Noname manuscript No.
(will be inserted by the editor)

Query Structuring and Expansion with Two-stage Term Dependence for Japanese Web Retrieval

Koji Eguchi · W. Bruce Croft

Received: date / Accepted: date

Abstract In this paper, we propose a new term dependence model for information retrieval, which is based on a theoretical framework using Markov random fields. We assume two types of dependencies of terms given in a query: (i) long-range dependencies that may appear for instance within a passage or a sentence in a target document, and (ii) short-range dependencies that may appear for instance within a compound word in a target document. Based on this assumption, our two-stage term dependence model captures both long-range and short-range term dependencies differently, when more than one compound word appear in a query. We also investigate how query structuring with term dependence can improve the performance of query expansion using a relevance model. The relevance model is constructed using the retrieval results of the structured query with term dependence to expand the query. We show that our term dependence model works well, particularly when using query structuring with compound words, through experiments using a 100-gigabyte test collection of web documents mostly written in Japanese. We also show that the performance of the relevance model can be significantly improved by using the structured query with our term dependence model.

Keywords Japanese web retrieval · Term dependence · Structured queries

1 Introduction

The structured query approach has been used to include more meaningful phrases in a proximity search query to improve retrieval effectiveness (Croft et al, 1991; Metzler

Koji Eguchi
Department of Computer Science and Systems Engineering
Kobe University
1-1 Rokkodai, Nada-ku, Kobe 657-8501, Japan
E-mail: eguchi@port.kobe-u.ac.jp

W. Bruce Croft
Department of Computer Science
University of Massachusetts, Amherst
Amherst, MA 01003-9264, USA

and Croft, 2005). Phrase-based queries are known to perform effectively, especially with large-scale collections such as the Web (Mishne and de Rijke, 2005; Metzler and Croft, 2005). This is caused by the fact that larger collections are in general noisier while they contain more information, and the fact that phrase-based structured queries can filter out many of such noisy documents. These methods can capture term dependencies that appear in a query. However, they did not distinguish different types of term dependencies such as: (i) long-range dependencies that may appear for instance within a passage or a sentence in a target document, and (ii) short-range dependencies that may appear for instance within a compound word in a target document. Capturing this kind of complex dependencies should be promising for any language; however, we believe it more promising especially for some languages, for instance Japanese, in which individual words are frequently composed into a long compound word and the formation of an endless variety of compound words is allowed.

Another technique, pseudo-relevance feedback, has been commonly used to address important notions of synonymy and polysemy in information retrieval (Buckley et al, 1994; Xu and Croft, 1996; Lavrenko and Croft, 2001; Zhai and Lafferty, 2001). It can significantly improve ad hoc retrieval results by expanding a query, assuming that all top-ranked documents retrieved in response to the query are relevant. Although pseudo-relevance feedback generally improves effectiveness by capturing the context of query terms in documents, it can occasionally add terms to the query that are not helpful. Recently, pseudo-relevance feedback was used to estimate multinomial models that were representative of user's interests, within the framework of probabilistic language models (Lavrenko and Croft, 2001; Zhai and Lafferty, 2001). Following Lavrenko and Croft (2001), we will use the term *relevance models* to describe such models. The combination of phrase-based query structuring and query expansion via the relevance model is promising, because each model has its own advantages: phrase-based query structuring can capture dependencies between query terms and the relevance model can handle mismatched vocabulary.

In this paper, we use the structured query approach using word-based units to capture compound words, as well as more general phrases, in a query. Our approach is based on a theoretical framework using Markov random fields (Metzler and Croft, 2005). Our work is the first attempt, to the best of our knowledge, to explicitly capture both long-range and short-range term dependencies for information retrieval. We further investigate how query structuring with term dependence can improve the performance of query expansion via a relevance model. Our experiments were performed using the 100-gigabyte web test collections that were developed in the NTCIR Workshop Web Task (Eguchi et al, 2003, 2004; Yoshioka, 2005) and mostly written in Japanese. This study is also the first attempt to thoroughly examine term dependencies in the Japanese language to formulate structured queries for web information retrieval.

The rest of this paper is structured as follows. Section **2** discusses problems of various kinds of term dependencies that appear in Japanese information retrieval, and research efforts to address the problems. Section **3** introduces the retrieval model that we use and the term dependence model using Markov random fields, which gives a theoretical framework to our investigation. Section **4** describes phrase-based query structuring with our two-stage term dependence model. Section **5** describes query expansion via a relevance model. Section **6** explains the test collections we use in this paper, and our experimental results. Section **7** concludes the paper.

2 Problem Statement and Related Research Efforts for Japanese Information Retrieval

We assume two types of dependencies of terms in a query: (i) long-range dependencies that may appear, for instance, within a passage or a sentence in a target document, and (ii) short-range dependencies that may appear, for instance, within a compound word in a target document. We believe that this assumption is realistic in any language; however, handling compound words may be different for different languages. Some languages, such as English, favor *open* compound words, in which multiple words are separated by spaces. In these languages, we must detect compound word boundaries; however, once a compound word is specified, the constituents of the compound word can be tokenized simply by the spaces. Some other languages usually use *closed* compound words, which are expressed without explicit word separators; typical examples are German, Swedish, Danish and Finnish. In these languages, we must find the constituents of a compound word, but detecting compound words should not be hard. In some East Asian languages that use ideograms, such as Japanese and Chinese, we must both segment individual words and detect compound words.

In this paper, we develop, in Section **4**, a *two-stage* term dependence model that captures long-range and short-range dependencies differently. We assume that: (i) global dependencies occur between query components that are explicitly delimited by separators in a query; and (ii) local dependencies occur between constituents within a compound word when the compound word is specified in a query component. These correspond to long-range and short-range dependencies, respectively. We experiment, in Section **6**, using a large-scale web document collection mostly written in Japanese; however, the two-stage term dependence model may be reasonable for other languages, if compound words and their constituents can be specified in a query, as mentioned above.

2.1 Problems in Processing the Japanese Language for Query Formulation

First, we consider what language units are appropriate for a query formulation process for the Japanese language. One question is, for example, how a compound noun with prefix or suffix words, such as “*ozon-sō*” (ozone layer), should be represented as a query. Possible ways are as follows.

(1)The compound noun should be used as it is.

(2)The compound noun should be decomposed into primitive words, such as “*ozon*” (ozone) and “*sō*” (layer).

(3)The suffix and prefix should be removed from the compound noun, but the dominant constituent word, such as “*ozon*” (ozone), should be kept.

(4)Adding to the original compound noun, the dominant constituent word should also be used, in the case of the example, “*ozon-sō*” (ozone layer) and “*ozon*” (ozone) are used.

Another question is whether or not other compound words, not containing a prefix or suffix, should be decomposed into their constituents, for example, whether a loan word expressed in *katakana* characters, “*ozon-hōru*” (ozone hole), should be decomposed into “*ozon*” (ozone) and “*hōru*” (hole) or not; and whether a general compound noun, “*kabushiki-tōshi*” (stock investment), should be decomposed into “*kabushiki*” (stock) and “*tōshi*” (investment) or not. Note that these compound words are expressed without a separator in the Japanese language when using the original Japanese characters, but not using the Latin alphabet as above. Thus, query terms input by a user are often expressed as such compound words, in which constituent words are not delimited by a separator. Possible ways are as follows.

(1)The compound word should be used as it is.

(2)The compound word should be decomposed into primitive words.

(3)Adding to the original compound word, the primitive constituent words should also be used.

We will discuss these issues and empirically specify appropriate language units (hereafter, *compound word models*) for the query structuring with two-stage term dependence in Section 4.1.

Furthermore, web search engines usually support query expression with a delimiter, even for users who use the Japanese language, so that a user can express multiple concepts that should reflect the user's information needs in a query, and the search engine can then construct a complex query. Using this function, the user can input a query consisting of multiple components, each of which is expressed as a compound word or a single word, such as "*ozon-sō ozon-hōru jintai*" (three components of 'ozone layer', 'ozone hole' and 'human body' with a space delimiter between each component), to search for the effects of destruction of the ozone layer and expansion of the ozone hole on the human body. This kind of query requires more global dependence between query components, such as between "*ozon-sō*" (ozone layer), "*ozon-hōru*" (ozone hole) and "*jintai*" (human body), and more local dependence between constituents within each compound word, in this case "*ozon*" (ozone) and "*sō*" (layer), or "*ozon*" (ozone) and "*hōru*" (hole). Note that this local dependence is tighter than global dependence, in general.

We will discuss the issue of how to formulate the query, taking into account both global dependence and local dependence, and develop appropriate models for this purpose, mainly in Section 4.2.

