Overlapping statistical word indexing: A new indexing method for Japanese text

OGAWA Yasushi, MATSUDA Toru
{ogawa, matsuda}@ic.rdc.ricoh.co.jp
Information and Communication R&D Center, RICOH Co., Ltd.
3-2-3 Shin-yokohama, Kouhoku-ku,
Yokohama, Kanagawa 222, JAPAN

Abstract

Because word boundaries are not apparently indicated in Asian languages including Japanese, word indexing cannot simply be applied. Although dictionary-based text segmentation techniques enable word indexing, they have some problems such as dictionary maintenance. N-gram indexing, another conventional indexing method, suffers from increase in index size. This paper proposes a new statistical indexing method. We first propose a segmentation method for Japanese text which uses statistical information of characters. It needs only a small amount of statistic information and computation, and does not need constant maintenance. We secondly propose a new indexing strategy which extracts some overlapping segments in addition to the segments extracted using the existing strategy. Thus it increases the effectiveness of retrieval.

1 Introduction

In recent information retrieval (IR) studies, ranking retrieved documents in order of their relevance to the query is considered an important function [22]. Most IR systems compute a document score based on weights of indexing units extracted from a given document and a query. An indexing method which extracts (or generates) indexing units from text is, therefore, one of the fundamental processes in IR systems, and greatly affects retrieval effectiveness [5].

Because text is written in natural language, indexing methods differ between languages. In English or other European languages, for example, words are usually used as indexing units, and large scale evaluation experiments have shown that word indexing is quite effective [9]. In some Asian languages such as Chinese, Japanese, and Korean (CJK), however, word boundaries are not indicated by delimiters such as spaces. Therefore the question of what kind of indexing units should be used for CJK has been studied extensively [4][6][12][14] [17][18][27].

One way to handle these languages is by using a large-scale dictionary and complex linguistic knowledge and segmenting the text into words[1] [6]. Once words are identified, techniques used in English are applicable, and one can expect to obtain the highly effective retrieval obtained with English IR systems. We call this method dictionary-based word indexing. One problem with this kind of indexing, however, is that the dictionary and linguistic knowledge require constant maintenance and make retrieval systems large and complicated.

The n-gram indexing [3] is therefore sometimes used in CJK. An n-gram is an *n* successive characters.[2] The n-gram indexing uses as indexing units n-grams extracted from the text, and retrieved documents are ranked according to weights of the n-grams [3][12]. The n-gram indexing is at least as effective as dictionary-based word indexing [6][12][17] [27], but the index size tends to be large because this kind of indexing extracts overlapping n-grams from text.

Another method that solves the problems of dictionary-based word indexing is one that uses a statistical segmentation of text. This statistical word indexing uses words as indexing units, but the words are identified using statistical information instead of a dictionary and linguistic knowledge [14]. But, because statistical segmentation is usually less accurate than dictionary-based segmentation, retrieval is also less effective [27]. In Japan, there are only a limited number of researchers investigating statistical segmentation methods [15][25][26], and a few IR systems using the statistical word indexing [1][17].

This paper focuses on the statistical word indexing for Japanese IR systems. It first proposes a statistical segmentation method for Japanese text. Because this is a simple method that uses statistical information of characters, it is free from the problems of the dictionary-based method. This paper also proposes a new indexing strategy that extracts, in addition to the basic segments determined using the conventional segmentation strategy that breaks text disjointly, *overlapping* segments that merge more than one basic segments. Because even when the conventional method fails

[1]It should be noted that what a word is in CJK not clear[6][14][16][12]. This paper defines a word as the smallest language unit which maintains meaning (sometimes called as a morpheme elsewhere).

[2]A single character is a special case of the n-gram when $n = 1$. So the character indexing [6][27] which treats single characters as indexing units is considered in this paper to be kind of n-gram indexing.

to identify some words, merged segments sometimes correspond to these words, this strategy can result in much more effective retrieval than can be obtained using the conventional dictionary-based word indexing and the n-gram indexing. In addition, to achieve the same level of retrieval effectiveness, the size of an index file is smaller that that needed when using the n-gram method, because non word segments are less likely to be extracted from text.

This paper is organized as follows. The next section describes dictionary-based word indexing and n-gram indexing. Section 3 explains a statistic word segmentation method for Japanese, and Sections 4 details a new indexing method. Section 5 describes an evaluation experiment and its results.

2 Conventional indexing methods

2.1 Dictionary-based word indexing

To apply the word indexing to Japanese, it is necessary to segment text into words by using a morphological analyzer. There are two types of parsing methods: one based on a small dictionary that contains only a limited number of functional words (closed-lexicon segmentation), and the other using a large dictionary that *ideally* includes all the words in the universe (open-lexicon segmentation). Since the meaning of a compound word, which is frequently found in Japanese text, can be expressed using other phrases or passages made up of its component words, component words need to be identified in retrieval. The closed-lexicon method, however, cannot identify component words in a compound word and therefore does not perform as well as open-lexicon segmentation does [12][17][18]. This paper thus addresses only the method using the open-lexicon segmentation, and it calls the word indexing using this kind of segmentation **dictionary-based word indexing**.

