NTCIR-5 Query Expansion Experiments using Term Dependence Models

Koji Eguchi
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
eguchi@nii.ac.jp

Abstract

This paper reports the results of our experiments performed for the Query Term Expansion Subtask, a subtask of the WEB Task, at the Fifth NTCIR Workshop, and the results of our further experiments. In this paper we mainly investigated: (i) the effectiveness of query formulation by composing or decomposing compound words and phrases of the Japanese language, which is based on a theoretical framework via Markov random fields, but taking into account special features of the Japanese language; and (ii) the effectiveness of the combination of phrase-based query formulation and pseudo-relevance feedback. We showed that pseudo-relevance feedback worked well, particularly when using query formulation with compound words.

Keywords: *Query Expansion, Term Dependence Model, Japanese Information Retrieval*

1 Introduction

Japanese text retrieval is required to handle several types of problems specific to the Japanese language, such as word compounds and segmentation [9]. To treat these problems, word-based indexing is typically achieved by applying a morphological analyzer, and character-based indexing has also been investigated. In earlier work, Fujii and Croft compared unigram-based and word-based indexing, and found their retrieval effectiveness comparable, especially when applied to text using *kanji* characters (*i.e.*, Chinese characters in Japanese) [9]. Following this work, many researchers investigated more complex character-based indexing methods, such as using overlapping character bigrams, sometimes mixed with character unigrams, for the Japanese language as well as for Chinese or Korean. Some researchers compared this kind of character-based indexing with word-based indexing, and found little difference between them in retrieval effectiveness [10, 1, 17]. Hybrid methods that combined word-based query formulation with character-based indexing were also studied [9, 18, 19]; however, the focus of these works was on how to improve efficiency while maintaining effectiveness in retrieval.

More recently, data fusion, overlapping indexing, and structured queries were investigated to handle the problems mentioned previously in text retrieval in Japanese or some other Asian languages. Some researchers used data fusion to achieve good retrieval effectiveness by combining several heterogeneous ranked lists produced independently, such as by word-based and character-based indexes, in response to a query [10, 23, 11]. Some earlier studies applied overlapping indexing, and investigated indexing methods based on overlapping n-grams, as described above, while others investigated indexing methods using both compound words and their constituent words [20, 21, 24].

The structured query approach has been used to include more meaningful phrases in a proximity search query to improve retrieval effectiveness [5, 14]. Phrase-based queries performed effectively, especially against large-scale and noisy text data such as typically appear on the Web [16, 14]. This paper focuses on structured queries as another approach to the problems in Japanese text retrieval. A few researchers investigated this approach to retrieval for Japanese newspaper articles [9, 17]; however, they emphasized how to formulate a query using n-grams and showed they performed comparably in retrieval effectiveness with the word-based approach. We are not aware of any studies that have applied structured queries to formulate queries reflecting Japanese compound words or phrases appropriately, or to effectively retrieve web documents written in Japanese.

In this paper, we use the structured query approach using word-based units to capture compound words, as well as more general phrases, into a query, and experiment using a large-scale web document collection mostly written in Japanese. We investigate query formulation by composing or decomposing compound words and phrases of the Japanese language, which is based on a theoretical framework via Markov random fields [14] but take into account special features of the Japanese language. We also investigate the effects of pseudo-relevance feedback for Japanese web documents and its combination with phrase-based queries.

Some of our experiments were performed while participating in the 'Query Term Expansion Subtask' [24] at the Fifth NTCIR Workshop, which was conducted from August 2004 to December 2005, as one of the subtasks of the 'WEB Task'. This paper also reports the results of further experiments. Because we focus on query formulation rather than retrieval models, we use 'Indri' as a baseline platform for our experiments. Indri is a scalable search engine platform that combines language modeling and an inference network framework [13, 15]. Indri supports structured queries simi-

Table 1. Feature functions and the corresponding Indri query language.

Feature	Type	Indri Expression
$f_T(q_i, D)$	Term	q_i
$f_O(q_i, q_{i+1}, ..., q_{i+k}, D)$	Ordered Phrase	$\#1(q_i q_{i+1} ... q_{i+k})$
$f_U(q_i, ..., q_j, D)$	Unordered Phrase	$\#\text{uw}N(q_i ... q_j)$

larly to 'INQUERY' [22]. This enables our experiments to take Japanese phrases including compound words into account, for instance by formulating queries expressed with compound words or phrases.

The rest of this paper is structured as follows. Section **2** briefly explains the task definition of the Query Term Expansion Subtask. Section **3** introduces Indri [13, 15] as the experimental platform, and the term dependence model via Markov random fields [14], which gives a theoretical framework to our investigation in this paper. Section **4** describes our phrase-based query formulation, which composes or decomposes Japanese compound words or phrases for effective searching of a large-scale Japanese text collection. Section **5** shows our experimental results using a training data set and a test data set. Section **6** concludes the paper.

2 Task

The Query Term Expansion Subtask was motivated by the question "Which terms should be added to the original query to improve search results?" In other words, this subtask focused on *query term expansion* [24] as an important aspect of query expansion techniques, but did not investigate query term reweighting or query structure construction[1]. This subtask also emphasized more detailed analysis of retrieval effectiveness rather than effectiveness averaged over all the topics specified by the organizers.

