrases. In *Proceedings of RIAO '97*, pages 200–214.

M. Shimohata and E. Sumita. 2002. Automatic paraphrasing based on parallel corpus for normalization. In *Third International Conference on Language Resources and Evaluation*, pages 453–457.

Y. Shinyama, S. Sekine, K. Sudo, and R. Grishman. 2002. Automatic paraphrase acquisition from news articles. In *Proceedings of Human Language Technology Conference (HLT2002)*, pages 40–46.

K. Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

T. Tokunaga, K. Kenji, H. Ogibayashi, and H. Tanaka. 2002. Selecting effective index terms using a decision tree. *Natural Language Engineering*, 8(2-3):193–207.

T. Tokunaga, H. Tanaka, and K. Kimura. 2003. Paraphrasing japanese noun phrases using character-based indexing. In *Proceedings of the Second International Workshop on Paraphrasing (IWP2003)*, pages 80–89.

K. Torisawa. 2001. A nearly unsupervised learning method for automatic paraphrasing of japanese noun phrase. In *The Sixth Natural Language Processing Pacific Rim Symposium (NLPRS2001) Post-Conference Workshop, Automatic Paraphrasing: Theories and Applications*, pages 63–72.