In this indexing method, a phrase "アジアの熱帯雨林保護" (protection of rain forests in Asia) can be segmented in the following way: "アジア"(Asia), "の"(in), "熱帯"(tropical), "雨"(rain), "林"(forests), and "保護"(protection). All of these words, or the words remaining after eliminating the functional word ("の": particle), are used as indexing units.

Dictionary-based word indexing has, however, the following problem[23]:

(1) Because dictionary-based segmentation cannot identify *unknown* or *new* words that are not registered in the dictionary, the dictionary needs to be maintained constantly. However, the maintenance requires human works and thus costs very much. In addition, since a morphological analyzer is not free from making errors even when the dictionary includes the complete list of words, and these errors deteriorate the effectiveness of retrieval.

(2) Because a morphological analyzer is generally heavy and complex software, it takes much time in indexing and combining the analyzer complicates IR systems.

2.2 N-gram indexing

An n-gram is a string of n successive characters extracted from text. **N-gram indexing**[3], ignoring word boundaries, extracts all the n-grams including overlapping ones as the indexing units, and retrieved documents are ranked according to weights of n-grams instead of words. Because there is no need to parse text by using a dictionary and other rules, it is free from the problems mentioned above. Although this method misses word-level semantics, it never lose words by parse errors at word segmentation. As a result, it is at least as effective as dictionary-based word indexing [6][12][17][27]. However, compared with the word indexing, because this method extracts many more indexing units, the index file becomes larger.

With bi-gram (n-gram when $n = 2$) indexing, the above phrase "アジアの熱帯雨林保護" is segmented as "アジ", "ジア", "アの", "の熱", "熱帯", "帯雨", "雨林", "林保", and "保護". Notice that out of nine bi-grams there are only two words, "熱帯"(tropical) and "保護"(protection), and one compound word "雨林"(rain forests).

When applying this method to Japanese, which has about 7000 distinct characters, n is usually set to 1 or 2 [2][8][11][19]. Because, as for the word length, words consisting of two characters are the most frequent in Japanese [16], the bi-gram indexing achieves better performance than does the uni-gram (n-gram when $n = 1$) indexing [17]. The bi-gram indexing, however, cannot handle efficiently single-character words which are frequently found in Japanese [19]. To process a single-character query "木"(tree), for example, it is necessary to retrieve all the bi-grams that contain "木" such as "大木"(big tree), "樹木"(which also means "tree") and so on. Because there are about 7000 different characters in Japanese, there are potentially 14000 bi-grams containing "木". Off course, not all of the 14000 bi-grams appear in real text, but still a large number of bi-grams need to be processed for a single character query, resulting in slower retrieval. In addition, the uni-gram indexing has an advantage for thesaurus effects: since kanji (a set of Chinese characters) is ideograms, if two words share the same kanji character, they have some conceptual relationship [6].

We therefore think it is better to combine n-gram ($n > 1$) indexing method(s) with the uni-gram indexing. We call this kind of indexing **multiple n-gram indexing**. When the uni-gram indexing and the bi-gram indexing are combined, what is extracted from the sample phrase is: "ア", "ジ", "ア", "の", "熱", "帯", "雨", "林", "保", "護", "アジ", "ジア", "アの", "の熱", "熱帯", "帯雨", "雨林", "林保", and "保護". This idea is similar to the combination of word-based indexing and character-based indexing [7], but because that combination requires natural language processing in order to identify words, we prefer the multiple n-gram indexing.

3 Statistical Word Indexing

Statistical word indexing uses words as indexing units, but text is segmented into words by using statistical information [14]. This section proposes a statistical segmentation method for Japanese text and explains its application to indexing.

3.1 Segmentation principle

One of the big advantages of a statistical word segmentation method is its robustness; Since the dictionary-based segmentation method is usually depend heavily on a dictionary to identify words, it cannot process unregistered words. On the other hand, as statistical methods process text using statistical information (data) of n-grams and so on, they can identify words that are unknown at the preparation of the statistical data. Therefore, the statistical methods do not require constant maintenance of statistical data as needed in the dictionary-based approach, and thus one can considerably reduce the maintenance cost.

There are several approaches in statistical segmentation [10][14][20][24][27]. We prefer a method that is based on the likelihood of a character pair(bi-gram) being a word boundary and breaks a given text at pairs (points) which have large likelihood [24][27]; it is easy to collect statistical data necessary for the segmentation and requires less computation [10][17].

In our previous study, we had developed a statistical segmentation method using bi-gram statistics [17]. The method used, as for bi-gram statistics measure, statistical probability (named **the segmentation probability**) that a given bi-gram to appear at word boundaries. It was used only in a post-processing of query word extraction, i.e. to identify component words in a kanji compound word which was extracted from query text using a closed lexicon parser [17]. Note that multiple n-gram indexing was used at document registration in our previous system.