The Query Term Expansion Subtask used a 100-gigabyte web document collection and 35 topics. The document collection consisted of 11,038,720 web documents that were gathered from the .jp domain and thus were mostly written in Japanese. This document collection, 'NW100G-01', was the same as that used for the NTCIR Web Retrieval Task conducted from September 2001 to October 2002 ('NTCIR-3 WEB') [8, 7] and for the NTCIR WEB Task conducted from March 2003 to June 2004 ('NTCIR-4 WEB') [6]. The 35 topics used for the Query Term Expansion Subtask were part of the topic set of the NTCIR-4 WEB; however, the relevance judgments were additionally performed by extension of the relevance judgment data of the NTCIR-4 WEB. This means the topic set of the NTCIR-3 WEB could be used for training the system parameters. The NTCIR-3 WEB topic set includes 47 topics. All the topics were written in Japanese. A topic example can be seen in [6, 24].

In the Query Term Expansion Subtask, only the title field in each topic was used to construct the query. The title field gives 1–3 terms that are simulated by the topic creator to be similar to the query terms used in real Web search engines. The definition of the title is different from the one used by the TREC Web Track [3] or the TREC Terabyte Track [2] in the following ways: (i) the terms in the title field are listed in their order of importance for searching, and they are delimited by commas; (ii) each of these terms is supposed to indicate a certain concept, and so it sometimes consists of a single word, but it may also consist of a phrase or a compound word; and (iii) the title field has an attribute, (*i.e.*, 'CASE') that indicates the kind of search strategies and can optionally be used as a Boolean-type operator [6]. These were designed to prevent as far as possible retrieval effectiveness evaluation from being influenced by other effects, such as the performance of Japanese word segmentation, but also to reflect as far as possible the reality of user input queries for current Web search engines. As for the CASE attribute in (iii) above, the retrieval effectiveness of the systems using this optional information should not be compared with other systems that do not use it.

[1] This paper considers these aspects, as well.

3 Retrieval Model and Query Language

3.1 Indri

We used Indri as a search engine platform for our experiments. The retrieval model implemented in Indri combines the language modeling [4] and inference network [22] approaches to information retrieval [13]. This model allows structured queries similar to those used in INQUERY [22] to be evaluated using language modeling estimates within the network. We omit further details of Indri because of space limitations. See [13] for the details.

3.2 Term Dependence Model via Markov Random Fields

Metzler and Croft developed a general, formal framework for modeling term dependencies via Markov random fields [14]. This subsection summarizes this term dependence model. Our proposal for query formulation that we describe in Section **4** is based on this framework but takes into account special features of the Japanese language.

Markov random fields (MRFs), also called undirected graphical models, are commonly used in statistical machine learning to model joint distributions succinctly. In [14], the joint distribution $P_\Lambda(Q, D)$ over queries Q and documents D, parameterized by Λ, was modeled using MRFs, and, for ranking purposes, the posterior $P_\Lambda(D|Q)$ was derived by the following ranking function, assuming a graph G that consists of a document node and query term nodes:

$$P_\Lambda(D|Q) \stackrel{rank}{=} \sum_{c \in C(G)} \lambda_c f(c), \qquad (1)$$

where $Q = q_1 ... q_n$, $C(G)$ is the set of cliques in an MRF graph G, $f(c)$ is some real-valued feature function over clique values, and λ_c is the weight given to that particular feature function.

Full independence ('fi'), *sequential dependence* ('sd'), and *full dependence* ('fd') are assumed as three variants of the MRF model. The full independence variant makes the assumption that query terms are independent of each other. The sequential dependence variant assumes dependence between neighboring query terms. The full dependence variant assumes that all query terms are in some way dependent on each other. To express these assumptions, the following specific ranking function was derived:

$$P_\Lambda(D|Q) \stackrel{rank}{=} \sum_{c\in T} \lambda_T f_T(c) + \sum_{c\in O} \lambda_O f_O(c) + \sum_{c\in O\cup U} \lambda_U f_U(c) \quad (2)$$

where T is defined as the set of 2-cliques involving a query term and a document D, O is the set of cliques containing the document node and two or more query terms that appear contiguously within the query, and U is the set of cliques containing the document node and two or more query terms appearing noncontiguously within the query. Here, the constraint, $\lambda_T+\lambda_O+\lambda_U = 1$ can be imposed. **Table 1** provides a summary of the feature functions and Indri query language expressions, which were proposed in [14].

4 Composing or Decomposing Japanese Compound Words and Phrases

In the phrase-based query construction (hereafter, *phrase expansion*) process for the Japanese language, we make a distinction between compound words containing prefix/suffix words, other compound words, and more general phrases.

