Study on the Combination of Probabilistic and Boolean IR Models for WWW Documents Retrieval

Masaharu Yoshioka Makoto Haraguchi
Graduate School of Information Science and Technology, Hokkaido University
N14 W9, Kita-ku, Sapporo-shi, Hokkaido, JAPAN
{yoshioka,mh}@ist.hokudai.ac.jp

Abstract

In this paper, we describe our information retrieval (IR) system that is used for the NTCIR-4 Web Task A. First, we introduce our IR system, which is based on the probabilistic IR model. This system is quite similar to the Okapi system, and uses both a word index and a phrase index comprising combinations of two adjacent words. Second, we propose a method for clarifying queries that combines the probabilistic IR model and the Boolean IR model. Since it is not easy to construct a Boolean query that covers all relevant documents, a mechanism for clarifying the Boolean query is required. In this paper, we propose "appropriate Boolean query reformulation for IR" (ABRIR) that support Boolean query formation and score documents based on combining probabilistic and Boolean IR models. Finally, we discuss the effectiveness of the method based on the results of experiments.

1 Introduction

In the NTCIR-4 Web Task [2], our team participated in the "Information Retrieval Task" (Task A) . There are two main items used to evaluate this task.

- Our baseline IR system, which is based on the probabilistic IR model.
- We propose "appropriate Boolean query reformulation for IR" (ABRIR) that support Boolean query formation and score documents based on combining probabilistic and Boolean IR models.

2 Our IR System, Based on the Probabilistic IR Model

Our system uses BM25 [4] as a basic probabilistic IR model, and ChaSen [3] as a morphological analyzer to extract index terms. We use a word index and a phrase index comprising combinations of adjacent words [5]. We employ pseudo-relevance feedback by using 5 top-ranked initially retrieved documents, and the Generic Engine for Transposable Association (GETA) tool [1] as a database engine.

We discuss the details of the system in the following sections.

2.1 Indexing Each Document

We used the organizer-prepared "cooked" data (i.e., processed raw data) to make an index for our IR system. Since some portions of the text were garbled and contained unnecessary tags, we normalized the data.

For such text, we checked the text coding by using raw metadata and converting it to EUC data. We also removed unnecessary tags, such as "<NWD:img>," from the text.

After text normalization, we applied the following procedure to extract the word index and phrase index from the text.

1. Morphological analysis
   We converted ASCII text characters into two-byte EUC codes by using KAKASI [2] as a code converter, and ChaSen as a morphological analyzer.

2. Extraction of index terms
   We extracted noun words (nouns, unknowns, and symbols) as index terms. We excluded numbers, prefixes, postfixes, and pronouns from the index terms. We removed "ー" from the end of a term when the length of the term was longer than two katakana characters. All alphabets were then normalized to one-byte ASCII codes and stored in lower case.

3. Extraction of phrasal terms
   We aimed to use compound nouns as phrasal terms, so we extracted phrasal terms from pairs of adjacent noun terms. In this case, we also

[1] http://geta.ex.nii.ac.jp/
[2] http://kakasi.namazu.org/

used prefixes, postfixes, and numbers for extracting phrasal terms [3].

2.2 Pseudo-Relevance Feedback

We used the five top-ranked documents for pseudo-relevance feedback.

However, when we normalized the score by the number of terms existing in the document, some text with fewer terms tended to score highly. For example, when the single query term is "TOEIC", a document containing only the term "TOEIC" scored higher. This may occur, for example, when the title of a page is "TOEIC" and the contents are Macromedia Flash or image objects. Since these documents are not useful for query expansion, we excluded documents containing fewer than four terms from the relevant documents list.

2.3 Term weighting

We used the BM25 weighting formula to calculate the score of each document.

$$\sum_{T \in Q} w^{(1)} \frac{(k_1+1)tf}{K+tf} \frac{(k_3+1)qtf}{k_3+qtf} \qquad (1)$$

Here, $w^{(1)}$ is the weight of a (phrasal) term T, which is a term or a phrasal term in query Q, and is calculated using

$$w^{(1)} = log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)} \qquad (2)$$

where N is the count of all documents in the database, n is the count of all documents containing T, R is the given number of relevant documents, and r is the count of all relevant documents containing T. In addition, tf and qtf are the number of occurrences of T in a document and in a query, respectively, and k_1, k_3, K are control parameters.