2.2 Research Efforts for Japanese Information Retrieval – Focusing compound words and segmentation

Japanese text retrieval is required to handle several types of problems specific to the Japanese language, such as compound words and segmentation (Fujii and Croft, 1993). 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 character unigram-based indexing and word-based indexing, and found their retrieval effectiveness comparable, especially when applied to text using *kanji* characters[1] (Fujii and Croft, 1993). Following this work, many researchers have 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 (Fujii and Croft, 1993; Jones et al, 1998; Chen and Gey, 2002; Moulinier et al, 2002). The focus of these studies was rather on how to

[1] The Japanese language is mainly expressed in *kanji*, *hiragana* and *katakana* characters. *Kanji* is derived from ancient Chinese characters. English alphabetic words are also sometimes used in a Japanese text, especially as proper nouns.

improve efficiency while maintaining effectiveness in retrieval. Some other researchers (i) used both phrases and their constituent words as well as individual words as an index (Ogawa and Matsuda, 1997; Kando et al, 1998); or (ii) made use of supplemental phrase-based indexing in addition to word-based indexing[2](Fujita, 1999), where phrase detection on targeted documents is required in advance. However, we believe this kind of approaches is not appropriate for the languages, for instance Japanese, in which individual words are frequently composed into a long compound word and the formation of an endless variety of compound words is allowed.

Meanwhile, the structured query approach does not require phrase detection on targeted documents in advance of searching (Croft et al, 1991; Metzler and Croft, 2005). A few researchers have investigated this approach to retrieval for Japanese newspaper articles (Fujii and Croft, 1993; Moulinier et al, 2002); however, they emphasized formulating a query using character n-grams and showed that this approach performed comparably in retrieval effectiveness with the word-based approach. We are not aware of any studies that have used structured queries to formulate queries reflecting Japanese compound words or phrases appropriately. We also have not seen any studies that used structured queries to effectively retrieve web documents written in Japanese. In this paper, we use the structured query approach using word-based units to capture, in a query, both term dependencies within a compound word and more general term dependencies.

3 Retrieval Model and Query Language

3.1 Retrieval Model

Indri is a search engine platform that can handle large-scale document collections efficiently and effectively (Metzler and Croft, 2004; Strohman et al, 2005). The retrieval model implemented in Indri combines the language modeling (Croft and Lafferty, 2003) and inference network (Turtle and Croft, 1991) approaches to information retrieval. This model allows structured queries similar to those used in *InQuery* (Turtle and Croft, 1991) to be evaluated using language modeling estimates within the network. Some of the query language operators supported in Indri are shown in **Table 1**, where the estimate of a document with respect to a query operator is referred to as a *belief.* Because we focus on query formulation rather than retrieval models, we use Indri as a baseline platform for our experiments. The efficiency of Indri operators is discussed in Strohman et al (2005). Our approach described in Section **4** is not limited to the platform of Indri, but can be implemented in any system where ordered and unordered phrase operators, such as those in **Table 1**, are workable. How indexes for ordered/unordered phrase operations can be implemented efficiently in any system is discussed in Strohman (2007). For example, most web search engines already index high-order n-grams, and our methods can be implemented with that type of index.

3.2 Term Dependence Model via Markov Random Fields

Metzler and Croft (2005) developed a general, formal framework for modeling term dependencies via Markov random fields, and showed that the model is very effective in a variety of retrieval situations using the Indri platform. This section summarizes

[2] This kind of approaches was also employed for English (e.g., Mitra et al (1997)).

Table 1 Indri query language.

Operator	Name	Description
$\#\mathtt{uw}N(\cdot)$	Unordered Phrase	Matches unordered text in which the terms appear unordered within a window of N terms
$\#\mathtt{od}M(\cdot)$	Ordered Phrase	Matches ordered text in which the terms appear ordered, with at most $(M-1)$ terms between each
$\#M(\cdot)$	Ordered Phrase	same as $\#\mathtt{od}M(\cdot)$
$\#\mathtt{combine}(q_1\ q_2 \cdots)$	Combine operator	Combines beliefs from other operators to form a single score for a document
$\#\mathtt{weight}(w_1 q_1\ w_2 q_2 \cdots)$	Weight operator	Combines beliefs from other operators to form a single score for a document, using weights to indicate which operators should be trusted most

this term dependence model. Markov random fields (MRFs), also called undirected graphical models, are commonly used in statistical machine learning to model joint distributions succinctly. In Metzler and Croft (2005), 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:

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

where $Q = t_1 ... t_n$, $C(G)$ 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. The sign '$\overset{\text{rank}}{=}$' indicates that the ranking of documents according to the left-hand side is equivalent to that according to the right-hand side.

Full independence[3] (*fi*), *sequential dependence* (*sd*), and *full dependence* (*fd*) are assumed as three variants of the MRF model. **Fig. 1** shows graphical model representation of each. The full-independence variant makes the assumption that query terms are independent of each other. The sequential dependence variant assumes dependence between query terms that appear contiguously, while 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:

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

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 2 provides a summary of the feature functions and Indri query language expressions proposed in Metzler and Croft (2005). In this table, $\#\mathtt{1}(\cdot)$ indicates exact phrase expressions. $\#\mathtt{uw}N(\cdot)$ is described in **Table 1**. Let us take an example query that consists of three words: "term dependence models". The following are Indri query expressions for this example according to the *sd* and *fd* models, respectively, each of which

[3] This is also referred to as the *term independence model* hereafter.

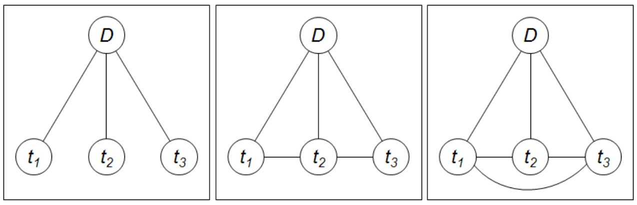

Fig. 1 Example Markov random field model for three query terms under various dependence assumptions: (left) full independence, (middle) sequential dependence, and (right) full dependence (Metzler and Croft, 2005).

Table 2 Metzler and Croft's Feature functions and the corresponding Indri query language.

Feature	Type	Indri Expression
$f_T(t_i, D)$	Term	t_i
$f_O(t_i, t_{i+1}, ..., t_{i+k}, D)$	Ordered Phrase	$\#\texttt{1}(t_i\ t_{i+1}\ ...\ t_{i+k})$
$f_U(t_i, ..., t_j, D)$	Unordered Phrase	$\#\texttt{uw}N(t_i\ ...\ t_j)$

is formulated in the form of "#weight (λ_T #combine($\cdots f_T^i \cdots$) λ_O #combine($\cdots f_O^i \cdots$) λ_U #combine($\cdots f_U^i \cdots$))":

$$
\begin{array}{lll}
\#\texttt{weight}(& \lambda_T\ \#\texttt{combine}(& \text{term dependence models }) \\
 & \lambda_O\ \#\texttt{combine}(& \#\texttt{1}(\text{ dependence models }) \\
 & & \#\texttt{1}(\text{ term dependence })\ ) \\
 & \lambda_U\ \#\texttt{combine}(& \#\texttt{uw}N_2(\text{ dependence models }) \\
 & & \#\texttt{uw}N_2(\text{ term dependence })\ )\ )
\end{array}
$$

$$
\begin{array}{lll}
\#\texttt{weight}(& \lambda_T\ \#\texttt{combine}(& \text{term dependence models }) \\
 & \lambda_O\ \#\texttt{combine}(& \#\texttt{1}(\text{ dependence models }) \\
 & & \#\texttt{1}(\text{ term dependence }) \\
 & & \#\texttt{1}(\text{ term dependence models })\ ) \\
 & \lambda_U\ \#\texttt{combine}(& \#\texttt{uw}N_2(\text{ dependence models }) \\
 & & \#\texttt{uw}N_2(\text{ term models }) \\
 & & \#\texttt{uw}N_2(\text{ term dependence }) \\
 & & \#\texttt{uw}N_3(\text{ term dependence models })\ )\ )
\end{array}
$$

where $\#\texttt{uw}N_\ell(\cdot)$ indicates phrase expressions in which the specified terms appear unordered within a window of N_ℓ terms, and N_ℓ is given by $(N_1 \times \ell)$ when ℓ terms appear in the window. The window-size parameter N_1 is determined empirically.

4 Query Structuring with Two-stage Term Dependence

In order to process the Japanese language or some other East Asian languages that use ideograms, we must know what language units are appropriate for phrase-based query structuring. We will discuss this issue in Section 4.1. We will also discuss, in Section 4.2, the issue of how to formulate a query, taking into account both the global dependence between query components that are separated by delimiters, and the local dependence between constituents of a compound word when the compound word is specified in a query component.

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 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 each prefix/suffix word usually expresses a specific concept only by being concatenated with the following or preceding word. Other compound words, that do not contain prefix/suffix words, may be used together with constituent words for query components, because both the compound words and often their constituent words convey specific meanings by themselves.