4.1 Compound Word Models

Compound words containing prefix/suffix words may only be treated in the same way as single words; otherwise, adding to these, the constituent words qualified by the prefix/suffix words may also be used for query components. At least, the prefix/suffix words themselves should not be used as query components independently, because they do not convey any meaning by themselves. Other compound words and their constituent words may be used for query components, because both the compound words and often their constituent words convey specific meanings by themselves. We assume the following models, sometimes distinguishing the compound words containing the prefix/suffix words and other compound words.

dcmp1 : Decomposes all compound words.

dcmp2 : Decomposes all compound words and removes the prefix/suffix words.

cmp1 : Composes each of all compound words as an exact phrase.

cmp2 : Composes each of the compound words containing the prefix/suffix words as an exact phrase, and each of other compound words as an ordered phrase with at most one term between each constituent word.

pfx1 : Composes each of the compound words containing the prefix/suffix words as an exact phrase, and decomposes other compound words.

pfx2 : Composes each overlapping bi-gram of the constituent words of the compound words containing prefix/suffix words as an exact phrase, and decompounds other compound words.

pfx3 : Combines the 'pfx1' and 'pfx2' and 'dcmp2'.

We mainly assume the 'pfx1' as a basic idea of expressing Japanese compound words in the rest of the paper, since we found some empirical evidences through experiments, as we describe in Section **5.3**. More general phrase models that we describe in the following subsection are mainly grounded in the idea of the 'pfx1' compound model.

4.2 Phrase Models

Phrase-based queries can often filter out noisy, irrelevant documents by reflecting their dependencies into a query using the term dependence model, as described in Section **3.2**. Moreover, in Japanese compound words, including the ones containing prefix/suffix words, the dependencies of each constituent word are stronger than in more general phrases. Therefore, we consider that these term dependencies should be treated as local within the compound words, and as global between the terms that consist of a whole query.

We assume the following dependence models reflecting the ideas that we described above as well as the term dependence model framework that we explained in Section **3.2**.

(1) *lsd* and *lfd*: Assume *local sequential/full dependence*, where the term dependencies are represented as local within each compound words, on the basis of the sequential/full dependence in Section **3.2**, and combines the beliefs (scores) about the resulting feature terms/phrases for each feature of f_T, f_O and f_U in Eq. (2).

The following is an example of Indri query expression, according to the 'lsd', on Topic 0015 in the NTCIR-3 WEB topic set, where the title field was described as the three parts of phrasal descriptions (a noun and two compound nouns in this case), "オゾン層 (ozone layer), オゾンホール (ozone hole), 人体 (human body)" and a morphological analyzer converted this into "オゾン (as a general noun) and 層 (as a suffix noun)," "オゾン (as a general noun) and ホール (as a general noun)," and "人体 (as a general noun)."

#weight(0.8 #combine(オゾン 層 オゾン ホール 人体)
0.1 #combine(#1(オゾン 層) #1(オゾン ホール) 人体)
0.1 #combine(#uw8(オゾン 層) #uw8(オゾン ホール) 人体))

where $\#1(q_i...q_j)$ and $\#\mathrm{uw}N(q_i...q_j)$ indicate the exact phrase expression and the phrase expression where the terms $q_i...q_j$ appear ordered or unordered within a window N terms, respectively.

(2) *lsd+* and *lfd+* : Assume modified models of the 'lsd' and 'lfd', respectively, where (i) for f_T and f_O, each compound word consisting of prefix/suffix word(s) is represented as an exact phrase and treated the same as the other words including the ones decompounded from the other compound words; and (ii) for f_U, the term dependencies are represented as local between the constituent words of general compound words or the exact phrases consisting of prefix/suffix word(s). Here in the f_O, the compound words other than the ones consisting of prefix/suffix words are expressed as the ordered phrases with at most one term between each constituent word.

The following is an example of Indri query expression according to the 'lsd+' on the same topic as shown above.

#weight(0.8 #combine(#1(オゾン 層) オゾン ホール 人体)
0.1 #combine(#1(オゾン 層) #od2(オゾン ホール) 人体)
0.1 #combine(#1(オゾン 層) #uw8(オゾン ホール) 人体))

where $\#\mathrm{od}M(q_i...q_j)$ indicates the phrase expression where the terms $q_i...q_j$ appear ordered, with at most $M-1$ terms between each.

(3) *glsd* and *glfd*: Assume *global and local sequential/full dependence*, where (i) for f_T and f_O, the term dependencies are represented the same as in the 'lsd+' and 'lfd+'; (ii) for f_U the term dependencies are represented, as global in a whole query, by coupling in an unordered phrase expression each combination of the terms including the constituent words of general compound words or the exact phrases consisting of prefix/suffix word(s).

(4) *glsd+* and *glfd+* : Assume alternative global and local sequential/full dependence, where (i) for f_T and f_O, the term dependencies are represented the same as in the 'lsd+' and 'lfd+'; (ii) for f_U, the term dependencies are represented, as global in a whole query, by coupling in an unordered phrase expression each combination of the phrasal descriptions in a query. Here in each unordered phrase expression in the f_U, the compound words containing prefix/suffix words are expressed as exact phrases, and other compound words are decomposed.

The following is an example of Indri query expression according to the 'glsd+'.

#weight(0.8 #combine(#1(オゾン 層) オゾン ホール 人体)
0.1 #combine(#1(オゾン 層) #od2(オゾン ホール) 人体)
0.1 #combine(#uw12(オゾン ホール 人体)
#uw16(#1(オゾン 層) オゾン ホール)))

In Section **5.4**, we investigate the effects of these models that take into account the special features of the Japanese language.

