Axis-specified Search: A Fine-grained Full-text Search Method for Gathering and Structuring Excerpts

Yasusi Kanada

Central Research Laboratory, Hitachi Ltd.
Higashi-Koigakubo 1-280, Kokubunji, Tokyo 185, Japan
E-mail: kanada@crl.hitachi.co.jp

ABSTRACT

A text search method, which is called an axis-specified search method, is proposed. This method is suitable for full-text searches of a large-scale text collection. In this method, in addition to specifying search strings, the user selects an axis from a predefined set. The system outputs excerpts and hyperlinks that are ordered along the axis. The search strings express the specific subject of the search, and the axis specifies a general-purpose method of ordering results. Short subtopics, which cannot be easily caught by statistical methods, are effectively gathered from the text collection. The user can get satisfactory results using a simple search string. Even if the number of results is very large, the user can easily survey them, because they are well structured. This method has been applied to an electronic encyclopedia and a newspaper database. In these applications, distributed descriptions that were related to each other could be gathered, and the user could discover their relationships from the results. For example, by specifying “semiconductor” for a search string and “year” for an axis, a table listing seven decades of semiconductor-related topics sorted by year was generated from newspaper issues published over a single year. By specifying “basin” for a search string and “area” (m^2) for an axis, descriptions of the world’s largest rivers were extracted from the encyclopedia and sorted according to their basin areas.

KEYWORDS: Information retrieval, full-text search, information extraction, information gathering, document classification, electronic encyclopedia, newspaper database.

1. INTRODUCTION

Large-scale full-text searching is becoming much more important because of the popularity of the Internet, intranets, CD-ROM and DVD-ROM. Through a full-text search method using an *N*-gram character index (e.g., an inverted index) or Patricia tree [Mor 68] [Fra 92], all the places in a document in which the search string occurs can be listed. However, a full-text search is usually used for finding *documents* whose subject is related to the search string. Most currently available full-text search systems only return whole document information but do not return information on each topic in the document. Searching for subtopics in documents, even those not directly related to the main subjects of the document, is often useful though, because a document usually contains subtopics or episodes and these may be much more important to users than the main topics. This type of text retrieval is called *fine-grained full-text searching* (FFS) in this paper.

There are four problems in FFS. The first problem is that it is difficult for an FFS system to find appropriate units of documents. In passage-retrieval systems, a document is divided into syntactic units such as sections or paragraphs. However, a syntactic unit may contain multiple topics, or a topic may be described in multiple syntactic units. It is almost impossible to divide a document into topics using current natural-language processing technology.

The second problem is an explosive increase of search results. If the number of results increases, it takes too much time for the user to survey all of them, unless they are structured properly. There are two causes of this increase. One is that amount of available text is increasing rapidly, causing the number of potential results to also increase. The other is that the number of possible subtopics is much larger than the number of documents.

The third problem is the difficulty of search string selection. In conventional database retrieval, the end-user accesses a database through professional searchers because elaborate selection and addition of strings in the query are essential in this case. However, in retrieval from the Internet or CD-ROM, the end-user directly searches the media because there is usually no expert who can help the user in this case. It is often difficult for users to refine their queries.

The fourth problem is restricting views. If the user successfully reduces the number of results, important issues related to the subject are often discarded. The user's exposure to the potential search results is, therefore, too narrow. When searching paper dictionaries or encyclopedias, the user may discover interesting information from articles other than the one searched for on the same page. This type of unexpected encounter is often very important. Electronic searches often exclude such a possibility. In other words, the conventional search methods function as point searches, but a bird's-eye view search or perspective search is required.

Methods for automatic classification of search results using clustering, which is based on a vector space model or a probabilistic model, have solved some of these problems. Clustering may make surveying a large number of search results easier. Search string selection may also become easier because the results that the user requires may be gathered into a cluster. Views may also be widened because the user may see clusters of less related issues.

However, there are two problems. First, it is probably impossible to handle short subtopics by the clustering-based methods because they are statistical methods [Hea 93]. Paragraphs may be classified by clustering, but smaller units of text are difficult to handle because of statistical errors. Second, the process of clustering does not necessarily reflect the user's intention because clustering is a bottom-up and data-driven method. It is usually difficult for the user to understand the meaning of each class in the results. In addition, the first FFS problem (the choice of appropriate units) is not solved because units of text used for statistical methods are syntactic and units that contain multiple topics are classified into only one class. Thus, the FFS problems mostly remain unsolved.

