NAIST-IS-DT0061201

Doctor's Thesis

Information Extraction and Retrieval Techniques for Task-Oriented Information Recommendation Systems

Hiromi itoh Ozaku

June , 2004

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

Doctor's Thesis
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
DOCTOR of ENGINEERING

Hiromi itoh Ozaku

Thesis committee: Masatsugu Kidode, Professor
Shunsuke Uemura, Professor
Yuji Matsumoto, Professor
Yasuyuki Kono, Associate Professor

Information Extraction and Retrieval Techniques for Task-Oriented Information Recommendation Systems*

Hiromi itoh Ozaku

Abstract

This thesis addresses issues related to retrieving information from electronic resources, such as the World Wide Web (WWW), whose text comprises diverse styles ranging from well-formed (e.g. news articles, in which text is scrupulously reviewed) to ill-formed (e.g. bulletin boards, chat-groups and personal diaries, which may include colloquial expressions, spontaneous outbursts, obsolete and inconsequential information).

For an idealized retrieval system architecture, I assume a kind of recommendation system. This architecture can retrieve information from electronic resources that is specifically needed by users, according to the "popular wisdom" associated with a particular task and without complicated procedures. Therefore, the recommendation system should be able to retrieve, correctly and efficiently, information related to the task from a wide variety of sources. The system also needs knowledge, in the form of a special database, about the task, to understand the users' needs and bases of recommendation as well as a support method to enable users to input keywords without hesitation.

Although many recommendation systems are available online, they do not consider users' purposes and informational contents. Traditional systems show only several documents with the users' input or statistically significant information. These traditional systems do not meet the users' needs. On the other hand, while many information retrieval systems are also available online, these too have several problems. For example, although online information retrieval systems

*Doctor's Thesis, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DT0061201, June , 2004.

provide very fast retrieval, the results are unsatisfactory because they show too much information. The precision rate of these systems is inadequate. One reason for this is that information retrieval technologies have been developed to retrieve information from well-formed text data, whose features are quite different from those of ill-formed electronic resources. The capabilities of retrieval technologies should therefore be checked for various features of electronic resources. Furthermore, there are problems of usability in current information retrieval systems. When users work with existing systems, they generally input certain keywords that are suited to their purpose. However, when they are unsure about the information they need, inputting keywords can be difficult. If they input inappropriate keywords, the retrieval systems do not work accurately. The systems should help users to input keywords that are appropriate for their intentions.

In this thesis, I describe a task-oriented information recommendation system to solve the problems mentioned above. To realize the proposed system, I have developed three modules: the analysis, retrieval, and selection modules.

I first investigate the performance of extraction and categorization techniques for electronic text resources other than well-formed text data. To develop two prototype support systems, I show the capabilities of extraction and categorization techniques, and propose new methods using these techniques from dynamic and ill-formed text data for the analysis module. Next, to exploit well-known techniques of information retrieval, I examine how to treat a huge body of data having various features, such as the WWW. The ability of several information retrieval techniques to process WWW data is also investigated, and an efficient retrieval method which I call the "SCORE method", is proposed for the retrieval module. I then focus on the task of recommending tourist routes, and propose a support method of inputting keywords for the selection module. I evaluate the method by extracting information for a recommendation system database, and show its effectiveness in supporting users retrieving information, from the perspective of usability.

Keywords:

Recommendation System for tourist routes, Information Retrieval, Keyword Extraction, Support tool for selecting queries

Contents

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

The World Wide Web (WWW) has become very popular since the late 20th century and the number of WWW documents has increased dramatically. Along with the popularization of the WWW, electronic texts can be readily obtained and a variety of users can access the WWW. Therefore, implementation of easy-to-use information retrieval systems is necessary for users.

Information retrieval technologies feature the latest breakthroughs using extraction and categorization technologies. Although many information retrieval systems which employ these technologies, are available, they have several problems.

First, although online information retrieval systems provide very fast retrieval, the results are unsatisfactory because they show too much information. One reason for this is that the technologies in the information retrieval systems have been developed in the area of efficiently using information from polished and well-formed documents, such as newspaper articles, and the features of these documents are quite different from these of other electronic text resources such as WWW data, e-mails, and network news. The WWW data comprise diverse styles ranging from well-formed to ill-formed. The well-formed data mean scrupulously reviewed data such as newspaper articles, and the ill-formed data comprise bulletin boards, chat-groups, and personal diaries, which may include colloquial expressions, spontaneous outbursts, obsolete, inconsequential information and so on. It is possible that these technologies, which work for the well-formed data, are ineffective for retrieving information from the ill-formed data. Therefore, these extraction and categorization technologies should be applied to the ill-formed data and their abilities should be tested from many directions. Second, problems of usability also exist in current information retrieval technologies. In general, when users use the available systems, they input certain keywords. However, when they are unsure about the information they need, recalling appropriate keywords is generally difficult. Even if appropriate keywords can be inputted easily, users have to search for information repeatedly to get enough needed information. Furthermore, users may need only a part of the information in a certain homepage, rather than the entire homepage. Using traditional information retrieval systems, users must first retrieve entire homepages and then extract the desired information from them. Therefore, searching is a time-consuming task. Ideally, the system should consider the users' purpose and extract only those parts of the information that are appropriate. A usable information retrieval system, which suits the users' purpose, needs to be created. For an easy-to-use information retrieval system, a new-style recommendation system should be developed.

1.2 Aim of the Thesis

The goal of this thesis is to design and explore the feasibility of an easy-to-use information retrieval system which applies extraction, categorization, and retrieval technologies, from ill-formed data such as the WWW data. By developing an easy-to-use information retrieval system, this thesis contributes to narrowing the digital divide, and to offering convenient WWW environments.

An easy-to-use information retrieval system has the following conditions:

- to retrieve target information with accuracy and efficiency;
- to retrieve information with few queries and interactions;
- to support task-oriented initial information development;
- to retrieve and organize up-to-date task-oriented information; and
- to recommend information which suits the users' needs.

If an information retrieval system can apply extraction, categorization, and reduction technologies suitably for handling data, it can fulfill these conditions. Easy-to-use information retrieval systems may then be developed.

To sharpen my aim, I propose a task-oriented information recommendation system with an example of an easy-to-use information retrieval system. In this thesis, I aim for the establishment of information extraction and retrieval techniques to develop a task-oriented recommendation system.

The task-oriented information recommendation system can show users results suited to a certain task, without complex procedures. The system has a database of "popular wisdom" (knowledge) associated with a particular task, and can help users search for information suited to their needs according to the database.

The key to achieving the proposed system is how to deal with information of the system, i.e., to analyze information suited to the users' needs, to retrieve information from ill-formed data, to use such information to show users the information and finally, to develop a database.

First I confirm the ability of the extraction and categorization technologies to analyze information suited to the users' needs from ill-formed data. I introduce two prototype support systems: one supports topic discovery, and the other supports conference program production for ill-formed data. These systems have been developed to analyze ill-formed data in detail and to retrieve information with a high degree of accuracy. I propose new methods of extraction and categorization from dynamic and ill-formed text data, and evaluate the performance of my proposed methods. Secondly, I confirm that the technologies are applicable to the huge amount of and variously styled documents on the WWW. I propose an efficient retrieval method, thermed the "SCORE method" for documents on the WWW. In response to the results of the first and the second experiments, I discuss possible development of the task-oriented recommendation system as an example of an easy-to-use system. Specifically, a sightseeing route production support system is discussed, and I examine how to fulfill the conditions for an easy-to-use information retrieval system. Finally, a support method to retrieve sightseeing information is proposed, and the efficiency of my proposed method is shown.

1.3 Design of Task-Oriented Recommendation Systems

When users retrieve information, inputting appropriate keywords so that they can extract information from the retrieval system accurately is important. However, it is difficult for users who do not have a concrete idea of which keywords are appropriate. I therefore propose a task-oriented recommendation system.

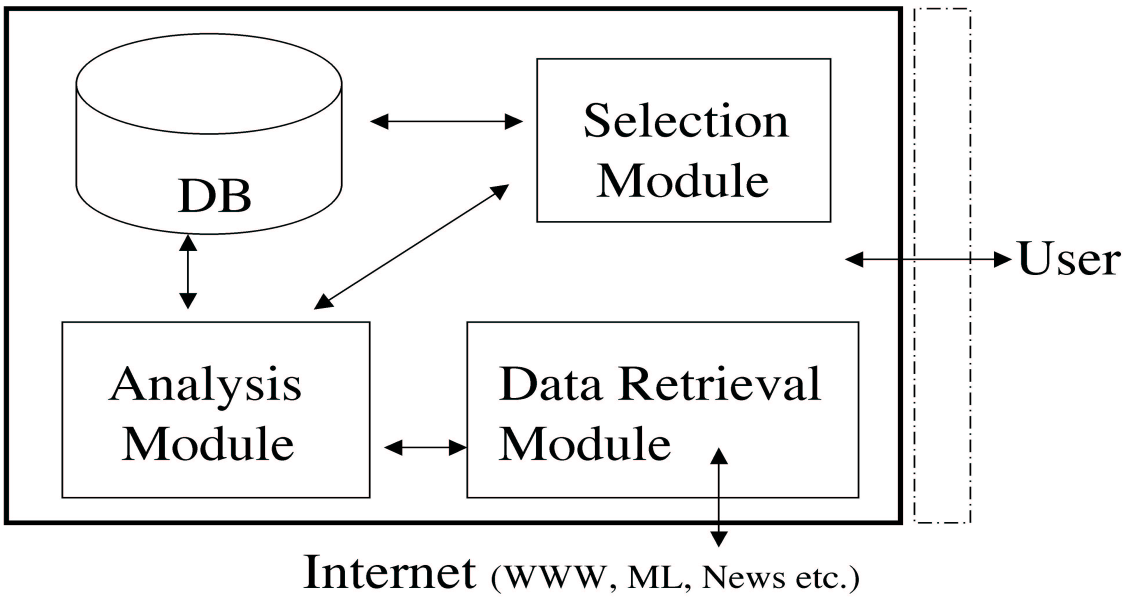

Figure 1.1. Task-oriented recommendation systems

The architecture of the task-oriented recommendation system is shown in Figure 1.1. This system has three modules: analysis, retrieval and selection module. Users need only to select some keywords from query candidates calculated by the input support module to get the necessary information about a certain task.

For example, if a user wants to watch a movie, he/she checks information about which movie has received the best reviews, and which movies are currently showing, and so on, before deciding which movie to see. Next he/she checks which theater is showing the selected movie, and how to get to the theater. I call the above process, which involves several retrieval procedures, “a task”. This process

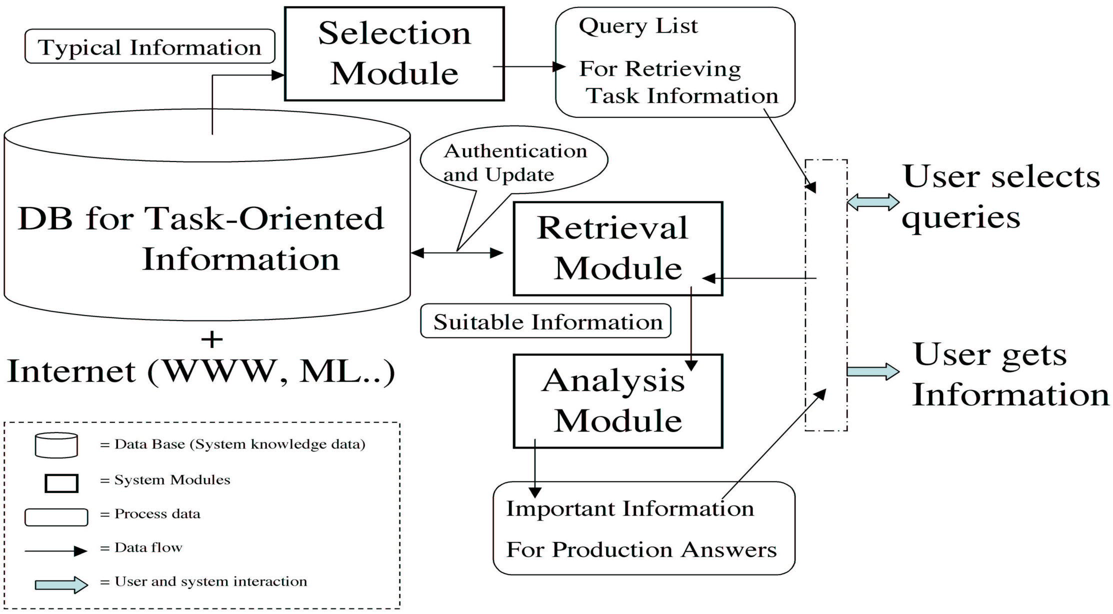

Figure 1.2. Data flow of a task-oriented recommendation system for users

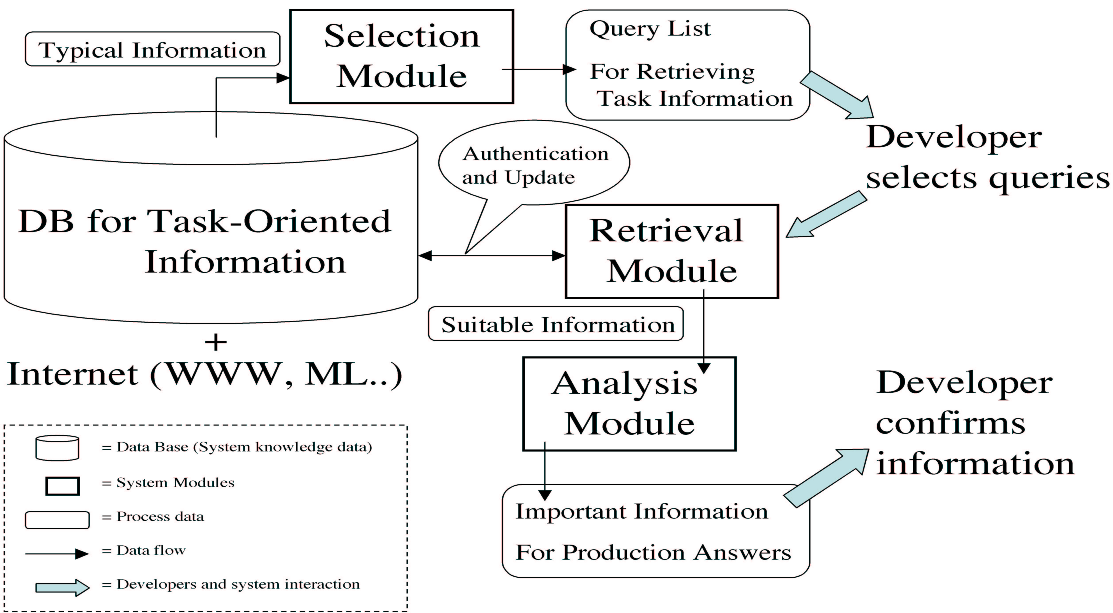

Figure 1.3. Data flow of a task-oriented recommendation system for developers

is very laborious, so the system should free users from having to carry out the process.

The data flow (Figure 1.2) of this system are follows:

- a user gets a query candidate list from the selection module;
- he/she selects some keywords from the list;
- the system retrieves information related to the task from the retrieval module;
- the analysis module extracts important information from the retrieved information;
- and he/she gets results according to his/her needs.

The key point of this system is useful not only for users, but for developers, to support a task-oriented initial information development and to retrieve and organize up-to-data task-oriented information shown in Figure 1.3.

In this thesis, I have experimented with methods of each module that meets conditions of an easy-to-use information retrieval system.

1.4 Outline of the Thesis

The thesis is organized as follows.

In Chapter 2, I introduce an overview of studies related to the issues about information retrieval research for various electronic data. Problems of information retrieval from electronic data are discussed.

In Chapter 3, I introduce two prototype systems that are able to find a topic and to produce a conference program, applying extraction, categorization, and retrieval techniques for ill-formed data. In the development of these support systems, I show that they can retrieve ill-formed data which have different features than those found in well-formed data, and can utilize these techniques as the analysis module of the task-oriented recommendation system shown in the lower left-hand of Figure 1.1.

In Chapter 4, the features of WWW data are discussed, and effective information retrieval techniques from a huge amount of data are explained. I perform a WWW retrieval task in a competitive conference. Using the results of this conference, I investigate the features of WWW data and discuss the ability of retrieval techniques to retrieve information from a huge amount of data. I propose the "SCORE method" as a retrieval method from a huge amount of data for the retrieval module of the system.

In Chapter 5, I concretely propose a task-oriented information recommendation system. The term "task" has many meanings, but in this work I focus on supporting the production of tourist routes as the "task." A support system as a task-oriented information recommendation system is introduced, and features of the task related to the production of tourist routes are discussed. In developing the task-oriented information recommendation system, it is important to retrieve the task-oriented information from the Internet with accuracy. To do so, selecting appropriate queries is very important. I therefore introduce a new method to support selecting appropriate queries for the user input support module (upper right-hand of Figure 1.1). The efficiency of the method is estimated in detail, and the process of retrieving information as bases of recommendation for the answer production module is explained.

Chapter 6 concludes this work and outlines directions for including a sample of an interface in future research.

Chapter 2

Overview of Related Studies

As computer systems have progressed, the opportunities for people to use them have increased. However, as the Internet has gained popularity, electronic data have been thrown into disarray as the result of increasing abundance. For example, the WWW data include diverse styles documents from polished and reviewed data as commercial messages and newspaper articles, to unrefined and freewheel data as bulletin boards, chat-groups and personal diaries. The computer systems should address variety of data.

Systems using natural language processing (NLP) technologies, such as extraction, categorization and retrieval have been developed for a long time, and are extremely precise to analyze documents. If these technologies apply suitably to practical applications, the applications must have been easy to help users getting needed information.

In Section 2.1, I introduce systems of supporting to find important information from unprocessed text resource, using extraction and categorization techniques. The rapid spread of the WWW has led to users demanding techniques for retrieving information from the WWW. In addition, many contest-type international conferences have been held to evaluate WWW retrieval systems. Current WWW retrieval techniques are briefly described in Section 2.2. The usage of the WWW has also been somewhat modified as retrieval technoiques have progressed. The historical context to WWW and to retrieval technology related research, as well as the current usages of the WWW in the latest mode are given in Section 2.3.

2.1 Support Systems for Getting Target Information

Major extraction and categorization techniques have been developed in the field of newspaper articles. With the development of extraction techniques, applied research for new electronic data other than newspaper articles has also progressed. In particular, the analysis and utilization of network news, e-mail, and the like are major targets for research.

The research of an "Automatic Digesting System" is well known as a successful system that applies extraction and summarization techniques [49]. However, although this system has used many heuristic rules of certain news groups, its application range is very limited. The Automatic Digesting System cannot deal with all articles that have abundant expressions even in one network news group.

Research exists to develop visual systems in relation to network news articles [43, 62]. These systems are effective in keeping up on trends of network news articles, but cannot interact directly with users and respond to their needs. Although there are plenty of other researches using network news articles, these researches are to understand exactly traditions of important sentences and relation structures of network news articles for extraction information [4].

"Knowledge Discovery System" research is available for e-mail users [3, 58]. This type of system is effective in extracting certain formatted data such as address information, meeting information, etc. This system also makes improvements using many heuristic rules, but has not been applied extraction and classification techniques effectively. On the other hand, several support systems have been developed [6, 14]. However, these systems are applied only to text classification and require human assistance on most processes; moreover, the accuracy of classification cannot be evaluated in detail.

Chapter 3 describes support systems that apply extraction, categorization and retrieval techniques. In addition, the abilities of the techniques are examined by questionnaires.

2.2 WWW Retrieval Systems

A huge amount of data is stored in the WWW. If factors affecting usability of a WWW retrieval system are considered, a quick response is of prime importance. There are two ways to achieve a quick response: one is to speed up the retrieval process, and the other is to divide data suitably for efficient retrieval.

The latter may be the more feasible way. Methods for merging results retrieved from multiple databases have been studied since 1995 in TREC[1][18]. In general, the most accurate method is the score normalization method. Many methods other than score normalization have been developed, such as the Interleaved method[60], the Raw Score method[10], the Feature Distance Ranking method[12]. These methods have been used for retrieval from a huge amount of data. In these cases, however, the data were well-formed data such as public documents. Furthermore, some researches pursue speed up in retrieval process such as Google, but they sacrifice the retrieval accuracy.

Researches to pursue precise retrieval are focus on retrieval queries [48, 61]. They are good to improve the retrieval precise. However, they do not make consideration to usability. The researches to gather only current information[50] and to retrieve only information suited a certain purpose[2] exist for achieving quick response and high accuracy. However, they do not show the objective precision rate.

My research is thought that retrieval precision is more important than retrieval speed, however it is not a way to ignore retrieval efficiency. To treat a variety of formed data such as WWW data, various methods have been developed in the recent TREC. However, these methods can only deal with in English. I investigated whether they can be used when retrieving Japanese texts. To do this, I join the 3rd NTCIR, which is a contest-type conference in Japan. The task using WWW data was carried out for the first time at the 3rd NTCIR. The results at the 3rd NTCIR are given in Chapter 4. I show the objective precision rate of my system.

[1] http://trec.nist.gov

2.3 Recommendation Systems

The following Information Retrieval Systems have evolved on the WWW [26]:

1. Human-wave tactics systems (1994-), such as Yahoo!, LookSmart
2. Large-scale systems (1996-), Lycos, AltaVista, Goo
3. Selection systems (1998-), Google, BIGLOBEsearch
4. Objective systems (2000-), Research Index, MySimon

Recently, the site of objective systems type have been increasing. These sites can be divided into two types, "Portal Sites" and "Domain Dependent Sites." A portal site is a site that lists several homepages of similar sites, for example, "JEITA Linguistic Site" for listing project names and related to language resources; "Eiga Portal" for listing information about movies, DVDs, and games. A domain-dependent site is a site developed for a certain purpose, for example, "駅すぱーと "to check train schedules, "CiteSeer" to search technical reference information, and "Amazon.com" for the distribution of books and DVDs. On either type of site, users usually find several homepages, extract a part of the information from each homepage, and consolidate several parts of the information fitting their interests, such as a user search for DVDs on Amazon.com. Then he/she may want to watch a movie instead of a DVD. In this case, the user checks the movie information and the theaters that play the movie on the Eiga portal. Next, he/she checks how to get to the theater, and checks the train schedule for the appropriate times to be able to get to the theater. Finally, he/she can go watch the movie.

I call this retrieval process "a task". I propose a task-oriented retrieval system. This system should be easy-to-use, able to show retrieval results using minimal inputs, and also able to process several retrieval results into a result suiting the task. This system is very similar to recommendation systems [13, 17]. These recommendation systems, however, have knowledge to recommend as a database, the database is updated manually. Furthermore, these systems do not care about users' intentions. These systems did not have ways that analyze users' inputs what their means. Then I think if the system can analyze users' inputs and purpose in detail, recommendation systems can match their need, and improve more easy-to-use.

In this thesis, I explain how to extract and retrieve information suited a task, how to treat the information in my proposed system for users use the system easily.

Chapter 3

Extraction and Categorization Techniques from Ill-formed Data

3.1 Introduction

To develop a task-oriented recommendation system, information handling technologies are needed. There are varieties of style information on the Internet, I focus on text information. Then, for handling text information, various technologies related Natural Language Processing (NLP) are needed. The NLP technologies have evolved dramatically in these days. However, when these technologies are made use of data on the Internet, they meet varied problems. Because the Internet data have varieties of style data including illegal, incorrect form, irregular cord and so on.

I have developed two experimental systems for handling special information from dynamic and ill-formed data such as network news, and for production of a conference program.