In this paper, we extend the method so as to index full text without using any parsers. Because, in Japanese, there are several character classes and their grammatical functions and usage patterns in writing are different from each other, the classes should be taken into consideration in segmentation [1][6][25]. Therefore, in our new method, the segmentation probability of a given character pair is at first determined according to the classes of the pair's constituent characters:

- The segmentation probability between characters of the different classes is set to 1.0 because such a point is, in most cases, a break between two words. The main exceptions are conjugational parts; the head is usually written using kanji but the conjugational part is written in hiragana. But, because conjugational parts are not very important in information retrieval, the segmentation of text at changes in class might work well.
- The segmentation probability between characters of the same class is determined as follows:

  hiragana: hiragana is a set of phonetic characters and is usually used to write conjugational parts or functional words such as particles, conjunctions and auxiliary verbs. Because most of these words are not useful for retrieval, we set the probability of hiragana pairs to 1.0 and segment a hiragana sequence into single characters.[3]

  katakana and kanji: katakana is another set of phonetic characters, and kanji is a set of Chinese ideograms. They are mainly used to write content words such as nouns and verb-heads. Because a katakana sequence and a kanji sequence sometimes form a compound word, these sequences need to be further segmented. We therefore segment them according to the segmentation probabilities between katakana pairs or kanji pairs estimated as described in the next subsection.

  ASCII: an ASCII character (letter of the English alphabet or numeric character) sequence generally forms a foreign word, a proper noun, or a number. Therefore we do not segment the sequence, and set the probability of ASCII pairs to 0.0. Exceptions are delimiters such as spaces and punctuation marks. They are excluded from the indexing units, and the probability of pairs whose first or second character is a delimiter is set to 1.0.

[3]There are hiragana sequences that form nouns such as "がん" (cancer), "ごみ"(refuse) and so on. However, because our retrieval system uses an inverted file which records positional information of indexing units in a document, it can correctly process queries with such hiragana words.

3.2 Estimating segmentation probability for kanji and katakan

To segment any kanji or katakana sequences, it is necessary to compute the segmentation probabilities between all possible kanji pairs and all possible katakana pairs. Collecting these probabilities requires a morphologically analyzed corpus in which words are identified manually or automatically. However, because there are nowadays several kinds of such corpora open to the public, IR system developers do not need to establish them. [4]

Given an analyzed corpus, the segmentation probability of a pair is determined by dividing the number of times the pair appeared at boundaries between two different words by the number of times the pair appeared in any place. It is almost impossible, however, to gather, from an existing corpus, the probabilities for all the possible pairs; Japanese has a large character set, and thus the number of possible pairs is enormous.

We therefore estimate a pair's segmentation probability by using statistical information about its constituent characters[17]. It is assumed that the segmentation probability of a character pair $c_i c_{i+1}$, $P_{seg}(c_i c_{i+1})$, is a product of the tail probability of the first character c_i, $P_{tail}(c_i)$, and the head probability of the second character c_{i+1}, $P_{head}(c_{i+1})$:

$$P_{seg}(c_i c_{i+1}) = P_{tail}(c_i) \times P_{head}(c_{i+1}) \qquad (1)$$

Here, $P_{head}(c)$ and $P_{tail}(c)$ are given by:

$$P_{head}(c) = \frac{\#(c \text{ appeared at the head of words})}{\#(c \text{ appeared at any place}} \qquad (2)$$

$$P_{tail}(c) = \frac{\#(c \text{ appeared at the tail of words})}{\#(c \text{ appeared at any place})} \qquad (3)$$

where $\#(x)$ represents the number of times of x's occurrences.

It is much more easy to compute the head and tail probabilities of all the possible characters than to compute the

[4]Even when they try to establish such a corpus by themselves, gathering statistical data is a process separate from indexing itself. Therefore, although a parser might be necessary to establish it, the maintenance of dictionaries does not become problem here.

Table 1: Example of head and tail probabilities.

Kanji		
	Head prob.	Tail prob.
熱	0.7541	0.3599
帯	0.2546	0.8573
雨	0.6866	0.7377
林	0.3629	0.8512
default	0.5859	0.5001

Katakana		
	Head prob.	Tail prob.
ア	0.4180	0.4279
イ	0.2394	0.1114
ウ	0.2785	0.0823
エ	0.5505	0.0451

segmentation probabilities of all the possible character pairs. Another advantage is that the amount of data needed is small: proportional to the size of the character set.

There may, however, still be characters that appeared, in a given corpus, only a few time or not at all, so statistical data cannot be obtained for all kanji characters.[5] To handle those less frequent characters, default values are introduced for the head and tail probabilities for kanji. The default values are computed as follows:

$$P_{head}^{default} = \frac{\sum_c \#(c \text{ appeared at the head of words})}{\sum_c \#(c \text{ appeared at any place})} \quad (4)$$

$$P_{tail}^{default} = \frac{\sum_c \#(c \text{ appeared at the tail of words})}{\sum_c \#(c \text{ appeared at any place})} \quad (5)$$

In our experiment we used a morphologically analyzed corpus, RWC-DB-TEXT-94-1.[6] It had been developed by the Real World Computing Society in Japan and its contents were about 100,000 Mainichi newspaper articles that appeared in 1994. Table 1 lists some of the head and tail probabilities gathered from the corpus. These values are used to compute the segmentation probability of the character pair "熱帯" as:

$$P_{tail}(\text{熱}) \times P_{head}(\text{帯}) = 0.3599 \times 0.2545 = 0.0916.$$

The probability of another pair "驟雨" (sudden shower), whose first character is a less frequent one, is computed using the default tail probability:

$$P_{tail}^{default} \times P_{head}(\text{雨}) = 0.5001 \times 0.6866 = 0.3434.$$

3.3 Simple indexing based on segmentation probabilities

Using the segmentation probabilities determined as explained above, text is segmented into *disjoint* parts at points whose segmentation probabilities are greater than a segmentation threshold T_{seg}. The resultant segments are used as the indexing units.

