Automatic Query Expansion for Japanese Text Retrieval

Chengfeng Han, Hideo Fujii, W.Bruce Croft
Computer Science Department
University of Massachusetts, Amherst, MA 01003
e-mail: {chan, fujii, croft}@cs.umass.edu

Abstract

Automatic query expansion methods for English text retrieval have been studied for a long time, with debatable success in many instances. In this paper, we study what the retrieval effectiveness will be achieved when we apply a successful automatic query expansion method for English text retrieval to Japanese text retrieval. Our experiments show that the automatic query expansion method also results in a notable improvement in Japanese text retrieval.

1 Introduction

Authors and searchers use a great variety of words to refer to the same thing, this is why expanding or modifying the users' queries can led to considerable improvement in retrieval results. Manual query expansion, semi-manual query expansion, and automatic query expansion have been studied to improve retrieval performance. In this paper, we focus on automatic query expansion.

Automatic query expansion or modification has been studied for nearly three decades, and a lot of methods have been proposed. The various methods can be basically classified into the following six groups[1]:

1. Based on syntactic information. The term relations are generated on the basis of linguistic knowledge and co-occurrence statistics [2][3]. The method uses a grammar and a dictionary to extract a list of terms for each term. A query is expanded by adding those terms which are most similar to any of the query terms. This produces only slightly better results than the original queries[4].

2. Based on relevance information. Relevance information is used to construct a global information structure, such as a pseudothesaurus [4][5] or a minimum spanning tree[6]. A query is expanded by means of this global information structure. The main problem about this method is that relevance information is not always available.

3. Automatic term classification. The similarities between terms are first calculated based on the association hypothesis and then used to classify terms by setting a similarity threshold

value [7][8][9]. A query is then expanded by adding all the terms of the classes that contain query terms. This method is too naive an approach to be useful[8][10][11].

4. Use of document classification. Documents are first classified and infrequent terms found in a document class are clustered in the same term class (thesaurus class) [12]. The indexing of documents and queries is enhanced either by replacing a term by a thesaurus class or by adding a thesaurus to the index data. However, the retrieval performance depends strongly on some parameters that are hard to determine [13]. This method is also much more expensive.

5. Concept based query expansion [1]. A similarity thesaurus is a term-vs-term similarity matrix which is constructed based on how the terms of the collection are indexed. A probabilistic method is used to estimate the probability of a term similar to a given query in the vector space model. This approach improves retrieval performance significantly.

6. Phrase based query expansion [14]. A term-vs-phrase association thesaurus is constructed by identifying phrases in the text and representing them by the terms that are closely associated with them (i.e.,occur in the same text windows). This method can also simulate all types of term-based association thesauri such as the term-vs-term, term-vs-noun_term, term-vs-verb_term and term-vs-adjective & adverb_term association thesauri. Queries are expanded by adding those phrases that are most related to the query from the association thesaurus. This approach also improves retrieval performance significantly.

Japanese has many different characteristics from English, such as syntax(e.g., S-O-V word order), agglutination (i.e., no space between words), phrase rules [15], pragmatics (e.g., the paragraph is less well defined).

Among the various methods, method 6 is simple and effective. The method can also simulate many types of association thesauri so that we can compare different types of association thesauri's effects to query expansion for Japanese text retrieval. The method can also specify the phrase rules and paragraph limit (i.e., we can specify the number of sentences within a paragraph). So we apply an approach based on this method to Japanese text retrieval and study what the retrieval effectiveness will be achieved.

In the following section, we show how such an association thesaurus is constructed. In section 3, we present a query expansion method. In section 4, many experiments have been done to evaluate this approach's effectiveness to Japanese text retrieval, and compare the effects of many different kinds of association thesauri to query expansion for Japanese text retrieval.

2 Automatically Constructing an Association Thesaurus

Building an association thesaurus is based on the assumption that concepts that have a similar lexical context may be related semantically. For example, the words "connectionism" and "neural networks" occur in similar lexical contexts, but rarely in the same documents.

We view a text as a structured object. Text objects consist of paragraphs. A paragraph consists of a set of sentences. It can be a natural text paragraph or a fixed number of sentences. A sentence contains phrases and words. Terms are defined as all words except stop words. A phrase do not

have to be a real phrase, it can be a sequence of terms which satisfy the set of specified phrase rules according to each term's part of speech tag.

Assume no two paragraphs are the same in the collection. We present a phrase by a vector $h_i = (p_{i1}, p_{i2}, \ldots, p_{in})^T$ in the paragraph vector space defined by all the paragraphs of the collection. The p_{ik}'s signify feature weights of the paragraph p_k with respect to the phrase h_i and n is the number of paragraphs in the collection. We define the feature weights p_{ik} as the phrase h_i's frequency of occurrence in the paragraph p_k.

We also present a term by a vector $t_j = (p_{j1}, p_{j2}, \ldots, p_{jn})^T$ in the paragraph vector space. Similarly, the p_{jk}'s signify feature weights of the paragraph p_k with respect to the term t_j and n is the number of paragraphs in the collection. The p_{jk} is defined as the term t_j's frequency of occurrence in the paragraph p_k.