In this chapter, I apply extraction, categorization and retrieval techniques to find topics from network news that include varieties of style data, and to produce a program for an annual meeting of the Association for Natural Language Processing. I propose methods to find target information from dynamic and ill-formed data, and show the efficient of my proposed methods.

3.2 Supporting Special Topic Discovery

In this section, I propose a Helpful Information Selection by Hunting On-line (HISHO[1]) system for retrieving information and showing topics from network news. This system can extract network news articles of interest to a user without requiring the use of keyword-like queries. In contrast to ordinary information retrieval and abstract generation systems, this system uses an "information context" to select articles from news groups on the Internet. This function is called a "Topic Search," and helps users grasp the thread of discussions using extract, categorization and retrieval techniques.

3.2.1 Features of Network News

In this section, I explain the difference of features between newspaper articles and network news articles.

Network news articles on the Internet are important information sources from which I can often obtain information relevant to my interests. During the last decade, the WWW on the Internet has become very popular. Even though homepages are often used nowadays for announcements instead of news articles, network news is heavily used and large numbers of articles are posted [63]. Network news has been thrown into disarray as a result of articles being posted in large quantities, both by inexperienced users posting articles irrelevant to some news groups, and by cross-posting, which causes articles to be sent around to many different news groups.

Two kinds of network news groups exist [43]: newswire-like groups, which I call the "announcement" type, and groups which facilitate discussions among users, which I call the "discussion" type. The discussion-type groups provide a huge number of articles related to a wide range of topics in each news group. These articles are usually written like dialogues, and topics, as well as keywords, change every day. Therefore, finding appropriate keywords and their synonyms is often quite difficult.

Nonetheless, much useful information in network news exists, and extracting necessary information from network news can be crucial to users. I have thus

[1] HISHO also means "secretary" in Japanese.

started to build a system that extracts Japanese network news articles which fit a user's interests, especially for articles in the discussion-type groups. An Internet news article consists of two parts: a "Header" area and a "Message" area [21]. A "Header" area consists of fields such as "Message-ID," "Reference," and "Subject." These fields usually identify the author, the domain in which the article belongs, and the posted date. Furthermore, they show the relations between articles, called references, when users reply to articles. The articles in the discussion-type groups are written like dialogues; therefore, many of them contain very little information. For example, some one-line messages contain only a joke or a brief indication that the user agrees or disagrees with something. Searching all of these messages is therefore not productive. Fortunately, users generally fill in the "Reference" field with some "Message-ID" automatically when they quote or reply to a certain article. HISHO analyzes the relations between groups of articles by comparing the "Reference" field and the "Message-ID".

The relations form a tree structure of articles, which I call a Reference Tree (RT). The minimum unit of comparison the HISHO should deal with is a group of articles or messages; that is, a Reference Tree (RT). Even when the information in the "Reference" and "Message-ID" fields is used, there are some articles that are related to an RT, but are not part of the RT. Often these are 'summary messages' explicitly summarizing a long thread, but having no 'header information' to connect them to the appropriate RTs because they were not written as direct responses to other articles. The articles summarized in a thread have usually expired as a result of the delay between their posting and the posting of the summary article. Some users also intentionally cut off the "References" of a summary article. Such deletion can also occur when beginners make mistakes or fail to use the follow-up command. Another kind of unconnected article is caused by cross-posting. A complete RT cannot then be built because the related articles cannot be found in the current news group. I believe that the summary articles are very important to Internet users, and that cross-posted articles are sometimes necessary to understand the line of a discussion. As stated previously, network news articles have special features different from newspaper articles.

3.2.2 Topic Searching

I have developed the HISHO system to search discussion-type groups for articles relevant to a user's interests without requiring the use of input queries.

In this section, I explain the function of HISHO to extract "information context" from network news in discussion-type groups. This function is called "Topic Search." Topic Search includes two processes: categorization and collection. The categorizing process can find a certain topic area in one "information context", and the collection process can gather "information context" related to user interests. The "information context" represents a group of network news. This structure is very similar in the RT that is mentioned in Section 3.2.1. So the RT can treat as the "information context". The categorizing process can find a certain topic area in one RT, and the collection process can gather RTs and articles related to user interests. Term-weighting methods are generally used for the categorization and collection of texts. Although morphological analysis is very useful in getting the best terms, I do not use it because network news has features different from newspaper articles, and were therefore unanalyzable by morphological analysis [2]. HISHO therefore uses *keywords* as character strings of Kanji, Katanaka, letters, or numbers. In the next section, I explain how to extract key strings what is called *keywords* in HISHO system.

3.2.2.1 Key String Extraction

It is assumed that nouns represent the features of an article better than verbs, adjectives, or other parts of speech, and that most of the nouns in articles consist strings of Kanji or Katakana, letters, or numbers, followed by Hiragana. If I cut the strings of Hiragana from the text, what is left will be either nouns or arbitrary string particles such as "は (wa)", " が (ga)", and " を (wo)". These particles are eliminated because these strings are not derived from nouns but have a verb or

[2]First, I tried to extract nouns using morphological analysis. I used Chasen version 2.0b6 and JUMAN version 3.61, but these were not suitable for network news articles, because network news articles include a variety of special code, such as one-byte characters, reference signals and garbage characters. Therefore I used key string extraction to get nouns from network news articles. However, I used morphological analysis, with later versions of Chasen, to extract keywords in other examinations in the following Chapters.

adjective stem.

However, filtering out particles that are not nouns is not adequate because the particles are not followed by a function word. Sometimes these key strings include meaningless strings, when for example a configuration of Kanji characters refers to the sentence "今日私はパーティを開く" (Today I hold a party). If the system wants to extract terms easily from this sentence, the system simply removes the Hiragana characters, punctuation marks, and so on. In the example above, the system therefore extracts three terms: "今日私 (today I)", "パーティ (party)", and "開 (hold)". These terms consist of Kanji and Katakana characters only. Then " 開 (hold)" is filtered out by using a function word. If there is a punctuation mark between "今日 (today)" and "私 (I)", the system can detect a word boundary. But "今日私 (today I)" consists only of Kanji characters, and the system cannot find a word boundary without morphological analysis. The problem of meaningless strings will be taken up in Section 3.2.2.6.

In the following sections, key strings represent *keywords* that consist of strings of Kanji, Katakana, letters, or numbers. Next I examine a test set consisting of articles, from December 1994 to April 1996, in two news groups (fj.life.health and fj.living). All the articles from these news groups are in the archives server of JAIST [64].

3.2.2.2 Categorizing Articles inside the RT

As explained in the previous section, HISHO first makes RTs automatically. Some RTs contain many articles, however, and it is possible that these RTs contain articles having various topics. HISHO can select articles with a common topic, and it can also find articles that change/shift that topic in a line of discussion by screening for topic-changing articles and topic-branching articles. I call an article in which the topic changes/shifts a Topic-Changing Article (TCA) and an article in which the topic branches a Topic-Branching Article (TBA).

The system can check whether or not there are TCAs in an RT. If the topic does not change in a series of articles, many of the same *keywords* tend to be used in all of the articles. On the other hand, if the topic changes, it is expected that *keywords* different from those in previous articles will be used after the topic change. HISHO therefore identifies a TCA (Figure 3.1) by looking for a transition in the frequency of *keywords*.

I utilize the following distinctive features to identify TCA.

1. In a TCA, the ratio of initial appearance *keywords* is higher than the ratio in the previous article.

2. When I split articles into two groups at a TCA, the *keywords* chosen in one group tend to appear more frequently in that group and less frequently in the other group.

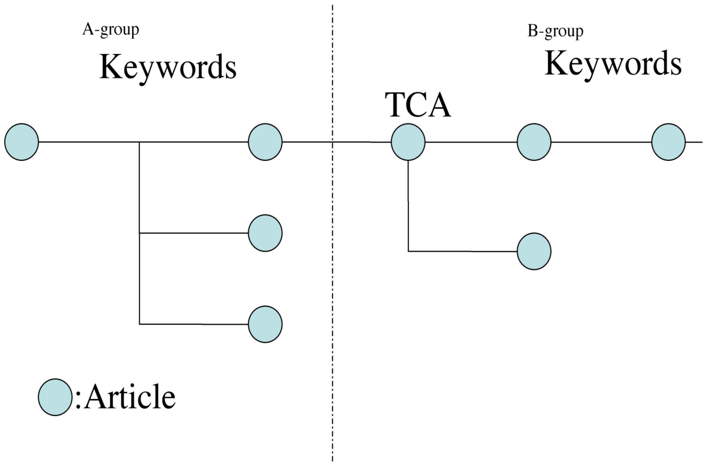

Figure 3.1. A sample of Topic cluster in an RT by a TCA

When several topics are discussed in one article, each is discussed in its own branch extending from that article. When a topic is not discussed clearly in its branch, it may be one of several topics discussed at several branches. As the branch of Figure 3.2 shows, when this happens, a group of articles in which the same topic is discussed overlaps a groups in which other topics are discussed. The article at a topic branching point is a TBA. Therefore, in the clustering of articles that branch from a certain article, the articles in the branches are allowed to belong to several clusters. My method compares pairs of articles and classifies articles according to their topics and then clusters them according to

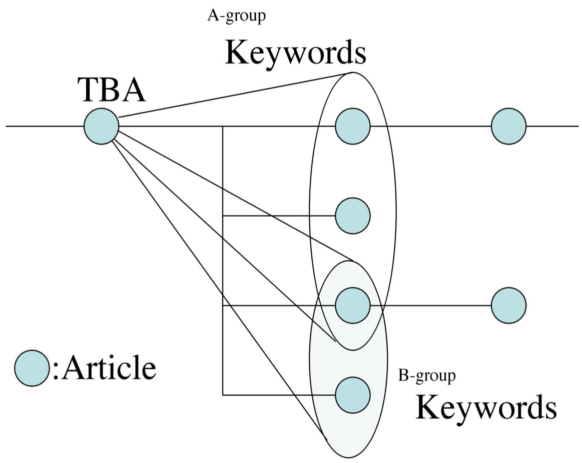

Figure 3.2. A sample of Topic cluster in an RT by a TBA

their topics. The branching article's topic is then assumed to contain the topics of these clusters. If several topics in the RT, are produced by clustering, my method presumes the branching article whose topics are separated at the branching point of the RT.

I utilize the following distinctive features to determine whether or not the topic discussed in the articles is the same. Two branches in which the same topic is discussed tend to quote the same parts of the branching original article, so they have same *keywords* according to quoting texts. Here, when the *keywords* of two articles connecting in one RT are compared, if the same *keywords* in the original article exist at a rate of over 60% in the *keywords* of the pursuant article, HISHO evaluate these articles have same topic.

I constructed RTs from a test set of about 10,000 articles; from these RTs, I selected 20 RTs containing about 400 articles as well as TCA. After cutting the headers and footers from the articles, I applied my methods for identifying TCAs and TBAs.

To evaluate my methods, I also had the TCAs and TBAs identified by three subjects. They identified TCAs and TBCs by actually reading the articles. I used TCAs and TBCs that at least two subjects identified, as correct answers in the examination, and then compared the output of my system. The results are listed in Table 3.1 [57].

Table 3.1. Results of TBA and TCA

	Recall	Precision
Topic-branching articles (TBA)	78%	82%
Topic-changing articles (TCA)	57%	94%

3.2.2.3 Collecting RTs

To find similar or related RTs, HISHO gives a score for each RT and compares the scores between an RT that suited the user's interests [3] and other RTs.

In an earlier study, I conducted experiments using character-weighting methods for collecting articles [22]. In those experiments the system gave, to all characters in an FT and in RTs, scores based on the frequency of occurrence and on common expressions in articles, and then compared the character lists of each RT and FT. If the list of an FT and an RT including the same *keywords*, the system added the *keyword* score of each list as the relative score between the FT and RT. The system selected related RTs according to their relative scores.

Since HISHO does not use morphological analysis, the *keywords* selected by HISHO are sometimes meaningless. Thus in the earlier study, the methods evaluated were quite different from term-weighting methods.

Although I was concerned that the lack of morphological analysis in the earlier experiments would affect the quality of the search, it had little or no effect on the results. I tried to apply two term-weighting methods to the collection process performed by HISHO.

In the following three sections I briefly review term-weighting methods (*tfidf* method and *score* method), describe my experiments using term-weighting methods in HISHO, and show the results of the experiments.

[3] I called an RT suited the user's interests an FT. The FT means the RT is focused user's interests on.

3.2.2.4 The *tfidf* Method

This term-weighting method is one of the most popular methods used in getting the features of articles. The *tfidf* method is used in the vector-space model [47]. The term weights are based on the frequency of a term in both a single document (term frequency tf) and the entire collection (inverse document frequency *idf*) [46]. The point of this method is that terms found frequently in relatively few documents are useful for getting the characteristics of a document. Terms that appear frequently in many documents are common terms and do not have any special meaning. These terms are given low scores according to their inverse document frequencies. By using the following formula, I can get one score S of a certain term j in one document D:

$$Sj = tf(D,j) \times \log(\frac{total\ Document\ number}{df(j)}) \qquad (3.1)$$

where $tf(D,j)$ is the frequency of the term j, and $df(j)$ is the number of documents that contain the term j.

3.2.2.5 The *select* Method and the HISHO Method

This term-weighting method was developed to select from newspaper articles keywords that could be used for classification of articles in a newspaper database. This kind of method generally uses some grammatical rules. For example, if a certain term occurs before a post-positional particle and is shown to be the subject word, it is assigned points, according to a point table for post-positional particles and idioms[4]. Each term is given a weight determined by comparing the point table and term frequency.

If the term score exceeds some threshold, the term becomes a keyword [29]. Ideally, an application using the *select* method has a list of general terms. I think to make the list using terms appear in the networknews articles constantly and liberally during a certain period. Then The *select* method with the list of general terms, called the HISHO method, extract terms in order to scores with the exception of terms in the list.

[4] In the discussion news groups, I could not find other forceful methods for selecting keywords without morphological analysis.

3.2.2.6 The Examination of Collection RTs

After the scores based on the features of articles are calculated, HISHO applies the vector-space model as the collection function.

I evaluated HISHO's ability to collect related RTs in the test set when using the *tfidf* method and when using the *select* method [39], and the HISHO method that is the *select* method with the list of general terms.

I chose six articles to serve as "input articles," and I manually selected related RTs as "answer RTs" from database of the network news arteicls. I collected "RTs" using "input articles" by three methods automatically, and compared the results of three methods and "answer RTs" manually.

The results are listed in Table 3.2.

Table 3.2. Results of Collection

	Recall	Precision
tfidf method	62%	88%
select method	68%	54%
HISHO method	71%	70%

The *tfidf* method gave excellent results. The *select* method gave results that may seem worse; however, I am currently more interested in recall rates than precision rates, because I think users would rather be presented with all the relevant articles than risk having relevant articles eliminated by the system. I think this would allow the user to browse more efficiently.

A significant problem with the *tfidf* method is that it calculates the term frequency of all the articles. News spools on a news server are modified every day, so the total number of articles changes every day. The calculation cost of the *tfidf* method is very high because this method requires the term frequency of all of the articles to be recalculated whenever the news spools are modified. The *select* method is preferable because it can get a score from each article.

For optimal performance, the *select* method needs a list of general terms. The list should include terms that appear frequently but do not have special meanings related to the article contents or topics. Terms like "hello," "these days," "I," and "guess" are common but do not say much about the topic of the article in which they appear. A list of such terms would be useful for avoiding unnecessary calculation when the *select* method is used.

I think a list of general terms can be made automatically and dynamically using term-frequency and heuristics for each news group. In my experiment I used as a general terms list the 50 terms that appeared most frequently in one year's worth of articles. When I repeated the earlier experiment with the *select* method, this time using the general terms list, I got an average precision rate of 70% and an average recall rate of 71%. This result shows the *select* method can improve using the general terms list. It means the HISHO method is the best method in this experiment.

Surprisingly, the list of high-frequency terms usually included even meaningless terms like "今日私 (today I)" [38]. In sum, the problem of meaningless terms can be relieved by using a list of high-frequency terms as a general terms list.

3.2.3 Conclusion

I used the results of the study mentioned above to improve the HISHO system. A prototype system was developed.

A user who accesses HISHO can see some reference trees with topic-changing articles (TCA) and topic-branching articles (TBA). The user can easily see where the same-topic articles are, and simply pushes a button to gather related articles. Next, HISHO starts calculating the relations. HISHO then shows some articles or RTs that have relations to or similarities with the article of interest. The user continues to read articles while HISHO changes the order of the article or the RTs. HISHO has other functions, such as finding hot topics, indicating news groups suited user's interests and so on, to help users to read network news easily.

I can demonstrate the potential of applying extraction and retrieval techniques to electronic data differently from newspaper articles to use features of the electronic data. However, I cannot evaluate HISHO itself from the users' side. Furthermore, I think that these techniques are successful because of the network news feature itself. In the following section, I will examine another support system using other electronic data that is different the network news.

3.3 Supporting Conference Program Production

In this section, I introduce another support system, which supports creating a conference program with extraction, categorization and retrieval techniques, and I evaluate the conference program produced by my system from the users' side with a questionnaire.

3.3.1 Features of Conference Applications

For some conferences, submitting applications for conference talks via the WWW has become common. When potential participants send in their applications, they include a title and an abstract for their talk. The abstract includes many technical terms, and its length varies from a few words to several hundreds words. Although the length of an abstract is about the same as that of a newspaper article, there are some different features between a newspaper article and an abstract. For example, an important sentence appears in the first line of a newspaper article, but can appear anywhere in an abstract. A newspaper article usually uses concrete expressions, but an abstract may include ambiguous expressions, and so on.

3.3.2 Experimental Production of a Conference Program

A conference program is a table of talks classified into sections, which include conditions of the talk, the time, and the room number. Sections also group talks with similar contents.

When producing a conference program, the following procedure is required.

A: to assemble applications, and to create a database

B: to divide applications based on the similarity of abstracts and titles

C: to give a session name to each group of similar applications

D: to fix the number of talks and rooms with the schedule of the sessions

In the procedures **B** and **C**, language processing technologies are applicable, i.e., the techniques of clustering data and attaching suitable names to the classified groups.

In the following sections, regarding the procedure of **B** and **C** as an automatic categorization and automatic labeling of conference applications (documents), I conducted an experiment to show clearly whether the traditional document clustering techniques were able to apply or not.

In the production processes, there are two approaches, namely:

1: First clustering applications, then decide session names from classified groups.

2: First decide session names, then classify applications fitting session names.

In both approaches, putting together approximately the same number of applications in one session is required, as well as reminders of session names from the titles and abstracts of the classified groups. I think that one condition of a good conference program is that it reminds its readers of a session name from the application title of each session, and a second is that the number of applications in each session is consistent.

I conducted the following experiments regarding each approach.

1: For the approach of first clustering applications, and then deciding session names from classified groups:

- using the clustering method [Experiment 1]

2: For the approach of first deciding session names, then classifying applications fitting session names:

- using a learning algorithm method [Experiment 2]
- using keywords' extraction and a conformity retrieval method [Experiment 3]

3.3.2.1 Experiment 1 (using a clustering method)

If the clustering method can fulfill the following conditions, a draft version of the conference program should easily be obtained.

Condition 1: The method can cluster applications into groups.

Condition 2: The method can generate groups that have approximately the same number of applications in each group.

Condition 3: The organizer can assign a suitable session name easily to each group that was generated by this method.

I tested two clustering methods, the top-down method [55] and the bottom-up method [6]. I attempted to cluster the applications of the 5th Annual Meeting of the Association for Natural Language Processing [5]. In this experiment, applications were first clustering into groups; I then decided session names from the classified groups; and finally, I created a conference program.

The top-down method [55] is recursively repeated to classify applications until the size of a group is one application. In the process, the method finds a word having maximum frequency of appearance in all applications in certain groups. Then, using the frequency of appearance of that word having maximum dispersity, the method divides the applications into two groups, one with a higher frequency of appearance of the word in each application, and the other with a lower frequency.

On the other hand, the bottom-up method [6] compares the similarities of each application or each group, and repeats the highest similarity pairs together until the number of groups becomes 1. This method uses "KL information" as a score of similarities for comparison.

Although these two clustering methods can make certain groups with certain semantic concepts, large variations appear in the number of applications included

[5]I got the applications from the program committee of the 5th Annual Meeting. Almost applications are written in Japanese. For tables of this thesis, Japanese titles were translated to English titles by the author. If the original title is English, the title has a * mark. In addition, all applications were numbered when they were submitted. The number described the column of "A-no." in tables.

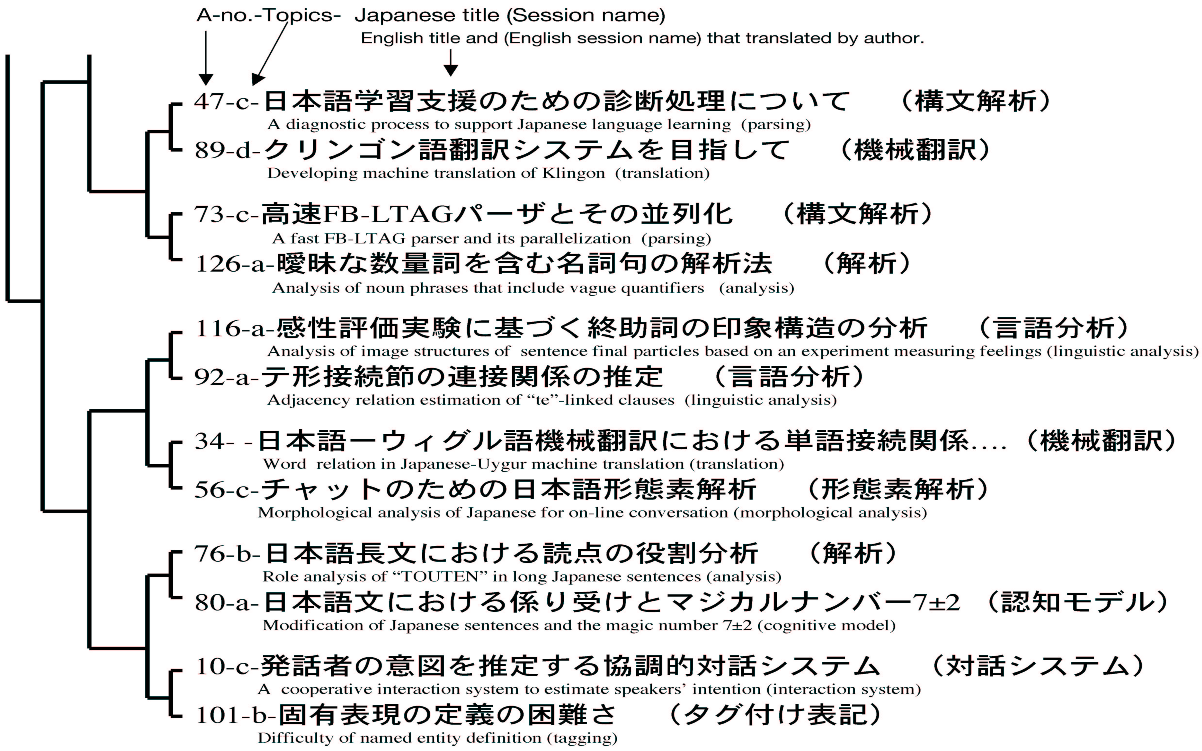

Figure 3.3. Sample results of the Top-down method

in each group. The applications in these groups (Figure 3.3) are compared with the actual talks in the sessions of the 5th conference meeting program. The groups produced by these classifications methods are quite different from the sessions of the 5th conference meeting program. Furthermore, it was difficult to assign a name of the group using the application's contexts in each group.

For the above reasons, I did not apply these classification methods to produce a conference program for the 6th Annual Meeting of the Association for Natural Language Processing.