ア (katakana)	
	0.1046
ジ (katakana)	
	0.0619
ア (katakana)	
	1.0000
の (hiragana)	
	1.0000
熱 (kanji)	
	0.0916
帯 (kanji)	
	0.5894
雨 (kanji)	
	0.2676
林 (kanji)	
	0.4761
保 (kanji)	
	0.0289
護 (kanji)	

Figure 1: Segmentation probabilities for the sample phrase.

For the phrase used in the example of the previous section, the segmentation probabilities are computed as shown in Figure 1. When $T_{seg} = 0.2$, the sentence is broken into "アジア", "の", "熱帯", "雨", "林", "保護", and the correct answer is obtained.

4 Overlapping Statistical Word Indexing

4.1 Problems with the simple statistical segmentation

In the statistical word indexing method, the threshold value controls the segmentation result. When the threshold value is large, text is not segmented at word boundaries with low probabilities, some words are therefore not extracted, and some relevant documents are missed at retrieval. When the threshold value is small, on the other hand, some words are unnecessarily divided into small parts or even into single characters. Because word semantics are lost in such a case, unsuitable documents are sometimes ranked inordinately high. In this way, the threshold value greatly influences the retrieval effectiveness:[7] there is an optimal value between 0 and 1, and an extreme value degrades the performance [17].

Even when the optimal value is used, however, the model used to estimate the segmentation probability is too simple to reflect various phenomena in natural language. Thus it is sometimes impossible to break text properly by simply controlling the threshold value. For example, the segmentation probabilities *for a compound* word "大使公邸" (an official residence of an ambassador) are computed as "大 0.1822 使 0.1652 公 0.0017 邸". Because the probability of a word "大使"(ambassador) is greater than the probability of "使公" which lies between component words "大使" and "公邸"(official residence), the simple method cannot correctly segment this compound word. The accuracy of the segmentation in the simple statistical method is not as much as that obtained in the dictionary-based one. Consequently, the simple statistical indexing is inferior to dictionary-based word indexing and/or to n-gram indexing. It should be

[5]Because the number of distinct katakana characters is small (about 90), such a less frequent katakana does not exit.

[6]You can get its order form via `http://www.rwcp.or.jp/WSWG/rwcdb/text/index.html`.

[7]The threshold value also affects the index size and the retrieval efficiency [17].

noted that even if a more sophisticated segmentation model is used, it is almost impossible to keep up with all the phenomena in natural language.

4.2 Overlapping indexing strategy

Because it is impossible to compute the *ideal* segmentation probability, the segmentation strategy — segmenting texts into disjoint parts — needs to be modified if we are to solve the problem described above. Our modification is as follows:

(1) Setting the segmentation threshold smaller than the optimal value for the simple strategy so as to prevent compound words remaining unsegmented. We call these small segments by **basic segments**.

(2) Merging more than one basic segment into a large segment if they satisfy a condition which will be explained later.

Although some basic segments are meaninglessly small and do not coincide with words, a merged segment sometimes constitutes a word which cannot be extracted by the simple method. In addition, some merged segments might correspond to compound words that represent the meaning of text more clearly than the case without them. The merged segments, in this way, improve the effectiveness of retrieval. As a merged segment overlaps the constituent basic segments, we name a indexing method that uses this new segmentation strategy **overlapping statistical word indexing**.

Because the new strategy generates more segments from text, the index file might become large. Thus, to limit this increase to a reasonable range, the condition mentioned in Modification (2) needs to be introduced. However, if we simply limit the number of segments to be merged, the possibility that non word segments are generated increases and the retrieval effectiveness might deteriorate. To limit the number of segments without decreasing the retrieval effectiveness, we again utilize the segmentation probability: we merge basic segments that any of the probabilities between two neighboring segments are less than a newly introduced merge threshold T_{merg}.[8]

When N-character text $c_1c_2\cdots c_N$ is given, segments are extracted in the following way:

(1) Set $i = 1$ and $j = 1$(the beginning of text).

(2) Find next c_j whose $P_{seg}(c_jc_{j+1}) > T_{seg}$, and extract a segment $c_i\cdots c_j$. If it fails to find such a character, extract a segment $c_i\cdots c_N$ and go to Step (4).

(3) If $P_{seg}(c_jc_{j+1}) \le T_{merg}$, go to Step (2).