With these definitions, we define the similarity between phrase h_i and term t_j by using a similarity measure such as the simple scalar vector product:

$$SIM(h_i, t_j) = t_j^T \cdot h_i = \sum_{k=1}^{n} p_{jk} \cdot p_{ik} \tag{1}$$

For each phrase h_i, the terms are chosen by setting a similarity threshold value. Thus, those terms form a feature vector representing the lexical context of the phrase h_i.

Since the Japanese words in a text are not separated by spaces, a segmentation program is needed to identify and extract terms from the documents. In our system, we use the JUMAN segmentation program[17] to extract terms from the Japanese databases. JUMAN can also get each term's part of speech tag.

Building an association thesaurus for Japanese databases mainly consists of the following five steps:

1. Use the Japanese segmentation program JUMAN to segment documents , get each term's root and each term's part of speech tag.

2. The program JPhrasefinder extracts phrases according to the set of specified phrase rules and the terms' part of speech tags.

3. For each phrase, JPhrasefinder collects the significant terms that occur in any document within a paragraph of n sentences and builds the term-vs-phrase association thesaurus.

4. Do some pro-processes such as data filtering.

5. The built association thesaurus is stored in JINQUERY[21].

JINQUERY is a Japanese version of INQUERY. INQUERY is based on an inference network-based probabilistic retrieval model which views information retrieval as an evidential reasoning process in which multiple sources of evidence about document and query content are combined to estimate the probability that a given document matches an information need [19]. INQUERY has demonstrated good retrieval effectiveness [19] for English text retrieval. JINQUERY also demonstrated good retrieval effectiveness [15] for Japanese text retrieval.

The thesaurus built by JPhraseFinder is collection dependent. JPhraseFinder automatically creates collection-specific phrase links. If there exists a collection which has a very good coverage over all topics, it is possible to produce a general thesaurus.

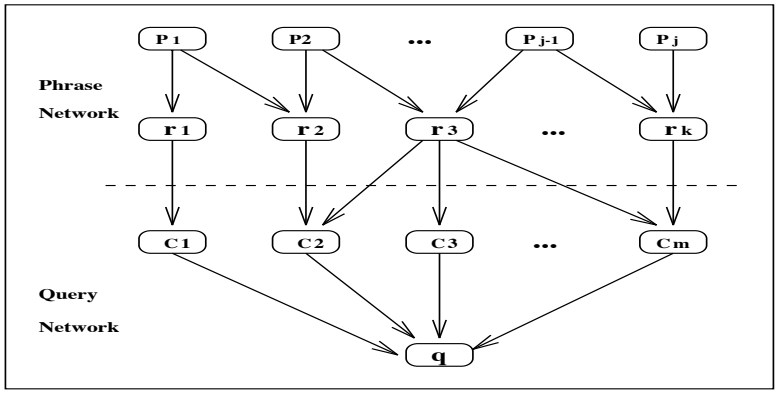

Figure 1: A simple phrase retrieval inference network

3 Automatically Expanding Queries

Three of the basic problems about automatic query expansion are:

- How to select suitable items for query expansion ?
- How to select weights of the selected items?
- How many items should be added into the original query?

Qiu has showed [1] that a query should be expanded by adding those terms that are most familiar to the query, rather than selecting terms that are similar to the query terms.

In our approach, we use JINQUERY to access the association thesaurus. We map phrases of the thesaurus into documents of JINQUERY, and terms into terms, the association between terms and phrases into the link from terms to documents. Thus, the association thesaurus is changed into a probabilistic inference network. Figure 1 illustrates a simple phrase retrieval inference network.

The similarity between a phrase and the query is denoted by $Simqp(Q, p)$. We define $Simqp(Q, p)$ by a constant k and the belief value of Q when the phrase p is instantiated as follows.

$$Simqp(Q, p) = k \cdot bel(Q \mid p) \tag{2}$$

Here, We present a simple example to show how phrases are selected and weighted by JINQUERY.

The inference network fragment shown in Figure 2 contains two phrases p1, p2 and three representation concepts. Phrase p1 is represented by the term r1 (*Japanese*) and term r2 (*automobile*), phrase p2 is represented by the term r2 (*automobile*) and term r3 (*industry*). A single query is

$$Q = \#WSUM(1.0\ 3.0\ Japanese\ 2.0\ automobile)$$

For the purpose of this example, we are using only features of a simplified form of the basic model shown in Figure 1. Arcs are drawn from a phrase only to representation concepts that have been assigned to that phrase. When a phrase is instantiated, it provides equal support for all members of the set of assigned representation concepts; all other representation concepts receive no