3.3.2.2 Experiment 2 (using machine learning method)

In this section, I explain the learning algorithm method.

If the tendency of the contexts of the applications for the 6th Annual Meeting is the same as that for the 5th Annual Meeting, the names of the sessions are likely to be similar. In this case, the tendency with which the session names and participating applications for the 5th Annual Meeting are learned can use the learning algorithm method, and a conference program for the 6th Annual Meeting can be created by using the result of the learning algorithm method. I used the maximum entropy method as the machine learning method.

The classification method using the maximum entropy is, first, to determine the probability that each application of the previous Annual Meeting will be assigned to each session by the maximum entropy method. Then the method is to classify applications for the current Annual Meeting into a certain session when the probability becomes maximum [23, 35].

In this experiment, I used the sessions of the 5th conference meeting program and the applications for the 5th Annual Meeting for learning relations, and classified the applications of the 6th Annual Meeting into the sessions of the 5th conference meeting program.

The maximum entropy method needs some features for learning probability. First, to create features, morphemes are obtained from the titles and abstracts of applications, using morpheme analyzer JUMAN [27]. In this case, only nouns are used among the morphemes as a feature to learn probability. Furthermore, keywords in the title are important in general, so the keywords in the title are used as another special feature to learn probability. When a participant submits

Table 3.3. Classified Results using the Maximum Entropy Model

A-No.	Topics	Title	Session Name	Probability
90	d	語彙化されたツリーオートマトンに基づく会話文翻訳システム A machine translation system using lexicalized tree automata based grammar	機械翻訳 (translation)	0．989
37	d	英日・日英機械翻訳の実力 Ability as English/Japanese machine translation	機械翻訳 (translation)	0．972
50	d	日英機械翻訳における名詞の訳語選択 Noun word selection in machine translation	機械翻訳 (translation)	0．887
38	d	情報検索における絞り込み語提示による検索者支援の試み A retrieval support system based on suggesting terms to the user	検索 (retrieval)	0．839
97	d	科学論文における要旨—本文間のハイパーリンク自動生成 An automatic hyperlink generation between an abstract and text in scientific papers	検索 (retrieval)	0．830
9	d	用例利用型翻訳のための類似用例検索手法 An similar-example retrieval for translation using examples	検索 (retrieval)	0．735

Table 3.4. Results in the case of low probability

A-No.	Topics	Title	Session Name	Probability
25	b	辞書定義文を用いた複合語分割 Compound word segmentation based on defining sentences in a dictionary	分類・他 (others) タグ付け (tagging) 辞書 (dictionary)	0．163 0．152 0．126
18	c,d	ＧＤＡタグを利用した複数文書の要約 Summarization from multiple documents using GDA-tag	分類・他 (others) 言語学 (linguistics) 抽出 (extraction)	0．237 0．179 0．156

an application to the conference, he or she generally chooses a conference topic that fits their research area. The conference topics are a list of interest fields of the conference. The participants usually select one or two topics from the list. The topics are then used as the features to learn probability.

In this conference, the topics are from **a** to **e**. **a** represents phonology, morphology, and syntax area, **b** represents computational lexicology, terminology, and the text database area, **c**, the language processing algorithm, morphological analysis, and syntactic parsing area, **d**, the machine translation, information retrieval, and interaction systems, and **e**, other research areas. I used 1818 features to learn probability through the maximum entropy method.

Tables 3.3 and 3.4 show sample results which were used to classify the applications of the 6th Annual Meeting using the maximum entropy method learned by the applications of the 5th Annual Meeting. For example, in Table 3.3, the sessions of "機械翻訳 (machine translation)" and "検索 (retrieval)" show the high probability and good results. On the other hand, in Table 3.4, the results of low probability concerned irrelevant or unclassifiable sessions. The maximum entropy method shows the results of low probability in the case of using limited data such as an abstract, especially when the abstracts describe a new field or all the contents of the paper are covered. Furthermore, some applications did not show any topics, whereas others showed many topics. In these cases, the features were not useful in learning probability. When the maximum entropy method is not provided with enough data for it to learn probability, the method uses low-trust features to learn probability. Therefore, the results of the maximum entropy method became worse. If there are few available features, the features need to have higher reliability.

In the maximum entropy method, the session names of the learning data and the classification data have to be the same. That is, a problem exists as to how applications related novel research should categorize to. This problem arises when classifying according to the supervised learning method[6].

[6]For this problem, the following improvement methods exist. First, to divide applications that have higher probability and applications that have lower probability to classify a certain session. The applications of higher probability fix to the classified session, because the applications of higher probability are traditional talks in the NLP research area. Then, new sessions are generated by my proposed method in Session 3.3.2.3 for the applications of lower

For the above reasons, I did not apply the maximum entropy method to create a conference program.

3.3.2.3 Experiment 3 (using keyword extraction and conformity retrieval method)

In this section, I explain keyword extraction and retrieval method to create a conference program.

The processes of this method are as follows:

- to extract keywords considered to be important because of high frequency of appearance in applications,
- to decide session names using the extracted important keywords,
- to retrieve applications with similar keywords to the session names,
- to classify applications that have a high similarity to the session names, and
- to produce a conference program draft.

To produce the 5th conference program in this Session, I extract keywords as candidates of session names from the applications for the 5th Annual Meeting, retrieve the applications to resemble the candidates for the session names, and use the retrieval results as the conference program draft of the 5th Annual Meeting. I call this method the "keyword extraction and retrieval method".

In conclusion, from the results of these experiments 1 to 3, the result of the keyword extraction and retrieval method are better than the results of either the classification method or the learning method. Therefore, I decided to produce the 6th conference program through the keyword extraction and retrieval method (see Section 3.3.3). I explain the keyword extraction and retrieval method in detail in the following (1) and (2).

---

probability, and the applications of lower probability classify to the new sessions. After that, I will be able to get a new draft of the conference program without fixing the session name. However, methods are needed to divide applications clearly into traditional research areas and applications in the leading-edge areas.

(1) Extraction of Session Names

In the first process of the keyword extraction and retrieval method, keywords are extracted as candidates for session names.

In this process, two ways are considered;

I: using only frequency of words

II: using a word score calculated through features of the word's appearance

For **I**, Kanji and Katakana characters were extracted as keywords from the title and the abstract of an application [29], and the frequency of the keywords was counted; candidates for the session name were selected from top 20 keywords in order to high-frequency.

For **II**, or the scoring method, keywords were extracted in the same way as in **I**, but then the keyword score using frequency was calculated by adding special scoring if stop words (は，が，を，について) followed the keywords, if the keywords appeared in the title[7], and if the keywords were a complex words [8]. Next, high scoring words were checked against the number of applications that included them, and the total of the high-scoring keywords was extracted (up to the rank 5th)[9]. Finally, the number of applications containing the high-scoring words was checked for every application.

For example, keywords such as "新聞記事 (newspaper articles)", " 構文 (syntax)","情報抽出 (information extraction)", " 従属節 (subordinate clause)", "情報 (information)", and " 抽出 (extraction)" were extracted from the application in Figure 3.4. The score of "新聞記事 (newspaper articles)" was 11 points, which consisted of a 5 title-score (the word was included in the title), 3 frequency-score (the word appeared 3 times in the application), 3 stop word-score (the word appeared before 「が」). The score of "情報 (information)" was 6 points, which consisted of 5 title-score, 1 frequency-score, and so on. Then I extracted the five highest-scoring keywords (e.g. "新聞記事 (newspaper articles)", and " 情

[7]If a keyword appears in the title, 5 points were added to the keyword score. For the position score, if the stop words followed the keyword, 3 points were added to the score of the keyword.

[8]For example, the keyword is a complex word such as "構文解析"(parsing), adding the frequency of the complex word to the frequency of the keywords of "構文"(syntax) and "解析"(analysis).

[9]$Score = termfrequency + positionscore + complexwordfrequency + stopwordscore$

⟨ TITLE ⟩ 新聞記事における書き出し文の構文 ⟨ /TITLE ⟩
⟨ ABSTRACT ⟩
情報抽出では定型的な文章である新聞記事が対象となることが多い。しかし、一般に、いつ・誰が・どこで、何を、どうしたといった５Ｗ１Ｈ型の固定的な表現で記述される書き出し文であっても、連体修飾節や連用節などの従属節を含む複雑な構造をとることがある。現在、これらの文から情報抽出を行うのは難しいと考えられている。一方で、どのような情報を抽出すべきかについても十分検討されていない。本研究では、このような複雑な文から抽出できる情報は何か、またどのような観点に着目して情報抽出を行うべきか明らかにする。そのため、新聞記事の書き出し文を対象にして、主節と従属節の関係に着目し分析を行う。⟨ /ABSTRACT ⟩

Figure 3.4. A sample of a result

報 (information)”), and the keywords became session name candidates that were extracted from the application of Figure 3.4. Furthermore, the number of applications including the session name candidates that were extracted from each application was counted, and the final session name candidates for the conference program were extracted from candidates that had higher numbers of applications including the session name candidates.

High-frequency words that were extracted using **I** were not suitable for session names. These words included words such as “システム (system)”, “本稿 (this paper)”, “我々(we)”, “利用 (utilization)”, “提案 (proposed)”, and “手法 (method)”. On the other hand, high-scoring words using **II** were suitable for session names such as “対話 (dialogue)”, “構文解析 (parsing)”, and “統語 (syntactic)”. Some keywords of the high-scoring words were used as session names of the 5th conference program. I therefore decided to use the keywords that were extracted using **II** in session names.

The keywords automatically extracted using **II** were 13 words of “システム (system)”, “対話 (dialogue)”, “統語 (syntactic)”, “情報検索 (information retrieval)”, “翻訳 (translation)”, “モデル (model)”, “解析 (analysis)”, “構文解析 (parsing)”, “抽出 (extraction)”, “辞書 (dictionary)”, “生成 (generation)”, “分類 (classification)”, and “手法 (method)”. Next, I selected 9 of these keywords that were considered to be appropriate for a session name. The session names of the 5th conference program for experiment were the 9 selected words of “対話 (dialogue)”, “情報検索 (information retrieval)”, “翻訳 (translation)”, “モデル (model)”, “解析 (analysis)”, “抽出 (extraction)”, “辞書 (dictionary)”, “生成

(generation)", and "分類 (classification)".

(2) Classification of applications into sessions

In this section, I explain how to classify applications into each session. This classification method used a "keyword vector."

A "keyword vector" uses a certain keyword in the abstract or title of an application as an element. The "keyword vector" consists of two vectors: one is a "session vector", which includes keywords of session name candidates and the similar words of the session name candidates. The other is a "lecture vector", which includes extracted keywords from each application.

First, I extracted the keywords that had a co-occurrence relation to the session name candidates from each application. The extracted keywords were the keywords of two higher ranks with the co-occurrence relation. The extracted keywords were the related words of the session name candidates, and elements of the session vector. However, if the extracted keywords appeared under five times in all the applications, then the keywords were omitted from the elements of the session vector. At this time, there was only session name candidate in the session vector [10]. Next, the similarity of each lecture vector and session vector was compared using the inner product of two vectors, and the most similar session vector was computed. The session name candidate in the session vector became the session name of the application that had the most similar lecture vector. Since this method means searching a session name by an application as a query, the keyword extraction and retrieval method is called it.

To compute the similarity of two vectors, I tried two kinds of scoring for the elements of the vector: one was the frequency of the keywords as elements in each vector, and the other was a score that is computed by II in Session (1) above.

In the case of using frequency, many of the applications to be classified in multiple sessions have the same similarity. On the other hand, in the case of using score that is given by the addition to the special points according to the appearance position in an abstract. A good result is obtained as shown in Table 3.5.

Therefore, I used the extraction and retrieval method with score to produce a conference program.

[10] The maximum number of elements in session vector was four words.

Table 3.5. Sample results of automatic classification

Session Name： 情報検索 (Information Retrieval)		
A-No.	Topic	Title
3	d	係り受け情報や語の意味情報を利用した日本語テキスト検索システム Japanese text retrieval system using semantic and dependency information
42	d	要素の順序関係から見た類似文最適照合検索 A retrieval method for similar sentences using element order relation
55	d	分類標数の相互参照に基づく多言語書誌データ検索システム A retrieval system for multilingual bibliographic data based on cross-referencing of library-book classification numbers
70	d	コンプリメントタームを用いた情報検索 Information retrieval using complement terms
81	d	情報検索の類似尺度を用いた検索要求文の単語分割 A word segmentation of retrieval order sentences in information retrieval using similarity scale
88	c	ニュース音声データベースの検索システムの試作 A retrieval system of broadcast news documents in a speech database

3.3.3 Production of the Conference Program

A draft of the 6th conference program is produced by the extraction and retrieval method through keyword scoring described in Section 3.3.2.

First, I transformed applications of the 6th Annual Meeting of the Association of Natural Language Processing into the data shown in Figure 3.5 [11].

The 20 words of the session name candidates were extracted using the method explained in Session 3.3.2.3 (1) for the conference program of the 6th Annual Meeting. Next, 9 words were selected manually for the session name: “対話 (dialogue)”, “要約 (summarization)”, “辞書 (dictionary)”, “コーパス (corpus)”, “検索 (retrieval)”, “抽出 (extraction)”, “解析 (analysis)”, “生成 (generation)”, and “翻訳 (machine translation)”. Next, all applications for the 6th Annual Meeting were automatically classified using the similar rates of session vector and lecture vector explained in Session 3.3.2.3 (2). However, some applications existed not to find any similar applications with any session names. These applications were allocated to “その他 (others)”.

In the above processes for extracting session names candidates and classifying applications, I did not consider conditions such as the number of meeting rooms or meeting schedules for the 6th conference program. The result of automatic classification (Table 3.6) was the draft of the conference program for the 6th Annual Meeting.

Manual adjustment required several hours. Seventeen applications changed by the organizers, with titles as shown in Table 3.7. Four of these titles did not have similarity with any sessions and were classified with the “その他 (others)” session. Table 3.8 shows the results of automatic classification and alteration by hand. If too many applications were classified into one session, I repeated to extract new keywords for session name from these applications and to classify applications [12]. However, appropriate session names could not be extracted from the applications in the session named “解析 (analysis)”. Thus, the “解析 (analysis)” session was divided into three sessions in the 6th conference program. The reason why the keyword extraction and retrieval method did not extract other

[11] For automation and increase in efficiency, I think applications need to change to a unified format. Here, XML format is used in experiments.

[12] In this process, “システム (system)” and “意味 (semantic)” were extracted as session names.

```
<APPLICATION>
<MAIL-ID>130</MAIL-ID>
<TYPE>講演発表</TYPE>
<TITLE> 大会プログラム自動生成に向けての一考察
</TITLE>
<AUTHOR id=1>
<KANJI>〇小作　浩美</KANJI>
<KANA>オザク　ヒロミ</KANA>
<AFFILIATION> 通信総合研究所
</AFFILIATION>
<NUMBER>123-456-7890</NUMBER>
</AUTHOR>
<AUTHOR id=2>
...
</AUTHOR>
<CATEGORY>d，e</CATEGORY>
<APPLIANCE>OHP</APPLIANCE>
<ABSTRACT>
　いくかの言語処理技術を利用して言語処理学会の
大会プログラムを自動作成することを試みた．
その結果と自動生成するにあたり明らかになった
問題点，改良点について報告する．
</ABSTRACT>
<ADDRESS>
住所：　 〒 651-2492 神戸市西区岩岡町岩岡 588-2
所属：　 通信総合研究所
氏名：　 小作　浩美
...
</ADDRESS>
</APPLICATION>
```

Figure 3.5. Sample data in XML format

Table 3.6. Sample results of automatic classification

Session Name： 対話 (dialogue)		
A-No.	Topic	Title
40	c	混合主導対話における音声認識誤りに対処するための対話管理 Dialogue management for correcting errors in speech recognition of mixed-initiative dialogues
59	c,d	制限知識下における効率的対話制御 Efficient dialogue control under limited knowledge conditions
85	c	道案内 WOZ システムとの対話における言い淀み表現の分析 Analysis of disfluency expressions in dialogues between users and WOZ system for directions
102	c	係り受け関係を用いた即時発話理解 ー音声対話メールシステムにおける手法ー Understanding real-time utterance using dependency structures
124	c	多重文脈に即応的な対話インターフェース：半可通 HANKATSU: Dialogue interface for multi-context model
Session Name： 辞書 (dictionary)		
A-No.	Topic	Title
15	a	概念体系における反対語の検討 Transactions of opposite words in concept classification
25	b	辞書定義文を用いた複合語分割 Compound Words Segmentation using dictionary definitions
32	b,d	翻訳システム用の辞書ツール Dictionary tool for machine translation system
110	a	既知形態素からなる未知複合語概念推定とその辞書登録 Concept identification of compound words consisted given morphemes and registration to dictionary
116	b	不要語リストを用いた RFC 英和辞書作成過程における課題 Problems in making an RFC English-Japanese dictionary using lists of meaningless words
117	b	ソフトウェア開発工程における用語構造と翻訳辞書作成過程における課題 Problems in term structures of software development and construction of a translation dictionary

Table 3.7. The applications were adjusted manually

A-No.	Topics	Title
11	c	語の重要度を考慮した談話構造表現の抽出 Extraction of dialogue structure expressions using word importance
13	d	直接引用表現を利用した要約知識の自動抽出の試み Automatic extraction of summarization knowledge using direct quote expressions
12	c	単語ラティス形式の音声認識結果を対象とした発話意図の認識 Communicative intention recognition for speech recognition of word lattice formations
19	c	Generating coherent text from finely classified semantic network*
39	c	日本語における口語体言語モデル Linguistic model for colloquial forms in Japanese
65	d	意味的共起関係を用いた動詞と名詞の同音意義語の仮名漢字変換 Kana-Kanji translation of homonyms using semantic co-occurrence relation
77	a	韓日語の副詞節の階層性に関する対照言語学的研究-南 (1974) の階層性モデルの観点から- A contrastive linguistics study for configurationality of adverb clauses in Korean and Japanese
83	a	FB-LTAG から HPSG への文法変換 Grammar conversion of FB-LTAG to HPSG
91	b	文節解析のための長単位機能語辞書 Compound function word dictionary for 'BUNSETSU' analysis
97	d	科学論文における要旨-本文間のハイパーリンク自動生成 Automatic hyperlinks between sentences in scientific papers
98	c	確率付き項構造による曖昧性解消 Disambiguation using structure with probability
118	a	コンピュータ西暦 2000 年対応の標準化におけるデータ，用語，処理，試験 Data, terms, procedures, tests for standardization of Y2K readiness
119	a	日本語待遇表現の評価実験による誤用とその認知について Abusage and recognition of polite expression in Japanese
121	d	図書館の自動リファレンス・サービス・システムの構築 Development of an automatic reference service system in a library
128	c	SGLR-plus による話者の対象認識構造を抽出する英語文パーザの試作 English parser to extract object recognition structure for speakers using SGLR-plus
130	d,e	大会プログラム自動生成に向けての一考察 A study of automatic conference program production
131	b	簡単なフィルターを用いた二言語シソーラスの自動構築 Automatic building of bilingual thesaurus using simple filter

Table 3.8. The session names of automatic classification and alterations manually

A-No.	Topic	Classification results	Alterations
11	c	抽出 (extraction)	解析 (analysis)
13	d	要約 (summarization)	抽出 (extraction)
12	c	翻訳 (machine translation)	理論 (theory)
19	c	その他 (other)	生成 (generation)
39	c	コーパス (corpus)	理論 (theory)
65	d	意味 (semantic)	システム (system)
77	a	その他 (other)	理論 (theory)
83	a	生成 (generation)	解析 (analysis)
91	b	解析 (analysis)	辞書 (dictionary)
97	d	生成 (generation)	システム (system)
98	c	システム (system)	解析 (analysis)
118	a	その他 (other)	理論 (theory)
119	a	その他 (other)	理論 (theory)
121	d	システム (system)	検索 (retrieval)
128	c	抽出 (extraction)	解析 (analysis)
130	d,e	抽出 (extraction)	理論 (theory)
131	b	抽出 (extraction)	コーパス (corpus)

session names from the applications classified into the "解析 (analysis)" session, was that many similar keywords appeared in these applications. This result is considered to be a limitation for choosing a session name using only frequency or position information, without using semantic information. Finally, the "その他 (others)" session was renamed "理論 (theory)" session manually. By the above-mentioned process, a draft of the 6th conference program was produced. I modified the draft according to fine adjustment of lecture's hopes, etc. I announced the 6th conference program to members of the Association of Natural Language Processing [13].

In this production procedure, this method is very helpful to decide session names and to cluster applications to the sessions. This method can shorten the time of the program production, and respond to the current tradition of the research area. The method should evaluate from the users' side. I describe the results of questionnaire survey in the following section.

3.3.4 Evaluation based on Questionnaire Survey

After the 6th conference, I sent out questionnaires about the program to presenters and participants. I circulated these questionnaires to audiences on the floor, asking their opinions of whether or not they thought the presentations suited the session names. I sent out questionnaires by e-mail to presenters and also asked for their opinions about whether or not their presentations suited the session name, and whether or not presentation they were interested in existed in the same session. I received answers from 79 of the 102 presenters to whom I sent questionnaires.

For the question about whether their presentations was suited to the session

[13] The session names extracted from the candidates made by automatic extraction were finally 対話 (dialogue), 要約 (summarization), 辞書 (dictionary), コーパス (corpus), 検索 (retrieval), 抽出 (extraction), 解析 (analysis), 生成 (generation), 翻訳 (translate), 意味 (semantic), システム (system). I did not use my method to create the poster session program; I utilized only the topic areas of applications emulating the poster session of the 5th program. To adapt the number of meeting places, I classified two sessions for posters, one for topics a and b, and the other for topics c, d and e. In the 6th program, there were 104 applications; three applications were canceled, and one application was registered after was created the program. The final number of applications was 102.

Table 3.9. The title replied not suitable to the session name

Answer No.	Including session name in the title	projection	get low score
1	○		
2	○		
3	○		
4	○	○	
5	○	○	
6		○	
7		○	○
8			○
9			○
10			
Total	5	4	3

name, 10 people replied that the session name was not suited their presentations, as shown in Table 3.9.

Four people out of ten replied that their presentations did not suit the session in which they had presentations. Actually, the session that the four people described to gather non-similar presentations, were in the "その他 (others)" session. Five additional people replied that their presentations were suited to the session. These presentations included session names. It seemed that "title-score" performed badly. Three titles did not include session names and the titles of the automatic classification results resulted in a low score.

There were 22 titles that did not include the session name, as shown in Table 3.10. Seven of these 22 got low scores and were classified into multiple sessions. Finally, presenters replied that three of the seven presentations did not suit the session.

According to the questionnaire with 79 valid replies, 69 people replied that the session name was well correlated to their presentation. Two of them replied that the session name was OK, but other presentations in the session seemed totally irrelevant to their presentation. Thus, 69 out of 79 responders (87%) were satisfied with this automatic production program, and I assume that the 6th conference program has been satisfactorily created for those who will make use of it.

Table 3.10. The titles did not include any session names of the 6th conference program.