5 Experiments

Our experimental setup is described in Section **5.1**. Using the NTCIR-3 WEB test collection as a training data set, we performed experiments using several types of stopwords, as described in Section **5.2**. We also investigated the effects of our compound word models and our phrase models, and attempted optimization of parameters in these models using the training data set, as described in Sections **5.3** and **5.4**. Moreover, we experimented using pseudo-relevance feedback with the phrase expansion, as described in Section **5.5**. Using the Query Term Expansion Subtask data for testing, we performed the experiments for our official submission as well as further experiments, as described in Section **5.6**.

5.1 Experimental Setup

We used the texts that were extracted from and bundled with the NW100G-01 document collection. In these texts, all the HTML tags, comments, and explicitly declared scripts were removed. We segmented each document into words using the morphological analyzer 'MeCab version 0.81'[2].

Before segmentation, we converted all two-byte characters, numerical characters, and spaces into the corresponding one-byte characters. We only used documents smaller than 20 megabytes for the segmentation, because the rest of the documents are likely to be binary files[3]. Moreover, we only used 1 MB of text data at the head of each document for efficiency. We did not use the part-of-speech (POS) tagging function of the morphological analyzer for the documents, because the POS tagging function requires more time. On completion of the morphological analysis, all Japanese words were separated by a space.

We used Indri to make an index of the web documents in the NW100G-01 document collection, using these segmented texts in the manner described above. We only used one-byte symbol characters as stopwords in the indexing phase, but we used several types of stopwords in the querying phase, as described in Section **5.2**, to enable querying even by phrases consisting of high-frequency words, such as "To be or not to be" in English[4]and to understand the effectiveness of the phrase expansion described in Section **4**. Earlier studies of Japanese text retrieval using word-based indexing often used the terms annotated by some types of POS, such as by 'noun' and 'unknown word'[5], and/or the terms expressed by non-*hiragana* characters for indexing [9]. This policy enables Japanese text retrieval systems to improve efficiency and effectiveness for general purposes, but this is considered to come at the expense

[2]⟨http://www.chasen.org/∼taku/software/mecab/src/mecab-0.81.tar.gz⟩.

[3]The 100-gigabyte document collection, NW100G-01, consists of HTML or plain text files following the 'Content-Type' information in individual HTTP headers and web pages [8, 7]. However, some binary files have 'text/html' or 'text/plain' as the Content-Type by mistake. The aim of this condition is to avoid using such inappropriate documents in the test collection

[4]A similar policy was adopted in earlier experiments using the term dependence model for English language [15, 14].

[5]Some Japanese morphological analyzers mark words that are not matched by a dictionary in morphological analysis with the POS 'unknown word'. Such words are likely to be newly coined, loan words, less frequent proper nouns, or misspelled words.

Table 2. Stopword classes.

	2-byte characters				1-byte characters		
	S_{jsb}	S_{unh}	S_{unk}	S_{bih}	S_{esb}	S_{ewd}	S_{una}
non	0	0	0	0	1	0	0
rlx1	1	1	1	1	1	1	0
rlx2	1	1	1	0	1	1	1
rgd	1	1	1	1	1	1	1

that some phrase queries do not work well because they contain nonindexed words. In Section **5.2**, we investigate an appropriate stopword setting for effective phrase expansion instead.

In the experiments described in the following sections, we only used the terms specified in the title field, as described in Section **2**. We performed morphological analysis using the 'MeCab' tool described at the beginning of this subsection, to segment each of the terms delimited by a comma, and to add POS tags. In this paper, the POS tags are used to specify prefix and suffix words because, in the phrase expansion process, we make a distinction between compound words containing prefix and suffix words and other compound words, as described in Section **4**. We did not use any query structure information provided as the CASE attribute in the title field, as we thought that users of current search engines tend not to use Boolean-type operators, even if a search engine supports them.

5.2 Impacts of Stopwords

We investigated the impacts of several types of stopwords in our experiments. As we did not use part-of-speech (POS) tagging when we performed indexing, we assumed the following classes of stopwords for the Japanese language.

S_{jsb}: Single two-byte characters, not including *hiragana*, *katakana*, or *kanji* characters.

S_{unh}: Single *hiragana* characters.

S_{unk}: Single *katakana* characters.

S_{bih}: All possible pairs of *hiragana* characters.

S_{esb}: Single one-byte characters.

S_{ewd}: 417 English stopwords[6].

S_{una}: Single one-byte alphabetical or numerical characters.

We evaluated the impacts of using some combinations of stopword classes that were likely to be appropriate for use in our further experiments, as shown in **Table 2**. In this table, the left column indicates the names of the stopword settings. Note that we converted all the two-byte alphabets and numerical characters into one-byte characters beforehand, as mentioned in the previous subsection, so some of these could be removed using S_{ewd} or S_{una}, but not by S_{jsb}, in our experimental setting.

In **Table 3**, we summarize the retrieval effectiveness of applying each of the stopword combinations described in Table 2 to the NTCIR-3 WEB test collection.

[6]Those were used in INQUERY [22].

Table 3. Impacts of stopwords using a training data set.