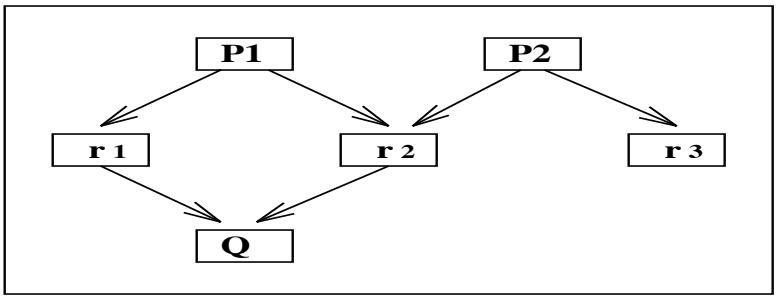

Figure 2: Inference network fragment

support. Techniques for estimating the representation nodes' beliefs are discussed in detail in [20]. Here, for the purpose of this example, we just use the following estimates [18] for the representation nodes' beliefs:

$$P(r_i = true \mid p_j = true) = 0.5 + (0.5 \cdot tf_{ij} \cdot idf_i) \tag{3}$$

$$P(r_i = true \mid all\ parents\ false) = 0.0 \tag{4}$$

where, if we use max_th_j to represent the maximum $SIM(h_j, t_i)$ value for any term t_i which associates the phrase h_j, and use fi to represent the term t_i's frequency of occurrence in the entire collection, then

$$tf_{ij} = \frac{SIM(h_j, t_i)}{max\_th_j} \tag{5}$$

$$idf_i = \frac{\log(collectionsize/f_i)}{log(collectionsize)} \tag{6}$$

Table 1 gives the terms' SIM and tf values for the two phrases and idf values on the collection. We assume that $max\_th_1 = 5$ and $max\_th_2 = 4$.

Table 1: SIM , tf and idf values

t_i	$SIM(p_1, t_i)$	$SIM(p_2, t_i)$	tf_{i1}	tf_{i2}	$idfi$
Japanese	3	0	0.6	0	0.77
automobile	3	2	0.6	0.5	0.18
industry	0	4	0	1.0	0.624

Thus, from equation (3), we have

$$P(Japanese = true \mid p1 = true) = 0.5 + 0.5 \times 0.6 \times 0.77 = 0.731$$

which results in a link matrix of

$$L_{Japanese} = \begin{pmatrix} 1.000 & 0.269 \\ 0.000 & 0.731 \end{pmatrix}$$

For the *automobile* node, we must compute beliefs for both parents, so

$$P(automobile = true \mid p1 = true) = 0.5 + 0.5 \times 0.6 \times 0.18 = 0.554$$

$$P(automobile = true \mid p2 = true) = 0.5 + 0.5 \times 0.5 \times 0.18 = 0.545$$

which result in a link matrix of

$$L_{automobile} = \begin{pmatrix} 1.000 & 0.455 & 0.446 & 0.446 \\ 0.000 & 0.545 & 0.554 & 0.554 \end{pmatrix}$$

The last column of this link matrix is unused since only one document can be instantiated at a time. It is set to the maximum of the individual document beliefs.

Using the same procedure, the link matrix for *industry* is

$$L_{industry} = \begin{pmatrix} 1.000 & 0.298 \\ 0.000 & 0.812 \end{pmatrix}$$

There are several ways to estimate the matrix at Q. We would generally estimate the matrix based on the frequency of each term in the query text. The link matrix can then be estimated as

$$L_Q = \begin{pmatrix} 1 & 0.6 & 0.4 & 0 \\ 0 & 0.4 & 0.6 & 1 \end{pmatrix}$$

Instantiating p1 results in

$$bel(r1) = 0.731 \quad bel(r2) = 0.554 \quad bel(r3) = 0.0$$

Instantiating p2 results in

$$bel(r1) = 0.0 \quad bel(r2) = 0.545 \quad bel(r3) = 0.812$$

Which gives

$$bel(Q \mid p1) = 0 \times 0.269 \times 0.446 + 0.4 \times 0.269 \times 0.554 + 0.6 \times 0.731 \times 0.446$$

$$+1 \times 0.731 \times 0.554$$

$$bel(Q \mid p2) = 0 \times 1.0 \times 0.455 + 0.4 \times 1.0 \times 0.545 + 0.6 \times 0.0 \times 0.455$$

$$+1 \times 0.0 \times 0.545$$

In practice, link matrices are replaced by a variety of canonical forms that are more efficient to store and compute[19].

Table 2 shows phrases discovered automatically by running JINQUERY on the association thesaurus. The floating numbers to the left of each phrase are belief values. The absolute magnitude of a belief value is not meaningful, but the relative magnitude of belief values is significant.

Query:1.1: #WSUM(1.0 3.0 日本 2.0 自動車)	
0.509971	鉛 メッキ 鋼
0.508772	日本 自動車 産業
0.507165	生産 会社
0.505467	ＧＥ
0.505430	自動車 市場
0.505430	自動車 流通
0.505324	自動車 電話
0.504911	ＭＤＣ 社
0.503601	自動車 部品 メーカー
0.503520	欧州 市場

<< Japanese >>

Query:1.1: #WSUM(1.0 3.0 Japanese 2.0 automobile)	
0.509971	lead plate steel
0.508772	Japanese automobile industry
0.507165	production company
0.505467	GE
0.505430	automobile market
0.505430	automobile distribution
0.505324	automobile telephone
0.504911	M D C company
0.503601	automobile part maker
0.503520	European market

<< English >>

Table 2: Phrases discovered automatically for query 1.1.