(4) Find next c_i whose $P_{seg}(c_ic_{i+1}) > T_{seg}$, and set $j = i$ and go to Step (2). If it fails to find such a character, the algorithm terminates.

[8]It should be noted that this segmentation strategy is not limited to our method using the segmentation probability. It is applicable to other methods such as one using bi-gram's mutual information[24].

Table 2: Parameters used in the experiments.

In-document frequency normalization constant	Kd	0.0,0.2,0.5,1.0,2.0,5.0
Document length constant	λ	0.0,0.2,0.4,0.6,0.8,1.0

When $T_{seg} = 0.10$ and $T_{merg} = 0.20$, the compound word "大使公邸" which the simple strategy fails to segment correctly is processed as follows: the basic segments ("大", "使" and "公邸") and the merged segments ("大使", "大使公邸" and "使公邸") are extracted. That is, two words — the component word "大使"(ambassador) and the compound word "大使公邸" — became to be additionally identified.

5 Performance Evaluation

5.1 Retrieval system

We are now developing an IR system for Japanese documents, and this system is called EXTRA (EXTensible RAnking retrieval system). By using object-oriented technology in implementation of EXTRA, it is able to deal with various indexing methods and ranking models. Thus we used it as a platform for evaluating the indexing method proposed here.

As for a ranking model, we used a probabilistic model originally proposed by Robertson [21] and modified to control the effect of the document length in normalizing the term's in-document frequencies [17]. The relevance value $r(D)$ of a document D is computed as:

$$r(D) = \sum_{t_i \in Query} \log(\frac{N}{df_i}) \cdot \frac{qf_i}{Kq + qf_i} \cdot \frac{tf_i}{Kd(\lambda\frac{L}{L_{ave}} + (1-\lambda)) + tf_i}, \quad (6)$$

where df_i is the document frequency of indexing units t_i, qf_i is the t_i's in-query frequency, and tf_i is the t_i's in-document frequency. Kq and Kd are constants for normalizing qf_i and tf_i, and N is the number of documents in the collection. L and L_{ave} are the length of the target document and the average document length in the collection, and λ is a constant for controlling of the effect of the document length. Note that $\lambda = 0.0$ means that the document length does not contribute the relevance value.

In this experiment, various combinations of Kd and λ shown in Table 2 are evaluated. Note that Kq in Formula (6) is fixed at 0.0, since the retrieval requests are rather short and Kq has no impact on effectiveness.

5.2 Test collection

Our experiment used the BMIR-J1 test collection[13]. BMIR-J1 was provided by the Database Construction Working Group for Evaluation of Information Retrieval Systems, under the SIG Database System of the Information Processing Society of Japan. By permission of the Nihon Keizai Shinbun, Inc., it was constructed using the Nikkei newspaper articles between Sept. 1, 1993 and Dec. 31, 1993. Its statistics is listed in Table 3. Although it is a small collection

Table 3: Statistics of BMIR-J1.

	queries	articles
number of items	60	600
average length (chr.)	12	703
minimum length (chr.)	2	102
maximum length (chr.)	28	3802
total size	—	872KB

as you can see, it is the only one available for evaluation of Japanese IR systems.

Retrieval effectiveness was measured using *recall*, the ratio of the number of relevant documents retrieved to the number of relevant documents in the entire collection, and *precision*, the ratio of the number of relevant documents retrieved to the total number of documents retrieved [5][22]. Furthermore, the amount of extracted indexing units — the number of distinct units and the number of total units — was measured.

5.3 Baseline Results

We first evaluated the performance of the conventional indexing methods: dictionary-based word indexing (Dict-word) and n-gram indexing (N-gram). To implement Dict-word, we used the "Chasen" Japanese morphological analyzer.[9] For evaluating N-gram, we tried several n values: the simple n-gram indexing for $n = 1, n = 2$, and $n = 3$; the multiple n-gram indexing for $n = 1 + 2$ and $n = 1 + 2 + 3$. Notice that the two methods are equivalent when $n = 1$.

Figure 2 shows recall-precision curves. As we have evaluated, for each indexing method, many combinations of Kd and λ and there is not enough space to show all of the results, only the best cases are plotted on the figure (parameter settings are shown in the legend of the figure).

In the simple N-gram, the best performance was achieved when $n = 2$. The reason why the bi-gram indexing performs better is that two-character words are the most frequent in Japanese and the bi-gram indexing captures them without exception. However, this performance was slightly inferior to the best case ($n = 1+2$) of the multiple N-gram. Average precision of 11 recall points [22] of the best simple N-gram was 0.462, and the best multiple N-gram yielded 0.483 (4.5% higher). These results confirmed our hypothesis that one can obtain better performance by combining more than one n-gram indexing. Dict-word was less effective than N-gram because compound words, especially katakana ones, were not properly segmented using the Chasen parser because its dictionary contains about 80,000 words and looks not enough to handle newspaper articles in BMIR-J1.

Table 4 lists the numbers of indexing units. Dict-word extracted the least number of total units and is considered to be the best method in terms of the inverted file size. As for simple N-grams, the number of distinct units increases as n increases, but the number of total units are almost the same because they extract n-grams at almost every character position in every document. Multiple N-grams accumulated

[9]It is one of the most widely-used parsers in Japan and available via http://cactus.aist-nara.ac.jp/lab/nlt/chasen.html.