The results of term extraction obtained in our system may vary owing to the results of the morphological analyzer. Therefore, we must minimize the effect of this difference. For example, suppose a phrasal term "AB" ("A"+"B") exists. When "AB" is registered in the dictionary of the morphological analyzer, term "AB" is extracted. When "A" and "B" are registered and "AB" is not, terms "A" and "B" and a phrasal term "AB" are extracted. In the latter case, in addition to "AB," terms "A" and "B" are also used to calculate the score. Therefore, phrasal terms should have lower weights than regular terms. For this purpose, we introduced a parameter $c(0 \leq c \leq 1)$ that is used for counting the phrasal terms in a query, where qtf is incremented by c (not 1) when a phrasal term is found.

[3] We did not include numbers for any phrasal index used to submit a result.

For the query expansion, we used Rocchio-type feedback: [6]

$$qtf = \alpha qtf_0 + (1-\alpha)\frac{\sum_{i=1}^{R} qtf_i}{R} \qquad (3)$$

where qtf_0 and qtf_i are the numbers of times T appears in the query and in the relevant document i.

We conducted retrieval experiments using the NTCIR-3 web test collection, and we set $k_1 = 1, K = \frac{dl}{avdl}, c = 0.3\alpha = 0.7$. Here, dl is the length of a document (the number of terms and phrasal terms) and $avdl$ is the average length of all documents. We set $k_3 = 1000$ for initial retrieval and $k_3 = 7$ for final retrieval.

2.4 Retrieval Procedure

The retrieval procedure of our IR system is as follows.

1. Morphological analysis
   We apply an identical morphological analysis process to generate an index of each document and to extract terms and phrases for the query.

2. Initial retrieval
   We apply the query to obtain the top-ranked documents. We set $R = r = 0$ to calculate the score of each document.

3. Pseudo-relevance feedback
   We select the five top-ranked documents as the relevant documents. When this set includes documents that have fewer than four terms, we remove them from the relevant documents list and include successively ranked documents.

   We do not use phrasal terms for the query expansion because they may be too specific for use with pseudo-relevance feedback [5]. When there are many terms in the relevant documents, we select the 300 terms that share the highest mutual information [7].

4. Final retrieval
   We apply the expanded query to obtain the final results.

2.5 Implementation

We implemented our baseline IR system using the Generic Engine for Transposable Association (GETA) tool. Since GETA cannot handle all documents as a single database, we divide the documents into eight subsets. In order to obtain an equivalent score from all databases, we share N, n, and $avdl$ for all of them. We apply a given query to all eight databases and merge the results.

2.6 Evaluation

Table 1 shows the evaluated result for submitted retrieved data (DBLAB-tt-2, DBLAB-ds-2), where documents judged to be "S" or "A" were classified as relevant (we have used this relevance judgement for all subsequent evaluations in this paper). Since we found some bugs after submitting a value and modifying the index (where we mistakenly used numbers to generate phrasal terms), we also included the evaluation results from the debugged system (where the parameter setting is identical to the submitted result). Survey type experiments are conducted with 35 topics selected by the organizers and target type ones are conducted with another 45 topics selected by the organizers.

Table 1. Evaluation of Results from Our System

	AvePrec	RPrec	Prec@10	Prec@20
tt (s)	0.216	0.242	0.394	0.313
tt (t)	0.228	0.247	0.360	0.333
ds (s)	0.189	0.218	0.351	0.359
ds (t)	0.231	0.251	0.371	0.339
tt-d (s)	0.223	0.254	0.411	0.361
tt-d (t)	0.215	0.232	0.344	0.306
ds-d (s)	0.200	0.234	0.383	0.341
ds-d (t)	0.235	0.242	0.378	0.333

"tt": title only, "ds": description only
"tt-d": title only (debugged)
"ds-d": description only (debugged)
"s": survey type, "t": target type
"AvePrec": average precision, "RPrec": R precision
"Prec@10,""Prec@20": Precision at 10, 20 documents

For most cases, our system performs better than average. However, in several cases it has poorer performance than average.

We assume that the quality of phrasal terms used in a query may affect the retrieval performance. For example, topic 0058 uses terms "存在論 (ontology)" = "存在 (onto-)"+ "論 (-logy)" in the title. In contrast, it uses "「存在とはなにか」ついて哲学的観点から…" that includes "哲学的観点 (philosophical aspect)" = "哲学 (philosoph-)" + "的 (-cal)" + "観点 (aspect)" in the description. Since "存在論 (ontology)" is a technical term in philosophy and artificial intelligence, "存在論 (ontology)" is a more appropriate word than "存在 (onto-,existence)." On the other hand, since "哲学的観点 (philosophical aspect)" is more important than "存在 (onto-, existence)", which is a common word, our system tends to neglect "存在 (onto-, existence)."