After phrases are discovered, we consider how phrases are weighted. In our approach, we choose the weight(Q, p) of an added item as $Simqp(Q, p)$. We specify a max_weight for the top ranked phrase p_t. According to equation (2), we get the constant k,

$$k = max\_weight/bel(Q \mid p_t) \tag{7}$$

Then we weight each phrase p_i according to the following equation:

$$weight(Q, p_i) = k \cdot bel(Q \mid p_i) \tag{8}$$

For the phrases in Table 2, if we specify that the weight of the top ranked phrase 'lead stell' is 0.8, from equation (7), we have

$$k = 0.8/0.509971 = 1.5687.$$

Then according to equation (8), we can get all the phrases' weights in Table 2. Table 3 shows final weights of the phrases in Table 2.

weight(Q, p_i)	phrase p_i
0.800000	lead steel
0.798111	Japanese automobile industry
0.795590	producer community
0.792926	GE
0.792868	automobile market
0.792868	automobile flow
0.792702	automobile telephone
0.792054	M D C community
0.789999	automobile part maker
0.789872	European market

Table 3: Weights for the phrases in Table 2

After we have determined how terms are selected and weighted, we take into account how many items should be added into the original query . Qiu[1] has showed that adding only those terms that are ranked in the top positions , rather than all the terms from the association thesaurus, can result in more significant effectiveness. In the next section, we will do experiments to study how the number of added phrases affects the retrieval effectiveness. We also study the following three expansion models:

DUP Only duplicate phrases from an association thesaurus are added into the queries.

A phrase is duplicate for a given query means that all terms of the phrase are subsets of the original query. For example, for the query "high frequency oscillators using transistors", the phrase "transistor oscillator", and the term "transistors" are duplicate with the query. The purpose of adding duplicate phrase is to see whether the evidence from the association thesaurus identifies the importance of query terms or phrases.

NODUP Only nonduplicate phrases are added into queries.

If any term of a phrase is not a subset of a given query, the phrase is nonduplicate with the query. For the sample query above, the phrase "npn transistors" , and terms "triode" and "anode" are nonduplicate with the query. The purpose of adding nonduplicate phrase is to see whether a thesaurus can provide some new information about queries.

BOTH Both duplicate and nonduplicate phrases are added into queries.

4 Experiments and their Results

There are various test collections in English, but unfortunately, there is no standard collection for Japanese currently. So, we used a test collection which has 1101 full-text newspaper articles on Japanese business and economics. It contains articles from three major Japanese newspapers, The Asahi, The Mainichi and Nihon-Keizai(Nikkei)-Shinbun.

In this experiment, we used 27 test queries. The queries are generally expressed as a sentence containing several keywords.

The program JPhrasefinder can produce different association thesauri by defining different phrase rules[1]. For example, if we define the phrase rule as {V}, JPhrasefinder will produce a term-vs-verb_term association thesaurus; if we define the phrase rule as {N}, a term-vs-noun_term association thesaurus will be built.

An important question is what range should be used to generate association data. Intuitively, a natural paragraph seems appropriate because it generally focuses on describing an individual topic. Because using a long paragraph would limit the performance of JPharasefinder, an experiment parameter, the paragraph limit, is set. The paragraph limit defines the maximum number of sentences allowed within a paragraph.

Experiments are conducted to address the following issues:

- How do different query expansion models (i.e., DUPONLY, NODUP, BOTH), different numbers of added items and max_weights for the top item from the association thesaurus for a given query affect query expansion's effectiveness?
- How do different types of terms, like verbs, adjectives & adverbs, and nouns affect retrieval performance?
- Compare the effects between the term-vs-term association thesaurus and the term-vs-phrase association thesaurus to query expansion.
- What are the best noun phrase rules?
- Compare the effects of association thesauri with different paragraph limits to query expansion.

According to above issues, our experiments are divided into the following four parts:

1. Test how different query expansion models (i.e., DUPONLY, NODUP, BOTH), different numbers of added items and max_weights for the top item from the association thesaurus for a given query affect query expansion's effectiveness[2].

   Table 4 shows the retrieval results of the expanded queries for different expansion models at 10 point recall-precision.

[1] In the phrase rule, we use individual letters represent the part of speech tags, i.e., N: noun, V: verb, J: adjective, D: adverb, R: prefix, O: postfix, A: adnoun. If a set of phrases is defined as {NN, NNN}, it means that any two or three consecutive nouns is taken as a phrase.

[2] all experiments in this part are based on an association thesaurus with the phrase rule {NNN, NN, N}, the paragraph limit 3. Because the results are sensitive to many parameters, in all experiments in section 4, correspond to the specified parameters, we take the best results which we got under all other parameters.