(a) Searching without query expansion				
	AvgPrec	R-Prec	P@10	P@20
rlx2	0.1550	0.1797	0.2021	0.1957
non	0.1545	0.1788	0.2000	0.1947
rgd	0.1543	0.1776	0.1979	0.1947
rlx1	0.1543	0.1776	0.1979	0.1936
(b) Searching with phrase expansion				
	AvgPrec	R-Prec	P@10	P@20
rgd	0.1609	0.1850	0.2000	0.1989
rlx1	0.1608	0.1850	0.1979	0.1989
rlx2	0.1601	0.1871	0.2043	0.2000
non	0.1598	0.1862	0.2085	0.2021
(c) The best result among NTCIR-3 WEB participations				
	AvgPrec	R-Prec	P@10	P@20
	0.1506	0.1707	0.2213	0.1968

In Table 3, the upper table (a) indicates the search results without query expansion, and the middle table (b) indicates the search results with phrase expansion but without pseudo-relevance feedback. In each of these tables, the rows are ranked in order of mean average precision. In this table and hereafter, AvgPrec, R-Prec, P@10, and P@20 refer to mean average precision, R-precision, precision for 10 ranked documents, and precision for 20 ranked documents, respectively. For reference, we put the best results from NTCIR-3 WEB participation [8] in the lower table (c) in Table 3. This shows that our baseline system worked well. For the experiments in Table 3 (b), we tested phrase expansion using the local sequential dependence model ('lsd') that we described in Section **4.2**, with $(\lambda_T, \lambda_O, \lambda_U) = (0.9, 0.1, 0.0)$ that maximized the mean average precision using the 'lsd' model. Note that stopword removal was only applied to the term feature f_T, not to the ordered/unordered phrase features f_O or f_U.

In the experimental results using phrase expansion in Table 3 (b), 'rgd' or 'rlx1' worked modestly better than others according to mean average precision, but the results were sensitive to the evaluation measures. As long as we do not have to specify the stopword setting, we used 'rgd' in the experiments hereafter, including our official submission to the Query Term Expansion Subtask, as described in Section **5.6** because, for pseudo-relevance feedback, which we use in Section **5.5**, using this stopword setting yielded fewer terms that were not related to the topic, such as terms with *hiragana* characters, but provided comparatively stable retrieval effectiveness.

5.3 Effects of Compound Word Models

We investigated the effects of compounding or decompounding of the query terms that were specified as constituent words of a compound word by the morphological analyzer, assuming the models as described in Section **4.1**. The experimental results using the NTCIR-3 WEB topic set are shown in **Table 4**. In this table, 'AvePrec$_a$' indicates the mean average precision over all the 47 topics, and 'AvePrec$_c$' indicates the mean average precision over 23 topics that include compound words in

Table 4. Effects of compounding or decompounding using a training data set.

	AvgPrec	%increase	AvgPrec$_c$	%increase
dcmp1	0.1545	0.0000	0.1589	0.0000
dcmp2	0.1508	-2.4165	0.1513	-4.8012
cmp1	0.1453	-5.9537	0.1401	-11.8294
cmp2	0.1486	-3.8085	0.1469	-7.5671
pfx1	0.1603	3.7589	0.1708	7.4686
pfx2	0.1603	3.7520	0.1708	7.4549
pfx3	0.1604	3.8292	0.1710	7.6081

the title field. '%increase' was calculated on the basis of the 'dcmp1', the result of retrieval by decompounding all compound words, which was also placed in Table 3 (a) as 'non'. In these experiments we didn't use stopword lists.

From the results using the 'cmp1' and 'cmp2', the naive phrase search using the compound words did not work well; however, from the results using the 'pfx1' and 'pfx2', it turned out that compounding the prefix/suffix words and decompounding other compound words worked well. As for the 'pfx3', we combined the 'pfx1', 'pfx2' and 'dcmp2' on the basis of Eq. (1), and optimized the weights for each of these features, changing each weight from 0 to 1 in steps of 0.1. In Table 4, we placed the results using the optimized weights for the features of the 'pfx1', 'pfx2' and 'dcmp2' in $(\lambda_{pfx1}, \lambda_{pfx2}, \lambda_{dcmp2}) = (0.7, 0.3, 0.0)$, which maximized the mean average precision. From the fact that the weight for the feature of the 'dcmp2' was optimized to be 0 and from the experimental result only using the 'dcmp2' shown in Table 4, it tuned out that the constituent words qualified by the prefix/suffix words almost did not contribute to the retrieval effectiveness by themselves without the prefix/suffix words. Moreover, the model combining the 'pfx1', 'pfx2' and 'dcmp2' did not improve the retrieval effectiveness, compared with the 'pfx1' or 'pfx2' alone, in spite of its complexity of the query.

5.4 Effects of Phrase Models

We investigated the effects of phrase expansion using our phrase models that we described in Section **4.2**, which were grounded in the empirical evidences through the experiments shown in the previous subsection, as well as in the theoretical framework explained in Section **3.2**.