The difference between these terms causes the quality of the initial retrieved results to vary so that the final results for retrieving the description are worse than average but the final results for retrieving the title are better than average.

Another problem arises from pseudo-relevance feedback with irrelevant and similar document sets. In topic 0006, our system retrieves quite similar documents (NW002999258, NW002999245, NW002999257, NW002999256, NW002999253) that contain formatted record data. Since these five documents have an almost similar term list, our query expansion method generates a bad query. In order to reduce the effect of irrelevant documents, we believe it is better to check for similarity among the top-ranked documents and to remove similar documents from the query expansion. One problem arises from our indexing method. Topic 0034 uses the following three terms "料理 (cooking)," "切り方 (cutting method)," and "名称 (name)" in the title. Since we do not use verbs for indexing, we do not identify "切り方 (cutting method)" as an index term of our system, and so the retrieved results of topic 0034 are poor. There are two possibilities for including "切り方 (cutting method)" as an index term. The first is to include verbs as index terms, while the second is to include phrasal terms made with noun postfixes. Since "方 (method)" is a noun postfix, "切り方 (cutting method)" can be included as the phrasal term "切り (cutting)" +" 方 (method)".

3 Combination of a Probabilistic IR model and a Boolean IR Model for Query Clarification

There are three major IR models: a probabilistic model such as that on which our proposed IR system is based, a vector space model; and a Boolean model [1]. The most distinctive differences between a Boolean model and other models are the assumption about the appropriateness of IR query term selection.

For example, a probabilistic model and vector space model may retrieve documents that do not contain user-specified query terms. In contrast, a Boolean model assumes that the user will select appropriate terms, and it retrieves documents that contain the user-specified query terms.

However, it is not easy to construct an appropriate Boolean query. For example, the user-constructed Boolean query defined in this test collection is not precise enough to retrieve all relevant documents, as we have showed in the retrieval results for the Boolean query.

Therefore, in this research, we propose our new IR system named ABRIR (Appropriate Boolean query Reconstruction for Information Retrieval) based on following two new proposed method.

- A method for constructing a Boolean query that

includes more relevant documents, by using information about relevant documents.

- Amethod for combining a probabilistic IR model and a Boolean IR model.

We discuss the details of each approach in the following sections.

3.1 Construction of a Boolean Query based on Relevant Documents

Since we assume that all relevant documents contain words that the user intends to retrieve, we select words that exist in all relevant documents. In order to remove common words, we use only words that exist in the original query. We construct a Boolean query by using these words with the original Boolean query.

The following procedure is used to construct a Boolean query. Figure 1 shows an example of this process.

1. Selection of Boolean candidate words
   We select all terms used in the original query that also exist in all relevant documents. We construct a Boolean "and" query by using the selected words. In this example, since "A" and "C" exist in all relevant documents, "A and C" is selected as a candidate query.

2. Modification of the Boolean query based on the initial query
   When we have created an original Boolean query, we relax the newly generated Boolean query. When there are one or more words in the initial query that are used within an "or" Boolean query, we expand the generated query by using this "or" information. In this example, since "C or D" exists in an original query, we modify the generated query to "A and (C or D)."

Since description query does not have an original Boolean query, we only apply first step to generate new boolean query.

We think methodologies proposed here is applicable not only for top-N ranked pseudo-relevance documents but also for user selected relevant documents. However, the meaning of this construction procedure is different according to the nature of relevant documents.

When we use user selected relevant documents, meaning of this procedure is simple. Since user selected ones should be included in the retrieved documents, it is necessary to construct a Boolean query that all selected ones can be satisfied.

In contrast, when we use top-N ranked pseudo-relevance documents, meaning of this procedure is different from former one. This method deals with coocurrence pattern of given query terms. In order to discuss this issue, we introduce

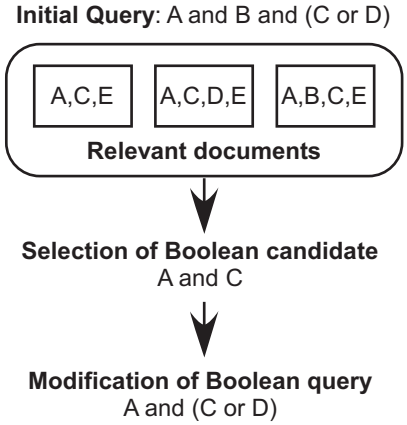

Figure 1. Boolean Query Construction

3.2 Modification of the Score Based on the Boolean Query

When we expand a query by using relevance feedback in the probabilistic IR model, there is a chance that documents without any original query terms will receive a higher score than documents with these terms. Since we assume that documents that do not satisfy the Boolean query may be less appropriate (compared with documents that satisfy the Boolean query), we subtract a penalty score from documents that do not satisfy the Boolean query.