Recall	Precision (% change)-27 queries			
	BASE	DUPONLY	NODUP	BOTH
10	71.9	76.6 (+6.5)	72.8 (+1.3)	75.3 (+4.8)
20	66.6	69.5 (+4.3)	66.9 (+0.4)	69.7 (+4.7)
30	56.8	59.3 (+4.4)	56.5 (-0.5)	58.5 (+3.0)
40	48.4	50.3 (+3.9)	49.1 (+1.4)	50.1 (+3.4)
50	41.7	43.5 (+4.5)	42.6 (+2.3)	44.0 (+5.6)
60	36.4	36.8 (+1.0)	37.0 (+1.4)	37.8 (+3.8)
70	33.2	33.4 (+0.5)	33.4 (+0.5)	33.6 (+1.1)
80	30.9	30.8 (-0.3)	31.6 (+2.3)	31.2 (+1.1)
90	25.0	25.1 (+0.1)	25.0 (-0.4)	26.0 (+3.9)
100	19.7	19.6 (-0.6)	19.8 (+0.7)	19.7 (-0.1)
avg	43.1	44.5 (+3.3)	43.5 (+0.8)	44.6 (+3.6)

Table 4: The retrieval results for different expansion models

First, the figures indicate that this automatic query expansion method yields a significant improvement (a 3.6% improvement for the collection which has only 1100 documents) in the retrieval effectiveness of Japanese text retrieval. Qiu [1], Jing and Croft[14] found that the performance improvement increase with the size of collection, so this gives promise of query expansion to Japanese text retrieval.

Among the three expansion models, DUPONLY gets a considerable improvement in retrieval results, NODUP gets a slight improvement in retrieval results, BOTH performs best.

In our experiments, we found that the retrieval performance is sensitive to the specified max_weight for the top item from the association thesaurus for a given query and the number of added nonduplicate items, and that the number of duplicate items discovered by JINQUERY is generally small, all duplicate items should be added to get better retrieval performance.

Figure 3 shows how the 'max_weight' in DUPONLY mode affects the retrieval effectiveness. Figure 4 shows how the 'max_weight' in NODUP mode affects the retrieval effectiveness. It seems that there is an optimum max_weight value. Further research should be done to find how to determine the optimum 'max_weight'.

Figure 5 shows some experimental results[3] about how the number of added nonduplicate items affects the retrieval performance. It seems that improvement by expanded queries increases when the number of additional terms increases. When the number of additional terms is between 14 to 21, the improvement remains constant. Once the number of additional terms gets to be larger than 22 , the improvement decreases. The results are similar to Qiu's conclusion[1]. Because our test collection size is very small, further experiments should be done on larger collections for more reliable conclusions.

[3]BOTH model is used, the max_weight for the top duplicate item is 0.11, the max_weight for the top nonduplicate item is 0.05.

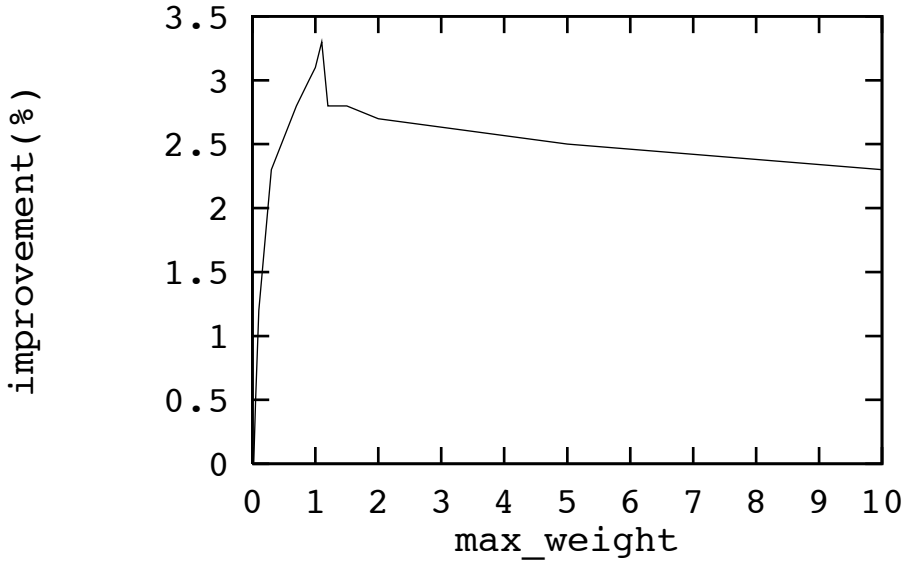

Figure 3: Improvement using expanded queries with different max_weights in DUPONLY model