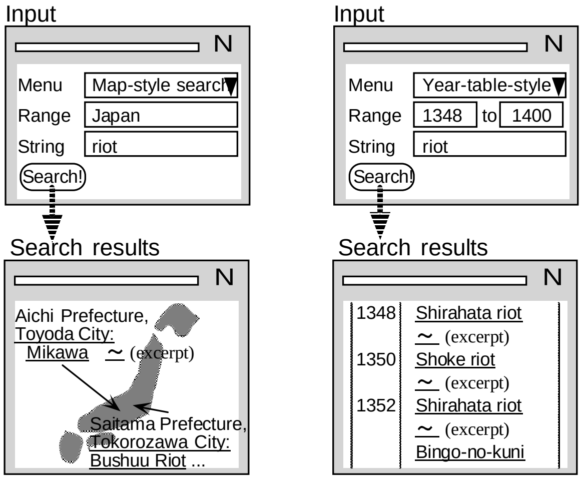
I believe that structuring of search results based on the user's intention is necessary to overcome the above text-search problems. Thus, I have developed an FFS method, which is called the *axis-specified search method*. In this method, the user selects an axis to structure the search results. The function of the axis-specified search is explained in Section 2. The outline and several important issues of implementation are explained in Section 3. Two applications of this method, i.e., Web-based encyclopedia and newspaper database, are discussed in Section 4. The results of case studies on these applications are discussed in Section 5. Related work is mentioned in Section 6, followed the conclusions in Section 7.

2. FUNCTION OF AXIS-SPECIFIED SEARCHES

Axis-specified searching is a fine-grained full-text search method. In an axis-specified search, the user selects an axis, and inputs search strings.[1] The search strings express the specific subject of the search, and the axis specifies a general-purpose method of ordering results. The full-text search results are then ordered along the axis and displayed in this order. They are placed in a space specified by the axis. The criterion for structuring search results is explicitly specified by the user, in contrast to clustering-based structuring methods [Cut 92] [Cut 93] [Mor 95], in which the structure is self-organized. The axes that the user can select are predefined by the search system. The space that is specified by the axis is called a *feature space*, and a value on the axis is called a *feature value*. More exactly, the axis-specified search method is a full-text search method where the user selects a feature space in which the searched texts labeled by the feature values that are extracted from the text are placed. The range of the feature value can also be specified by the user. Therefore, unnecessary results, whose feature values are outside of the range, are omitted.

An example is shown in **Figure 1**(a). The user searches an encyclopedia by specifying "riot" as a search string. The user selects the geometrical axes and specifies Japan as a range in the feature space. Then, excerpts from the encyclopedia articles are sorted according to geographical positions. The excerpts contain Japanese geographical names and contain or are close to the search string. Alternatively, the system can display a map on which the geographical names and the excerpts from the search results are overlaid. In this example, "Aichi Prefecture", "Toyoda City" (which is in Aichi Prefecture), "Saitama Prefecture", and "Tokorozawa City" (which is in Saitama Prefecture) are geographical names. "Mikawa" and "Bushuu Riot" are the article headings in the encyclopedia. Excerpts follow these headings in the search results.

The geographical names are not necessarily bibliographical information, nor are they always included in the main topic of the document, but they may be in-



Menu: Map-style search
Range: Japan
String: riot

Menu: Year-table-style
Range: 1348 to 1400
String: riot

1348	Shirahata riot ~ (excerpt)
1350	Shoke riot ~ (excerpt)
1352	Shirahata riot ~ (excerpt) Bingo-no-kuni

(a) A search with the geographical axes

(b) A search with the year axis (year-table-style search)

Figure 1. Function of the axis-specified search — two examples (translated from Japanese)

[1] Multiple axes can logically be specified in a search. However, only one axis can be specified in our current implementation mainly because of human interface design difficulty.