We apply the penalty based on the importance of the word. In a probabilistic IR model, we use this score for calculating a penalty score for each word, since part of equation 1, where $w^{(1)}\frac{(k_3+1)qtf}{k_3+qtf}$, shows the importance of the word in the query. We use a control parameter β to calculate the penalty score.

$$Penalty(T) = \beta * w^{(1)}\frac{(k_3+1)qtf}{k_3+qtf} \qquad (4)$$

For "or" Boolean query, we use the highest penalty from all "or" terms as the overall penalty.

We now describe how to calculate the penalty by using the Boolean query discussed in Figure 1. First, we calculate the penalty score for all words ("A," "C," and "D"). We assume $Penalty(C) \geq Penalty(D)$ in this case. Documents not possessing"A," "C," or "D" terms receive the penalty $Penalty(A)+Penalty(C)$. Documents possessing only the "C" term receive $Penalty(A)$.

3.3 Implementation

We implement ABRIR (Appropriate Boolean query Reconstruction for Information Retrieval) based on our baseline IR system discussed in previous section. The GETA tool has a mechanism for applying the Boolean "and" operation, but not for applying the

Boolean "or" operation by itself. In previous experiments, when the system retrieved the size of the top-ranked documents for each database, we could find the desired size of top-ranked documents for the total database. However, when we apply the Boolean "or" operation on the retrieved results and reject documents from them, the desired size of the top-ranked documents for each database may not be large enough to retrieve documents of the desired size from the entire database.

3.4 Evaluation

We also apply ABRIR to NTCIR4 web test collection. We construct initial Boolean queries from topic description for title retrieval task. When given terms are split into two or more index words by ChaSen, we use the last phrase for an initial Boolean query in order to avoid constructing complicated Boolean queries. For example ("利用者 (利用 (use)-者 (-er))" or "新人研究者 (新人 (new)-研究 (research)-者 (-er))") is a query described in the topic, an initial Boolean query is ("!c 利用者 (user)" or "!c 研究者 (researcher)") [4].

We submitted results based on query reconstruction method and modification of score method (DBLAB-tt-1, DBLAB-ds-1). However, since some bugs exist in the IR system that is used for generating the submitted results, the retrieved results are almost the same as the results obtained using the probabilistic IR model. Therefore, we conducted a new retrieval experiment to confirm its effectiveness. Table 2 shows the results of this experiment.

Table 2. Evaluation Results for Our System with Boolean Construction

	AvePrec	RPrec	Prec@10	Retrieved
tt-b (s)	0.200	0.236	0.431	1843
tt-b (t)	0.218	0.255	0.371	1451
tt-o (s)	0.153	0.184	0.374	1685
tt-o (t)	0.212	0.247	0.378	1390
ds-b (s)	0.155	0.196	0.370	1327
ds-b (t)	0.220	0.246	0.387	1166

"tt-b": title only with Boolean construction
"tt-o": title only by using original Boolean query
"ds-b": description only with Boolean construction
"Retrieved": number of relevant retrieved documents

Since more documents are retrieved by using the Boolean constructions than by using the original user-constructed Boolean query, we have confirmed that the original Boolean query is stricter than the constructed query. There were 158 more "Retrieved" documents from the survey task (1843-1685 from 35 topics: 3893 relevant docuemnts), and 61 more from the target task only (1451-1390 from 45 topics : 2891 relevant documents). This strategy is useful mainly for survey-type retrievals.

When we compare the results with those from the probabilistic IR model only, this system performs worse for "Average Precision" and "RPrec" values. This problem arises owing to the difference in the number of relevant retrieved documents (ones for the debugged system; tt-d (s) 2166, tt-d (t) 1843, ds-d (s) 2177, and ds-d(t) 1616), and implies that the given Boolean query is not precise enough to represent the user's retrieval intention.

Figures 2 and 3 show the recall-precision graph of the retrieved results using different Boolean queries for survey type retrieval. This method improves the performance precision, especially for smaller recall value. We assume our Boolean queries works well to reduce the side-effects where a more important word tends to mask the existence of another important word. However, this restriction is too strict to collect all relevant documents.