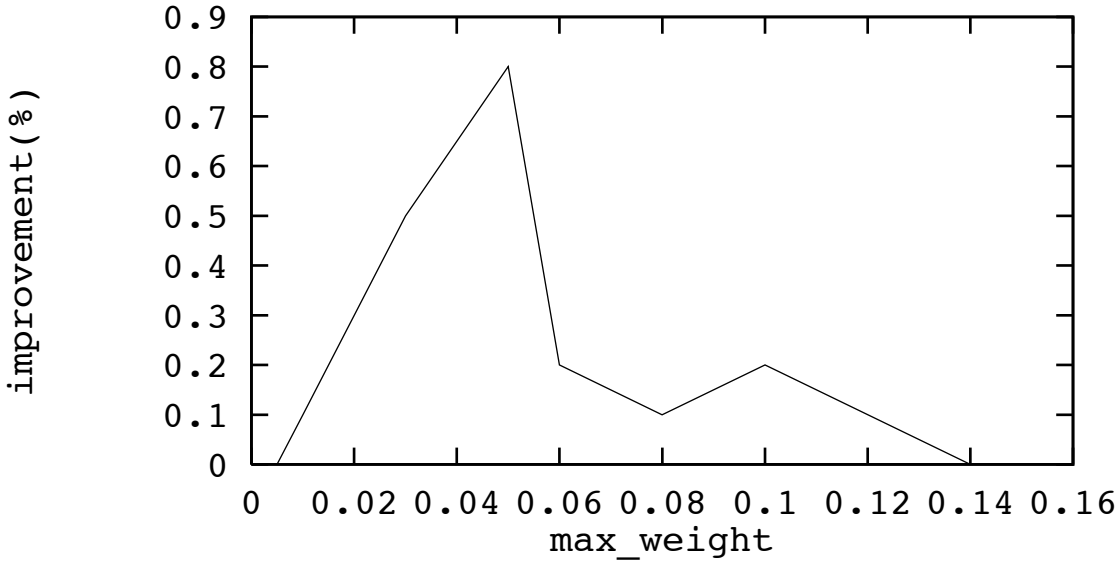

Figure 4: Improvement using expanded queries with different max_weights in NODUP model

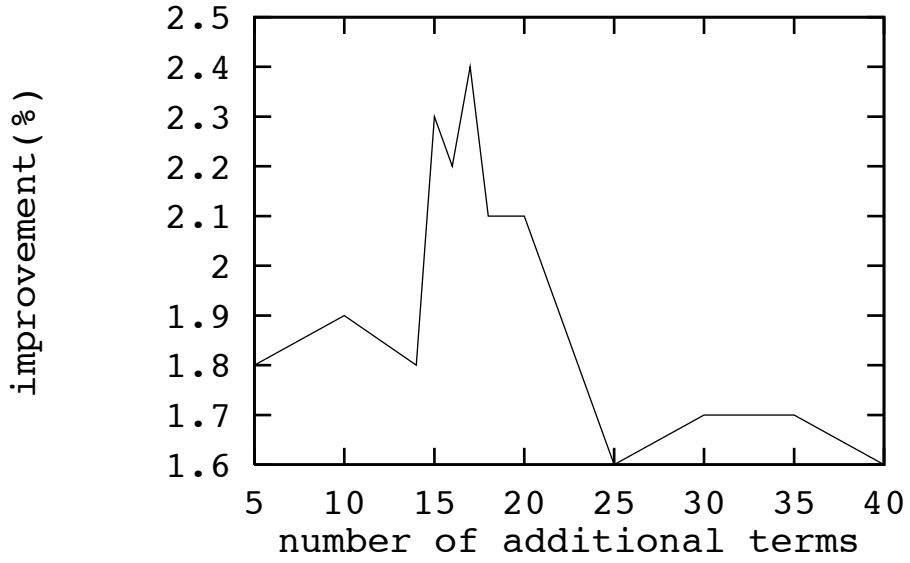

Figure 5: Improvement using expanded queries with various numbers of added nonduplicate items

2. Test on the effects of all kinds of term-based association thesauri to query expansion.

   Most existing automatic thesauri are based on term-vs-term co-occurrence data. Here, we simulate a term-vs-term association thesaurus by defining a set of phrase rules as {N, J, D, V}, i.e., all terms are 'phrases'. In order to study how different types of terms, like verbs, adjectives and adverbs, and nouns affect retrieval performance, we also simulate term-vs-noun_term, term-vs-verb_term, and term-vs-adjective & adverb_term association thesauri.

   Table 5 gives a summary of results[4] on effects of term-vs-term, term-vs-noun_term, term-vs-verb_term, and term-vs-adjective & adverb_term association thesauri to query expansion.

[4]all thesauri are built with the paragraph limit 3 in this part.

Recall	Precision(% change)-27 queries				
	BASE	{N}	{J,D}	{V}	{N,J,D,V}
10	71.9	73.4 (+2.0)	72.1 (+0.2)	71.9 (-0.1)	72.3 (+0.6)
20	66.6	69.1 (+3.8)	66.8 (+0.3)	66.6 (-0.0)	69.0 (+3.7)
30	56.8	56.3 (-0.9)	56.9 (+0.2)	57.2 (+0.8)	55.4 (-2.4)
40	48.4	50.3 (+3.8)	48.4 (+0.0)	48.9 (+1.0)	48.8 (+0.7)
50	41.7	43.0 (+3.1)	41.5 (-0.4)	42.1 (+1.0)	42.7 (+2.6)
60	36.4	37.8 (+3.6)	36.4 (-0.0)	36.5 (+0.2)	37.4 (+2.6)
70	33.2	34.8 (+4.8)	33.2 (-0.1)	33.2 (+0.0)	34.9 (+5.0)
80	30.9	30.8 (-0.3)	30.9 (-0.1)	30.9 (+0.1)	31.2 (+0.8)
90	25.0	25.1 (+0.2)	25.2 (+0.4)	25.0 (-0.3)	25.2 (+0.7)
100	19.7	19.4 (-1.2)	19.7 (+0.0)	19.7 (+0.3)	19.7 (+0.2)
avg	43.1	44.0 (+2.1)	43.1 (+0.1)	43.2 (+0.3)	43.7 (+1.4)