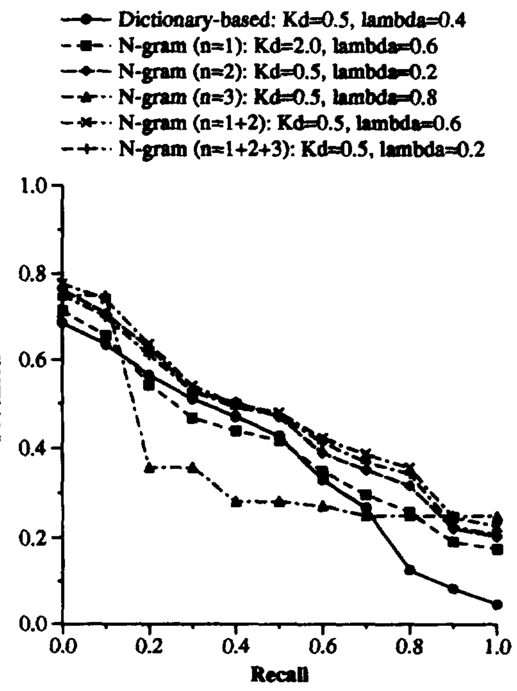

Figure 2: Recall-precision curves for the conventional methods.

Table 4: Number of indexing units for the conventional methods.

	number of distinct units	number of total units
Dictionary-based Word	15,768	236,830
N-gram (n=1)	2,123	438,119
N-gram (n=2)	54,082	437,519
N-gram (n=3)	173,324	436,919
N-gram (n=1+2)	56,205	875,638
N-gram (n=1+2+3)	229,529	1312,557

indexing units by its component indexes: $n = 1 + 2$ doubles and $n = 1+2+3$ triples the number of total units compared with simple N-grams.

5.4 Results of simple statistical word indexing

In evaluating the performance of the simple statistical indexing, we used various values for the segmentation threshold T_{seg}: 0.00, 0.05, 0.10, 0.15, 0.20, and 1.00. The best results (and their parameter settings) for each threshold value are shown in Figure 3. As one can see from this figure, precision was considerably low when $T_{seg} = 1.00$. It corresponds to the simple segmentation based on only character classes, meaning the necessity of introducing the segmentation probability. The performance increased with decreasing T_{seg}, and became highest when $T_{seg} = 0.15$. Further decreases in T_{seg}, however, yielded lower precision. This is just what we would expect from the arguments in Section 4.1. Average

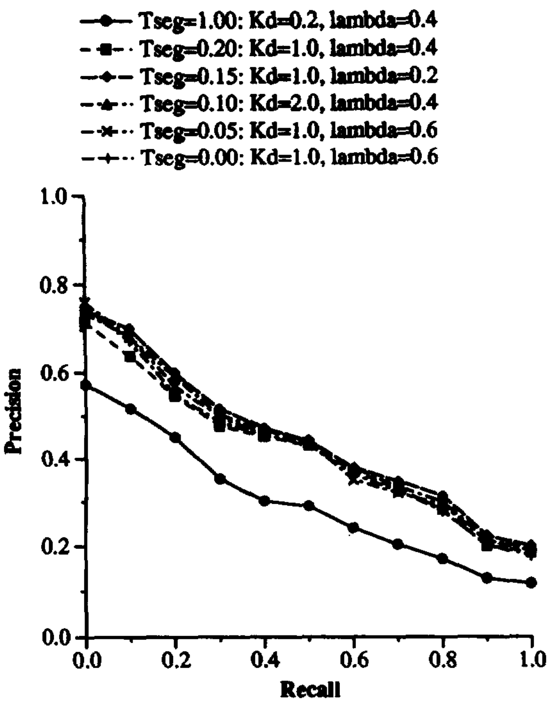

Figure 3: Recall-precision curves for the simple statistical word indexing.

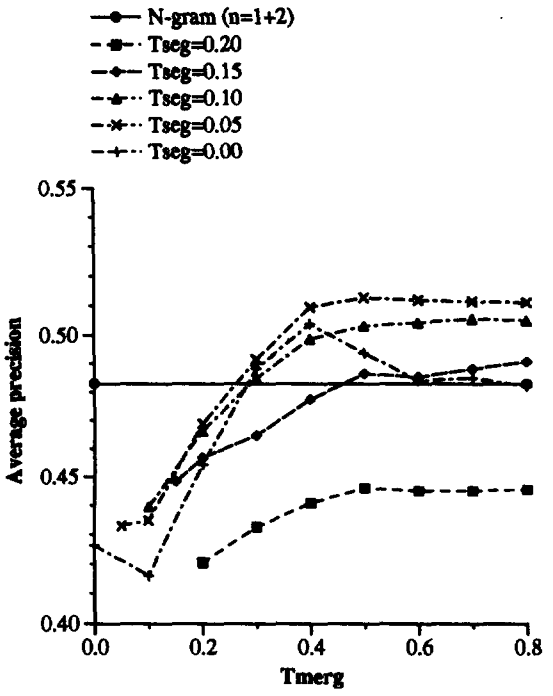

Figure 4: Average precisions for various T_{merg}s.