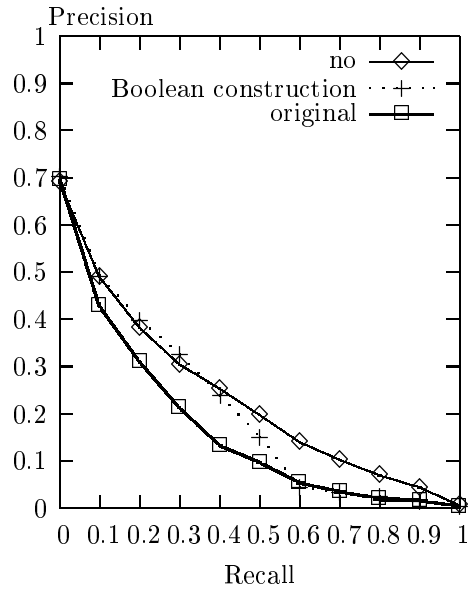

Figure 2. Recall-Precision Graph for Different Boolean Queries (title only, survey)

We have also conducted retrieval experiments using the score modification method. Since constructed Boolean queries perform better than original queries, we use them for calculating the penalties. Table 3 shows the results of this method with different β values.

From this result, we confirmed that the penalty calculation improves the retrieval results. Since we conducted the experiment using a 'buggy' IR system to determine the value of β, the chosen parameter value $\beta = 0.2$ is not ideal. In the title-only experiment,

[4] "!c" is a prefix for phrase index.

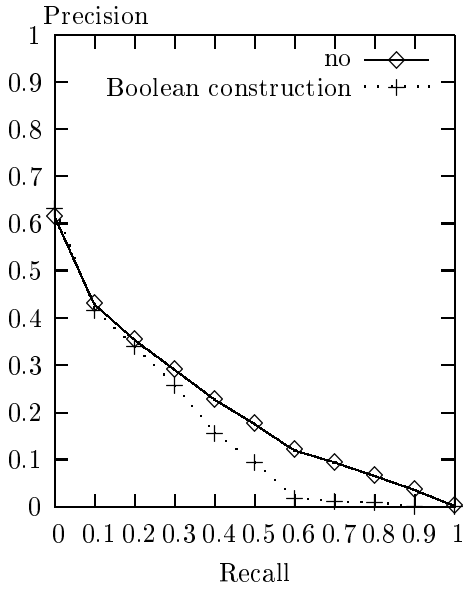

Figure 3. Recall-Precision Graph for Different Boolean Queries (description only, survey)

Table 3. Evaluation Results for Our System with Penalties

	AvePrec	RPrec	Prec@10	Prec@20
tt-0.2 (s)	0.229	0.255	0.420	0.364
tt-0.2 (t)	0.225	0.249	0.376	0.334
ds-0.2 (s)	0.207	0.235	0.391	0.349
ds-0.2 (t)	0.226	0.244	0.381	0.343
tt-1.0 (s)	0.241	0.263	0.431	0.376
tt-1.0 (t)	0.239	0.259	0.394	0.348
ds-1.0 (s)	0.218	0.242	0.389	0.346
ds-1.0 (t)	0.238	0.251	0.385	0.341
tt-2.0 (s)	0.241	0.265	0.429	0.380
tt-2.0 (t)	0.241	0.260	0.389	0.348
ds-2.0 (s)	0.211	0.237	0.394	0.346
ds-2.0 (t)	0.234	0.251	0.388	0.341

"tt-β" : title only $\beta = 0.2, 1.0, 2.0$,
"ds-β" : description only $\beta = 0.2, 1.0, 2.0$,

the case where $\beta = 2.0$ offers the best performance. In contrast, the results for the case where $\beta = 1.0$ in the description-only experiment has better performance than that of $\beta = 2.0$. We assume this difference comes from the quality of the given Boolean query, because our constructed Boolean query used for the description offers worse performance than the titles query, in terms of the relevant retrieved document sizes. Therefore, we think the estimation of an appropriate value for β should be based on a user model which has information how correctly a user may describe a Boolean query.

Figures 4 and 5 show recall-precision graphs of the retrieved results for different values of β for survey type retrieval. This method improves performance, especially for recall values of 0.1 to 0.6. From this result, we believe this method is useful for reducing the side effects where words of higher importance tend to mask the existence of other important words, without filtering relevant documents that do not satisfy the Boolean query.