We define the following models, sometimes distinguishing compound words containing the prefix/suffix words from other compound words.

(1) *dcmp1*: Decomposes all compound words.
e.g., "*ozon-sō*" (in English, 'ozone layer') is decomposed into "*ozon*" and "*sō*".

(2) *dcmp2*: Decomposes all compound words and removes the prefix/suffix words.
e.g., "*ozon-sō*", in which "*sō*" is a suffix noun, is converted into "*ozon*".

(3) *cmp1*: Composes each compound word as an exact phrase.
e.g., "*ozon-sō*" is used as an exact phrase ("#1(*ozon sō*)" in the Indri query language, as shown in **Table 1**).

(4) *cmp2*: Composes each compound word that contains prefix/suffix words as an exact phrase, and each other compound word as an ordered phrase with at most one term between each constituent word.
e.g., "*ozon-sō*" that contains a suffix word and "*ozon-hōru*" (in English, 'ozone hole') that does not contain prefix/suffix words are expressed as phrases in different manners. In the Indri query language, the former is expressed as "#1(*ozon sō*)", the same as in *cmp1*; and the latter is expressed as "#od2(*ozon hōru*)".

(5) *pfx1*: Composes each of the compound words containing prefix/suffix words as an exact phrase, and decomposes other compound words.
e.g., "*ozon-sō*" is used as an exact phrase, the same as in *cmp1*; on the other hand, "*ozon-hōru*" is decomposed into "*ozon*" and "*hōru*".

(6) *pfx2*: Composes each overlapping word-based bigram of the constituent words of the compound words containing prefix/suffix words as an exact phrase, and decomposes other compound words.
e.g., "*dai-kyū-jō*" (in English, 'article nine' such as of the Constitution) in which "*dai*" and "*jō*" are prefix and suffix words, respectively, is decomposed into a couple of exact phrases, "*dai-kyū*" and "*kyū-jō*" ("#1(*dai kyū*)" and "#1(*kyū jō*)", respectively, in the Indri query language).

(7) *pfx3*: Linearly combines *pfx1*, *pfx2* and *dcmp2*.

The combined model *pfx3* was defined to investigate how each of the three component models contributes to retrieval effectiveness by changing weights for the component models. We discuss the details in Section *6.3.1*.

We mainly assume *pfx1* as the basic technique for expressing Japanese compound words in the rest of the paper, because we found some empirical evidence through experiments to support its use, as we describe in Section *6.3.1*. More general term dependence models that we describe in Section 4.2 are grounded, in part, in the idea of the *pfx1* compound word model.

4.2 Two-stage Term Dependence Model

In compound words that often appear for instance in Japanese, the dependencies of each constituent word are tighter than more general term dependencies. Therefore, we consider that these term dependencies should be treated as global between query components that make up a whole query and as local within a compound word when the compound word appears in a query component. Metzler and Croft's term dependence model, which we summarized in Section 3.2, gives a theoretical framework for this study, but must be enhanced when we consider more complex dependencies as mentioned above. We propose *two-stage term dependence model* that captures term dependencies both between query components in a query and between constituents within a compound word. To achieve the model mentioned above, we extend the term dependence model given in Eq. (2), on the basis of Eq. (1), as follows:

$$P_\Lambda(D|Q) \overset{\text{rank}}{=} \sum_{c_q \in T(Q)} \lambda_T f_T(c_q) + \sum_{c_q \in O(Q)} \lambda_O f_O(c_q) + \sum_{c_q \in O(Q) \cup U(Q)} \lambda_U f_U(c_q) \quad (3)$$

where

$$\begin{aligned} f_T(c_q) &= f_T^*\left(\{c_t\}_{c_t \in T(q_k), q_k \in c_q}\right) \\ f_O(c_q) &= f_O^*\left(\{c_t\}_{c_t \in O(q_k), q_k \in c_q}\right) \\ f_U(c_q) &= f_U^*\left(\{c_t\}_{c_t \in O(q_k) \cup U(q_k), q_k \in c_q}\right) \quad . \end{aligned} \quad (4)$$

Here, Q consists of query components $q_1 \cdots q_k \cdots q_m$, and each query component consists of individual terms $t_1 \cdots t_n$. $T(Q)$, $O(Q)$ and $U(Q)$ express the clique sets with (global) dependence between query components consisting of a whole query. $T(Q)$ is defined as the set of 2-cliques involving a query component and a document D, $O(Q)$ is the set of cliques containing the document node and two or more query components that appear contiguously within the whole query, and $U(Q)$ is the set of cliques containing the document node and two or more query components appearing noncontiguously within the whole query. Moreover, $T(q_k)$, $O(q_k)$ and $U(q_k)$ express the clique sets with (local) dependence between individual terms consisting of a query component q_k, and can be defined similarly as $T(Q)$, $O(Q)$ and $U(Q)$. We then define the feature functions f_T^*, f_O^* and f_U^* so that some treatments on Japanese compound words are reflected, as we will describe later in this section, and that the global dependence between query components are reflected as well. Hereafter, we assumed that the constraint $\lambda_T + \lambda_O + \lambda_U = 1$ was imposed independently of the query. When Q consists of two or more query components and each of which has one word, Eq. (3) is equivalent to Eq. (2). The model given by Eq. (2) can be referred to as *single-stage term dependence model.* When we ignore the dependencies between query components in f_O^* and f_U^*, Eq. (3) represents dependencies only between constituent terms within each query component, which can be referred to as *local term dependence model*; otherwise, Eq. (3) expresses the *two-stage term dependence model.*

According to Eq. (3), we assumed the following instances, considering special features of the Japanese language.

Two-stage term dependence models

(1) *glsd*$^+$ expresses the dependencies on the basis of the *sequential dependence* (see Section 3.2) both between query components and between constituent terms within a query component, assuming dependence between neighboring elements.

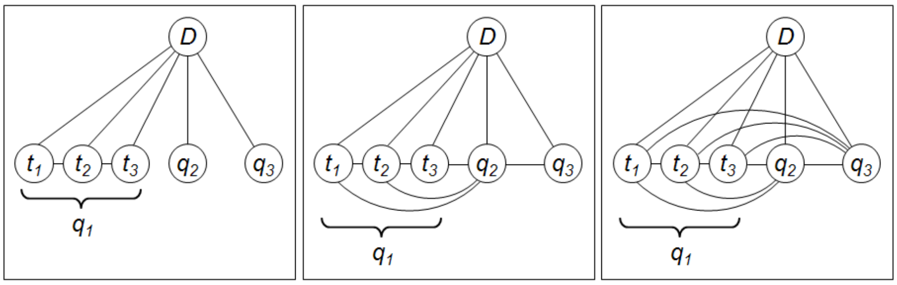

Fig. 2 Example graphical models of two-stage term dependence model with three query components (q_1, q_2 and q_3), one of which consists of a compound word with three constituent terms (t_1, t_2 and t_3), under three assumptions corresponding to *lsd*$^+$ (left), *glsd*$^+$ (middle), and *glfd*$^+$ (right).

The beliefs for the resulting feature terms/phrases for each of f^*_T, f^*_O and f^*_U are combined as in Eq. (3). A graphical model representation of this model is shown in the middle of **Fig. 2**.

(2) *glfd*$^+$ expresses the dependencies between query components on the basis of the *full dependence* (see Section 3.2), assuming all the query components are in some way dependent on each other. It expresses the dependencies between constituent terms within a query component on the basis of the sequential dependence, even in this model. A graphical model representation of this model is shown in the right of **Fig. 2**.

Here in f^*_T, f^*_O and f^*_U, each compound word containing prefix/suffix words is represented as an exact phrase and treated the same as the other words, on the basis of the empirical results reported in Section *6.3.1.* Moreover, in f^*_O, each general compound word (not containing prefix/suffix words) is expressed as an ordered phrase with at most one term between each constituent word. Let us take an example from the NTCIR-3 WEB topic set (Eguchi et al, 2003), which is written in Japanese. The title field of Topic 0015, as shown in **Fig. 3**, was described as three query components, "*ozon-sō*, *ozon-hōru*, *jintai*" (which mean 'ozone layer', 'ozone hole' and 'human body'). A morphological analyzer converted this to "*ozon*" ('ozone' as a general noun) and "*sō*" ('layer' as a suffix noun), "*ozon*" ('ozone' as a general noun) and "*hōru*" ('hole' as a general noun), and "*jintai*" ('human body' as a general noun). The following are Indri query expressions for this example according to the *glsd*$^+$ and *glfd*$^+$ models, respectively, in the form used in Section 3.2.

```
#weight( $\lambda_T$ #combine( #1( ozon sō ) ozon hōru jintai )
         $\lambda_O$ #combine( #1( ozon sō ) #od2( ozon hōru ) jintai )
         $\lambda_U$ #combine( #uw$N_4$( #1( ozon sō ) ozon hōru )
                              #uw$N_3$( ozon hōru jintai ) ) )

#weight( $\lambda_T$ #combine( #1( ozon sō ) ozon hōru jintai )
         $\lambda_O$ #combine( #1( ozon sō ) #od2( ozon hōru ) jintai )
         $\lambda_U$ #combine( #uw$N_4$( #1( ozon sō ) ozon hōru )
                              #uw$N_3$( ozon hōru jintai )
                              #uw$N_3$( #1( ozon sō ) jintai )
                              #uw$N_5$( #1( ozon sō ) ozon hōru jintai ) ) )
```

Local term dependence models

⟨TOPIC⟩
⟨NUM⟩0015⟨/NUM⟩
⟨TITLE CASE="c" RELAT="1-2"⟩ *ozon-sō*, *ozon-hōru*, *jintai* ⟨/TITLE⟩
. . .
⟨/TOPIC⟩

(a) An extract transliterated in the Latin alphabet from an original sample topic.