Using the NTCIR-3 WEB test collection, we optimized each of the phrase models, changing each weight of λ_T, λ_O and λ_U from 0 to 1 in steps of 0.1, and changing the window size N for the unordered phrase feature as 2, 4, 8, 50 or ∞ times of the number of words that appeared within the window, as in the experiments performed in [14]. Additionally, we used $(\lambda_T, \lambda_O, \lambda_U) = (0.9, 0.05, 0.05)$ for each N value above. The results of the optimization that maximized the mean average precision over all the 47 topics ('AvePrec$_a$') are shown in **Table 5**. This table includes the mean average precision over 23 topics that contain compound words in the title field as 'AvePrec$_c$'. '%increase' was calculated on the basis of the 'base', the result of retrieval not using phrase expansion that was also placed in Table 3 (a) as 'rgd'. Through the optimization, the global and local sequential dependence model 'glsd+' worked best when $(\lambda_T, \lambda_O, \lambda_U, N) = (0.9, 0.05, 0.05, \infty)$. Among the models using the full dependence, 'glfd+' worked best when $(\lambda_T, \lambda_O, \lambda_U, N) = (0.9, 0.05, 0.05, 50)$.

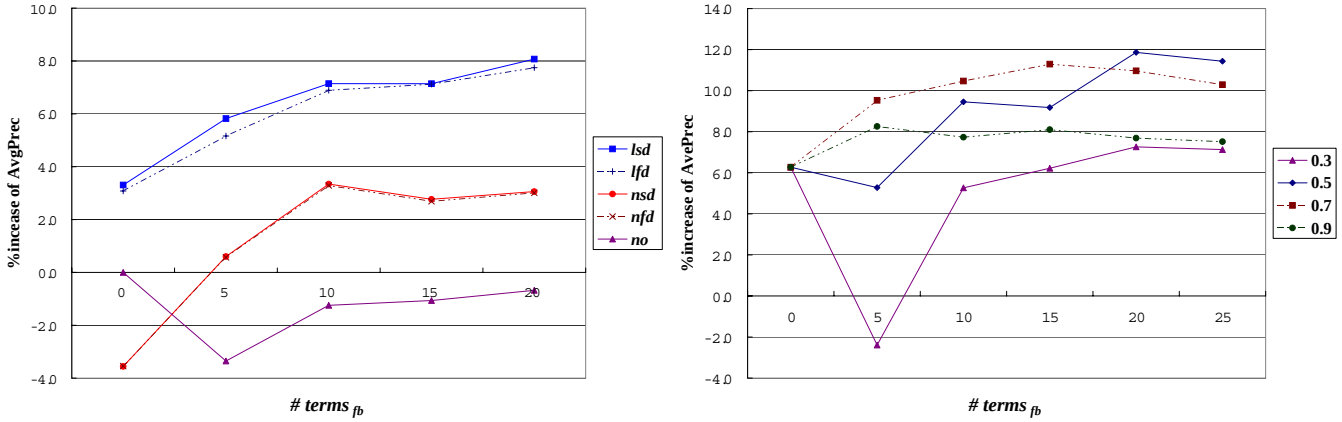
Table 5. Optimization results using a training data set.

	AvgPrec	%increase	AvgPrec$_c$	%increase
base	0.1543	0.0000	0.1584	0.0000
lsd	0.1609	4.3149	0.1720	8.5862
lsd+	0.1624	5.2319	0.1749	10.4111
lfd+	0.1619	4.9120	0.1739	9.7744
glsd	0.1633	5.8346	0.1756	10.8446
glfd	0.1625	5.3230	0.1775	12.0136
glsd+	0.1640	6.2731	0.1776	12.0740
glfd+	0.1626	5.4140	0.1769	11.6788

5.5 Effects of Pseudo-Relevance Feedback with Phrase Expansion

We carried out experiments on the combination of phrase expansion and pseudo-relevance feedback, using the NTCIR-3 WEB topic set, in order to investigate the effectiveness of this combination. Pseudo-relevance feedback was implemented in Indri, based on Lavrenko's relevance model [12]. The results are shown in **Figure 1** (a). In this graph, the horizontal axis indicates the number of feedback terms ($\#terms_{fb}$), and the vertical axis shows the percentage increase in mean average precision compared to the mean average precision in the case not using phrase expansion or pseudo-relevance feedback, the same as the 'base' in Table 5. For comparison, we naively applied either the sequential dependence or the full dependence models[7] to each of the phrasal descriptions delimited by commas in the title field of a topic, and combined the beliefs (scores) about the resulting structure expressions using '#combine' operator of Indri. The explanatory note indicates which results used the naive applications of the sequential dependence model ('nsd') or the full dependence model ('nfd'), the local sequential dependence model ('lsd') or the local full dependence model ('lfd'). Here, the parameters were set as $(\lambda_T, \lambda_O, \lambda_U, N) = (0.9, 0.05, 0.05, 4)$, and the original query weight for pseudo-relevance feedback and the number of feedback documents were set as $weight_{orig} = 0.7$ and $\#docs_{fb} = 10$. In these experiments and hereafter, the feedback weight for pseudo-relevance feedback was set as $(1 - weight_{orig})$. For baseline comparison, we also performed experiments with pseudo-relevance feedback without phrase expansion. As shown in Figure 1 (a), the pseudo-relevance feedback did not work well, directly, for the 100-gigabyte web document collection. A simi-

[7] For this experiment, we used the same tool that was used in the experiments in [14], provided by the authors of this paper.



(a) A preliminary comparison with baseline results.

(b) Using optimized phrase expansion.

Figure 1. Pseudo-relevance feedback with phrase expansion using a training data set.

Table 6. Phrase expansion using a test data set.