Table 5: Summary of results on all types of term-based association thesauri

From above experimental results, term-vs-noun_term thesaurus performs better than the term-vs-verb or the term-vs-adjective & adverb_term thesaurus, the term-vs-verb_term thesaurus performs better than the term-vs-adjective & adverb_term thesaurus. The term-vs-term thesaurus, which ignores part of speech tags, is slightly less effective than the term-vs-noun_term thesaurus, but performs better than the term-vs-verb or the term-vs-adjective & adverb_term thesaurus.

It seems that nouns are most useful for query expansion, verbs less, adjectives & adverbs least.

We also concluded that the term-vs-phrase thesaurus in part 1 performs much better than all types of term_based thesauri.

3. Test what are the best noun phrase rules?

   There are many noun phrase rules used to identify the noun phrases for Japanese text[15], but what are the best noun phrase rules? In the following experiments, three sets of phrase rules, are used to produce different term-vs-phrase association thesauri. Table 6 presents the experimental results[5] on the three term-vs-phrase association thesauri.

[5]all thesauri are built with the paragraph limit 3 in this part.

Recall	Precision (% change)-27 queries			
	BASE	{NNN NN}	{NNN,NN,N,NV,VN,RNN, NNO,RNO,JRNN,JNN,JNNO}	{NNN NN N}
10	71.9	75.9 (+5.5)	75.9 (+5.5)	75.3 (+4.8)
20	66.6	69.2 (+3.8)	69.3 (+4.1)	69.7 (+4.7)
30	56.8	59.0 (+3.9)	59.3 (+4.4)	58.5 (+3.0)
40	48.4	50.0 (+3.2)	50.3 (+3.9)	50.1 (+3.4)
50	41.7	43.0 (+3.3)	43.5 (+4.5)	44.0 (+5.6)
60	36.4	36.6 (+0.4)	36.6 (+0.4)	37.8 (+3.8)
70	33.2	33.4 (+0.6)	33.4 (+0.6)	33.6 (+1.1)
80	30.9	30.9 (-0.1)	30.9 (-0.1)	31.2 (+1.1)
90	25.0	25.0 (+0.0)	25.1 (+0.1)	26.0 (+3.9)
100	19.7	19.7 (+0.0)	19.6 (-0.6)	19.7 (-0.1)
avg	43.1	44.3 (+2.8)	44.4 (+3.0)	44.6 (+3.6)

Table 6: Summary of results on the term-vs-phrase thesauri with different sets of noun phrase rules

The purpose of removing single noun from the set of phrase rules is to reduce the amount of the association data because many phrases are composed of single nouns. It is expected that some important single-noun phrases would be lost because of this. But how big would the loss be? The Table 6 shows that the loss is not significant for Japanese text. Further experiments should be done on larger collections to confirm this conclusion.

Fujii and Croft mentioned some phrase rules for Japanese text retrieval in their paper[15], we use these phrase rules as our second set of phrases rules, but the experimental results show that the addition of adjective, prefix, postfix, and adnoun to the noun phrase rules does not enhance retrieval performance.

Our experiments show that the association thesaurus based on the noun-only noun phrase rules performs best. We may even remove single noun from the phrase rules to reduce the amount of association data if we can confirm that the individual nouns do not play a very important role for query expansion in Japanese text retrieval.

4. Experiments on association thesauri with different paragraph limits.

   JPhrasefinder has shown that a small paragraph limit can significantly reduce the amount of association data. The experiments on association thesauri with different paragraph limits are to find an appropriate paragraph limit. Table 7 reports the results of query expansion using three thesauri with different paragraph limits. PL-n means that the association thesaurus is generated with the paragraph limit n.

Recall	Precision(% change)-27 queries			
	BASE	PL-1	PL-3	PL-5
10	71.9	75.2 (+4.6)	75.3 (+4.8)	75.7 (+5.2)
20	66.6	70.1 (+5.2)	69.7 (+4.7)	69.3 (+4.1)
30	56.8	60.1 (+5.9)	58.5 (+3.0)	59.2 (+4.3)
40	48.4	51.8 (+6.9)	50.1 (+3.4)	50.0 (+3.3)
50	41.7	45.2 (+8.4)	44.0 (+5.6)	43.2 (+3.6)
60	36.4	38.8 (+6.4)	37.8 (+3.8)	36.8 (+0.8)
70	33.2	34.7 (+4.5)	33.6 (+1.1)	33.5 (+0.9)
80	30.9	32.5 (+5.3)	31.2 (+1.1)	30.8 (-0.5)
90	25.0	26.5 (+5.7)	26.0 (+3.9)	25.1 (+0.2)
100	19.7	19.6 (-0.2)	19.7 (-0.1)	19.6 (-0.6)
avg	43.1	45.4 (+5.5)	44.6 (+3.6)	44.3 (+2.9)

Table 7: Summary of results on association thesauri with different paragraph limits

From above experimental results, the association thesaurus with paragraph limit 1 produces much better results than the other thesauri. This may be explained by the fact that the sentence is well organized in Japanese.