⟨TOPIC⟩
⟨NUM⟩0015⟨/NUM⟩
⟨TITLE CASE="c" RELAT="1-2"⟩ozone layer, ozone hole, human body⟨/TITLE⟩
⟨DESC⟩I want to learn about the effects destruction of the ozone layer and expansion of the ozone hole have on the human body⟨/DESC⟩
. . .
⟨/TOPIC⟩

(b) An extract translated in English from the above sample topic.

Fig. 3 A sample topic and its English translation.

(3) *lsd*$^+$ indicates the model obtained by ignoring the dependencies between query components in *glsd*$^+$. A graphical model representation of this model is shown in the left of **Fig. 2**.

(4) *lfd*$^+$ indicates the model obtained by ignoring the dependencies between query components and applying the *fd* model to constituent terms within each query component.

In these models, each compound word containing prefix/suffix words is represented as an exact phrase and treated the same as the other words, in the same manner as the cases of *glsd*$^+$ and *glfd*$^+$. The following is an example of an Indri query expression according to *lsd*$^+$ on Topic 0015.

#weight(λ_T #combine(#1(*ozon sō*) *ozon hōru jintai*)
λ_O #combine(#1(*ozon sō*) #od2(*ozon hōru*) *jintai*)
λ_U #combine(#1(*ozon sō*) #uwN_2(*ozon hōru*) *jintai*))

Our two-stage term dependence models are based on the framework of Markov Random Field model from the following view. When we suppose an MRF graph consisting of a document as a root node and all decomposed query terms as leaf nodes, ignoring query components, the set of all cliques that contain the document node and multiple query terms that appear contiguously can be enumerated, and then the following limitations can be applied to the clique set. We assume that (1) individual word-to-word dependencies across different query components (e.g., dependence between *hōru*" and "*jintai*" in the case of Topic 0015) are disregarded; however, (2) only word-to-word dependencies within each query component (e.g., dependence between "*ozon*" and "*sō*" or between "*ozon*" and "*hōru*") and (3) component-to-component dependencies (e.g., dependence among all elements of two or more query components, for instance, among "*ozon*", "*hōru*" and "*jintai*", are kept and the corresponding features are considered as in Eq. (3). When simply applying Metzler and Croft's single-stage term dependence model, which was reviewed in Section 3.2, to all decomposed query terms in a whole query, not only the meaningless dependencies as mentioned in (1) are involved, but it is also obvious that the number of combinations of the query terms (i.e., the number of cliques) exponentially increases. Our two-stage term dependence models correspond to considering both (2) and (3), and our local term dependence models correspond to only considering (2), both of which are expected to improve retrieval effectiveness with reasonable efficiency. In Section *6.3.2*, we investigate the effects of the models defined in this section.

5 Query Expansion via Relevance Models

Lavrenko and Croft (2001) formulated relevance models that explicitly incorporated relevance into the language modeling. Metzler et al (2004) modified the relevance models as a pseudo-relevance feedback function in the framework of inference network-based retrieval models. In this paper, we follow this method of pseudo-relevance feedback, as briefly described below.

Given an initial query Q_{st}, we retrieve a set of $\#docs_{fb}$ documents and form a relevance model from them. We then form Q_{rm} by wrapping the #`combine` operator of Indri, around the most likely $\#terms_{fb}$ terms from the relevance model that are not stopwords. Finally, an expanded query is formed that has the following form:

$$Q_{new} = \#\texttt{weight}(\nu Q_{st} \quad (1.0 - \nu) Q_{rm}) \tag{5}$$

where #`weight` indicates an Indri operator as described in **Table 1**. The parameter ν controls a balance between the original query Q_{st} and expanded query Q_{rm}. In this paper, we formulate Q_{st} using the two-stage term dependence models, instead of using the term independence model, and reformulate Q_{new} using the relevance model-based pseudo-relevance feedback as above.

6 Experiments

An overview of the test collections we use is given in Section 6.1. Our experimental setup is described in Section 6.2. Using the NTCIR-3 WEB test collection as a training data set, we investigated the effects of the compound word models and the two-stage term dependence model, and attempted to optimize the parameters in these models using the training data set, as described in Sections *6.3.1* and *6.3.2*, respectively. Using the NTCIR-5 WEB Task data, we performed the experiments with two-stage term dependence model for testing, as described in Section *6.3.3*. Moreover, we experimented using pseudo-relevance feedback with the two-stage term dependence model over the training and testing data sets, as described in Sections *6.4.1* and *6.4.2*, respectively.

6.1 Data

We used a 100-gigabyte web document collection for experiments. The document collection consisted of web documents 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-3 Web Retrieval Task (‘NTCIR-3 WEB’) (Eguchi et al, 2003), for the NTCIR-4 Web Task (‘NTCIR-4 WEB’) (Eguchi et al, 2004), and for the NTCIR-5 Web Task[4](‘NTCIR-5 WEB’) (Yoshioka, 2005).

We used the topics and the relevance judgment data of the NTCIR-3 WEB for training the system parameters.[5] We used the topics and the relevance judgment data

[4] Query Term Expansion Subtask.

[5] For the training, we used the relevance judgment data based on the *page-unit document model* (Eguchi et al, 2003) included in the NTCIR-3 WEB test collection.

Table 3 Test collections used.

collection size	ca. 100 gigabytes
# of documents	11,038,720
# of NTCIR-3 WEB topics (for training)	47
# of NTCIR-5 WEB topics (for testing)	35

Table 4 Proportion of languages in a Web document collection NW100G-01.

Language	Proportion
Japanese	90. %
English	8.3 %
Simp. Chinese	0.05%
Korean	0.03%
Trad. Chinese	0.02%
West European	0.01%
Other Languages *	0.01%
No Text Content	0.78%
Not Identified	0.02%

(*) Russian, East European, Thai, Hebrew, Arabic, and Turkish

that were used in NTCIR-5 WEB[6] for testing. All the topics were written in Japanese. The numbers of topics can be seen in **Table 3**. A topic example can be seen in **Fig. 3**. A summary of the document collection is shown in **Table 3**. In the NW100G-01 collection, the proportion of the estimated number of pages in each language (Eguchi et al, 2003) is shown in **Table 4**. The title field of each topic gives 1–3 query components that are suggested by the topic creator to be similar to the query terms used in real Web search engines. This definition of the title is different from the one used by the TREC Web Track (Craswell and Hawking, 2003) or the TREC Terabyte Track (Clarke et al, 2004) 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 compound word; and (iii) the title field has an attribute (i.e., 'CASE' and 'RELAT') that indicates the kind of search strategies and can optionally be used as a Boolean-type operator (Eguchi et al, 2004). 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. In this paper, we only used the title fields of the topics. We did not use any query structure information provided as the attributes 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.

6.2 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

[6] The topics were a subset of those created for the NTCIR-4 WEB, Informational Retrieval Subtask. The relevance judgments were additionally performed by extension of the relevance data of the NTCIR-4 WEB. The task was motivated by the question "Which terms should be added to the original query to improve search results?" The objectives of this paper are different from those of that task; however, the data set is suitable for our experiments.

Table 5 Effects of compounding or decomposing of terms in queries in experiments using training data.

	AvgPrec$_a$	%chg	AvgPrec$_c$	%chg
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

analyzer 'MeCab version 0.81'.[7] We did not use the part-of-speech (POS) tagging function of the morphological analyzer for the documents, because it requires more time.[8] On completion of the morphological analysis, all Japanese words were separated by spaces. We used Indri to make an index of the web documents in the NW100G-01 document collection, using these segmented texts described above. We only used one-byte symbol characters as stopwords in the indexing phase, to enable querying even by phrases consisting of high-frequency words, similar to "To be or not to be" in English, and to understand the effectiveness of the phrase-based query structuring described in Section 4.2. Instead, we used in the querying phase several types of stopwords only over the term feature f_T, but not over the ordered/unordered phrase features f_O or f_U, in Eqs. (2) and (3). As for the types of stopwords we used, see Eguchi (2005).