	AvgPrec	%increase	AvgPrec$_c$	%increase	AvgPrec$_o$	%increase
non	0.1405	0.0000	0.1141	0.0000	0.1852	0.0000
lsd+	0.1521	8.2979	0.1326	16.2563	0.1852	0.0000
lfd+	0.1521	8.2389	0.1325	16.1407	0.1852	0.0000
glsd+	0.1503	6.9576	0.1313	15.1167	0.1823	-1.5496
glfd+	0.1588	13.0204	0.1400	22.6950	0.1906	2.9330

lar phenomenon was found in [15] for a web document collection in English.

We also experimented using the global and local sequential dependence model 'glsd+' using the optimal parameters, which performed best in the phrase expansion experiments in Table 5, with the pseudo-relevance feedback. For the pseudo-relevance feedback, we used top-ranked 10 documents. The results are shown in **Figure 1** (b). In this figure, the explanatory note indicates the original query weight for pseudo-relevance feedback ($weight_{orig}$). The horizontal axis and the vertical axis are the same as in Figure 1 (a).

5.6 Official Submission and Further Experiments

We participated in the Query Term Expansion Subtask after the first half of the workshop period, long after the other participating groups completed their dry runs[8]. However, we decided to submit some run results that were positioned as baselines for our further experiments. A part of the evaluation results of our official submission can be seen as 'non' in **Table 6**, and in the top three rows in **Table 7**. In these tables, 'AvePrec$_a$', 'AvePrec$_c$' and 'AvePrec$_o$' indicate the mean average precision over all the 35 topics, that over 22 topics that include the compound words in the title field, and that over 13 topics that do not include the compound words, respectively.

As further experiments using the same topics, we performed the phrase expansion using each of the lsd+, lfd+, glsd+ and glfd+ models, which worked well with the optimal parameters as shown in the previous subsection. For evaluation, we used the relevance judgment data that were provided by the organizers of this subtask. The results are shown in Table 6. '%increase' was calculated on the basis of the result of retrieval not using phrase expansion.

Using the same data set, we also carried out experiments on the combination of phrase expansion and pseudo-relevance feedback. For the phrase expansion we used the glsd+ and glfd+ models with the optimal parameters. For the pseudo-relevance feedback, the original query weight ($weight_{orig}$) were set as 0.5 or 0.7, and the number of the terms that were newly added to the query ($\#terms_{ad}$) were set as 10 or 20. Note that the definition of the $\#terms_{ad}$ is different from that of the $\#terms_{fb}$ used in the previous subsection, which was defined as the number of the terms that were used to add to or reweight the original query terms. The definition of the $\#terms_{ad}$ was an indicated condition for the result submission to this subtask. We used top-ranked 10 documents for the feedback ($\#docs_{fb}$).

The results are shown in Table 7. '%increase' was calculated on the basis of the result of retrieval not using pseudo-relevance feedback for each model. These results suggest that, by combining with the phrase expansion, the pseudo-relevance feedback works effectively especially for the queries that include compound words, probably due to the good performance of the initial retrieval, but also works well for the other queries.

6 Conclusions

In this paper we found that (i) our phrase expansion worked well, and (ii) the combination of phrase expansion and pseudo-relevance feedback was more effective than phrase expansion alone, in our experiments using a 100-gigabyte test collection of web documents mostly written in Japanese. Our phrase expansion composes or decomposes compound words and phrases of the Japanese language, taking its special features into account. This paper focused only on short queries in accordance with the NTCIR-5 Query Term Expansion Subtask. Phrase expansion using longer natural language-based queries is a future task.

[8]The author is one of the organizers of the WEB Task at the Fifth NTCIR Workshop; however, the Query Term Expansion Subtask was separately coordinated by another organizer and thus the author was involved in it only as a participant.

Table 7. Pseudo-relevance feedback with phrase expansion using a test data set.

$(weight_{orig}, \#docs_{fb}, \#terms_{ad})$	AvgPrec	%increase	AvgPrec$_c$	%increase	AvgPrec$_o$	%increase
no phrase expansion	0.1405	0.0000	0.1141	0.0000	0.1852	0.0000
(07, 10, 10)	0.1523	8.4403	0.1216	6.5623	0.2044	10.3984
(07, 10, 20)	0.1539	9.5223	0.1231	7.8851	0.2060	11.2293
glsd+ only	0.1503	0.0000	0.1313	0.0000	0.1823	0.0000
(0.7, 10, 10)	0.1662	10.6427	0.1459	11.1069	0.2007	10.0768
(0.7, 10, 20)	0.1665	10.8367	0.1446	10.1170	0.2036	11.7141
(0.5, 10, 10)	0.1676	11.5573	0.1507	14.7307	0.1963	7.6884
(0.5, 10, 20)	0.1695	12.8389	0.1489	13.3843	0.2045	12.1740
glfd+ only	0.1588	0.0000	0.1400	0.0000	0.1906	0.0000
(05, 10, 10)	0.1707	7.4823	0.1520	8.6218	0.2022	6.0661
(05, 10, 20)	0.1768	11.3225	0.1557	11.2587	0.2123	11.4017

Acknowledgments

We thank W. Bruce Croft and Donald Metzler for valuable discussions and comments, and David Fisher for helpful technical assistance with Indri. This work was mainly carried out while the author was visiting the University of Massachusetts, Amherst. This work was supported in part by the Overseas Research Scholars Program and the Grants-in-Aid for Scientific Research (#17680011) from the Ministry of Education, Culture, Sports, Science and Technology, Japan, and in part by the Telecommunications Advancement Foundation, Japan and the Center for Intelligent Information Retrieval.