5 Conclusion

In this paper, we applied an approach based on the phrased-based query expansion method to Japanese text retrieval, and did some experiments on the collection with 1100 documents. Our experiments showed:

- The method also results in considerable improvement in Japanese text retrieval.
- Retrieval performance is sensitive to the weights of the added items and the number of the added additional terms.
- For Japanese text retrieval , the term-vs-phrase thesauri perform much better than all types of term_based thesauri, the association thesaurus based on the noun-only noun phrase rules produces the best improvement, and individual nouns may not help to improve retrieval performance very much.
- For Japanese text, it seems that nouns are most useful for query expansion, verbs less, adjective & adverbs least.
- For Japanese text retrieval, the association thesaurus with paragraph limit 1 produces much better results than thesauri with other paragraph limits.

In our future research, we will do further experiments on larger databases to conform conclusions we got here. We will also do some experiments to compare this method's effects to Japanese text retrieval and English text retrieval.

Qiu[1], Jing and Croft[14] have showed that the larger the collection is, the better the association thesaurus performs, better results will be expected on association thesauri built on larger collections.

References

[1] Qiu Yonggang and Frei,H.P., "Concept based query expansion", SIGIR' 93, p160-169, 1993.

[2] Grefenstette, G., "Use of syntactic context to produce term association lists for retrieval", SIGIR' 92, p89-97, June 1992.

[3] Ruge, G., "Experiments on linguistically-based term associations", Information Processing and Management, 28(3), p317-32, 1992.

[4] Salton,G., "Experiments in automatic thesaurus construction for information retrieval", Information Processing, 1, p115-123, 1971.

[5] Salton, G., "Automatic term class construction using relevance - a summary of work in automatic pseudoclassification", Information Processing & Management, 16(1), p1-15, 1980.

[6] Smeaton, A.F., Van Rijsbergen, C.J., " The retrieval effects of query expansion on a feedback document retrieval system", The Computer Journal, 26(3), p239-46, 1983.

[7] Lesk, M.E., "Word-word association in document retrieval systems", American Documentation, 20(1), p27-38, 1969.

[8] Minker, J., Wilson, G.A., Zimmerman, B.H., "An evaluation of query expansion by the addition of clustered terms for a document retrieval system", Information Storage and Retrieval, 8(6), p329-48, 1972.

[9] Sparck-Jones, K., Barber, E.B.," What makes an automatic keyword classification effective?", J.of the ASIS, 18, p166-175, 1971.

[10] Peat, H.J., Willett, P., "The limitation of term co-occurrence data for query expansion in document retrieval system", J. of the ASIS, 42(5), 378-83, 1991.

[11] Spark-Jones, K., "Notes and references on early classification work", SIGIR Forum, 25(1), p10-17, 1991.

[12] Croch, C.J., "An approach to the automatic construction of global thesauri", Information Processing and Management, 26(5), p629-40, 1990.

[13] Croch, C.J., Yong, B., "Experiments in Automatic Statistical Thesaurus Construction", SIGIR' 92, p77-87, June 1992.

[14] Yufeng Jing, Croft W.B., "An association thesaurus for information retrieval", UMass Technical Report, p94-7, 1994.

[15] Fujii H., Croft W.B., "A comparison of indexing techniques for Japanese text retrieval", SIGIR' 93, p237-47, 1993.

[16] Fujii, H., Croft, W.B., "Comparing the Retrieval Performance between English and Japanese text databases", 2 nd Annual Workshop on Very Large Corpora, 1994.

[17] Matsumoto Y., Kurohasi S., Myoki Y., etc., "Users' guide for the JUMAN system, a user-extensible morphological analyzer for Japanese", Nagao Laboratory, Kyoto University, 1991.

[18] Croft, W.B., Turtle, H., "Text retrieval and inference", In Text-Based Intelligent System, Paul Jacobs (ed.), Lawrence Erlbaum, New Jersey, p127-156.

[19] Turtle, H., Croft, W.B., "Evaluation of an inference network-based retrieval model", ACM Transactions on Information Systems, 9(3), p187-222, 1991.

[20] Turtle, H. "Inference networks for document retrieval", PH.D thesis, University of Massachusetts at Amherst, 1990.

[21] Callan, J.P., Croft, W.B., and Harding, S.M., "The INQUERY retrieval system", Proceedings of the 3rd International Conference on Database and Expert System Applications, p78-83, 1992.

[22] Croft, W.B., Turtle, H., and Lewis, D., "The use of phrases and structed queries in information retrieval", SIGIR' 91, p32-45, 1991.

[23] Ogawa, R., Kikuchi, Y. Takahasi, K., "Recent developments in full-text database technologies", IPSJ Joho-shori, 33(4), p404-412, 1992.

[24] Sasaki, Y., "An automatic query construction experiment", Proc. of the 25th Annual Meeting on Information Science and technology, p43-48, 1988.