Table 5: Number of indexing units for the simple statistical word indexing.

T_{seg}	number of distinct units	number of total units
1.00	22,533	271,267
0.20	8,986	300,875
0.15	7,274	317,502
0.10	5,489	337,986
0.05	3,856	359,156
0.00	2,193	392,472

precision of the best case (T_{seg} = 0.15) was 0.448. It was 7.2% worse than the best multiple N-gram, and only 3.0 % worse than the best simple N-gram.

Table 5 lists the numbers of indexing units for the various T_{seg} values. According to the decrease in T_{seg}, text is broken into smaller parts. This is why the number of distinct units decreased while the number of total units increased. Even in the case of extracting the largest number of total units (T_{seg} = 0.00), the number was less than any of N-grams. This means that this simple method outperforms the n-gram indexing from the viewpoint of the index size.

5.5 Results of overlapping statistical word indexing

We measured the performance for various combinations of T_{seg} and merge threshold T_{merg}(0.10, 0.20, $\cdots$ 1.00). It should be noted that only combinations which satisfy $T_{merg} \geq T_{seg}$ were evaluated because no segments are merged otherwise.

Figure 4 represents average precisions for various T_{seg} and T_{merg} combinations. Average precision increased with increasing T_{merg}, and for all T_{seg} reached their peeks around T_{merg} = 0.5. The best performance (average precision = 0.513) was obtained when T_{seg} = 0.05 and T_{merg} = 0.50. This T_{seg} was, as we expected, smaller than the optimal value (T_{seg} = 0.15) for the simple segmentation strategy. Compared to the average precision (0.483) obtained with the best conventional method, N-gram (n=1+2), the proposed method outperforms it by 6.2%. The performance gain increased to 11.0% when compared to the best simple N-gram (n = 2).

The effect of T_{merg} is illustrated in Figure 5, which shows recall-precision curves obtained when T_{seg} = 0.05. With increasing T_{merg}, precision increased at low or middle recall ranges. This is because newly extracted (compound) words helped clearly represent the meaning of text and mainly improved the order of highly ranked documents.

Table 6 lists the numbers of indexing units for various T_{merg} values when T_{seg} = 0.05. As T_{merg} increased, more segments were merged and extracted, and both the number of distinct units and the number of total units increased. The number of total units was, however, still 42% smaller than it was with the best multiple N-gram, and only 16% larger than with the best simple N-gram. That is, compared with the best conventional method, the proposed method showed the better retrieval effectiveness with the smaller index.

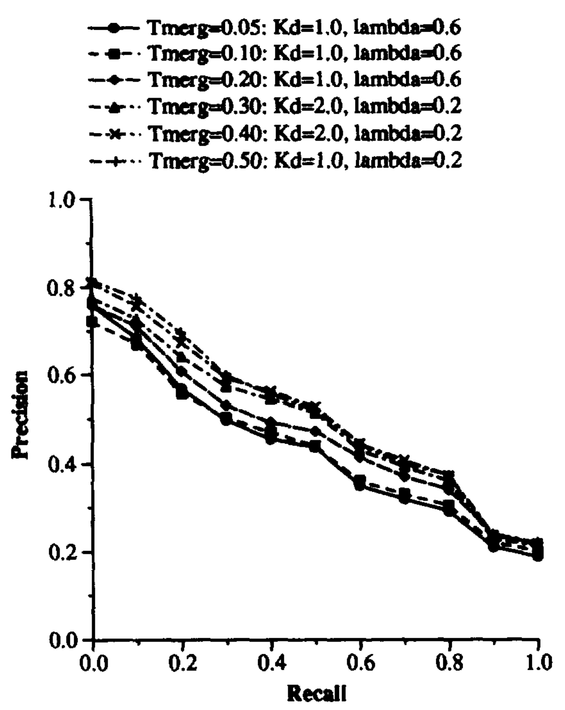

Figure 5: Recall-precision curves for the overlapping statistical word indexing.

Table 6: Number of indexing units for the overlapping statistical word indexing.

T_{merg}	number of distinct units	number of total units
0.05	3,856	359,156
0.10	5,834	377,933
0.20	11,693	416,559
0.30	20,106	460,411
0.40	28,609	490,918
0.50	35,524	510,572

6 Conclusion

This paper proposes a new method for statistical indexing of Japanese documents. We have developed a statistical segmentation method that uses the head and tail probabilities of characters as well as character classes. This method is free from constant maintenance of data as needed in the dictionary-based segmentation method. In addition, because it requires only a small amount of statistical information and computation at segmentation, IR systems can be made compact. We also proposed an overlapping indexing strategy for the statistical segmentation method. This method extracts words that cannot be extracted when the conventional segmentation strategy is used, and it increases the effectiveness of retrieval. It also helps reduce the index size below the size needed for n-gram indexing.