References

[1] A. Chen and F. C. Gey. Experiments on cross-language and patent retrieval at NTCIR-3 Workshop. In *Proc. of the 3rd NTCIR Workshop*, 2002.

[2] C. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2004 Terabyte Track. In E. M. Voorhees and L. P. Buckland, editors, *Proc. of the 13th Text REtrieval Conference (TREC 2004)*, 2004.

[3] N. Craswell and D. Hawking. Overview of the TREC 2003 Web Track. In E. M. Voorhees and L. P. Buckland, editors, *Proc. of the 12th Text REtrieval Conference (TREC 2003)*, pages 78–92, 2003.

[4] W. B. Croft and J. Lafferty, editors. *Language Modeling for Information Retrieval*. Kluwer Academic Publishers, 2003.

[5] W. B. Croft, H. R. Turtle, and D. D. Lewis. The use of phrases and structured queries in information retrieval. In *Proc. of the 14th Annual International ACM SIGIR Conference*, pages 32–45, Chicago, USA, 1991.

[6] K. Eguchi, K. Oyama, A. Aizawa, and H. Ishikawa. Overview of the Informational Retrieval Task at NTCIR-4 WEB. In *Proc. of the 4th NTCIR Workshop*, 2004.

[7] K. Eguchi, K. Oyama, E. Ishida, N. Kando, and K. Kuriyama. Evaluation methods for web retrieval tasks considering hyperlink structure. *IEICE Transactions on Information and Systems*, E86-D(9):1804–1813, 2003.

[8] K. Eguchi, K. Oyama, E. Ishida, N. Kando, and K. Kuriyama. Overview of the Web Retrieval Task at the Third NTCIR Workshop. In *Proc. of the 3rd NTCIR Workshop*, 2003.

[9] H. Fujii and W. B. Croft. A comparison of indexing techniques for Japanese text retrieval. In *Proc. of the 16th Annual International ACM SIGIR Conference*, pages 237–246, Pittsburgh, USA, 1993.

[10] G. J. F. Jones, T. Sakai, M. Kajiura, and K. Sumita. Experiments in Japanese text retrieval and routing using the NEAT system. In *Proc. of the 21st Annual International ACM SIGIR Conference*, pages 197–205, Melbourne, Australia, 1998.

[11] N. Kummer, C. Womser-Hacker, and N. Kando. Handling orthographic varieties in Japanese IR: Fusion of word-, n-gram-, and yomi-based indices across different document collections. In *Proc. of the 2nd Asia Information Retrieval Symposium*, Jeju Island, Korea, 2005.

[12] V. Lavrenko. *A Generative Theory of Relevance*. PhD thesis, University of Massachusetts, Amherst, 2004.

[13] D. Metzler and W. B. Croft. Combining the language model and inference network approaches to retrieval. *Information Processing and Management*, 40(5):735–750, 2004.

[14] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proc. of the 28th Annual International ACM SIGIR Conference*, pages 472–479, Salvador, Brazil, 2005.

[15] D. Metzler, T. Strohman, H. Turtle, and W. B. Croft. Indri at TREC 2004: Terabyte Track. In *Proc. of the 13th Text Retrieval Conference (TREC 2004)*, 2004.

[16] G. Mishne and M. de Rijke. Boosting web retrieval through query operations. In *Proc. of the 27th European Conference on Information Retrieval Research*, pages 502–516, Santiago de Compostela, Spain, 2005.

[17] I. Moulinier, H. Molina-Salgado, and P. Jackson. Thomson Legal and Regulatory at NTCIR-3: Japanese, Chinese and English retrieval experiments. In *Proc. of the 3rd NTCIR Workshop*, 2002.

[18] Y. Ogawa. Effective and efficient document ranking without using a large lexicon. In *Proc. of 22th International Conference on Very Large Data Bases*, pages 192–202, Bombay, India, 1996.

[19] Y. Ogawa and H. Mano. RICOH at NTCIR-2. In *Proc. of the 2nd NTCIR Workshop*, 2001.

[20] Y. Ogawa and T. Matsuda. Overlapping statistical word indexing: A new indexing method for Japanese text. In *Proc. of the 20th Annual International ACM SIGIR Conference*, pages 226–234, Philadelphia, USA, 1997.

[21] M. Toyoda, M. Kitsuregawa, H. Mano, H. Itoh, and Y. Ogawa. University of Tokyo/RICOH at NTCIR-3 Web Retrieval Task. In *Proc. of the 3rd NTCIR Workshop*, 2002.

[22] H. R. Turtle and W. B. Croft. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3):187–222, 1991.

[23] P. Vines and R. Wilkinson. Experiments with Japanese text retrieval using mg. In *Proc. of the 1st NTCIR Workshop*, 1999.

[24] M. Yoshioka. Overview of the NTCIR-5 WEB Query Expansion Task. In *Proc. of the 5th NTCIR Workshop*, 2005 (to appear).