In the experiments described in this paper, we only used the title fields of the topics, as described in Section 6.1. We performed morphological analysis using the MeCab tool described at the beginning of this section to segment each of the query components delimited by commas within each title field, and to add POS tags to the segmented words. Here, the POS tags[9]are used to specify prefix and suffix words that appear in a query because, in the query structuring process, we make a distinction between compound words containing prefix/suffix words and other compound words, as described in Sections 4.1 and 4.2. Note that we only used the POS information to specify the type of a compound word: whether it contains prefix or suffix words; and thus, once the type was specified, we did not use the POS information itself for querying.

6.3 Experiments on Two-stage Term Dependence Model

6.3.1 Effects of Compound Word Models

We investigated the effects of compounding or decomposing of the query terms that were specified as constituent words of a compound word by the morphological analyzer, using the models described in Section 4.1, to empirically determine appropriate language units for the phrase-based query structuring that we discussed in Section 4.2. The experimental results using the NTCIR-3 WEB topic set are shown in **Table 5**.

[7] ⟨http://sourceforge.net/projects/mecab/⟩.

[8] Regardless of whether the POS tagging function is chosen or not, the resulting segmentation is the same, in this case of the MeCab tool.

[9] As suffix words, we used suffix nouns, suffix verbs and suffix adjectives; and as prefix words, we used nominal prefixes, verbal prefixes, adjectival prefixes and numerical prefixes, according to the part-of-speech system used in the MeCab tool.

In this table, 'AvgPrec$_a$' indicates the mean average precision over all 47 topics, and 'AvgPrec$_c$' indicates the mean average precision over the 23 topics that include compound words in the title field. '%chg' was calculated by the percentage of difference between performance of a target method and that of a baseline method, being divided by the baseline performance. As the baseline, we used *dcmp1*, the result of retrieval by decomposing all compound words. In the experiments in this section we did not use stopword lists, for simplicity.

From the results using *cmp1*, the naive phrase search using compound words did not work well. Comparing *pfx1* with *cmp1* or *cmp2*, we can see that the case of decomposing general compound words that do not contain prefix/suffix words turned out better than the case of not decomposing those. Furthermore, comparing *pfx1* with *dcmp1*, we can see that the case of using compound words containing prefix/suffix words as exact phrases was better than the case of decomposing those. Therefore, it turned out that compounding the prefix/suffix words and decomposing other compound words work well, such as in *pfx1* or *pfx2*. As for *pfx3*, we linearly combined *pfx1*, *pfx2* and *dcmp2*, with weights, 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 5**, we show the results using the optimized weights for the features of *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 *dcmp2* was optimized to be 0 and from the poor performance result only using *dcmp2* shown in **Table 5**, it is apparent that the constituent words qualified by the prefix/suffix words contributed very little to the retrieval effectiveness by themselves without the prefix/suffix words. Moreover, the above model combining *pfx1* and *pfx2* did not improve the retrieval effectiveness, compared with *pfx1* or *pfx2* alone, in spite of the complexity of the query. Based on these considerations and the fact that the performance of the *pfx1* and *pfx2* was almost the same, we used the simpler *pfx1* compound word model as the basic way of expressing Japanese compound words, and extended this idea to the query structuring using more general term dependence models that we discussed in Section 4.2.

6.3.2 Experiments on Two-stage Term Dependence Model for Training

We investigated the effects of query structuring using the local term dependence and the two-stage term dependence that we described in Section 4.2. These approaches are grounded in the empirical evidence through the experiments shown in Section *6.3.1*, as well as in the theoretical framework explained in Section 3.2. Using the NTCIR-3 WEB test collection, we optimized each of the models, defined in Section 4.2, 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_1 for the unordered phrase feature as 2, 4, 8, 50 or ∞ times the number of words specified in the phrase expression. Additionally, we used $(\lambda_T, \lambda_O, \lambda_U) = (0.9, 0.05, 0.05)$ for each N_1 value above. The results of the optimization that maximized the mean average precision over all 47 topics ('AvgPrec$_a$') are shown in **Table 6**. This table includes the mean average precision over 23 topics that contain compound words in the title field as 'AvgPrec$_c$'. '%chg' was calculated on the basis of *fi*, the result of retrieval with the term *independence* model. After optimization, the *glsd*$^+$ model worked best when $(\lambda_T, \lambda_O, \lambda_U, N) = (0.9, 0.05, 0.05, \infty)$. while the *glfd*$^+$ model worked best when $(\lambda_T, \lambda_O, \lambda_U, N) = (0.9, 0.05, 0.05, 50)$. In the experiments in this section, stopword removal was only applied to the term feature f_T, not to the phrase features f_O or f_U. The impact of stopwords on our models is discussed in Eguchi (2005).

Table 6 Optimization results using training data.

	$\mathrm{AvgPrec}_a$	%chg	$\mathrm{AvgPrec}_c$	%chg
fi	0.1543	0.0000	0.1584	0.0000
lsd$^+$	0.1624	5.2319	0.1749	10.4111
lfd$^+$	0.1619	4.9120	0.1739	9.7744
glsd$^+$	0.1640	6.2731	0.1776	12.0740
glfd$^+$	0.1626	5.4140	0.1769	11.6788
naive-lsd	0.1488	-3.5551	0.1472	-7.0743
naive-lfd	0.1488	-3.5427	0.1473	-7.0496
ntcir-3	0.1506	-2.3774	0.1371	-13.4680

For comparison, we naively applied Metzler and Croft's single-stage term dependence model, using either the sequential dependence or the full dependence variants defined in Section 3.2, to decomposed words within each of the query components delimited by commas in the title field of a topic, and combined the beliefs about the resulting structure expressions using the #`combine` operator shown in **Table 1**.[10] We show the results of these as *naive-lsd* and *naive-lfd*, respectively, in **Table 6**. These are different from our local term dependence models, *lsd*$^+$ and *lfd*$^+$ that are based on the empirical results reported in Section *6.3.1*, involving treatments on Japanese compound words. The results in **Table 6** suggest that Metzler and Croft's model must be enhanced to handle the more complex dependencies that appear in Japanese queries. For reference, we also show the best results from NTCIR-3 WEB participation (Eguchi et al, 2003), as *ntcir-3*, at the bottom of **Table 6**. This shows that even our baseline system worked better than the *ntcir-3* results.

Our two-stage term dependence and local term dependence models worked well especially for the queries that contain various compound words, such as on Topic 0060: "*sekai-ju*", "*hokuō shinwa*" and "*namae*" (in English, 'the World Tree', 'Norse mythology' and 'name'). These three query components are expressed using a compound word having a suffix word "*ju*", another compound word having no prefix/suffix words, and a general noun, respectively. How to formulate structured queries in this case is very similar to the example shown in Section 4.2. In this case, the mean average precision of the *fi*, *lsd*$^+$ and *glsd*$^+$ models were 0.2162, 0.5164 and 0.5503, respectively, and so our local term dependence model *lsd*$^+$ worked well and our two-stage term dependence model *glsd*$^+$ worked the best. Our two-stage term dependence models also worked for a part of (but not all of) the queries that do not contain compound words, such as on Topic 0019: "*ume*", "*meisho*" and "*Tokyo*" (in English, 'plum tree', 'place of interest' and 'Tokyo'), where no compound words appear in the Japanese query. In such a case, the local term dependence models behave the same as the full independence model *fi*, and the mean average precision of both the *fi* and *lsd*$^+$ models was 0.1202 and that of *glsd*$^+$ was 0.1318. The difference between the performance of *glsd*$^+$ and *glfd*$^+$ depends on topics.

6.3.3 Experiments on Two-stage Term Dependence Model for Testing

For testing, we used the models optimized in Section *6.3.2*. We used the relevance judgment data, for evaluation, that were provided by the organizers of the NTCIR-

[10] In another way of applying Metzler and Croft's model to all decomposed words in a whole query, ignoring boundaries across query components, the number of combinations of the words exponentially increases. Actually, in our preliminary experiments using some Japanese queries, the searching by this simple application did not accomplish within feasible time.

Table 7 Test results of phrase-based query structuring.

	AvgPrec$_a$	%chg	AvgPrec$_c$	%chg	AvgPrec$_o$	%chg
fi	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

'*' indicates statistical significant improvement over *fi*, *lsd*$^+$, *lfd*$^+$ and *glsd*$^+$ where $p < 0.05$ with two-sided Wilcoxon signed-rank test.