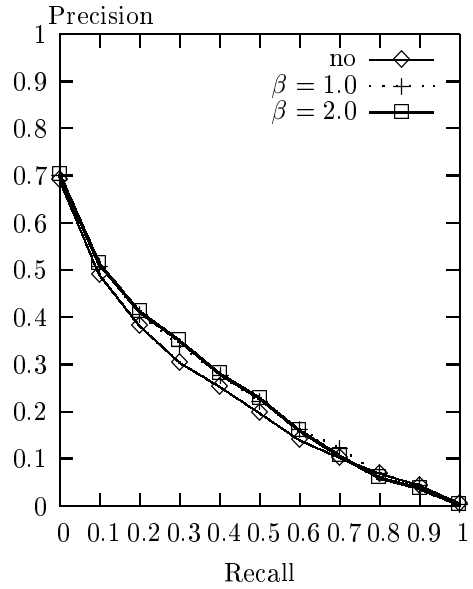

Figure 4. Recall-Precision Graph for Different β (title only, survey)

4 Conclusion

In this paper, we have presented our IR system, based on Okapi with a compound noun index. This system performs better than average over the NTCIR-4 web test collection. We also proposed a new method for combining a probabilistic IR model and a Boolean IR model, and verified this method using the NTCIR-4 web test collection. We confirmed that a user-constructed Boolean query is not precise enough to represent the retrieval intention, and proposed a

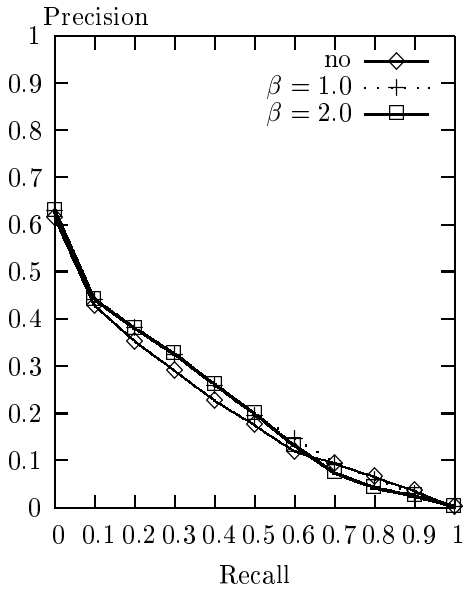

Figure 5. Recall-Precision Graph for Different β (description only, survey)

method for constructing Boolean queries based on relevant documents to improve the retrieval performance. We also confirmed that calculating a penalty based on the Boolean query improves the retrieval performance. Our system performs very well, especially for survey-type retrieval.

For future work, we plan to use a thesaurus for constructing more expressive Boolean queries.

Acknowledgments

We would like to thank the organizers of the NTCIR Web Task for their efforts in constructing this test data. This research was partially supported by a Grant-in-Aid for Scientific Research on Priority (2), 15017202 from the Ministry of Education, Culture, Sports, Science, and Technology, Japan.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[2] K. Eguchi, K. Oyama, A. Aizawa, and H. Ishikawa. Overview of the informational retrieval task at ntcir-4 web. In *Working Notes of the Fourth NTCIR Workshop Meeting*, 2004. http://research.nii.ac.jp/ntcir-ws4/NTCIR4-WN/WEB/NTCIR4WN-OV-WEB-A-EguchiK.pdf.

[3] Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, H. Matsuda, K. Takaoka, and M. Asahara. *Morphological Analysis System ChaSen version 2.2.1 Manual*. Nara Institute of Science and Technology, 2000.

[4] S. E. Robertson and S. Walker. Okapi/Keenbow at TREC-8. In *Proceedings of TREC-8*, pages 151–162, 2000.

[5] M. Toyoda, M. Kitsuregawa, H. Mano, H. Itoh, and Y. Ogawa. University of tokyo/ricoh at ntcir-3 web retrieval task. In *Proceedings of the Third NTCIR Workshop on research in information Retrieval, Automatic Text Summarization and Question Answering*, 2002. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings3/NTCIR3-WEB-ToyodaM.pdf.

[6] M. Uchiyama and H. Isahara. Implementation of an IR package. In *IPSJ SIGNotes, 2001-FI-63*, pages 57–64, 2001. (in Japanese).

[7] M. Yoshioka and M. Haraguchi. Construction of personalized and purpose-oriented thesaurus. In *Proceedings of Asian Association for Lexicography '03 (ASIALEX)*, pages 461–466, 2003.