A-No.	Topics	Titles
3	d	かぎ括弧で囲まれた表現の種類の自動判別 Automatic type discriminant for expressions enclosed within parentheses
8	b	概念体系における反義概念の検討 A study of concept of antonymy in concept classification
12	c	単語ラティス形式の音声認識結果を対象とした発話意図の認識 Communicative intention recognition for speech recognition of word lattice formations
19	c	Generating coherent text from finely classified semantic network*
28	d	文分割による連体修飾節の言い換え Paraphrasing of adnominal clauses using sentence segmentation
35	c	ＨＰＳＧの複数の文脈自由文法へのコンパイル Compile multiple context-free grammar using HPSG
39	c	日本語における口語体言語モデル Linguistic model for colloquial form in Japanese
41	d	ワールドワイドウェブを利用した住所探索 Address search on the World Wide Web
42	c	LR 表への複数の接続制約の組み込みによる一般化 LR 法の拡張 Expansion of an LR parser integrating multiple connection constraints into an LR table
47	d	n-gram モデルと IDF を利用した統計的日本語文短縮 Statistic clipping of Japanese sentence using an n-gram model and IDF
57	c	グルーピング法による GLR パーザの効率的な実装 Effective development of GLR parser using a grouping method
58	c	キーワードの活性度の変化を用いたテキスト中の単語と話題の対応付け Indexing words and topics in text using changes degree of keyword activity
65	d	意味的共起関係を用いた動詞と名詞の同音意義語の仮名漢字変換 Kana-Kanji translation of homonyms using semantic co-occurrence relation
77	a	韓日語の副詞節の階層性に関する対照言語学的研究-南 (1974) の階層性モデルの観点から- A contrastive linguistics study for configurationality of adverb clauses in Korean and Japanese
83	a	FB-LTAG から HPSG への文法変換 Grammar conversion of FB-LTAG to HPSG
98	c	確率付き項構造による曖昧性解消 Disambiguation using structure with probability
101	d	ニュース速報記事の前文情報との照合に基づく見出し文の言い替え Paraphrasing of news headlines based on verifying pre-information of news bulletins
118	a	コンピュータ西暦 2000 年対応の標準化におけるデータ，用語，処理，試験 Data, terms, procedures, tests for standardization of Y2K readiness
119	a	日本語待遇表現の評価実験による誤用とその認知について Abusage and recognition of polite expression in Japanese
125	a,c	人間の文処理は左隅型である：埋め込み構造と記憶負荷 Embedding structure and memory loading
129	a	形容詞的ふるまいをする「名詞＋の」について About adjectival action of 'Noun + NO'
131	b	簡単なフィルターを用いた二言語シソーラスの自動構築 Automatic building of bilingual thesaurus using simple filter

3.3.5 Discussion

In the experiment of this program production, I considered the tendency of session names, the restrictions on the number of produced session conditions, and the increase in efficiency of program production.

3.3.5.1 The Tendency of Session Names

In the method of extraction and retrieval for program production, the following processes are needed:

1 Extraction of keywords important to producing the program.

2 Calculation of keyword scores by the scoring method.

3 Retrieval of applications using session similarities.

In the process of calculating scores, I considered the appearance position of keywords. If a keyword appeared in the title, it got a high score. This score represents titles given by authors that directly matched their research.

The majority of titles were included in keywords of the session names for the 6th conference program. However, 29 out of 102 titles were not included in the session names. Twenty two of these 29 did not include any keywords using the session names. This means that 80% of the titles included a session name keyword. If session name candidates can be retrieved from all titles and applications classified in a session, I believe I have developed a appropriate program. In the 5th conference program, the majority of titles were included in the session names [14].

As a title can easily be included in session name, it is clear that the keyword containing a field is easily included in a title. Of course, some titles were classified to other sessions by considering information in the abstract, even if the title was included in the session name. My proposed method can assign talks to the appropriate session, different from the session name that was included in the title, even if the title is included in the session name. Concretely, seven

[14] In the 5th conference program, there were 79 titles that had a session name, out of 107 applications.

titles corresponded to the above case: that is, the titles were included in the session names in the experiment to produce the 6th conference program. Since the validity of a session name was not checked in the questionnaire, I think that a comparatively good program was created to compare with the past programs.

3.3.5.2 Restrictions on Conditions of Program Production

I examined the ability of extraction, categorization and retrieval techniques to produce a conference program. However, in reality, restrictions for actual program production, for example restrictions such as time and place limitation, should also be considered.

In this program creation, I set the conditions at my own discretion. I think there are other important points to consider in the preceduers of programs production. I should investigate the suitability of other restrictions on conditions on program production in more detail.

If I can pigeonhole the restriction conditions as heuristic knowledge, I can produce the conference programs more efficiently.

3.3.5.3 The Efficiency of Program Production

The most time-consuming process in this program creation is reforming from registrations into XML-data, because the registrations have a variety of styles and characters code. The conference applications did not fit a certain format, so I had to reform the applications manually. For automatic generation, the input data should have a uniform format. I can cope with this problem easily by using the WWW. Conference staff should improve the conference's services for users, with the cooperation of all participants[24].

3.3.6 Conclusion

In this research, I applied extraction, categorization and retrieval techniques to produce an actual conference program. In the process of program production, I showed the ability of these techniques using the keyword extraction and retrieval method. Furthermore, if registration forms are unified using WWW technologies, I may also shorten the process of program production.

In Section 3.3.3, I showed that my method successfully produced the 6th conference program. I got good results that 87% of the response shown gratulant from the questionnaire described in Session 3.3.4. Furthermore, since the manual work in program production processes other than data creation was only several hours, my method can be said to achieve an increase in efficiency.

Two problems remain. The first problem is of classification, namely that an application for two or more sessions must be classified into one session. The second problem is mistaken classification, because the title includes words of the session name. To solve these two problems, the technology of extracting a more exact keyword, even from a short abstract, is required. Furthermore, I should deal with applications other than those written in Japanese; produce perfect automatic program production; define the restriction conditions of real program production; and conduct a validity investigation of session names.

3.4 Conclusions of Chapter 3

I have shown that information handling technologies can support users efficiently in the Session 3.2 and Session 3.3. This means the potential of the extraction and categorization techniques is widely applicable. However, these experiments utilized small data sets compared with WWW data. In the next chapter, I try to measure the ability of information retrieval techniques to deal with large-scale data.

Chapter 4

Information Retrieval Techniques from WWW Data

4.1 Introduction

The World Wide Web (WWW) has become very popular, and the number of WWW documents has increased dramatically. Along with the popularization of the WWW, the load on information retrieval systems is increasing rapidly. Usually, an information retrieval system is made up of multiple machines for distributing the load of retrieval processes. Three critical points of research for WWW retrieval dealing with a huge amount of data are follows [9];

1. Understanding resource descriptions
2. Selecting appropriate resources
3. Merging results

To raise the precision of information retrieval, I am especially interested in point 3; that is, how to merge retrieval results effectively.

Ongoing research regarding point 3 discusses how to effectively merge certain results from multiple information retrieval systems. That research is generally classified into two groups. One group is called "Metasearch" research. The other group I have named, "Database-search" research.

In Metasearch, search engines with different retrieval methods are used and multiple results from those engines are merged effectively. In general, Metasearch utilize some ready-made search engines by using different retrieval methods. Metasearch has the advantages of finding answers to check a wide range of homepages by using different retrieval methods. Metasearch is research used to utilize different multiple search engines efficiently, rather than for the improvement of retrieval precision. For example, Metasearch research is used to reduce communication costs [32, 51], to improve effectiveness in utilizing features of retrieval systems, and to select appropriate retrieval systems [54].

On the other hand, Database-search use search engines with a certain retrieval method and multiple results from those engines are merged with a high degree of accuracy. Database-search has the advantages of finding answers with a high precision, if Database-search systems utilize the best precision retrieval method and the best precision merging method. In this section, I discuss several properties of the Database-search.

In Database-search, the most popular type of merging method is called "Score Normalization". Score Normalization is used to gather all information in a document and in term frequency, for example, from each database of each system, and to normalize each resultant score using the gathered information, and then to get final results. However, the load of gathering all information from all systems is very high, especially when applied to a WWW environment. Therefore, normalization is impractical when used with WWW information retrieval systems. However, research to select and merge databases that have a strong possibility of including answers, and a strong possibility of calculating an approximate solution using a part of the database is strong, but focuses mainly on point 2: selecting appropriate resources. The purpose of this research is to investigate newspaper articles, and not large amounts of data, such as WWW data.

In this chapter, I establish the property of WWW documents and various retrieval measurements, and compare merging methods using large data that is released as Web task data of NTCIR3[1]. The result of the experiment clearly shows that merging results using the information retrieval methods impervious

[1] NTCIR stands for NII NACSIS Test Collection for IR Systems Project. http://research.nii.ac.jp/ntcir/index-en.html

database size as typified by Okapi and SMART method, are efficient as same as using the score normalization. Especially when extracting information from the data, such as the WWW data, is not biased word appearances, the Okapi and SMART method have the same effect as the score normalization. That is shown using Okapi and SMART methods, it is possible to retrieve information from large data in high precision. I will report a series of experiments and then analyze results.

4.2 Retrieval Methods and Merging Methods

In this section, I report certain retrieval methods and merging methods in my investigation. This investigation is a kind of the database search. The process of this investigation is following; First, a huge amount WWW data is divided to some databases. Then using a certain single retrieval method, information as results of the each database is retrieved from the each database. These results from all databases are merged by certain single merging method, and the final retrieved result is obtained. I inform each precision rate of certain retrieval methods, consider the feature of WWW documents and the best merging method. In addition, this situation is thought from real WWW situation. It means that if WWW retrieval system is achieved in the real world, only one retrieval system can not manage whole WWW data. So the retrieval system should share burdens of retrieval process with some machines and merge retrieved results of the individual machines. There are other researches such as metaserach and parallel computation for ways to share burdens and to merge results. However I want to compare the precisions of the well-known retrieval methods to retrieve information from WWW data with accuracy. Then I decide the investigation process as mentioned above, and experiments in order to the database search.

In my investigation, I use three retrieval methods in section 4.2.1, four merging methods in section 4.2.3, and Small collection (10GB) of NTCIR3 web task data as targeted data. In the experiment, I utilize only text area of NTCIR3 web task data. That means the tag information, control code and so on, omitted from NTCIR3 data before the experiment. At that time, data size is 2.1GB and the number of files is about 1.5 million. The targeted data is formatted as Figure 4.1.

```
<NW:DOC>
  <NW:META>
    <NW:DOCID>NW000054231</NW:DOCID>
    <NW:URL>http://www.crl.go.jp/</NW:URL>
    <NW:DATE>Wed, 18 Apr 2001 04:11:32 GMT</NW:DATE>
    ...
  </NW:META>
  <NW:DATA>
    <NW:DSIZE>873</NW:DSIZE>
    通信総合研究所（CRL）
    …
    CRL とは。
    …
    安心して暮らしやすい国民生活のために、…
    </NW:DATA>
</NW:DOC>
```

Figure 4.1. Data Sample

4.2.1 Retrieval Methods

I utilize three retrieval methods. The retrieval methods are Okapi, INQUERY, and SMART. The Okapi and INQUERY method are shown higher precision in Web task of TREC8 [19]. The SMART method did not join the Web task in TREC, but it is gotten higher precision in several other tasks in TREC [8]. In TREC8, the Web task targeted data include Japanese data, query format however dose not includes Japanese query. So I investigate these retrieval methods to apply in Japanese environment.

(1) Okapi (BM25)

Okapi method is a probabilistic retrieval model that is developed by S.E. Robertoson et.al. When Query Q and Text D_i are given, Okapi method made up probability $P(T|Q, D_i)$ that was adapted the Text D_i to the Query Q. In my investigation, I use the following formula that is called BM25 [45].

$$BM25(Q, D_i) = \sum_{T \in Q} w^{(1)} \times \frac{(k_1+1)tf}{K+tf} \times \frac{(k_3+1)qtf}{k_3+qtf} \tag{4.1}$$

Here; T is a word included in query Q. tf is number of T included in text D_i. qtf is number of T included in query Q. $w^{(1)}$ is weight of T defined by the following formula.

$$w^{(1)} = \log \frac{N - n + 0.5}{n + 0.5} \tag{4.2}$$

N is number of text in targeted data. n is number of text that include word T. K is a value of following formula.

$$K = k_1((1-b) + b\frac{dl}{avdl}) \tag{4.3}$$

k_1, b, k_3 are constant numbers set at experimentally. Here, I use $k1 = b = 1$, $K3 = 1000$[2]. And dl is text length of D_i, $avdl$ is average text length in targeted data. Here, the text length means number of word in the text.

(2) SMART

SMART is a search system which the research group centered on G.Salton developed, and uses the retrieval model established as a vector space model. SMART expresses a document and a search query as a vector which consists of dignity of each word, and has the feature is to calculate the degree of similar using the inner product between a document vector and a query vector[52, 65].

When a certain query Q and a document D_i are given and a certain word $T(t_1 \leq T \leq t_m)$ is contained in both of the query Q and the document D_i, a score $SMART(Q, D_i)$ is calculated by the following formula;

$$SMART(Q, D_i) = \sum_{k=1}^{m} (q_{iT} \times d_{iT}) \tag{4.4}$$

$D_i = (d_{i1}, d_{i2}, ...d_{it})$, d_{iT} is calculated by the following formula;

[2]This constant number is pursuant to setting of the IR package [59]. In this experiment, I utilized the IR package version 1.47.

$$d_{iT} = \frac{\frac{1+\log(tf)}{1+\log(avtf)}}{(1-slope)\times pivot + slope \times utf} \tag{4.5}$$

here, tf is the frequency of word T contained in document D_i.
$avtf$ is the average frequency of words contained in one document.
$pivot$ is the average frequency of whole words in one document.
utf is the number of words in document D_i [3].
And the score of $slope$ was defined as 0.25 from A.Shinhal's experiment report[52].

$$q_{iT} = \frac{1+\log(qtf)}{1+\log(avqtf)} \times \log\frac{N}{n} \tag{4.6}$$

here, qtf is the frequency of word T contained in query Q.
$avqtf$ is the average frequency of words contained in query Q.
N is the number of whole documents in the document set for search.
n is the number of the document that contains in T.

(3) INQUERY

INQUERY is a search system which the group centered on W.B.Croft developed, and a measure of retrieval based on Bayes type inference network. Using this measure, the document rank order is determined by the certainty factor $B(Q|D_i)$ of query Q, when document D_i is given. The certainty factor is calculated by the following formula[1];

$$INQ(Q, D_i) = \sum_{T\in Q\cap D_i} (0.4 + 0.6 \times \frac{tf}{tf+0.5+1.5\times\frac{dl}{avdl}} \times \frac{\log\frac{N+0.5}{n}}{\log N+1}) \times \frac{qtf}{\sum_{T\in Q} qtf} \tag{4.7}$$

n is the number of the document that contains in word T.
N is the number of whole documents in the document set for search.

[3]For convenience of application, I experiment using $advl$ of Okapi method in Formula 4.3 instead $pivot$ of SMART method, and dl of Okapi method in Formula 4.3 instead utf of SMART method.

dl is the length of the document D_i or the number of words in the document D_i.
$avdl$ is the average length of the documents or the average number of words in the document in the document set for search.
tf is the number of word T contained in document D_i.
qtf is the number of word T in query Q [4].

4.2.2 Features of Retrieval Methods

If the system retrieves information from divided databases, the system needs to merge all results from each database. At first, the normalization method is one of prevailing methods. The normalization method is a method of changing a retrieval score of each retrieval result from multiple databases that it may become equivalent to the retrieval result from one database, and then obtaining a final retrieval result. Although the normalization method is explained in Section 4.2.3, when treating huge numbers data, the normalization method is not a realistic method. The normalization method is however extremely precise to retrieve information from a database of newspaper articles. Then I find a method to retrieve information without normalization, and to retrieve information with absolute precision as well or better than normalization method. In this section, I explain the features of retrieval methods that use my experiments.

A value of tf is not effected on a normalization method. Because tf is a term frequency of a certain word in each document, and is fixed from each document. Here, I normalize the multiple databases in order to equate with the single database. Then the total document number N and the document number n that contain a certain word is fluctuated in the normalization. Therefore if much difference exists between a normalization score and a score each retrieval result from multiple databases, a value of idf is effected to retrieval scores. The difference of idf becomes evident in comparing figures shown in Section 4.2.1.

[4] To normalize qtf divided by the query length $\sum_{T \in Q} qtf$ in Formula 4.7, I omitted to divide by the query length since the query length does not affect the ranking of a score.

The *idf* means that Formula 4.2 of Okapi is;

$$\log \frac{N-n+0.5}{n+0.5} \tag{4.8}$$

then, Formula 4.6 of SMART is;

$$\log \frac{N}{n} \tag{4.9}$$

and, Formula 4.7 of INQUERY is;

$$\frac{\log \frac{N+0.5}{n}}{\log N+1} \tag{4.10}$$

When it is assumed that the distribution of the frequency of appearance of a word is invariable in the entire database, if a size of database becomes α times, the total document number N and the document number n that contain a certain word also becomes α times generally.

In this situation, the *idf* of SMART is invariability for the total document number N. The formula of Okapi is;

$$\begin{aligned} \log \frac{\alpha(N-n)+0.5}{\alpha n+0.5} &\simeq \log \tfrac{\alpha(N-n)}{\alpha n} \\ &= \log \tfrac{N-n}{n} \\ &\simeq \log \tfrac{(N-n)+0.5}{n+0.5} \end{aligned} \tag{4.11}$$

This formula shows *idf* of Okapi is almost invariability for the total document number N.

On the other hand, the formula of INQUERY is;

$$\begin{aligned} \frac{\log \frac{\alpha N+0.5}{\alpha n}}{\log \alpha N+1} &\simeq \tfrac{\log \frac{N+0.5}{n}}{\log \alpha N+1} \\ &= \tfrac{\log \frac{N+0.5}{n}}{\log \alpha+\log N+1} \end{aligned} \tag{4.12}$$

The denominator of $\log \alpha$ shows this *idf* is not invariability for the total document number N.

I confirm a value of *idf* each figure using virtual numbers. For example, the total document number N changes 1,000 to 100,000,000. The document number that contain a certain word n changes 1 to 100,000. The *idf* of SMART becomes

Table 4.1. Value of *idf* in the Okapi and the INQUERY

N	n	Value of Ex.(4.8)	Value of Ex.(4.10)
1,000	1	6.501790046	6.908255154
10,000	10	6.858014663	2.09163582
100,000	100	6.901772242	1.23239082
1,000,000	1,000	6.906255404	0.8735419263
10,000,000	10,000	6.90670483	0.676545069
100,000,000	100,000	6.906749784	0.5520495829

the fixed value of 6.907755279. The *idf* of Okapi and INQUERY is shown in Table 4.1. The *idf* of Okapi closes in on the fixed value, the *idf* of INQUERY progressively diminishes.

When targeted documents are divided into multiple databases at random, the value of the *idf* in Okapi and SMART method is equal to the fixed value that is calculated in normalization method at any sized databases. On the other hand, the value of the *idf* in INQUERY have a probability of a quite difference to the value in normalization.

If whatever words are equably found in targeted documents, the precision of Okapi and SMART method should become equivalent to the normalization method, because the value of *idf* in Okapi and SMART is not effected from size of databases as same as in the normalization method. I confirm the above mentioned things to compare precisions of the retrieval method of Okapi, SMART, INQUERY, and the normalization method. In this experiment, I divide the targeted documents into multiple databases and retrieve information from the multiple databases using the retrieval methods.

4.2.3 Merging Methods

For getting a retrieval result from multiple databases, it needs to put together multiple results from each database in some way. I use three retrieval methods explained in Section 4.2.1. Then merging methods to put together retrieval results from the databases by each retrieval method are used four following ways:

(1) Score Normalization (SN，正規化)

First, this method extracts information that is needed to compute score from all the databases such as frequency of words, total number of the entire document, and average of document length. Then, this method calculates similarity score using the extracted information and make a final result. This method gets the most accurate result to be equivalent to retrieve information from single database, because this method uses the information compiled by multiple databases. This method suffers from the disadvantage of the high cost of compiling information of multiple databases beforehand.

(2) Score

This method is to compile result applying to score of each retrieval result directly. If words appears biased in targeted documents, each score of retrieval results can not be compared directly. If whatever words are equably found in targeted documents, each score of retrieval results can be compared directly. I believe whatever words are equably found in the WWW data. So I confirm it by the experiment.

(3) Weighted Score (WS)

The weighted score is to calculate tendency of appearance words in each database, change score in accordance with the tendency, and then get a result according to recalculated scores. This method is approximate score normalization. It is not normalized information in whole the databases. It is normalized information in each database, so there is not disadvantage of the high cost of compiling information as same as the score method[10].

This method is calculated the score w of a certain document by a following formula:

$$w = 1 + |C| \times \frac{s - \overline{s}}{\overline{s}} \tag{4.13}$$

Here, $|C|$ is number of the targeted database, s is a score of a certain document in a certain database, $\overline{s}$ is average score of the entire documents in the database.

(4) Top

This method is just listing retrieval results according to the ranking results. First, it is gathered same ranking results from retrieval results in each database, then is listed at random in each ranking result[40]. This method introduced for case of available to use only ranking information of WWW retrieval.

Table 4.2. Average Precision and Precision of 10, 20 docs in the Okapi+SN, the SMART+SN and the INQUERY+SN (5DB Same Size per URL)

		qp-cont	qp-wlink
Okapi+SN	Ave P	0.1834	0.1572
	P@10	0.1739	0.2383
	P@20	0.1554	0.2106
SMART+SN	Ave P	0.1386	0.1179
	P@10	0.1891	0.2383
	P@20	0.1413	0.1872
INQUERY+SN	Ave P	0.1479	0.1067
	P@10	0.1739	0.1936
	P@20	0.1413	0.1553

4.3 Experiments

4.3.1 How to Test and Evaluate of Retrieval Methods

For retrieving information quickly and efficiently from a huge database, it is important to divide a database into appropriate sized data sets of multiple machines, to retrieve from multiple data sets at each machine, and to merge the whole results from multiple machines into a final result efficiently. When WWW documents are collected, automatic crawling systems are generally utilized. At that time, a crawling system collects WWW documents in the order of links in a WWW document. First, the crawling system collects all documents in a certain URL[5]. Then the system moves other URLs in the order of links in the URL. So the database of the crawling system are made on a URL. Then, I experiment in a case of dividing a targeted database on a URL and in a case of dividing the database at random These cases are considered to run a check on the situation as whatever words are equably found in the WWW data. Furthermore, I experiment in a case of dividing the database into same sized data sets and in a case of dividing the database into different sized data sets. These cases are considered to

[5]URL stands for Uniform Resource Locator and means WWW address.

investigate whether the influence of data size is correct or not such as I explained INQUERY is effect of data size in Section 4.2.2.

In this experiment, the data of the small collection (10GB) at a Web task in NTCIR3 was divided into several data sets in order to URLs and at random. Then by Okapi, SMART, and INQUERY methods, 2000 documents from each data set were retrieved for all queries in the NTCIR3. These retrieved documents were merged by Score Normalization (SN), Score, Weight Score (WS), and Top methods. I got the 1000 documents for a final result of each query. Then I compared a average precision, precisions of top 10 and top 20 documents [6]. I utilized the queries of the survey retrieval task in NTCIR3. The Japanese and English sample query is shown in Figure 4.2. In this experiment, I used only DESC (DESCRIPTION) in the query. The DESC represents the most fundamental description of the user's information needs in a single sentence.

The average precision (Ave P) of 96 topics, precisions of top 10 and top 20 documents (P@10, P@20) are calculated by trec_eval [7] and the NTCIR3 answers in two cases of considering content only in a document and considering links. The NTCIR Web Task tries to take two other assumptions, which assume hyper-linked pages or a passage to be an information unit, into the relevance assessment. The case of " considering content only" means that the assessor judges the relevance of a page only on the basis of the entire information given by it, as conventionally performed. The case of "considering links" means when the assessor judges the relevance of a page, the assessor can browse other pages that connected from the page under judging within one click distance.

For comparing abilities of the retrieval methods of Okapi, SMART, and INQUERY, the results for retrieving information from 5 data sets divided by URLs are shown in Table 4.2. Furthermore, I estimated a two-tailed t-test (t-test) for the average precisions in the case of considering content only of Okapi+SN, SMART+SN and INQUERY+SN [8].