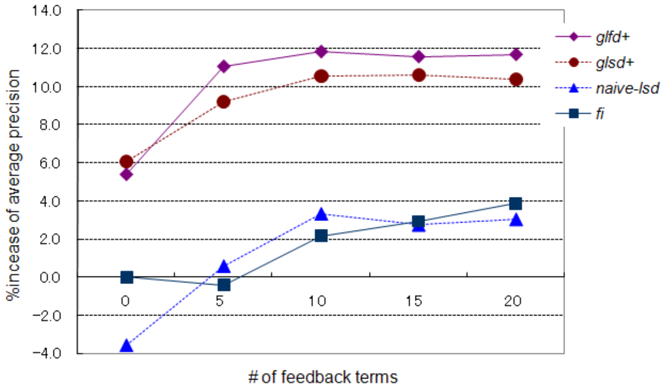

Fig. 4 A preliminary comparison with baseline results using training data

5 WEB task. The results are shown in **Table 7**. In this table, 'AvgPrec$_a$', 'AvgPrec$_c$' and 'AvgPrec$_o$' indicate the mean average precisions over all 35 topics, over the 22 topics that include compound words in the title field, and over the 13 topics that do not include the compound words, respectively. '%chg' was calculated on the basis of the result of retrieval with the term *independence* model (*fi*).

The results show that our two-stage term dependence models, especially the *glfd*$^+$ model, gave 13% better performance than the baseline (*fi*), which did not assume term dependence, and also better than the local term dependence models, *lsd*$^+$ and *lfd*$^+$, which only assumed local dependence within a compound word. The advantage of *glfd*$^+$ over *fi*, *lsd*$^+$ and *lfd*$^+$ was statistically significant at the two-sided 5% level, where the Wilcoxon signed-rank test was used, in average precision over all the topics. The results of 'AvgPrec$_c$' and 'AvgPrec$_o$' imply that our models work more effectively for queries expressed in compound words.

6.4 Experiments on Pseudo-Relevance Feedback with Two-stage Term Dependence Model

6.4.1 Experiments on Pseudo-Relevance Feedback for Training

We carried out preliminary experiments with a combination of phrase-based query structuring and pseudo-relevance feedback, using the NTCIR-3 WEB topic set, to investigate how this combination works. Pseudo-relevance feedback was implemented in Indri, based on Lavrenko's relevance models (Lavrenko and Croft, 2001), as described in Section **5**. The results are shown in **Fig. 4**. In this graph, the horizontal axis indicates the number of feedback terms ($\#terms_{fb}$), and the vertical axis shows the

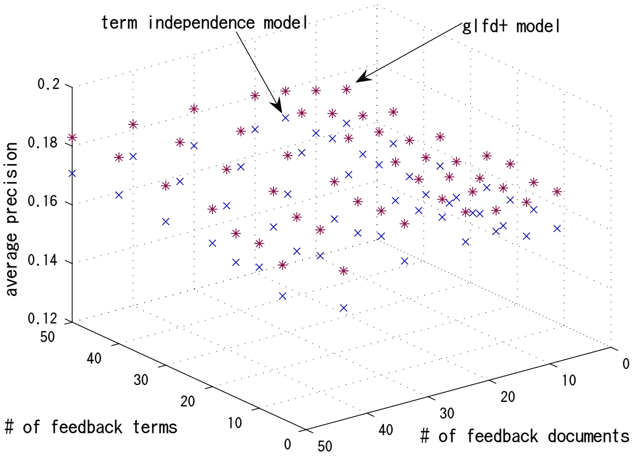

Fig. 5 Pseudo-relevance feedback with phrase-based query structuring using training data.

percentage increase in mean average precision compared to the mean average precision without phrase-based query structuring or pseudo-relevance feedback. For comparison, we naively applied the single-stage term dependence model in the same manner as *naive-lsd* in Section *6.3.2*.[11]

The explanatory note indicates which results used the naive applications of the single-stage term dependence model, *naive-lsd*, and the two-stage term dependence models, *glfd*$^+$ and *glsd*$^+$. For baseline comparison, we also performed experiments with pseudo-relevance feedback using the term *independence* model (*fi*), which assumes that query terms are independent of each other. Here, the parameters were set as $(\lambda_T, \lambda_O, \lambda_U, N) = (0.9, 0.05, 0.05, 50)$ for the *glfd*$^+$ model, and $(\lambda_T, \lambda_O, \lambda_U, N) = (0.9, 0.05, 0.05, \infty)$ for the *glsd*$^+$ model, each of which maximized the mean average precision when pseudo-relevance feedback was not applied. The original query weight for pseudo-relevance feedback and the number of feedback documents were set as $\nu = 0.7$ and $\#docs_{fb} = 10$. **Fig. 4** suggests that Metzler and Croft's term dependence model should be enhanced to handle the more complex dependencies that appear in queries with compound words.

We also experimented using the two-stage term dependence models *glsd*$^+$ and *glfd*$^+$ with pseudo-relevance feedback, changing the weights of $\#docs_{fb}$ and $\#terms_{fb}$ to 5, 10, 15, 20, 30, 40 and 50, respectively, and ν to 0.3, 0.5, 0.7 and 0.9. The results show that the *glfd*$^+$ model worked better than the *glsd*$^+$ model when combined with pseudo-relevance feedback. Actually, the *glfd*$^+$ model worked 3.7% better than the *glsd*$^+$ model in average precision on average over all the combinations of the parameters (with a maximum of 17.4%). The mean average precision over all topics with optimized values for ν for each combination of $(\#docs_{fb}, \#terms_{fb})$ is shown in **Fig. 5**. In this figure, the results for the term independence model, i.e., the *fi* model, are also shown as a baseline. Selected evaluation values are shown in **Table 8**, where 'AvgPrec$_a$', 'AvgPrec$_c$' and 'AvgPrec$_o$' indicate the mean average precision over all the 47 topics, that over 23 topics that include the compound words in the title field, and that over 24 topics that do not

[11] We tested using both the sequential dependence model and the full dependence model. The results of these two models were almost the same in this context. Only the result using the sequential dependence model is shown, as *naive-lsd*, in **Fig. 4**.

Table 8 Pseudo-relevance feedback with phrase-based query structuring using training data.

	ν	AvgPrec$_a$	[%chg$_f$]	(%chg$_t$)	AvgPrec$_c$	[%chg$_f$]	(%chg$_t$)	AvgPrec$_o$	[%chg$_f$]	(%chg$_t$)
fi only	1.0	0.1543	[+0.00]	(+0.00)	0.1584	[+0.00]	(+0.00)	0.1503	[+0.00]	(+0.00)
(10, 05)	0.9	0.1598	[+3.59]	(+3.59)	0.1664	[+5.03]	(+5.03)	0.1535	[+2.14]	(+2.14)
(10, 10)	0.9	0.1597	[+3.53]	(+3.53)	0.1655	[+4.44]	(+4.44)	0.1542	[+2.61]	(+2.61)
(10, 20)	0.9	0.1620	[+5.02]	(+5.02)	0.1693	[+6.85]	(+6.85)	0.1551	[+3.18]	(+3.18)
glsd$^+$ only	1.0	0.1640	[+0.00]	(+6.27)	0.1776	[+0.00]	(+12.07)	0.1509	[+0.00]	(+0.41)
(10, 05)	0.7	0.1685	[+2.77]	(+9.22)	0.1892	[+6.53]	(+19.40)	0.1487	[-1.47]	(-1.06)
(10, 10)	0.7	0.1706	[+4.04]	(+10.57)	0.1864	[+4.95]	(+17.62)	0.1555	[+3.02]	(+3.45)
(10, 20)	0.5	0.1730	[+5.51]	(+12.12)	0.1894	[+6.64]	(+19.51)	0.1573	[+4.23]	(+4.66)
glfd$^+$ only	1.0	0.1626	[+0.00]	(+5.41)	0.1769	[+0.00]	(+11.68)	0.1489	[+0.00]	(-0.91)
(10, 05)	0.7	0.1714	[+5.38]	(+11.08)	0.1886	[+6.57]	(+19.01)	0.1549	[+4.03]	(+3.07)
(10, 10)	0.7	0.1726	[+6.12]	(+11.86)	0.1854	[+4.76]	(+16.99)	0.1603	[+7.66]	(+6.68)
(10, 20)	0.5	0.1742	[+7.12]	(+12.92)	0.1896	[+7.17]	(+19.68)	0.1595	[+7.06]	(+6.08)

In the left column, '$(\cdot, \cdot)$' indicates ($\#docs_{fb}$, $\#terms_{fb}$).

include the compound words, respectively. '%chg$_f$' and '%chg$_t$' were calculated on the bases of (i) no pseudo-relevance feedback, i.e., $\nu = 1.0$, for each model; and (ii) the term independence model alone, i.e., the *fi* model without pseudo-relevance feedback, respectively. **Fig. 5** and **Table 8** show that the phrase-based query structuring method works better than the baseline that does not assume term dependence at all, for almost every combination of ($\#docs_{fb}, \#terms_{fb}$). These figure and table also show that, for a sufficient number of feedback documents, more feedback terms give more effective retrieval performance in general, but at the expense of searching cost.