We have implemented the proposed indexing method on EXTRA, which is a retrieval system for Japanese documents, and have evaluated it by using the BMIR-J1 test collection. The experimental results confirmed that the method performs better than the conventional indexing methods — dictionary-based word indexing and n-gram indexing — from the viewpoints of retrieval effectiveness and index size.

References

[1] H. Akama and F. Konishi. Application of frame association to Japanese informatin retrieval. In *Proc. of Int. Workshop on Information Retrieval with Oriental Languages*, pages 27–34, 1996.

[2] T. Akamine and S. Fukushima. Flexible string inversion method for high-speed full text search (*in Japanese*). In *Proc. of Advanced database system symp. '95*, pages 35–42. Information Processing Society of Japan, 1996.

[3] W.B. Cavnar. Using an n-gram-based document representation with a vector processing retrieval model. In *Proc. of 3rd TREC*, pages 269–277, 1995.

[4] L.F. Chien. Fast and quasi-natural language search for gigabits of Chinese texts. In *Proc. of 18th ACM SIGIR Conf.*, pages 112–121, 1995.

[5] W.B. Frakes and R. Basza-Yates. *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall, New Jersey, 1992.

[6] H. Fujii and W. B. Croft. A comparison of indexing techniques for Japanese text retrieval. In *Proc. of 16th ACM SIGIR Conf.*, pages 237–246, 1993.

[7] H. Fujii and W. B. Croft. Comparing the retrieval performance of English and Japanese text databases. In *Proc. of 2nn Workshop on Very Lage Corpora*, pages 87–97, 1994.

[8] K. Furuse, K. Asada, and A. Iizawa. Implementation and performance evaluation of compressed bit-sliced signature files. In *Proc. of 6th Int. Conf. on Information Systems and Data Management*, pages 164–177, 1995.

[9] D. Harman, editor. *The 4th Text REtrieval Conference (TREC-4)*. National Institute of Standards and Technology, 1996.

[10] K. Kageura. Bigram statistics revisited: A comparative examination of some statistical measures in morphological analysis of Japanese kanji sequences. `http://www.dcs.shef.ac.uk/ kyo/llc1.ps`, 1996.

[11] Y. Kawashimo et al. Development of full text search system Bibliotheca/TS (*in Japanese*). In *Proc. of the 45th Zenkoku Taikai, 3*, pages 241–242. Information Processing Society of Japan, 1992.

[12] J.H. Lee and J.S. Ahn. Using n-grams for Korean text retrieval. In *Proc. of 19th ACM SIGIR Conf.*, pages 216–224, 1996.

[13] K. Matsui et al. Test collection for information retrieval systems from the viewpoint of evaluating system functions. In *Proc. of Int. Workshop on Information Retrieval with Oriental Languages*, pages 42–47, 1996.

[14] J.Y. Nie, M. Brisebois, and X. Ren. On Chinese text retrieval. In *Proc. of 19th ACM SIGIR Conf.*, pages 225–233, 1996.

[15] T. Nishino and T. Fujisaki. A stochastic parsing of kanji compound words (*in Japanese*). *Transactions of Information Processing Society of Japan*, 29(11):1034–1042, 1988.

[16] The National Language Research Institute of Japan. Studies on the vocabulary of modern newspapers (in Japanese). Report 37, 1970.

[17] Y. Ogawa. Effective and efficient document ranking without using a large lexicon. In *Proc. of 22nd VLDB Conf.*, pages 192–202, 1996.

[18] Y. Ogawa, A. Bessho, and M. Hirose. Simple word strings as compound keywords: An indexing and ranking method for Japanese texts. In *Proc. of 16th ACM SIGIR Conf.*, pages 227–236, 1993.

[19] Y. Ogawa and M. Iwasaki. A new character-based indexing method using frequency data for Japanese documents. In *Proc. of 18th ACM SIGIR Conf.*, pages 121–129, 1995.

[20] J.M. Ponte and W.B. Croft. USeg: A retargetable word segmentation procedure for information retrieval. Technical Report 96-2, University of Massachusetts, 1996.

[21] S.E. Robertson and S. Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proc. of 17th ACM SIGIR Conf.*, pages 232–241, 1994.

[22] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[23] G. Salton and M. Smith. On the application of syntactic methodologies in automatic text analysis. In *Proc. of 12th ACM SIGIR Conf.*, pages 137–150, 1989.

[24] R. Sproat, C Chih, and W. Gale. A stochastic finite-state word-segmentation algorithm for Chinese. In *Proc. of 34th ACL Conf.*, pages 66–73, 1994.

[25] M. Suzuki. Japanese sentence segmentation algorithm using character patters based on the statistical investigation (*in Japanese*). *Transactions of Institue of Electronics, Information and Communication Engineers, Japan*, J79-D-II(7):1236–1243, 1996.

[26] K. Takeda and T. Fujisaki. Automatic decomposition of kanji compound words using stochastic estimation (*in Japanese*). *Transactions of Information Processing Society of Japan*, 28(9):952–961, 1987.

[27] P. Vines et al. Indexing for Chinese text retrieval. In *Proc. of 1st Australian Document Computing Conf.*, pages 85–89, 1996.