[6] The average precision is to check from the top ranked document, when finding adapted document, to calculate a precision at the point, then finally to get the average using all precisions[25].

[7] ftp://ftp.cs.cornell.edu/pub/smart/trec_eval.v3beta.shar

[8] The t-test in this chapter is estimated by using average precision pair of each query. The number of query is 47, then this t-test is a two-tailed t-test of the degree of freedom $n = 46$.

```
<TOPIC>
<NUM>0008</NUM>
<TITLE CASE=’’b’’>サルサ，学ぶ，方法</TITLE>
<DESC>サルサを踊れるようになる方法が知りたい</DESC>
<NARR>
 <BACK>最近はやっているサルサという踊りを
 学ぶためにどうすればよいか具体的な方法が
 知りたい．例えば教室に通うという場合には,
 その場所や授業形態など，具体的な内容を必
 要とする． </BACK>
 <RELE>具体的な方法の表記のない，流行であること
 のみを扱った文書は不適合とする．</RELE>
</NARR>
<CONC>サルサ，習う，方法，場所，カリキュラム</CONC>
<RDOC>NW011992774, NW011992731, NW011992734</RDOC>
<USER>大学院修士 1 年，女性，検索歴 2.5 年</USER>
</TOPIC>

<TOPIC>
<NUM>0008</NUM>
<TITLE CASE=’’b’’>Salsa, learn, methods</TITLE>
<DESC>I want to find out about methods for learning how to dance the salsa</DESC>
<NARR>
 <BACK>
 I would like to find out in detail how best to learn how to dance the salsa,
 which is currently very popular.
 For example, if I should go to dance classes, I need detailed information
 such as where I should go and what the class would be like.
 </BACK>
 <RELE>
 Documents simply saying that it is popular without giving any detailed
 information are irrelevant.
 </RELE>
</NARR>
<CONC>Salsa, learn, methods, place, curriculum</CONC>
<RDOC>NW011992774, NW011992731, NW011992734</RDOC>
<USER>1st year Master’s student, female, 2.5 years search experience</USER>
</TOPIC>
```

Figure 4.2. NTCIR3 Web task Sample Query in Japanese and English

Table 4.3. Value of Paired t-test in the Okapi, the SMART and the INQUERY

	SMART+SN	INQUERY+SN
Okapi+SN	0.0048 **	0.0301 *

Table 4.4. Results for Average Precision without considering links (5DB, 10DB, 20DB per URL)

Ave P		SN	Score	WS	Top
Okapi	5DB	0.1834	0.1841	0.1833	0.1350
	10DB	0.1834	0.1788	0.1788	0.1207
	20DB	0.1834	0.1775	0.1675	0.0963
SMART	5DB	0.1386	0.1341	0.1379	0.1044
	10DB	0.1386	0.1320	0.1402	0.0977
	20DB	0.1386	0.1292	0.1411	0.0809
INQUERY	5DB	0.1476	0.1497	0.1404	0.1236
	10DB	0.1476	0.1498	0.1493	0.1115
	20DB	0.1476	0.1519	0.1496	0.0950

Table 4.5. Results for Precision at 10 docs without considering links (5DB, 10DB, 20DB per URL)

P@10		SN	Score	WS	Top
Okapi	5DB	0.1739	0.1717	0.1739	0.1565
	10DB	0.1739	0.1717	0.1826	0.1500
	20DB	0.1739	0.1739	0.1826	0.1326
SMART	5DB	0.1891	0.1978	0.1848	0.1478
	10DB	0.1891	0.1870	0.1848	0.1370
	20DB	0.1891	0.1761	0.1848	0.1065
INQUERY	5DB	0.1739	0.1739	0.1674	0.1696
	10DB	0.1739	0.1696	0.1630	0.1609
	20DB	0.1739	0.1717	0.1609	0.1304

Table 4.6. Results for Average Precision without considering links (5DB Same Size per URL and at Random)

Ave P		SN	Score	WS	Top
Okapi	URL	0.1834	0.1841	0.1833	0.1350
	Random	0.1834	0.1842	0.1848	0.1519
SMART	URL	0.1386	0.1341	0.1379	0.1044
	Random	0.1386	0.1351	0.1350	0.1305
INQUERY	URL	0.1476	0.1497	0.1404	0.1236
	Random	0.1476	0.1490	0.1495	0.1438

At the P@10, the t-test did not indicate a statistically significant difference between any methods. The t-test indicated a statistically significant difference at the Ave P between Okapi+SN and others. The results are shown in Table 4.3 [9]. These results show the P@10 is not difference in three retrieval methods, but the Ave P of considering content only shows the good method in order of Okapi, INWUERY, and SMART.

4.3.1.2 Merging Experiments for Same Sized Data Sets

I explained the experiments for the same sized data sets. I examined in the case of dividing 5, 10, 20 data sets. In the case of 5 data sets (5DB), the size of a data set is about 500MB. The size is about 250MB for 10 data sets (10DB), about 125MB for 20 data sets.

I investigated four merging methods that is SN, Score, WS and Top for the data sets dividing by URLs of 5DB, 10DB, and 20DB. The Ave P of considering content only (qp-cont) is shown in Table 4.4, the 10@P of qp-cont is shown in Table 4.5. And I estimated the t-test. In the case of the merging methods of SN, Score and WS, there is not any difference between the number of data sets, but the merging method of Top is statistically significant at the 1% level when the number of data sets is different. To show the difference of precision in 5DB, 10DB and 20DB, the recall and precision curves are shown in Figures 4.3 and

[9] The single * mark means statistically significant at the 5% level, the double ** mark means statistically significant at the 1% level.

Table 4.7. Results for Precision at 10 docs without considering links (5DB Same Size per URL and at Random)

P@10		SN	Score	WS	Top
Okapi	URL	0.1739	0.1717	0.1739	0.1565
	Random	0.1739	0.1739	0.1717	0.1630
SMART	URL	0.1891	0.1978	0.1848	0.1478
	Random	0.1891	0.1870	0.1891	0.1826
INQUERY	URL	0.1739	0.1739	0.1674	0.1696
	Random	0.1739	0.1739	0.1717	0.1696

Table 4.8. Results for Average Precision without considering links (5DB Diff Size at Random)

Ave P		SN	Score	WS
Okapi	qp-cont	0.1840	0.1843	0.0929
	qp-wlink	0.1579	0.1592	0.0785
SMART	qp-cont	0.1367	0.1371	0.0691
	qp-wlink	0.1179	0.1163	0.0588
INQUERY	qp-cont	0.1488	0.1489	0.0752
	qp-wlink	0.1071	0.1076	0.0539

Table 4.9. Results for Precision at 10 docs without considering links (5DB Diff Size at Random)

P@10		SN	Score	WS
Okapi	qp-cont	0.1739	0.1717	0.1239
	qp-wlink	0.2404	0.2383	0.1596
SMART	qp-cont	0.1891	0.1870	0.0935
	qp-wlink	0.2383	0.2362	0.1191
INQUERY	qp-cont	0.1696	0.1674	0.0913
	qp-wlink	0.1894	0.1872	0.1128

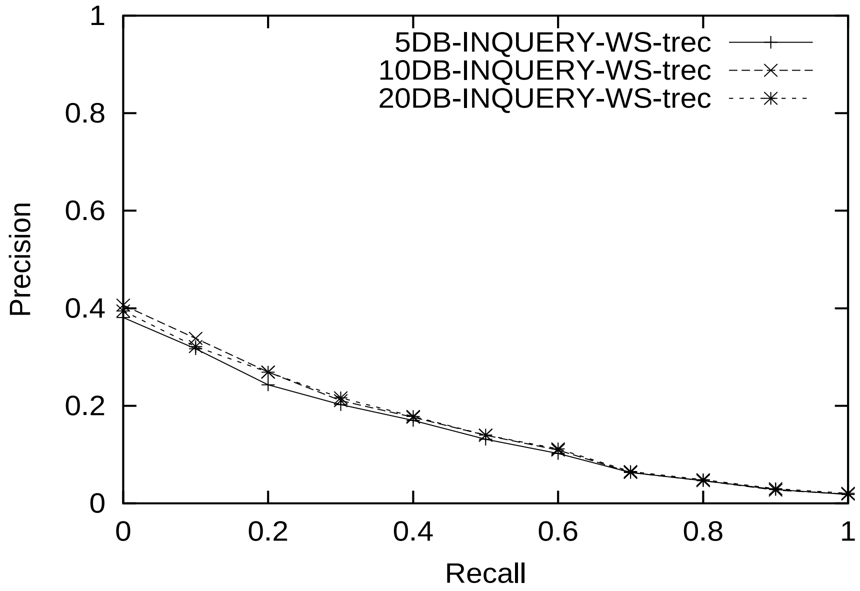

Figure 4.3. INQUERY+WS's Recall-precision curves without considering links (5DB, 10DB, 20DB)

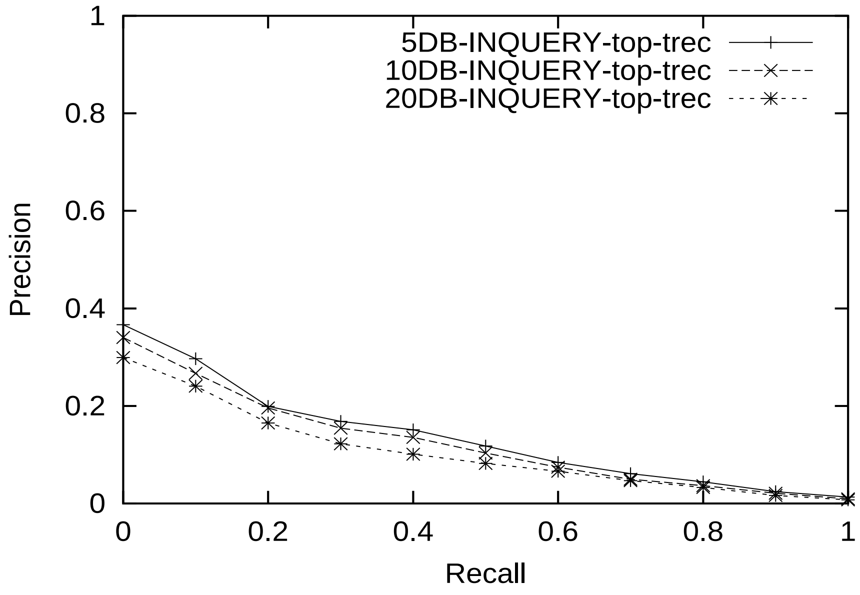

Figure 4.4. INQUERY+Top's Recall-precision curves without considering links (5DB, 10DB, 20DB)

4.4. As shown by the result of t-test, rates of precision in Top are worsened in order to rise numbers of data sets.

I investigated the merging methods in the case of dividing at random and by URLs. The results are shown in Tables 4.6 and 4.7. I also estimated t-test in these cases. There are not statistically significant in any cases with SN, Score and WS, except for Top merging method.

The above mentioned things mean the Top merging method is effected of the number of data sets and the dividing way as at random or by URLs. On the other hand, the SN, Score and WS merging methods are not effected of the number of data sets and the dividing way.

4.3.1.2 Merging Experiments for Different Sized Data Sets

In this section, I experimented in the case of dividing data into different sized data sets. At first, the NTCIR3 documents were divided into 5 data sets that the size of each data set is about 50MB, 100MB, 250MB, 600MB and 1GB. There are also two ways to divide at random and by URLs. Here, I omitted the Top merging method because the Top was effected by the size of data sets in Section 4.3.1.1. The Ave P and P@10 are shown in Tables 4.8 and 4.9.

In the case of different sized data sets, I think some sort of difference to each retrieval method. I estimated the t-test of the Ave P and P@10 for the merging methods as SN, Score and WS along with in the case of dividing into 5 same sized data sets. The results were shown in Table 4.10. In this case of dividing data into 5 different sized data sets, the results of t-test in the SN and Score are no difference as indicated on the Table 4.10. The result of the WS is statistically significant at the 1% level. It means that although the WS normalizing approximately in each data sets is effective in the case of dividing data into same sized data sets, is nonfunctional in the case of dividing data into different sized data sets.

In the case of different sized data sets, the precisions of Okapi+SN, INQUERY+SN and SMART+SN were compared. The results of recall-precision curves were shown in Figure 4.5. The precision of the Okapi method got the best result as indicated on the figure 4.5. Furthermore, the precisions of Okapi, INQUERY and SMART with each merging method were compared in the different sized data sets. The recall-precision curves of the INQUERY were shown

Table 4.10. Results of Paired t-test on all merging methods (5DB Diff Size at Random)

		SN	Score	WS
Okapi	P@10	1	0.17803	0.0051 **
	Ave P	0.8104	0.7223	$1{,}55\times10^{6}$**
SMART	P@10	1	0.7040	1.52×10^{5} **
	Ave P	0.6108	0.6918	6.99×10^{7} **
INQUERY	P@10	0.3979	0.1732	2.21×10^{5} **
	Ave P	0.6102	0.7246	2.59×10^{6} **

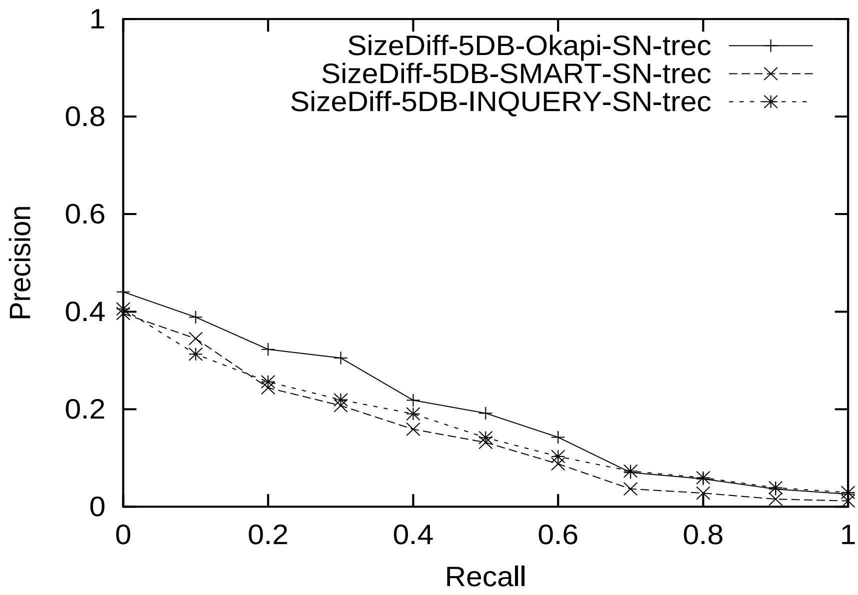

Figure 4.5. SN's Recall-precision curves without considering links (5DB Diff Size at Random)

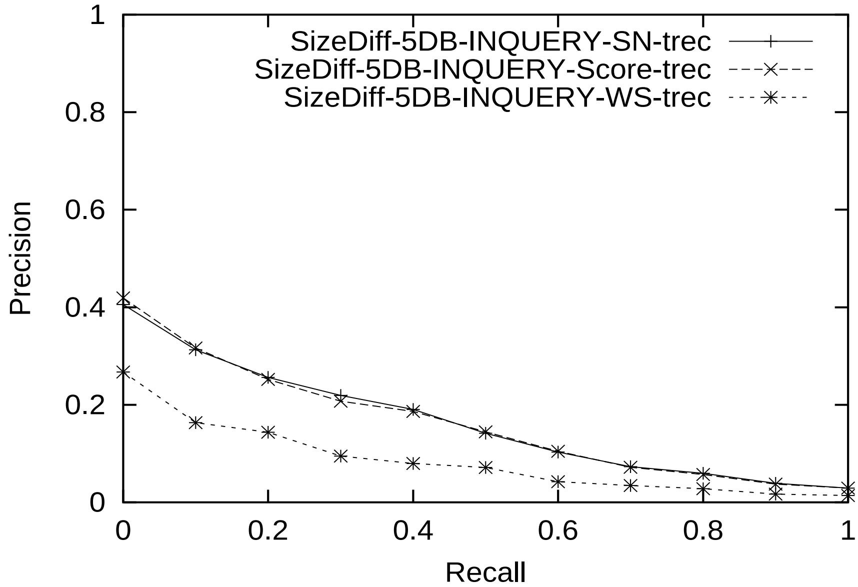

Figure 4.6. INQUERY's Recall-precision curves without considering links (5DB Diff Size at Random)

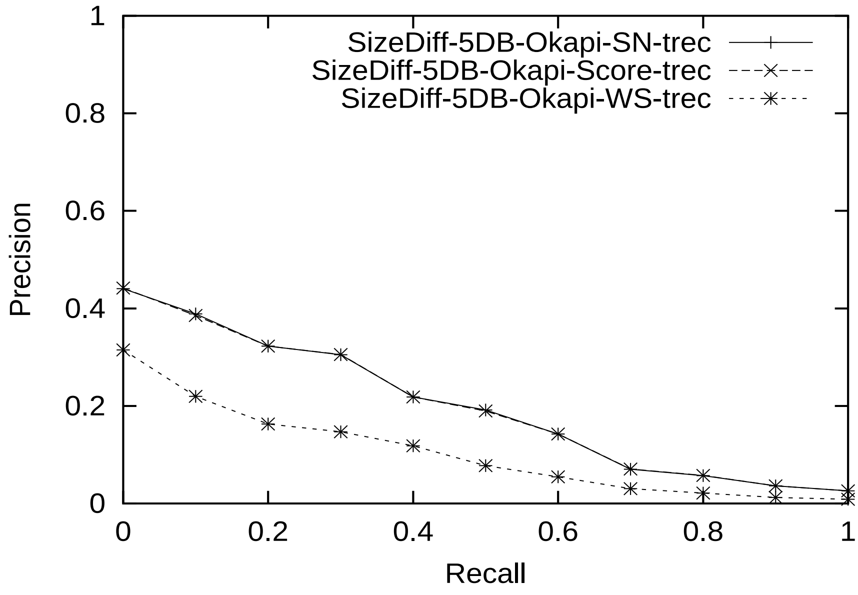

Figure 4.7. Okapi's Recall-precision curves without considering links (5DB Diff Size at Random)

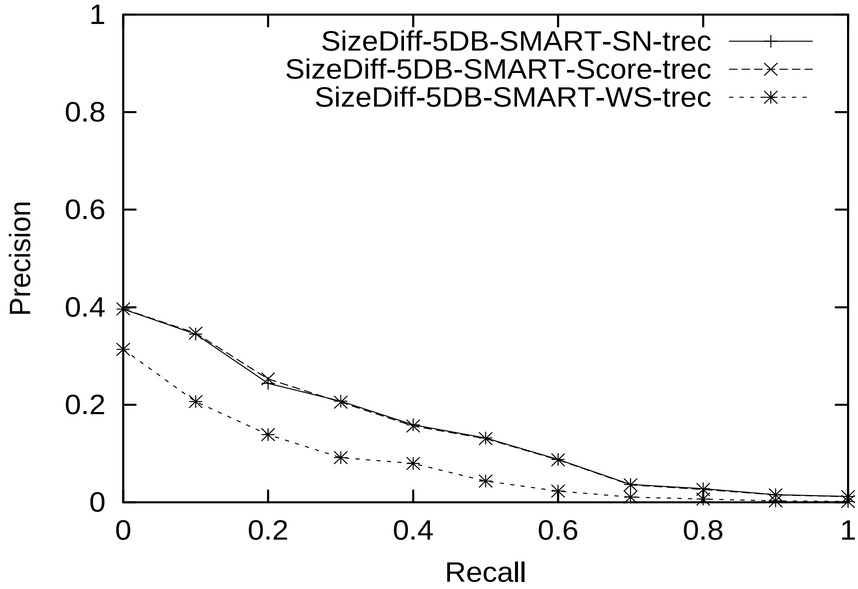

Figure 4.8. SMART's Recall-precision curves without considering links (5DB Diff Size at Random)

in Figure 4.6, the curves of the Okapi were shown in Figure 4.7, and the curves of the SMART were shown in Figure 4.8. The results of the SN and Score were overlapped in any Figures. So it means that the SN is equal to the Score.

4.3.2 Comparing Rank Order

I explained in Section 4.2.2 that if much difference is existed between a normalization score and a score each retrieval result from multiple databases, a value of *idf* is effected to retrieval scores. The difference of a *idf* value can be examined to compare scores of all retrieval methods. However, to compare scores in all queries between retrieval methods is cumbersome procedure. Then I compared the Ave pre of Score merging method in all retrieval methods.

The results in Section 4.3.1 showed the precisions of the SN and Score were not any difference in any retrieval methods for the different sized data sets. The t-test result of INQUERY+SN and INQUERY+Score was not statistically significant. This t-test result is a perverse effect. I think this result is involved that the assessment of the trec_eval is complied with ranking orders regardless scores of the retrieval results. Then I compared the Kendall's rank correlation coefficients of each retrieval methods with the SN. This means that to compare the entire ranking order of all answers in three pairs of Okapi+SN and Okapi+Score, INQUERY+SN and INQUERY+Score, SMART+SN and SMART+Score and to show comparison using the Kendall's rank correlation coefficients. Higher average of the coefficient means higher degree of positive correlation. Smaller dispersion of the coefficient means higher degree of positive correlation more precisely.

The results of the Kendall's rank correlation are shown in Table 4.11. The correlations of the SN and Score in any retrieval methods are high degree, the correlations in the Okapi and SMART are higher degree than the correlation in the INQUERY. Additionally the dispersion of the INQUERY+SN and INQUERY+Score is bigger than the dispersion of the Okapi+SN and Okapi+Score, SMART+SN and SMART+Score. I think that the reason for this is to exist some queries that dropped to a lower precision to a fault with the INQUERY owing to different sized data sets. In this case, although the average coefficient has not been affected, the dispersion of the coefficient has been bigger in consequence of occurrence of the lower precisions. On the other hand, the results of the Okapi+SN and Okapi+Score, the SMART+SN and SMART+Score are shown higher degree of the correlations, and smaller dispersions. Then these methods have not been affected the size of the data sets. Furthermore, to make pairs of the Kendall's rank correlation coefficients of the Okapi, SMART and INQUERY, I estimated the t-test. The result of the t-test between the Okapi and SMART is 0.0635, so not statistically significant. The result of the t-test between Okapi and INQUERY is 3.31×10^{22}. The result between SMART and INQUERY is 6.767×10^{23}, then statistically significant at the 1% level. The results of this experiment are shown that the Okapi and SMART are completely unaffected the size of data sets, the INQUERY is affected by the size of data sets.

4.4 Discussion

I concentrate my discussions on the INQUERY+SN, INQUERY+WS, Okapi+SN, Okapi+Score, SMART+SN and SMART+Score. Because the SN method is well known to the public as getting higher precision, and "the INQUERY+WS gets

Table 4.11. Average and Variance of Kendall's rank correlation coefficents (5DB Diff Size at Random)

	Okapi	SMART	INQUERY
Average	0.97565	0.97799	0.8250
Variance	0.00016	4.59×10^{23}	0.00484

higher precision" is evident in the related works [53]. Furthermore, I think that the WWW data has a feature such as the distribution of the frequency of appearance of a word is invariability in the entire database, because the retrieval results of dividing follow URLs and at random, make no difference in the experiments of Section 4.3. If this is true, a value of *idf* stays constant, so the precisions of the Okapi+Score and SMART+Score are equal to the precisions of the Okapi+SN and SMART+SN as noted in Section 4.2.2.