6.4.2 Experiments on Pseudo-Relevance Feedback for Testing

We carried out experiments on the combination of phrase-based query structuring and pseudo-relevance feedback. For the phrase-based query structuring we used the *glsd*$^+$ and *glfd*$^+$ models with the optimal parameters. For evaluation, we used the relevance judgment data that were provided by the organizers of the NTCIR-5 WEB task. For the pseudo-relevance feedback, we used the top-ranked 5 and 10 documents ($\#docs_{fb}$), and 5, 10 and 20 terms ($\#terms_{fb}$) for feedback. We used the optimized value of the original query weight ν, which we obtained from training, in Section 6.4.1, corresponding to each pair of ($\#docs_{fb}$, $\#terms_{fb}$). The results are shown in **Table 9**. In this table, 'AvgPrec$_a$', 'AvgPrec$_c$' and 'AvgPrec$_o$' 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. We performed significance tests on 'AvgPrec$_a$' on the bases of (i) no pseudo-relevance feedback, i.e., $\nu = 1.0$, for each model; and (ii) the term independence model alone, i.e., the *fi* model without pseudo-relevance feedback, as shown in this table.[12] '%chg$_f$' and '%chg$_t$' were calculated on the bases of (i) and (ii) mentioned above, respectively.

As shown at the top of the results for the *glfd*$^+$ model (when $\nu = 1.0$) in this table and at the top of the results of the *fi* model, the *glfd*$^+$ model alone worked 13% better than the *fi* model alone in mean average precision, in total. This is the same as reported in **Table 7**. The combination of the phrase-based query structuring and the pseudo-relevance feedback achieved statistically significant improvements over the phrase-based query structuring alone, under certain conditions of ($\#docs_{fb}, \#terms_{fb}$). The results in **Table 9** also show that, by combining with phrase-based query structuring, the pseudo-relevance feedback works effectively both for queries that include

[12] The results of significance tests on 'AvgPrec$_c$' or 'AvgPrec$_o$' are not presented, since 22 or 13 topics make it difficult to achieve statistical significance.

Table 9 Pseudo-relevance feedback with phrase-based query structuring using test data.

	ν	AvgPrec$_a$	[%chg$_f$]	(%chg$_t$)	AvgPrec$_c$	[%chg$_f$]	(%chg$_t$)	AvgPrec$_o$	[%chg$_f$]	(%chg$_t$)
fi only	1.0	0.1405	[+0.00]	(+0.00)	0.1141	[+0.00]	(+0.00)	0.1852	[+0.00]	(+0.00)
(05, 05)	0.9	0.1577	[+12.23*]	(+12.23**)	0.1346	[+18.01]	(+18.01)	0.1967	[+6.21]	(+6.21)
(05, 10)	0.9	0.1565	[+11.40*]	(+11.40**)	0.1333	[+16.86]	(+16.86)	0.1957	[+5.71]	(+5.71)
(05, 20)	0.9	0.1553	[+10.52*]	(+10.52**)	0.1324	[+16.04]	(+16.04)	0.1940	[+4.76]	(+4.76)
(10, 05)	0.9	0.1583	[+12.66*]	(+12.66**)	0.1360	[+19.22]	(+19.22)	0.1959	[+5.82]	(+5.82)
(10, 10)	0.9	0.1575	[+12.14*]	(+12.14**)	0.1360	[+19.24]	(+19.24)	0.1939	[+4.73]	(+4.73)
(10, 20)	0.9	0.1565	[+11.39*]	(+11.39**)	0.1345	[+17.89]	(+17.89)	0.1937	[+4.61]	(+4.61)
glsd$^+$ only	1.0	0.1503	[+0.00]	(+6.96)	0.1313	[+0.00]	(+15.12)	0.1823	[+0.00]	(-1.55)
(05, 05)	0.7	0.1652	[+9.97]	(+17.62**)	0.1430	[+8.88]	(+25.34)	0.2029	[+11.30]	(+9.57)
(05, 10)	0.7	0.1655	[+10.14*]	(+17.81**)	0.1418	[+7.94]	(+24.25)	0.2057	[+12.83]	(+11.08)
(05, 20)	0.5	0.1709	[+13.77]	(+21.69**)	0.1477	[+12.50]	(+29.50)	0.2102	[+15.32]	(+13.53)
(10, 05)	0.7	0.1674	[+11.44*]	(+19.19**)	0.1473	[+12.14]	(+29.09)	0.2016	[+10.58]	(+8.87)
(10, 10)	0.7	0.1656	[+10.21*]	(+17.88**)	0.1453	[+10.66]	(+27.39)	0.1999	[+9.67]	(+7.97)
(10, 20)	0.5	0.1680	[+11.80]	(+19.58)	0.1474	[+12.21]	(+29.18)	0.2029	[+11.29]	(+9.56)
glfd$^+$ only	1.0	0.1588	[+0.00]	(+13.02**)	0.1400	[+0.00]	(+22.70)	0.1906	[+0.00]	(+2.93)
(05, 05)	0.7	0.1730	[+8.96*]	(+23.14**)	0.1505	[+7.49]	(+31.89)	0.2111	[+10.78]	(+14.03)
(05, 10)	0.7	0.1750	[+10.20*]	(+24.55**)	0.1512	[+8.01]	(+32.52)	0.2152	[+12.94]	(+16.25)
(05, 20)	0.5	0.1784	[+12.33]	(+26.96**)	0.1538	[+9.89]	(+34.82)	0.2199	[+15.38]	(+18.76)
(10, 05)	0.7	0.1728	[+8.80*]	(+22.97**)	0.1517	[+8.39]	(+32.99)	0.2083	[+9.32]	(+12.52)
(10, 10)	0.7	0.1714	[+7.95]	(+22.01**)	0.1496	[+6.91]	(+31.17)	0.2082	[+9.25]	(+12.45)
(10, 20)	0.5	0.1762	[+11.01]	(+25.46**)	0.1552	[+10.90]	(+36.07)	0.2118	[+11.14]	(+14.39)

In the left column, '$(\cdot,\cdot)$' indicates ($\#docs_{fb}$, $\#terms_{fb}$). In the column of AvgPrec$_a$, '*' and '**' indicate statistically significant improvements over no pseudo-relevance feedback (i.e., $\nu = 1.0$) for each model and over the term independence model alone (i.e., the *fi* model when $\nu = 1.0$), respectively, where $p < 0.05$ with the two-sided Wilcoxon signed-rank test.

Table 10 Example feedback terms in cases average precision was increased and decreased.

Topic 80 [161.61% increased]: *iro, shinri-teki-kōka* (color, psychological effect)			Topic 62 [92.85% increased]: *kafun-shō, yobō-hō* (hay fever, preventive measure)			Topic 19 [77.08% decreased]: *geijutsu-sakuhin,yōroppa,chūsei* (work of art, Europe, medieval)		
term (translation)		weight	term (translation)		weight	term (translation)		weight
iro	(color)	0.000031	*kafun*	(pollen)	0.000088	*bungaku*	(literature)	0.000035
hito	(man)	0.000014	*-shō* *	(fever of)	0.000058	*-seiki* *	(century of)	0.000026
shikisai	(color)	0.000011	*jōhō*	(information)	0.000020	*sakuhin*	(work)	0.000020
karā	(color)	0.000009	*hito*	(man)	0.000013	*kenkyū*	(research)	0.000019
-teki *	(-like)	0.000008	*-hō* *	(method of)	0.000012	*-gaku* *	(-logy)	0.000011
sinri	(psychology)	0.000006	*hisan*	(dispersal)	0.000012	*-shi* *	(history of)	0.000010
-kyū *	(-class)	0.000005	*pēji*	(page)	0.000011	*ippan*	(general)	0.000010
kōka	(effect)	0.000005	*sugi*	(cedar)	0.000009	*gendai*	(modern)	0.000009
ao	(blue)	0.000004	*chiryō*	(treatment)	0.000009	20	(20)	0.000008
aka	(red)	0.000003	*yobō*	(preventive)	0.000008	19	(19)	0.000007
mi	(look)	0.000003	*arerugī*	(allergy)	0.000007	*shōsetsu*	(novel)	0.000007
nikansuru	(regarding)	0.000003	*keisai*	(publication)	0.000005	*shi*	(poetry)	0.000007
-ka *	(-ize)	0.000003	*-teki* *	(-like)	0.000005	*runesansu*	(Renaissance)	0.000006
toiu	(called)	0.000003	*taisaku*	(measure)	0.000005	*bijutsu*	(art)	0.000006
-gaku *	(-logy)	0.000003	*-nen* *	(year of)	0.000004	18	(18)	0.000006
seikaku	(character)	0.000003	*-nichi* *	(day of)	0.000004	*hyōron*	(critique)	0.000006
supōtsu	(sport)	0.000003	*saito*	(site)	0.000004	*itaria*	(Italy)	0.000005
chōsa	(survey)	0.000002	*kusuri*	(medicine)	0.000004	*bunka*	(culture)	0.000005
shiken	(test)	0.000002	*kenkō*	(health)	0.000004	*chūsei*	(medieval)	0.000005
-kurai *	(or so)	0.000002	*nitsuite*	(about)	0.000004	*supein*	(Spain)	0.000005

Topic numbers and the corresponding queries with English translations in parentheses are indicated on top of the table. Whether and how much average precision was increased or decreased comparing with that without pseudo-relevance feedback are also shown in brackets. In the columns of feedback terms, '*' indicates suffixes in Japanese, while no prefixes appeared in these examples.

compound words and those that do not include compound words. Combining with the pseudo-relevance feedback, the *glfd*$^+$ model worked 9–15% better than the *fi* model, in mean average precision, under the same conditions of ($\#docs_{fb}, \#terms_{fb}$) in the results shown in **Table 9**.

Table 10 shows example feedback terms in cases when average precision was increased and decreased using *glfd*$^+$ model under the condition that ($\#docs_{fb}, \#terms_{fb}$) $= (10, 20)$ and $\nu = 0.5$. Whether and how much average precision was increased or decreased comparing with that without pseudo-relevance feedback are also indicated in this table. Topics 80 and 62 were the most successful cases and we can see that feedback terms include synonyms of query terms. For instance, "*shikisai*" and "*karā*" are synonyms of "*iro*" (color) that appear in the query of Topics 80. On the other hand, Topic 19 was the worst case partially because some numeric words hurt retrieval effectiveness.

7 Conclusions

In this paper, we proposed new phrase-based query structuring methods, which are based on a theoretical framework using Markov random fields. Our two-stage term dependence model captures both the global dependence between query components explicitly delimited by separators in a query, and the local dependence between constituents within a compound word when the compound word appears in a query component. We found that query structuring using our two-stage term dependence model worked 13% significantly better in mean average precision than the baseline that did not assume term dependence at all, and better than using models that only assumed either global dependence or local dependence in the query. The experimental results also imply that our models work more effectively for queries expressed in compound words, which are often used in the Japanese language.

As another contribution of this paper, we investigated how query structuring with term dependence could improve the performance of query expansion via a relevance model. We demonstrated through a series of experiments that the combination of the term dependence model and the relevance model was more effective than either the term dependence model or the relevance model alone. When we tested the two-stage term dependence alone, as mentioned above, this model worked 13% better in mean average precision than the baseline with the term independent model. When we tested the two-stage term dependence with the relevance model, this model worked 8–12% better in mean average precision than the baseline with the two-stage term dependence alone. Consequently, we achieved a significant 22–27% gain in mean average precision from the term independence model without query expansion, when we combined the term dependence model and the relevance model.

The two-stage term dependence model may be reasonable for other languages, if query components can be specified in a query, for example “ ‘ozone hole’ ‘human body’ ”. In this paper we assumed that the query components were delimited by separators in a query. This model is also applicable if we perform phrase detection on a long query or natural language input and we consider the resulting phrases to be query components. The application to natural language-based queries in either Japanese or English, employing an automatic phrase detection technique, is worth pursuing as future work.

Acknowledgements We thank Donald Metzler and David Fisher for valuable discussions and comments. This work was supported in part by the Grants-in-Aid for Scientific Research (#19024055 and #18650057, and #20300038) from the Ministry of Education, Culture, Sports, Science and Technology, Japan, and in part by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the sponsor.

References

Buckley C, Salton G, Allan J, Singhal A (1994) Automatic query expansion using SMART: TREC 3. In: Proceedings of the 3rd Text Retrieval Conference, pp 69–80, Gaithersburg, Maryland, USA

Chen A, Gey FC (2002) Experiments on cross-language and patent retrieval at NTCIR-3 Workshop. In: Proceedings of the 3rd NTCIR Workshop, Tokyo, Japan

Clarke C, Craswell N, Soboroff I (2004) Overview of the TREC 2004 Terabyte Track. In: Proceedings of TREC 2004, Gaithersburg, Maryland, USA

Craswell N, Hawking D (2003) Overview of the TREC 2003 Web Track. In: Proceedings of TREC 2003, pp 78–92, Gaithersburg, Maryland, USA

Croft WB, Lafferty J (eds) (2003) Language Modeling for Information Retrieval. Kluwer Academic Publishers

Croft WB, Turtle HR, Lewis DD (1991) The use of phrases and structured queries in information retrieval. In: Proceedings of ACM SIGIR 1991, Illinois, USA, pp 32–45

Eguchi K (2005) NTCIR-5 query expansion experiments using term dependence models. In: Proceedings of the 5th NTCIR Workshop, Tokyo, Japan

Eguchi K, Oyama K, Ishida E, Kando N, Kuriyama K (2003) Overview of the Web Retrieval Task at the Third NTCIR Workshop. In: Proceedings of the 3rd NTCIR Workshop, Tokyo, Japan

Eguchi K, Oyama K, Aizawa A, Ishikawa H (2004) Overview of the Informational Retrieval Task at NTCIR-4 WEB. In: Proceedings of the 4th NTCIR Workshop, Tokyo, Japan

Fujii H, Croft WB (1993) A comparison of indexing techniques for Japanese text retrieval. In: Proceedings of ACM SIGIR 1993, Pittsburgh, Pennsylvania, USA, pp 237–246

Fujita S (1999) Notes on phrasal indexing: JSCB evaluation experiments at NTCIR ad hoc. In: Proceedings of the First NTCIR Workshop, Tokyo, Japan, pp 101–108

Jones GJF, Sakai T, Kajiura M, Sumita K (1998) Experiments in Japanese text retrieval and routing using the NEAT system. In: Proceedings of ACM SIGIR 1998, Melbourne, Australia, pp 197–205

Kando N, Kageura K, Yoshioka M, Oyama K (1998) Phrase processing methods for Japanese text retrieval. SIGIR Forum 32(2):23–28

Lavrenko V, Croft WB (2001) Relevance based language models. In: Proceedings of ACM SIGIR 2001, New Orleans, Louisiana, USA, pp 120–127

Metzler D, Croft WB (2004) Combining the language model and inference network approaches to retrieval. Information Processing and Management 40(5):735–750

Metzler D, Croft WB (2005) A Markov random field model for term dependencies. In: Proceedings of ACM SIGIR 2005, Salvador, Brazil, pp 472–479

Metzler D, Strohman T, Turtle H, Croft WB (2004) Indri at TREC 2004: Terabyte Track. In: Proceedings of TREC 2004, Gaithersburg, Maryland, USA

Mishne G, de Rijke M (2005) Boosting web retrieval through query operations. In: Proceedings of the 27th European Conference on Information Retrieval Research, Santiago de Compostela, Spain, 2005, pp 502–516

Mitra M, Buckley C, Singhal A, Cardie C (1997) An analysis of statistical and syntactic phrases. In: Proceedings of RIAO 97, Montreal, Canada, pp 200–214

Moulinier I, Molina-Salgado H, Jackson P (2002) Thomson Legal and Regulatory at NTCIR-3: Japanese, Chinese and English retrieval experiments. In: Proceedings of the 3rd NTCIR Workshop, Tokyo, Japan

Ogawa Y, Matsuda T (1997) Overlapping statistical word indexing: A new indexing method for Japanese text. In: Proceedings of ACM SIGIR 1997, Philadelphia, Pennsylvania, USA, pp 226–234

Strohman T (2007) Efficient processing of complex features for information retrieval. PhD thesis, University of Massachusetts, Amherst

Strohman T, Turtle H, Croft WB (2005) Optimization strategies for complex queries. In: Proceedings of ACM SIGIR 2005, Salvador, Brazil, pp 219–225

Turtle HR, Croft WB (1991) Evaluation of an inference network-based retrieval model. ACM Transactions on Information Systems 9(3):187–222

Xu J, Croft WB (1996) Query expansion using local and global document analysis. In: Proceedings of ACM SIGIR 1996, Zurich, Switzerland, pp 4–11

Yoshioka M (2005) Overview of the NTCIR-5 WEB Query Expansion Task. In: Proceedings of the 5th NTCIR Workshop, Tokyo, Japan

Zhai C, Lafferty J (2001) Model-based feedback in the language modeling approach to information retrieval. In: Proceedings of ACM CIKM 2001, Atlanta, Georgia, USA, pp 403–410