In the case of the same sized data sets, the precisions of Okapi, SMART and INQUERY with SN, Score and WS are not difference. Then the merging methods of SN, Score and WS are considered about the same abilities for the same sized data sets. On the other hand, in the case of the different sized data sets, the precisions with SN and Score are better than the precision with WS, the result of t-test is statistically significant at the 1% level. Then the WS is not efficient in the different sized data sets.

To consider general WWW situation, multiple data collecting systems gather WWW data in parallel. The size of the database in each collecting system is differentiated according to the network circumstance and the abilities of the collecting systems. The case of the different sized data sets is about equal to the real WWW situation.

If the WWW data has a feature such as the distribution of the frequency of appearance of a word is invariability in the entire database, a value of *idf* is a very important factor in retrieval methods. Then there is a difference of a part of *idf* in the formula of the Okapi, SMART and INQUERY. The formula of INQUERY (4.12) is affected on the size of database, the formula of Okapi (4.11) is not affected on the size of database. The value of *idf* of SMART is a constant number, so is not affected on the size of database. In the case of the different sized data sets, I found Okapi+Score and SMART+Score was equal to Okapi+SN and SMART+SN as discussed in Section 4.3.1. Herefrom, to retrieve information from WWW data with Okapi+Score and SMART+Score is possible to be lower cost and same level precision to compare with Okapi+SN and SMART+SN. This means that the information retrieval system omits normalizing process when the system retrieves information from multiple databases.

In the experiments of this chapter, it can be said that the WWW data has a feature such as the distribution of the frequency of appearance of a word is invariability in the entire database unlike the database of newspaper articles. I think the database of newspaper articles has a certain deviations of the frequency of appearance of a word according to the contexts, the deviation in the WWW data may be countered by huge volumes of data. Then, if the retrieval methods such as Okapi and SMART no effect of the size of database are employed to the retrieval system, it is possible to merge multiple results using scores of the retrieval method directly without score normalization.

Finally, I compare the precisions of the Okapi and SMART methods. Each method can retrieve information no effect of the size of database. The average precision of the Okapi is better than the SMART. On the other hand, The precision of top 10 documents of the SMART is better than the Okapi. For retrieval WWW document, users usually check only at the head of retrieval results. Then it can be said that the SMART is better suited for WWW retrieval. However, I think that this results come from a value of *slope* in the SMART formula (4.5). The precision of top 10 documents of the Okapi can be improved to coordinate a value of b in the Okapi formula (4.3). I implemented a value of utf and $pivot$ in the SMART as a value of dl and $avdl$ in the Okapi. Then the average precision of the SMART gets worse. Whichever retrieval methods are not major difference of the precision, the cost of calculation, and can retrieve information from the WWW data.

4.5 Conclusions of Chapter 4

I explain about the merging methods to multiple retrieval results and the retrieval methods from multiple data sets for retrieving information efficiency and with high accuracy from a huge amount of data. In the research for newspaper articles, the best method is estimated to be the score normalization method. However, the normalizing process is a heavy process, and needs a great deal of calculating cost for a huge amount of data. Then I experiment the retrieval methods without normalization are equal the ability to the score normalization.

In the experiments, I found the Okapi, SMART method using the score directly has the ability as same as the Okapi, SMART method with score normalization. It shows the WWW data has a feature such as the distribution of the frequency of appearance of a word is invariability in the entire database. Of course, it seems this results came from the WWW data using NTCIR. The real WWW data changes the whole time. The results may be changed according to collecting methods or the WWW situation. In this point, I should continue to research in detail.

The precision of the Okapi+Score is equal to the precision of the best systems participated in the NTCIR3. However this precision for WWW data is by no means satisfactory to compare the precision for newspaper articles. To improve the precision for WWW data, the method using the link information [11] and using Relevant Feedback [7, 33] have been developing. These methods have several problems

In this retrieval process, users should input appropriate keywords as queries. However, query selection is difficult for users that have vague purposes. Then I think that selecting a keyword suited users' purpose is important for retrieving information from WWW data efficiently. Next chapter, the research of this point is explained and evaluated.

Chapter 5

Task-Oriented Information Recommendation Systems

5.1 Introduction

The recent proliferation of the Internet has enabled us to easily obtain vast amounts of electronic texts. However, finding the information that we really require has become more difficult, and concurrent development in search technologies is necessary.

In general, a Word Wide Web (WWW) information retrieval system needs a keyword for a query. Query selection is difficult for users, and a short query is not effective to retrieve the information[20].

To support users in selecting queries, recommendation systems[44] and decision support systems[17] have been developed, using databases as domain knowledge. The traditional systems are available only manually and not up-dated.

Since the information on the Internet changes from time to time, the databases of these systems should be modified according to the changes on the WWW. Therefore, an automatic renewal of the database is very important. Furthermore, the value of the Internet information may also change rapidly. For example, service information regarding the time schedules of a certain shop, or term-limited discount ticket information must be accessed within a deadline. If recommendation systems can aggressively extract the information valuable to users, the systems will be able to introduce the recommendation worthy to use.

I have been developing such a recommendation system with a limited task that I can utilize value of information aggressively with minimal inputting queries. If given a limited purpose, the system has the advantage of recommending appropriate queries to the users for their retrieving information effectively, and gathering information precisely. In this research, I have been developing the support system for making up the tourist routes in Nara.

The system can help tourists get information and prepare their itineraries. I believe that the technology involved in the retrieval of tourist route information is important from the following research aspects:

1. Query selection, is difficult for visitors who do not know the sightseeing area.

2. Various formats are provided through each home page of travel companies, the individual diaries, and municipalities.

3. Event information has an ever-changing value that needs to be assessed and updated.

A support system for tourists should extract data that is related to (3) information with constantly changing values. In other words, when users look for sightseeing information, the support system will help them check the value of the information. Therefore, the system is expected to judge the value of information. The ever-changing value information is associated with event information such as festival schedules for the period when a certain exhibition is open. In the first step, I attempted to extract event information from the Mainichi Shinbun (a Japanese daily newspaper) automatically. Because newspapers have a certain format to describe event information, it is easy to figure out if articles have real event information or not.

Session 5.2, describes the support system for producing tourist routes. Then my experiments of event information extraction using features of keyword appearance, are described in Section 5.3.

5.2 Support Systems for Producing Tourist Routes

In general, travelers usually check the places of interest, event information, how to get there, and prepare a rough itinerary in advance. Recently, a variety of sightseeing information has increased on the Internet and in particular, I have focused on tourist access of this content.

Several tour simulation systems[1] currently exist on WWW, with individual sightseeing information databases. Users will select particular places from the database and simulate tour routes, but they must first do a Google search related to their travel destination to check which location is the best to visit. If users did not have enough information about sightseeing spots, users have difficulty in getting new sightseeing information. This process of checking places is both time-consuming and onerous. For example, when users want to go to “奈良 (Nara)”, they input “奈良 (Nara)” and “観光 (sightseeing)” to the robot type search engine such as Google system and get 90,000 homepages as a result. If users use the category type search engine such as Yahoo system, they get 124 homepages[2]. If these results had ranking information, the ranking is not always true to match users’ needs. Users should continue to find out their target information from the results. Thus, selection of valuable sightseeing places or event information using these search engines is not easy. Especially, users who are not familiar to 奈良 (Nara), will have a trouble to search or select the keywords of event information in 奈良 (Nara).

If a user found the latest event information on WWW, then the user inputs the event information with a tour simulation system. In general, the tour simulation system has individual database, which takes forever to update the database. So the user cannot get appropriate sightseeing routes in spite of inputting the latest event information. Some systems are available which introduce some sightseeing routes including the latest event places. The recommended sightseeing routes are under control of the site administrators, and the routes cannot be changed in

[1]For example, at the site of Japan tourism association (http://www.nihon-kankou.or.jp), users can create samples of tourist routes.

[2]I checked this results in the end of October 2003

compliance with the users wishes.

To sum up, the traditional support system has the following problems:

1. The users who do not know about sightseeing spots have some disadvantages with the system.

2. The system is not flexible enough for the users as mentioned above.

3. The users cannot access to the latest information on the WWW through this system.

To settle the above problems, I have developed a support system helping the tourists with producing their routes. In fact, the system can help tourists find information and prepare their itineraries[37]. In this research[37], I have showed that the following key-points were important for consideration of, the easy-to-use support system for sightseeing routes production.

A The system can update the database of sightseeing information automatically or the system manager can update the database easily and quickly.

B The system can deal with information related "Time Information" for producing sightseeing routes.

C The system can tell directions between sightseeing places.

I think the key-point C can be realized by extracting place information obtained from WWW; for example, "Mapion", "MapFan", and other navigation sites. Furthermore, several touring activity models are proposed in the research area of civil engineers[31]. Especially, there are many important cultural assets in Nara area, many researchers are under intense study of touring activity of Nara. Then, if users can select tourist spots of Nara, the system can help them to product touring routes using the touring activity models. Regarding the key-point B, several researchers have focused on extracting time information [30, 36], and I realized to modify my proposed extraction methods to fit the support system for producing sightseeing route that suited users' schedules.

Here, in order to solve problem 3 described above, it is important (related point A) to collect "in-season" sightseeing event information from WWW. For this purpose, I obtain at first the keywords which will lead me to search a sightseeing event appropriately, then I will collect the event information on WWW using the keywords as queries.

However, it is difficult for users to select appropriate keywords. Then I have noted the following points for the ease of keywords selection.

Since sightseeing events are in general held cyclically, I have assumed that events or keywords related to the events will appear independently each in news articles or mail magazines including data information, by which I will be able to extract event information efficiently using keywords that appear periodically in newspaper corpora, as a retrieval query. I tried to extract event information using periodic words as a query, and found that the information on the events performed periodically may have a certain similarity of description form, co-occurrence relation words, etc. similarly with the information on the events performed irregularly. Also I examined the query expansion that extracts co-occurrence relation words with periodic words to retrieve the event information.

5.3 Experiment: Event Information Extraction

In this section, I explain preparatory text of event information extraction, my idea to extract event information, evaluation experiment of my idea, and the results of the evaluation experiment. For this experiment, I used about 30,000 articles from the Mainichi Shinbun (Nara local edition) for 7 years from 1996 to 2002.

5.3.1 Preparatory Test: determining whether users could recall keywords

In Section 5.2, I explained that users who were not familiar to sightseeing places, should be in trouble searching or selecting keywords of events information in the sightseeing places. In order to confirm whether the above mentioned things are true, I conduct the preparatory test that users can recall the event words as a query of the retrieval systems. The sightseeing event information can be retrieved efficiently using the word showing events, such as a festival name as a reference keyword. Then I sent emails to 50 people who live in the Kansai area and in the Kanto area, to ask if they can produce event words related to festivals and exhibitions in Nara [3].

I received 30 answers. Sixteen answers were received from the Kansai area (KS-respondents), 14 answers were received from the Kanto area (KT-respondents).

The average number of event words that KS-respondents could recall was 3.1 words, and that of KT-respondents was 1.4 words. The KS-respondents produced variety words from the traditional festivals (e.g., Omizutori and Ohchamori), to the newfangled festival (e.g., the Basara-festival). The 7 KT-respondents, on the other hand, could not produce any words related to events in Nara. The 7 KT-respondents only recalled traditional, well-known events (e.g., Yamayaki, Tsunokiri). So, I found the event of Nara was not known other than the limited things that were historical buildings.

People who live far from sightseeing spots cannot recall many terms for events. It is difficult for visitors who are not familiar with sightseeing spots to input appropriate keywords in order to retrieve appropriate sightseeing information.

5.3.2 Idea of Event Information Extraction

For users who were unsure of sightseeing spots, recalling keywords that were related to sightseeing events was difficult in Section 5.3.1. Therefore, I have generated two hypotheses as follows:

As a premise, users who are unsure of sightseeing spots can not recall event keywords. Events usually are held periodically, the events or the related words appear periodically in the data which date information like newspaper or mail magazines is carrying out clearly. So if the words which appear periodically are collected, the word related to an event is automatically collectable[41]. Then

Hypothesis 1: Users could extract event information using a list of periodic

[3]The Kansai area of Japan lies in the middle of Japan's main island, Honshu. The Kanto area is located at the east of Kansai area in Honshu. The Kansai area includes the prefectures of Nara, Wakayama, Kyoto, Osaka, Hyogo, and Shiga. The Kanto area is comprised primarily of Tokyo and the surrounding area. Its boundary is nearly the same as that of the Kanto plain. The Kansai area is often compared with the Kanto area.

terms that describe in Section 5.3.3, more accurately than by using recall terms.

Next, some events held irregularly. The information on the event held periodically may have a certain similarity of description form, co-occurrence relation words, etc. with the information on the event held irregularly. Then,

Hypothesis 2: Users could extract irregular event information using co-occurrence relation words with periodic words.

I verified these two hypotheses in extracting event articles from the Mainichi Shinbun using four types of queries such as;

Pattern 1: keywords that users could produce, i.e., "recall words" in Section 5.3.1 （Recall Words, R）[4],

Pattern 2: keywords that users selected from the list of periodic words, i.e., "Select Words" in Section 5.3.3 （Select Words, S）,

Pattern 3: keywords that had 5 words added using recall words (R) at the query expansion (Recall word Expansion, RE),

Pattern 4: keywords that had 5 words added using select words (S) at the query expansion (Select word Expansion, SE).

Consequently, if the following things can be said, the above mentioned hypotheses are true. To compare with results of the pattern 1 and the pattern 2, if the results in the pattern 2 get better than the results of the pattern 1, it can be said that the hypothesis 1 is true. To compare with results in the case of without expansion (the pattern 1 and 2) and with expansion (the pattern 3 and 4), if the results of with expansion get better than the results of without expansion, it can be said that the hypothesis 2 is true.

In the following sections, I explain the method of extraction periodic words and the method of keyword expansion, and report the experiment results.

[4] For users who could not produce any words in Section 5.3.1, I use "奈良 (Nara)" as a Recall Word.

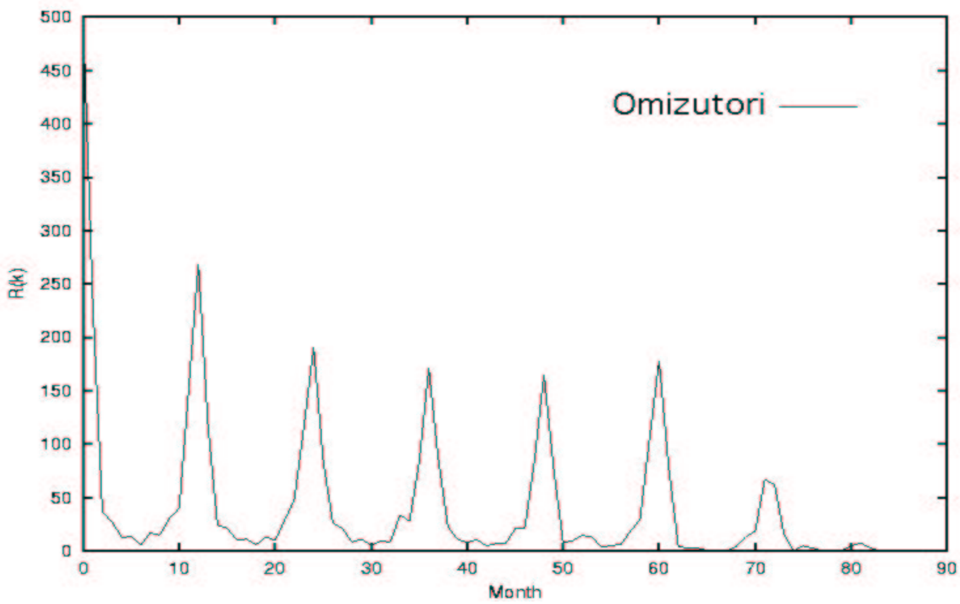

Figure 5.1. Autocorrelation Function Results (OMIZUTORI)

5.3.3 Periodic Word Extraction

I built the list of periodic words used in the Pattern 2, to collect the words that appeared periodically in Mainichi Shinbun.

Events were held during certain seasons at some resorts. For example, the Omizutori is a famous festival in Nara, is held in the spring every year. These events are always reported in the newspapers just before and after they begin. Therefore, terms related to these events appear periodically in news articles or e-mail advertisements. I conducted a preparatory test to verify the periodicity of event terms. I employed the autocorrelation function. The function was able to extract periodicity for the appearance of the term, which was widely used to detect periodicity in speech signals[16]. Autocorrelation function $R(k)$ is given by formula (5.1).

$$R(k) = 1/N \sum_{n=0}^{N-1} x(n) \times x(n+k) \tag{5.1}$$

$$(k = 0, 1, 2, ..., N-1)$$

$x(n)$ is frequency of a certain word in nth month. At the autocorrelation function result $R(k)$ of a certain word, the value of k in case $R(k)$ takes maximum, is the temporary cycle of the certain word. When values are several times the value of k, if $R(k)$ have peak value and the values of $R(k)$ between peak values are below a certain threshold[5], at that time, the certain word is extracted as a periodic word. For example, shown in Figure 5.1[6], "お水取り (Omizutori)" have peak values at $k = 12$ and peak values appear periodically and the values between peak values are below the threshold value, "お水取り (Omizutori)" are extracted as periodic words. As periodic words, I extracted "お水取り (Omizutori)", "綱引き (tug-of-war)" "歳旦祭 (Saitan festival)", "修二会 (Shunie)", "除夜祭 (Joya festival)", etc. that are candidate for event terms [7].

Next, for getting "Select word (S)" of Pattern 2 in Section 5.3.2, I send the respondents (30 people) the list of periodic words, I ask them to select words that they think event words, from the list. I obtained 19 answers. Twelve were from KS-respondents, and seven were from KT-respondents.

5.3.4 Query Expansion

For the pattern 3 in Section 5.3.2, I extracted co-occurrence relation words "recall word expansion (RE)" with "recall word (R)". I extracted 5 RE words each R word. For the pattern 4 in Section 5.3.2, I extracted co-occurrence relation words "select word expansion (SE)" with "select word (S)". I also extracted 5 SE words each S word. I extracted co-occurrence keywords using the log likelihood ratio [15] that is calculated by the formula (5.2).

Log likelihood ratio λ of terms v and w is the likelihood ratio that is calculated by the maximum likelihood estimator of a case where term v is subordinate to term w, and a case where term v is independent of term w [59].

---

[5] In this experiment, I set the threshold value 50 experientially.

[6] In this figure, the score of $R(k)$ is noticeably normalized by 0 to 500. I used the Mainichi Shinbun for 7 years, so $N = 82$ (7 years x 12 months).

[7] Using function 5.1, I extracted 189 words for the periodic words. Of course, not all extracted terms are event terms. I wanted to obtain keywords such as a few useful hints that would support users in making tourist plans. I do not discuss precise extraction of event terms using periodicity. In addition, these terms are extracted by Chasen 2.2.9 [28] using a special dictionary adding Nara place names.

$$\lambda = 2\sum_{i,j} f_{ij}\left\{log\frac{f_{ij}}{F} - log\frac{f_{i.}f_{.j}}{F^2}\right\}(i,j \in 1,2)$$

Here, $f(v,w)$ is the number of document contained in both word v and w. $f(x)$is the number of document contained in word x. F is the number of whole document, then; $f_{11} = f(v,w)$,$f_{12} = f(v) - f(v,w)$,$f_{21} = f(w) - f(v,w)$, $f_{22} = F - f_{11} - f_{12} - f_{21}$. And $f_{i.} = f_{i1} + f_{i2}$, $f_{.j} = f_{1j} + f_{2j}$.

5.3.5 Objective Articles and Evaluation

At the experiment in Section 5.3, I used abut 30,000 articles from the Mainichi Shinbun (Nara local edition) for 7 years from 1996 to 2002.

I determined the destination articles that were extracted from Mainichi articles with the IR package [8] version 1.54[59]. The queries were keywords from event titles in the sightseeing data extracted pages on the Nara prefectural site [9]. I checked the accuracy of all retrieved articles manually, then I got 1,425 articles as the objective articles, are shown in Figure 5.2. I used "precision at top n documents (Precision)"[25] i.e., "document-level precision" after 5, 10, 15, 20, 30, 100, 200, 500, and 1000 documents(articles) respectively retrieved [10].

5.3.6 Experiment at Pattern 1 and Pattern 2

I extracted articles using the recall words (R) of pattern 1 and the select words (S) of pattern 2, from Mainichi Shinbun. Then I calculated the precision at top n documents (precision) to compare with the objective articles that were explain in Section 5.3.5. The results are shown in Figure 5.3. The vertical axis shows the top number of articles and the lateral axis shows the precision at top n articles (precision) in Figure 5.3. Using R of pattern 1, average precision rate of nine precision rates each at the top n document is 18%, the maximum precision is 25% at the top 5 articles. On the other hand, using S of pattern 2, the average

[8]The package is available at http://www.crl.go.jp/jt/a132/members/mutiyama/index.html.

[9]http://yamatoji.pref.nara.jp

[10]This evaluation measure can be computed using "trec_eval", a program to evaluate TREC results, which is available at ftp://ftp.cs.sornell.edu/pub/smart/trec_eval.v3bata.shar.

```
<DOC NAME=''19960405-M2L-1141''>
<DATE>19960405</DATE>
<TITLE>６日から「桜まつり」ーー大和高田市</TITLE>
<TEXT>
大和高田市築山の築山児童公園と同市大中の大中公園で
６日から２１日まで恒例の「桜まつり」が開かれる.
（途中略）
カラオケ大会や福引きなどもある.
</TEXT>
</DOC>
```

Figure 5.2. Sample of Answer Articles

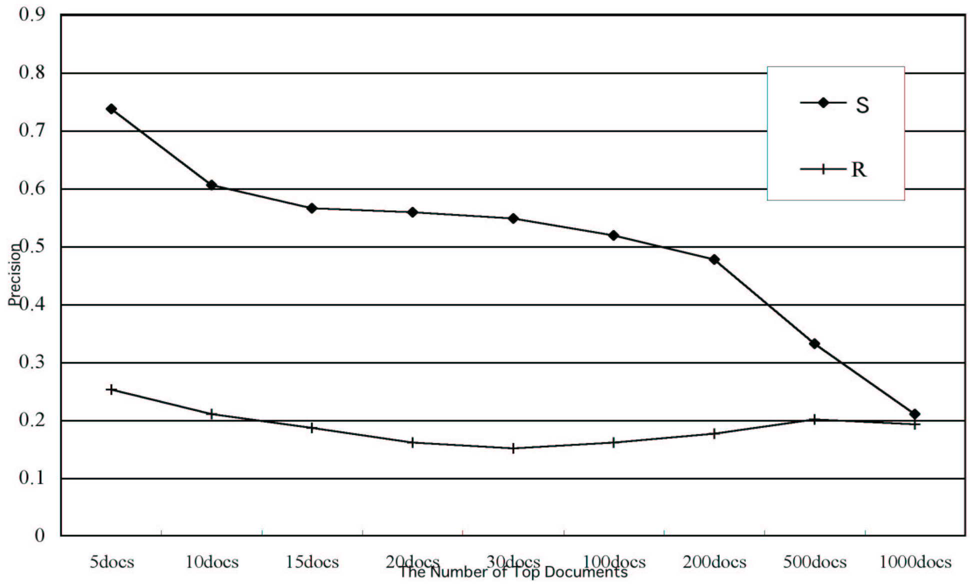

Figure 5.3. The precision using R and S as a query

precision is 51%, the maximum precision is 74% at the top 5 articles. The S of pattern 2 is better than the R of pattern 1. The precision rates of R and S for all at the top documents were compared using the t-test. I estimated a two-tailed t-test (t-test) for the R and S precision pair. The t-test indicated a statistically significant difference between S and R [11].

5.3.7 Experiment at Pattern 3 and Pattern 4

Figure 5.4 shows the results of the precision in the case of extraction using only recall words (R) of pattern 1, recall words with query expansion (RE) of pattern 3, only select words (S) of pattern 2, and select words with query expansion (SE) of pattern 4.

In anticipation, I thought that the precision of RE and SE became better than the precision of R and S. Because some irregular events are contained in the sightseeing data extracted pages on the Nara prefectural site, then the objective articles explained in Section 5.3.5 are also contained in some irregular events. As the hypothesis 2, if users can extract irregular events using query expansion, the precisions with query expansion are expected to be better than the precisions without query expansion. However, as shown in Figure 5.4, RE of pattern 3 is more accurate than R of pattern 1. The t-test did not show any significant differences. S of pattern 2 is more accurate than SE of pattern 4. The difference is statistically significant. This result overturns the hypothesis 2, i.e., the query expansion is ineffective in improving the accuracy to extract event information.

5.4 Discussion

5.4.1 Hypothetical Verification

The result in Section 5.3.6 shows that the results using S of pattern 2 is better than the results using R of pattern 1. This result means the hypothesis 1 can be said appropriate. On the other hand, the result in Section 5.3.7 shows that

[11] I simple say "statistically significant" if the difference in precision (such as the precision at top n documents) between two results is statistically significant at the 1% level, based on a two-side t-test of the null hypothesis of equal means.

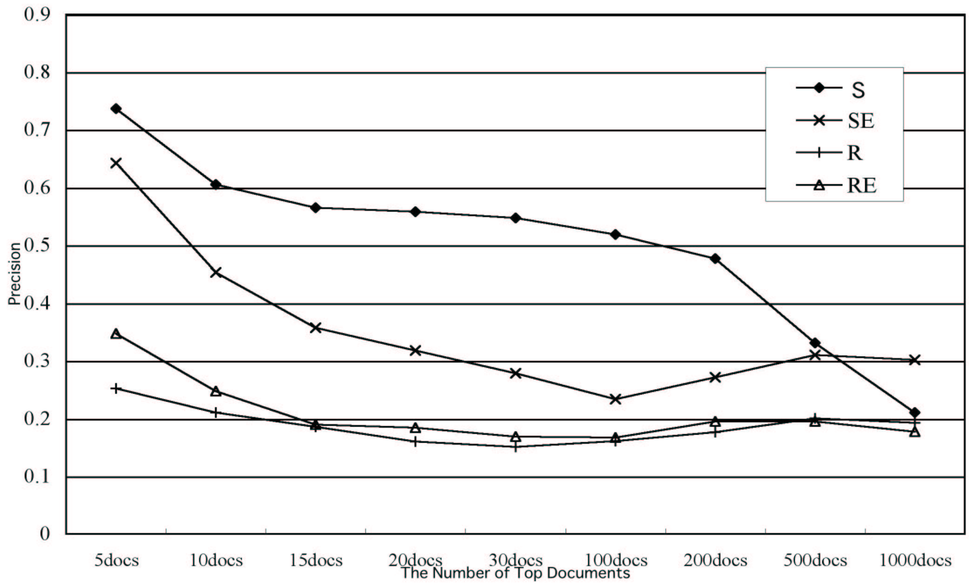

Figure 5.4. The precision of R, RE, S, and SE

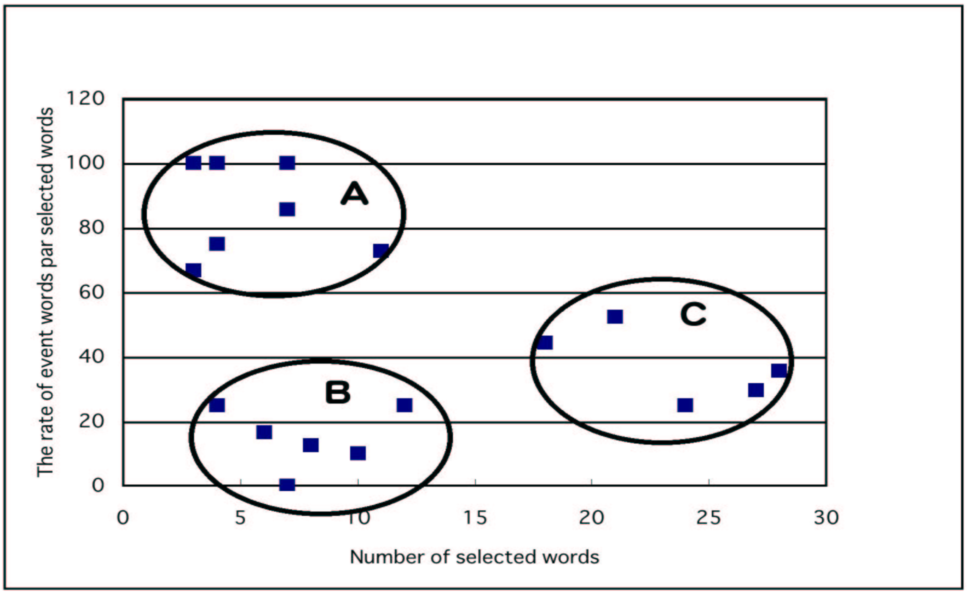

Figure 5.5. A distribution of respondents

the results using query expansion (pattern 3 and pattern 4) are worse than the results without query expansion (pattern 1 and pattern 2). So, the hypothesis 2 can be said inappropriate in this experiment.

At the experiments in Section 5.3.6 and 5.3.7, I analyze all replies together. Therefore, I thought to get the result that the hypothesis 2 is inappropriate. In fact, I thought that the knowledge of people who live near sightseeing spots and the knowledge of people who live far from sightseeing spots are quite different in the preparatory test of Section 5.3.1. Then I analyze tendencies in selecting from a list of periodic words.

5.4.2 Distribution of Respondents for Selected Words

The tendency for whether the event word is chosen correctly from the list of periodic words, can be divided into three groups. Figure 5.5 shows the analysis. The 7 respondents in group A, selected various words but mostly correct event words. All A respondents were from the Kansai area, and knew a lot about Nara. The respondents in group B could only select a few words. The respondents in group B were from the Kanto area, and did not know much about Nara. The respondents in group C, were from the Kanto area, but some of them had lived in Kansai, and some of them had visited Nara many times. Therefore, they knew more about place names in Nara than group B. The list of periodic words included place names. The respondents in group C selected place words as event words by mistake.

The purpose of my system was to support people who did not know about the Nara area, and I targeted groups B and C. I then analyzed the respondents in group A who knew Nara, and the respondents in group B+C who did not know about Nara.

5.4.3 Results of Pattern 1 and Pattern 2 Divided by Group

Figure 5.6 plots precision using R of group A (A-R) and group B+C (BC-R). Members of group A could recall correct event words, so the ratio of the precision was very high compared to that of group B+C. The difference is statistically

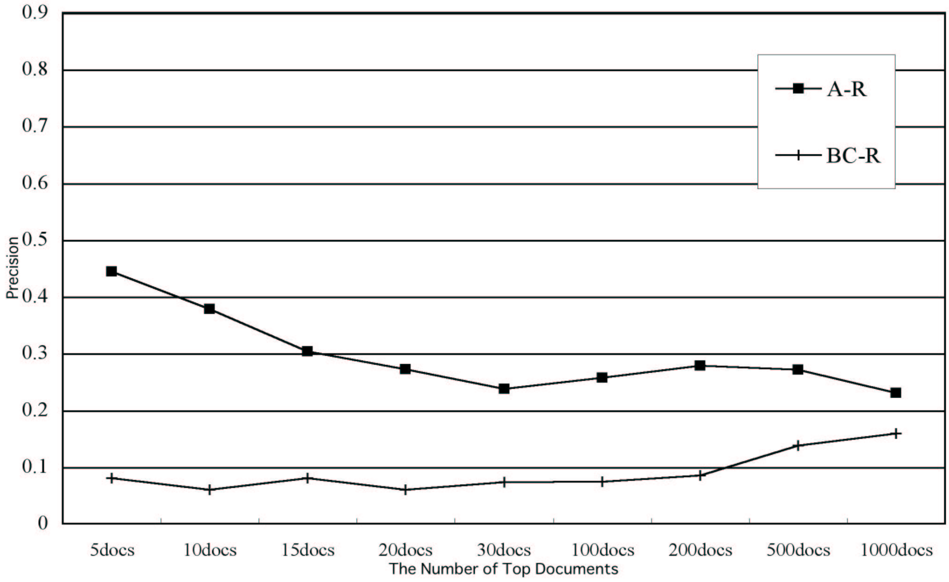

Figure 5.6. Precisions using R of group A and B+C

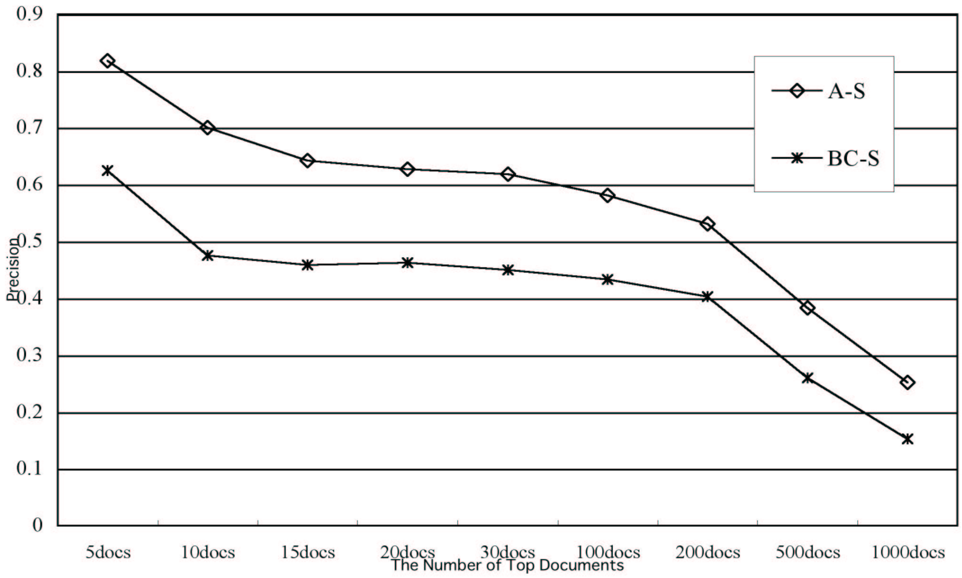

Figure 5.7. Precisions using S of group A and B+C

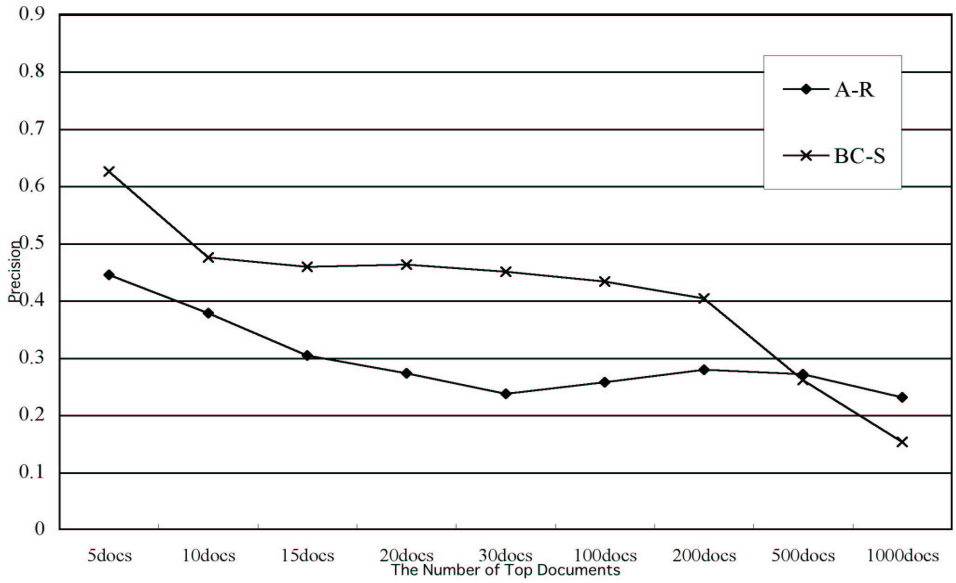

Figure 5.8. Precision of A-R and BC-S

significant.

Precision in using S of pattern 2 is shown in Figure 5.7. The precision of group A (A-S) is better than that of B+C (BC-S). However, the difference between the ratio in group A and group B+C is small, compared to the difference in precision using S between group A and group B+C. In this case of pattern 2, the difference is also statistically significant.

In the case of each pattern, the precision of group A is better than the precision of group B+C. The results of A-R and A-S and the results of BC-R and BC-S in Figure 5.6 and Figure 5.7, show the results of the pattern 2 are better than the results of the pattern 1.

This result indicates that people who do not know about sightseeing spots in detail can extract event information easily and correctly if they use the list of periodic words. Therefore, in extracting event information, the periodic words are useful for visitors who are unaware of sightseeing spots.

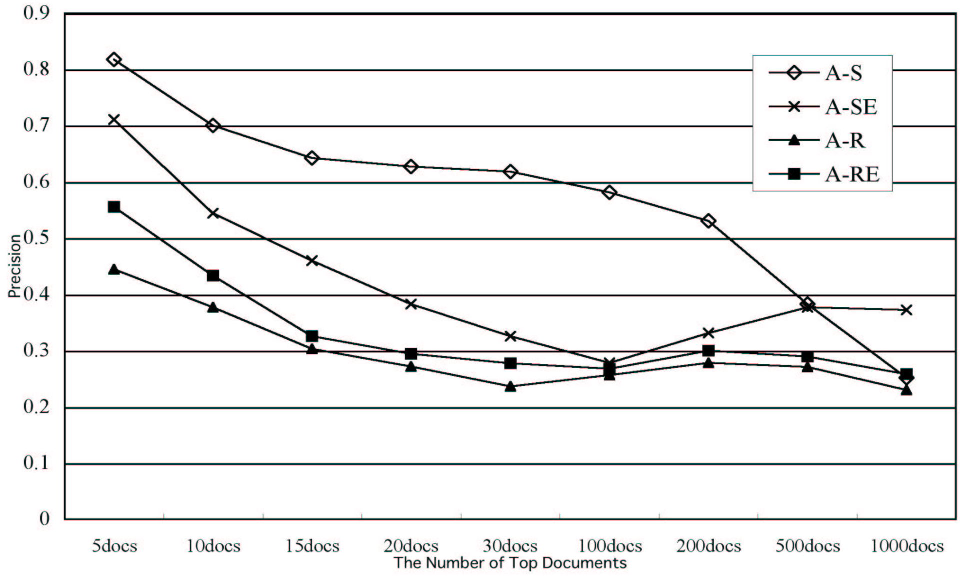

Figure 5.9. Precision of group A

5.4.4 Comparison of Different Groups in Patterns 1 and 2

BC-A is a more notable improvement than BC-R. The precision using S in group B+C (BC-S) and the precision using R in group A (A-R) are compared in Figure 5.8. The precision of BC-S is better than the precision of A-R. The difference is statistically significant. This result also indicates that people who do not know about Nara in detail can extract event information easily and correctly if they use the list of periodic words. This result means the hypothesis 1 can also be said appropriate as same as the result in Section 5.3.6.

5.4.5 Results using Query Expansion by Group

Figure 5.9 and Figure 5.10 show the results of pattern 1 (R), pattern 2 (S), pattern 3 (RE), and pattern 4 (SE) for each group A and B+C.

In group A, the result of A-RE is better than the result of A-R. The difference is statistically significant. However, to compare with the results of A-S and A-SE

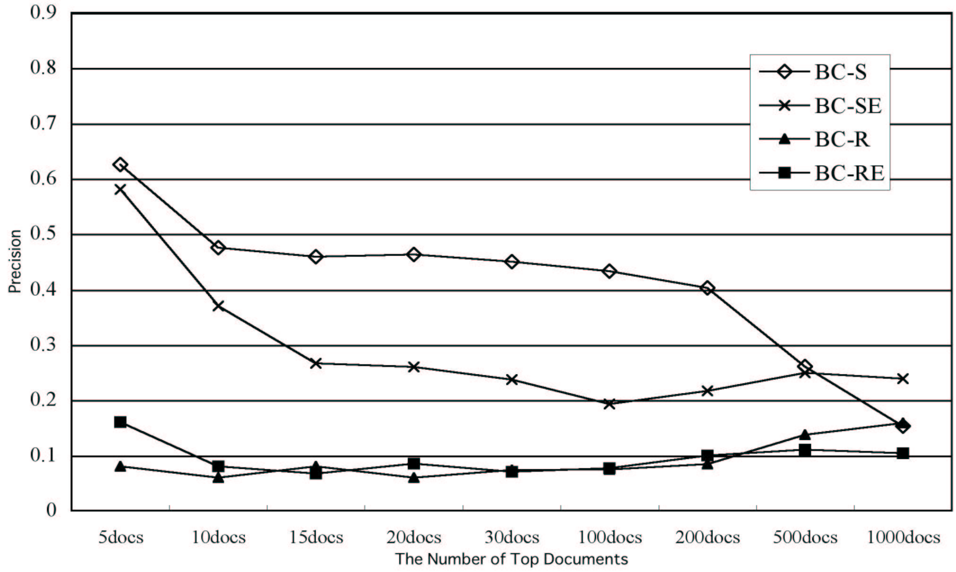

Figure 5.10. Precision in group B+C

in group A, and the results of BC-S and BC-SE in group B+C, the results using query expansion are worse than the results without query expansion. In these cases, the differences are statistically significant at 5% level of significance in each group. Furthermore, the results of BC-R and BC-RE are not difference, the result of *t*-test is not any significant differences. This result means the hypothesis 2 can also be said inappropriate as same as the result in Section 5.3.7.

These results show that query expansion does not have an effect except for keywords that are almost correct answers. When using query expansion, it is necessary to stringently restrict conditions.

5.4.6 Extraction of New Event Information

As shown in Figure 5.9, although the precisions at top 5 documents of A-SE and A-S do not have a difference so much, the precision at top 100 documents of A-SE is reduced by half as same as the precision of A-R.

I think query expansion is ineffective using wrong event words. In group A, wrong event words are not contained, then this result of A-SE has a reason in

```
<DOC NAME=''20010407-M2L-26258''>
<DATE>20010407</DATE>
<TITLE>談山神社であす「神幸祭」薄墨桜の下で春のひとときを</TITLE>
<TEXT>
藤原鎌足を祭る桜井市多武峰の談山神社は８日午後１０時から
「神幸（じんこう）祭」を営む.
（途中略）
「鎌足さん」と参拝者が一緒になって薄墨桜の下で
春のひとときを楽しむ.
</TEXT>
</DOC>
```

Figure 5.11. The event article in the articles considered to be errors

others. I think the objective articles explained in Section 5.3.5, are not perfect. I confirm extracted articles at top 100 documents in the case of A-SE whether event articles or not. In particular, I collected 1900 articles to top 100 among the articles that were extracted at the experiment of pattern 4 in Section 5.4.5. Next, I deleted objective articles and duplicate articles from 1900 articles. 324 articles are evaluated non-event articles. I had four subjects which were manually checked whether the 324 articles was event articles or not. In result, there are 25 articles that all subjects judged as an event article. There are 114 articles that some one in four subjects judged as an event article. This means that the articles of 35% in articles that the system judged as an error article are event articles that did not posted on the Nara prefectural site. Actually, after checking into 25 articles that all subjects judged as an event article, all articles are not posted on the Nara prefectural event information site. The 18 articles are posted on the page from event information of the Nara site. The 7 articles are not posted on anyplace of the Nara site, and irregular event articles shown in Figure 5.11. On the strength of this, I found that query expansion have possibilities to extract irregular event information.

5.5 Conclusions of Chapter 5

In order to develop the easy-to-use retrieval system, the following points are important;

1. To narrow down the purpose of the system is necessary
2. The input from a user is lessened, or otherwise is supported
3. The system tailor retrieval results to the purpose
4. Appropriate data for the purpose is always updated

I decided the purpose as "a task" to produce the sightseeing tourists routes. I have developed the method supporting the user by showing the periodic word list for their easy of input. The method is shown effective and has possibilities to update the database by extracting new data.

My methods applied effectively the techniques of extraction keywords, information retrieval and query expansion. Thus, I could show that the support method can contribute not only to data collection but to update and tailor retrieval results as the main purpose of the system.

Chapter 6

Conclusions

This final Chapter discusses some important aspects of information extraction and retrieval techniques in task-oriented recommendation systems, and suggests applications and directions for future research in this field.

6.1 Major Contributions

This thesis focuses on issues related to handling information by means of extraction, categorization, and retrieval techniques, and on developing an easy-to-use system for producing tourist routes as an example of a task-oriented recommendation system, as shown in Figure 6.1. This system mainly consists of three modules including: 1) a Selection Module to support the user in generating queries for retrieving required information; 2) an Analysis Module for analyzing information in detail; and 3) a Retrieval Module with highly precision for retrieving information related to a particular task.

The Selection Module gets typical information as input, and outputs a list of query candidates. The Analysis Module gets texts related to the task, and outputs task-oriented keywords. The Retrieval Module inputs queries including keywords selected from the query candidates by the user, and outputs information related to the task from database, the Web, e-mails, network news articles and so on. For example, in the task of making tourist routes, the Selection Module gets event information with schedules, and outputs event-related words as query candidates. The Analysis Module gets information related to sightseeing places, and outputs detailed event information including event names, schedules, and venues, such as “お水取り (OMIZUTORI)” as the event name, 12th March 2004 as the schedule, and “東大寺二月堂 (Todaiji Nigatsudou)” as the venue. The Retrieval Module gets queries selected from the candidates or input as a user request, and finally outputs information related to the sightseeing places, events, and so on.

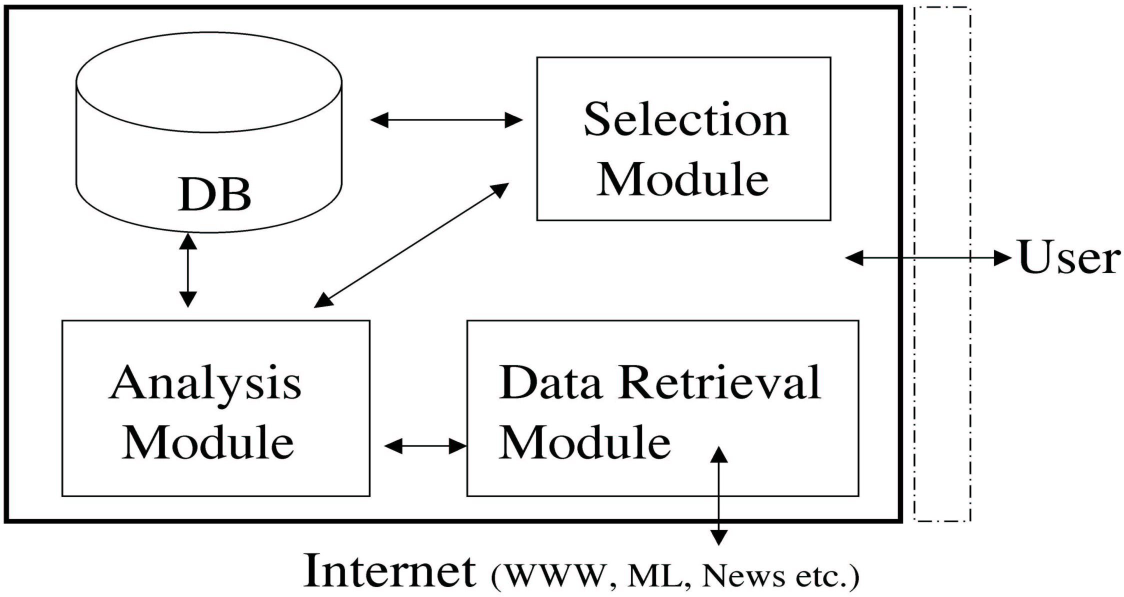

Figure 6.1. A task-oriented recommendation system

The proposed information retrieval system should take account of the user’s purposes and arrange retrieval results in response to his/her needs. To absorb the user’s needs, it is important to analyze the results of his/her input correctly. As a first step towards achieving this, I have proposed in Chapter 3 two support systems using extraction and categorization techniques, and have described applications confirming their utility.

The techniques presented in Chapter 3 contribute to analyzing and processing dynamic and ill-formed data, and were incorporated into the analysis module, as shown in Figure 6.1. In Chapter 4, I extended the retrieval system to the WWW. A retrieval method called the “SCORE method,” was developed for effective extraction and retrieval of a user’s required information from a huge amount of data. This retrieval method was shown to be capable of retrieving information efficiently and correctly from a huge amount of ill-formed data, and of integrating multiple retrieval results into one result effectively. The method was incorporated into the retrieval module of the system as shown in Figure 6.1.

When a user retrieves information for a specific purpose, the retrieval process is composed of a sequence of procedures in response to the user's purpose and the extent of their knowledge. I have called these procedures, collectively, a "task." Developing the task-oriented information retrieval system allowed the system to predict the user's needs, without interactions, and also allowed users to prepare a database that fit the task. The task, in this study, was exemplified by the production of a sightseeing tourist route.

A conventional information retrieval system generally requires the user to input many keywords, although he/she normally prefers to minimize such interactions. The interactions themselves generally do not take into consideration the user's intentions. If he/she cannot input appropriate keywords into the system at the outset, he/she cannot get the information he/she requires. Furthermore, it may be difficult, especially for inexperienced users, to recall appropriate keywords, or keywords that are related to their needs. I have therefore proposed the Selection Module of the task-oriented recommendation system as a method of support for users inputting queries. A method to extract potential keywords related to sightseeing information was proposed, and its effectiveness in retrieving sightseeing information was demonstrated in Chapter 5. The novel feature of this approach is that information extraction techniques, involving keyword features of events related to sightseeing tours such as festivals, are usually held in a cycle. Thus, these extraction techniques, which require minimal input by the user, are applied to both retrieving information and organizing the results in a way that suits the users' needs.

In Chapter 1, I defined the five conditions of an easy-to-use information retrieval system as follows:

- to retrieve target information with accuracy and efficiency;
- to retrieve information with few queries and interactions;
- to support task-oriented initial information development;

- to retrieve and organize up-to-date task-oriented information; and
- to recommend information which suits the users' needs.

In this thesis, I have proposed a task-oriented information recommendation system for a sample of an easy-to-use system. To focus on the task of sightseeing tourist route production, I have performed specific experiments on my system. I have proposed methods which satisfy the first four conditions, and I have shown, by experimentation, the capability of a task-oriented information retrieval system. However, I have not yet implemented the complete recommendation system. An Interface Module that meets the fifth condition, defined above remains to be implemented. In the next section, I present my ideas about the Interface Module, and discuss future prospects.

6.2 Further Directions

I established techniques to correctly retrieve information related to the task in Chapter 5. The Interface Module further needs to extract or filter information related to the task in detail, and to organize and produce sightseeing routes by matching this detailed information with the information regarding users' needs.

First, techniques for detailed information extraction have been developed using "named entity recognition" technology. If a system has a database of correct event information, it can retrieve the time and location information with an accuracy of about 90% using the named entity recognition technique[56], for which tools (such as Rex[42], NExT[34], and Bar[5]) are publicly accessible. By applying these tools to my system, it should be possible to develop a method to extract event-names information correctly from event information data, as well as time/date/location information, for implimenting the fifth condition. Next if the system can retrieve detailed information, it can organize the database easily, for example according to time information. The system would simply need to compare the time information in a particular event's data with the current time, to update the database and ensure that it has only current event information, complete with time and place information. If the system incorporates the relation of places to tourist activity models[31] as tourist knowledge, it can produce a sightseeing route according to the users' wishes.

To improve the proposed system and to evaluate its usability requires further investigation, including an automatic database-updating algorithm and methods for producing sightseeing routes automatically.

The methods relating to the operation of task-oriented information can be applied to agent systems. Agent systems should be able to respond to any possible situation, but they require knowledge to respond to multiple situations. A certain situation can equate with a certain task; if databases for several tasks are combined, they can respond to multiple situations. Therefore, the method proposed in this study has a high potential for accumulating knowledge, which will enable users to confirm the adequacy of the agent's knowledge. Furthermore, the development of ubiquitous computers and robots has recently been proceeding rapidly, and related technologies are available for diverse applications. In this area, the technologies for understanding a user's situations and for interactions with users are very important. Therefore, information handling techniques are absolutely essential in the future.

List of Publications

Journal Papers

[1] **Hiromi itoh Ozaku**, Masao Utiyama, Hitoshi Isahara, Yasuyuki Kono and Masatsugu Kidode, "An Event Information Retrieval Method Using Features of Keyword Appearance in Newspaper Corpora," Transactions of the Japaneses Society for Artificial Intelligence, vol.19, no.4, pp.225–233, April 2004 (in Japanese).

[2] **Hiromi itoh Ozaku**, Masao Utiyama, Hitoshi Isahara, Yasuyuki Kono and Masatsugu Kidode, "A comparative Study on Merging Results from WWW retrieval System," IPSJ Transactions on Databases, vol.44, no.SIG8, pp.78–91, June 2003 (in Japanese).

[3] **Hiromi itoh Ozaku**, Masao Utiyama, Masaki Murata, Kiyotaka Uchimoto and Hitoshi Isahara, "Supporting Conference Program Production Using Natural Language Processing Technologies," Journal of Natural Language Processing, vol.9, no.5, pp.131–148, October 2002 (in Japanese).

[4] Kiyotaka Uchimoto, Qing Ma, Masaki Murata, **Hiromi itoh Ozaku**, Masao Utiyama and Hitoshi Isahara, "Named Entity Extraction Based on A Maximum Entropy Model and Transformation Rules," Journal of Natural Language Processing, vol.7, no.1, pp.63–90, April 2000 (in Japanese).

[5] Masaki Murata, Qing Ma, Kiyotaka Uchimoto, **Hiromi itoh Ozaku**, Masao Utiyama and Hitoshi Isahara, "Information Retrieval Using Location and Category Information," Journal of Natural Language Processing, vol.7, no.1, pp.141–160, April 2000 (in Japanese).

[6] Kiyotaka Uchimoto, Satoshi Sekine, Masaki Murata, **Hiromi itoh Ozaku** and Hitoshi Isahara, "Term recognition using corpora from different fields," International Journal of theoretical and applied issues in specialized communication, vol.6, no.2, pp.233–256, 2000.

International Conferences

[4] **Hiromi itoh Ozaku**, Masao Utiyama, Hitoshi Isahara, Yasuyuki Kono and Masatsugu Kidode, "Study on Merging Multipule Results from Information retrieval system," Working Notes of the Third NTCIR workshop Meeting,pp.39–46,October 2002.

[5] Kiyotaka Uchimoto, Masaki Murata, Qing Ma, **Hiromi itoh Ozaku** and Hitoshi Isahara, "Namaed Entity Extraction Based on A Maximum Entropy Model and Transformation Rules," The 38th Anuual Meeting of the Association for Computational Linguistics (ACL2000), pp.326–335, October 2000.

[6] **Hiromi itoh Ozaku**, Kiyotaka Uchimoto, Masaki Murata and Hitoshi Isahara, "Automatic Recommendation of Hot Topics in Discussion-type Newsgroups," The Fifth International Workshop in Information Retreival with Asian Languages (IRAL2000), pp.203–204,September 2000.

[7] **Hiromi itoh Ozaku**, Kiyotaka Uchimoto, Masaki Murata and Hitoshi Isahara, "Topic Search for Intelligent Network News Reader HISHO," The 2000 ACM Symposium on Applied Computing (SAC2000), pp.28–33,March 2000.

[8] Masaki Murata, Kiyotaka Uchimoto, **Hiromi itoh Ozaku**, Qing Ma, Masao Utimoto and Hitoshi Isahara, "Japanese probablistic information retrieval using location and category information," The 5th International Workshop on Information Retrieval with Asian Languages (IRAL2000), pp.203–204, September 2000.

[9] **Hiromi itoh Ozaku**, Kiyotaka Uchimoto and Hitoshi Isahara, “Improvement of Intelligent Network News Reader HISHO,” The 3rd International Workshop in Information Retreival with Asian Languages (IRAL1998), pp.122–129, 1998.

Domestic Workshops

[1] **Hiromi itoh Ozaku**, Eiko Yamamoto, Masao Utiyama, Hitoshi Isahara, Yasuyuki Kono and Masatsugu Kidode, “Extraction and Organization of Event Information for Computer Assisted Production of Tourist Routes,” The 18th Annual Conference of the Japaneses Society for Artificial Intelligence, May 2003 (in Japanese).

[2] **Hiromi itoh Ozaku**, Masao Utiyama, Hitoshi Isahara, Yasuyuki Kono and Masatsugu Kidode, “Extraction of Relation between Event Words and Time Information for Task Oriented WWW retreival system,” The 9th Annual Conference of the Association for Natural Language Processing, pp.663–666, March 2003 (in Japanese).

[3] **Hiromi itoh Ozaku**, Masao Utiyama, Hitoshi Isahara, Eiki Fujimoto, Yasuyuki Kono and Masatsugu Kidode, “A Comparative Study on Merging Results from WWW Retrieval System,” Text Processing for Information Access Symposium, PP.81–88, Feburary, 2003 (in Japanese).

[4] **Hiromi itoh Ozaku**, Yasuyuki Kono, Hitoshi Isahara and Masatsugu Kidode, “A new data controle method to embed ’time’ function in the assist system makeing a tour route,” The 45th Annual Meeting of Information Processing Society of Japan, pp.647–652, March 2002 (in Japanese).

[5] **Hiromi itoh Ozaku**, Yasuyuki Kono and Masatsugu Kidode, “A Study of the Usability for the Assist System to Make Tourist Routes,” The Symposium of Human Iterface, pp.429–432, October 2001 (in Japanese).

Acknowledgments

I had productive days and could satisfying experiences at NAIST. Thank you very much for all members at Artificial Intelligence Laboratory.

The one person to whom I am most indebted is my supervisor, Professor Masatsugu Kidode. He has introduced me the fundamentals of Artificial Intelligence technology and guided me the way to conduct research and joys of research. I am deeply grateful to Associate Professor Yasuyuki Kono who gave me thoughtful, valuable advice and continually encouraged me. Without his infinite patience and generous support, I would not have been able to complete the doctoral thesis.

I would like to express my gratitude to the members of my thesis committee: Professor Shunsuke Uemura and Professor Yuji Matsumoto for their helpful comments and suggestions.

Next, I am thankful to all members at Communications Research Laboratory. Especially, I wish to thank Dr. Hitoshi Isahara who gave me an opportunity to study at NAIST, and Dr. Masao Utiyama who gave me fruitful advice and precise comments for my research.

I am deeply grateful to Professor Ian Smith for his helpful comments and friendly supports. I am thankful to Professor Worman Dee and Ms. Miwako Furuta for their helpful advice. I wish to thank Ms. Yukari Tanimura and Ms. Hiromi Mochizuki for their kind supports. I also thank many people who contributed to this thesis by giving helpful advice.

Finally, I would like to thank my husband, Gen Itoh for his consistent care, continual encouragement and generous support. I thank my daughters Haruka, Sayaka for their consistent encouragement and patience.

References

[1] J. Allan, J. Callan, F. F. Feng, and D. Malin. INQUERY and TREC-8. In *TREC8; Proceedings of Text Retrieval Conference*, pp. 551–600, 2000.

[2] Y. Ariyoshi and T. Fukushima. Development of Web Search Engines Specialized to Purpose and Individual. *Journal of the Japanese Society for Artificial Intelligence*, Vol. 16, No. 4, pp. 520–524, 2001. (in Japanese)

[3] H. Asano, T. Kato, and A. Takagi. Extraction of Sender Information from E-mails Based on Local Pattern Matching of Signatures and Its Application to Address Book Management. *Journal of Information Processing Society Japan*, Vol. 39, No. 7, pp. 2196–2206, 1998. (in Japanese)

[4] H. Asano and M. Nagata. NetNews Area Analysis Using Surface Informaton. IPSJ SIGNotes Natural Language NL-99-132-004, Information Processing Society of Japan, 1999. (in Japanese)

[5] Bar: model collection for chasen and yamcha, http://chasen.naist.jp/˜masayu-a/p/bar. (checked in Mar. 2004)

[6] L. D. Baker and A. K. McCallum. Distributional Clustering of Words for Text Classification. In *SIGIR'98: Proceedings of the 21st Annual International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 96–103, 1998.

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

[8] C. Buckley and J. Walz. The TREC-8 Query Track. In *TREC8; Proceedings of Text Retrieval Conference*, pp. 65–76, 1999.

[9] J. Callan. *Chapter 5: Distributed information retrieval*. Kluwer Academic Publishers, pp. 127–150, 2000.

[10] J. P. Callan, Z. Lu, and W. B. Croft. Searching Distributed Collections with Inferencd Networks. In *SIGIR'95; Proceedings of the 18st Annual International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 21–28, 1995.

[11] N. Craswell, D. Hawking, and S. Robertson. Effective site finding using link anchor information. In *SIGIR01; Proceedings of the 24th Annual International Conference on Research and Development in Information Retrieval*, pp. 250–257, 2001.

[12] N. Craswell, D. Hawking, and P. Thistlewaite. Merging results from isolated search engines. *The 10th Australasian Databese Conference*, pp. 189–200, 1999.

[13] J. Delgado and N. Ishii. Content + Collaboration = Recommendation. *The workshop on Recommender Systems AAAI 1998*, pp. 37–41, 1998.

[14] S. T. Dumais and J. Nielsen. Automating the Assignment of Submitted Manuscripts to Reviewers. In *SIGIR'92: Proceedings of the 15st Annual International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 233–244, 1992.

[15] Ted Dunning. Accurate Methods for the Statistics of Surprise and Coincidence. *Association for Computational Linguistics*, Vol. 19, No. 1, pp. 61–74, 1993.

[16] T. Ehara. *Users digital signal processing.* Tokyo Denki University Press, 1991. (in Japanese)

[17] K. Fujimoto, H. Kazawa, H. Sato, A. Abe, and K. Matsuzawa. DSIU Systems: Decision Support for Internet Users. *Journal of the Japanese Society for Artificial Intelligence*, Vol. 15, No. 1, pp. 61–64, 2000. (in Japanese)

[18] D. Harman. Overview of the fourth text retrieval conference (TREC-4). In *TREC4: Proceedings of Text REtrieval Conference*, pp. 1–24, 1995.

[19] D. Hawing, E. Voorhees, N. Craswell, and P. Bailey. Overview of the TREC-8 web track. In *TREC8; Proceedings of Text REtrieval Conference*, pp. 131–150, 1999.

[20] Y. Hayashi and Y. Obashi. Recent and Future Technical Trends in Search Sevices on the World Wide Web. *Magazine of Information Processing Society*, Vol. 39, No. 9, pp. 861–865, 1998. (in Japanese)

[21] M. Horton and R. Adams. Standard for interchange of usenet messages, 1987. Network Working Group Request for Comments:1036 (RFC 1036).

[22] H.Ozaku and H. Isahara. Intelligent Network News Reader. In *IROL'96: Proceedings of the workshop on Information Retrieval with Oriental Languages*, pp. 126–131, 1996.

[23] H. Inui, K. Uchimoto, M. Murata, and H. Isahara. Classification of Open-Ended Questionnaires based on Predicative. IPSJ SIGNotes Natural Language NL-128-25, Information Processing Society of Japan, 1998. (in Japanese)

[24] S. Kaneda, T. Itoh, and S. Harashima. Development of IEICE Submitted Paper Management System. *Technical Report, The Journal of Institute of Electronics Information and Communication Engineers*, 2000. (in Japanese)

[25] K. Kishida, M. Iwayama, and K. Eguchi. Methodology and Pragmatics of Retrieval Experiments at NTCIR Workshop. *Pre-meeting Lecture at the NTCIR-3*, pp. 1–25, 2002. (in Japanese)

[26] Y. Koseki and T. Fukushima. New Generation technologies for Internet Search Portals. *The 2001st symposium of the information society*, 2001. (in Japanese)

[27] S. Kurohashi and M. Nagao. Japanese Morphological Analysis System JUMAN Version 3.61 Users Manual, 1999.

[28] Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, H. Matsuda, K. Takaoka, and M. Asahara. Japanese Morphological Analysis System ChaSen version 2.2.1. 2000.

[29] Y. Miyamoto, S. Kato, Y. Goto, and H. Hayashi. Development of an automatic keyword-extracting system. *ISCIE, the 37th Annual Conference of the Institute of Systems, Control and Information Engineers*, pp. 107–108, 1993.

[30] S. Mizobuchi, T. Sumitomo, M. Fuketa, and J. Aoe. A method for understanding japanese time expressions. *Journal of Information Processing Society*, Vol. 40, No. 9, pp. 3408–3419, 1999. (in Japanese)

[31] S. Morichi, T. Hyodo, and N. Okamoto. A study on touring activity models for one-day car trip. *The Journal of Infrastructure Planning*, pp. 63–70, 1992. (in Japanese)

[32] A. Morimura and Y. Kiyoki. An integration method between www search engines. IPSJ SIGNotes DataBase System DB-125-15 125, Information Processing Society of Japan, 2001. (in Japanese)

[33] M. Murata, Q. Ma, and H. Isahara. Applying multiple characteristics and techniques to obtain high levels of performance in information retrieval. In *NTCIR3; Proceedings of the NTCIR Workshos 3 (CLIR)*, pp. 87–92, 2002.

[34] NExT, a Named Entity Extraction Tool, http://www.ai.info.mie-u.ac.jp/˜next/next_en.html. (checked in Mar. 2004)

[35] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. *IJCAI'99: Workshop on Machine Learning for Information Filtering*, pp. 61–67, 1999.

[36] Y. Obata, H. Watanabe, and T. Kawaoka. An intelligent mechanism to understand time in a simple sentence. *Technical Report of IEICE*, pp. 1–6, 2001. (in Japanese)

[37] H. Ozaku, Y. Kono, and M. Kidode. A study of the usability for the assist system to make tourist routes. *The symposium of Human Interface*, pp. 429–432, 2001. (in Japanese)

[38] H. Ozaku, K. Uchimoto, and H. Isahara. Characteristics of common terms in discussion type internet news and its application to the news reader -hisho-. *The 4th Annual Meeting of the Association for Natural Language Processing*, 1998. (in Japanese)

[39] H. Ozaku, K. Uchimoto, and H. Isahara. Improvement of Intelligent Network News Reader HISHO. In *IRAL'98: Proceedings of the 3rd International Workshop on Information Retreval with Asian Languages*, pp. 122–129, 1998.

[40] H. Ozaku, M. Utiyama, H. Isahara, Y. Kono, and M. Kidode. Study on merging multiple results from information retrieval system. *NTCIR3; Working Notes of Web*, 2002.

[41] H. Ozaku, E. Yamamoto, M. Utiyama, H. Isahara, Y. Kono, and M. Kidode. Extraction and Organaization of Event Information for Computer Assisted Production of Tourist Routes. *The 17th Annual Conference of the Japanese Society for Artificial Intelligence*, 2003. (in Japanese)

[42] Rosette(R) Entitiy Extractor (REX), http://www.basistech.com/products/rex.html. (checked in Jan. 2004)

[43] E. Renninson. Galaxies of news: An approach to visualizing and understanding expansive news landscapes. In *UIST94*, 1994.

[44] P. Resnick and H. R. Varian. Recommender Systems. *Communications of the ACM*, Vol. 40, No. 3, pp. 66–72, 1997.

[45] S. E. Robertson and S. Walker. Okapi/keenbow at trec-8. In *TREC8; Proceedings of Text Retrieval Conference*, pp. 151–162, 2000.

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

[47] G. Salton, A. Wong, and C.S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, pp. 613–620, 1975.

[48] M. Sasaki and H. Shinnou. Query expansion using latent contextual document relevance. IPSJ SIGNote Fundamental Infology FI-68-10 10, Information Processing Society of Japan, 2002. (in Japanese)

[49] M. Sato, S. Sato, and Y. Shinoda. Automatic digesting of the netnews. *Journal of Information Processing Society Japan*, Vol. 36, No. 10, pp. 2371–2379, 1995. (in Japanese)

[50] N. sato, M. Uehara, Y. Sakai, and H. Mori. A distributed search engine for fresh information retrieval. *Journal of Information Processing Society Japan*, Vol. 43, No. 2, pp. 321–331, 2002. (in Japanese)

[51] N. Sato, M. Uehara, Y. Sakai, and H. Mori. A distributed search engine for fresh information retrieval. *Information Processing Society of Japane*, Vol. 43, No. 2, pp. 321–331, 2002. (in Japanese)

[52] A. Shinhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *SIGIR96: Proceedings of the 19th Annual International Conference on Research and Development in Information Retrieval*, pp. 21–29, 1996.

[53] L. Si and J. Callan. Using sampled data and regression to merge search engine results. In *SIGIR02; Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval*, pp. 19–26, 2002.

[54] A. Sugiura and O. Etziomi. Query routing for web search engines. In *Architecture and experiments, Proceedings of 9th World Wide Web Conference*, 2000.

[55] H. Tanaka. An efficient document clustering algorithm based on the topic binder hypothesis. In *NLPRS'97: Proceedings of the 4th Natural Language Processing Pacific Rim Symposium*, pp. 387–392, 1997.

[56] K. Uchimoto, Q. Ma, M. Murata, H. Ozaku, M. Utiyama, and H. Isahara. Named entity extraction based on a maximum entropy model and transformation rules. *Journal of Natural Language Processing*, Vol. 7, pp. 63–90, 2000.

[57] K. Uchimoto, H. Ozaku, and H. Isahara. A method foridentifying topic-changing articles in discussion-type newsgroups within the intelligent network news reader hisho. *NLPRS 1997: The Natural Language Processing Pacific Rim Symposium*, pp. 378–380, 1997.

[58] H. Ueda, Y. Yanagisawa, M. Tsukamoto, and S. Nishio. Applying knowledge discovery techniques to trend analysis of electronic mail. *Journal of Infor-*

mation Processing Society Japan, Vol. 41, No. 12, pp. 3285–3294, 2000. (in Japanese)

[59] M. Utiyama and H. Isahara. Implementation of an ir package. IPSJ SIGNotes Fundamental Infology FI-63-8, Information Processing Society of Japan, 2001. (in Japanese)

[60] E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird. The Collection Fusion Problem. In *TREC3: Proceedings of Text REtrieval Conference*, pp. 95–104, 1994.

[61] J. Xu and B. Croft. Query Expansion Using Local and Global Document Analysis. In *Proceedings of the 19th International Conference on Research and Development in Information Retrieval*, pp. 4–11, 1996.

[62] J. Yabe, S. Takahashi, and E. Shibayama. Visualizing emantic content and relationships of a thread of news articles. In *Proceedings of the 14th Annual Meeting of Japan Society for Software Science and Technology*, pp. 129–132, 1997. (in Japanese)

[63] News articles in these days, fj.news.lists.

[64] http://mitsuko.jaist.ac.jp/fj/. (checked in Apr. 2001)

[65] Geta ver.3, 2003. http://geta.ex.nii.ac.jp/e/index.html. (checked in Dec. 2003)