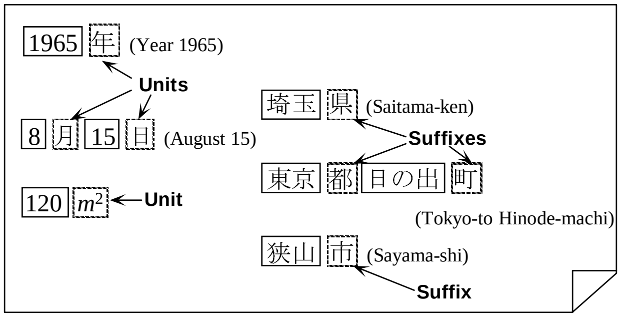
cluded in subtopics. Thus, subtopics, as well as subject topics, are ordered along the geographical axis. The user can see excerpts, or summarized information, in the search results. The user can also follow hyperlinks, which are underlined in the figure, and see the original text. If the hyperlink embedded in the excerpt is clicked, the original text around the extracted text is displayed. (See Figure 6.) If the hyperlink embedded in the article heading is clicked, the beginning of the article or the whole article is displayed.

More concretely, an axis can be specified by one of the two methods: by units of a quantity or by the ending or beginning of words. These methods are explained below in reference to **Figure 2**.

The first method is called a *quantity search*. In this method, an axis is specified in units of a quantity. For example, if the axis is the year, date or time, then, “年” (year), “日” (day), “時” (hour), and other units of time are used as the units (Figure 2(a)). Quantities with these units are extracted from the text collection and sorted. Although time may be expressed in many different units, they can be interchanged. So they can be theoretically put on the same axis. To filter this sorted result, the text is retrieved by a full-text search for the search strings. If the search string does not appear in the document in which the units were found, or it appears but is not close enough to the quantity in the text, the item is removed from the results list. An example using this method is shown in Figure 1(b). This figure shows an example of a year-table-style search, which is a type of quantity search. “Riot” is specified as a search string, and 1348 and 1400 are specified for the range of years. The results are displayed as a year table.

Some other units that are useful for a quantity search are:

- “円” (yen) or “銭” (sen) — Units of Japanese money that are interchangeable.
- “ドル” (dollar) or “セント” (cent) — Units of money that are interchangeable. These units are also interchangeable to Yen, but are not constantly changeable because the exchange rate is floating. These units should, therefore, probably be separated from yen or sen in the quantity search.
- “人” (nin) or “名” (mei) — Units of a number of people (equal units).
- “頭” (tou), “羽” (ba or wa), “尾” (bi), “匹” (hiki, biki or piki), etc. — Units of a number of animals.
- “個” (ko), “冊” (satsu), “種” (shu), etc. — Units of a number of objects.
- m, km, mm, light year, etc. and km/h — Units of length, distance, or velocity. Units of length and distance are interchangeable. Although km/h is not interchangeable to these other units, it is included here because sometimes it is difficult to distinguish this unit from km.
- m^2, mm^2, acres, hectares, etc. — Units of area that are interchangeable.
- m^3, mm^3, *l*, m*l*, etc. — Units of volume that are interchangeable.
- Hz, kHz, MHz, etc. — Units of frequency that are interchangeable.
- ℃ — A unit of temperature.



(a) Extraction of a Japanese date or quantity expression

(b) Extraction of Japanese geographical names

Figure 2. Extraction of units and suffixes from text — examples

The knowledge used to extract a quantity is not very specialized, so the implementation is easy.

The second method is called the *subword search*. In this method, an axis is specified by the ending or beginning of words. For example, in Japan, geographical names usually have suffixes (Figure 2(b)). Names of prefectures have “県”, which is pronounced “ken” and means prefecture, as suffixes, although there are exceptions. “埼玉県” (Saitama-ken) is an example. Tokyo has “to” as a suffix, i.e., “東京都” (Tokyo-to). Cities, towns and villages also usually have suffixes. “狭山市” (Sayama-shi, where “shi” means city) and “日の出町” (Hinode-machi, where “machi” means town) are examples. Therefore, many, but not all, Japanese geographical names can be extracted only using syntactic information. Although the above suffixes do not directly specify axes, the geographical names can be placed in a single-dimensional space of lexicographically sorted names, or in a two-dimensional space of longitude and latitude. The outline of the search procedure is similar to that of the first method. Figure 1 (a) is an example of using the subword-search method.

Some other suffixes that are useful for a subword search are listed below.

- “川” (kawa or gawa, which means river), “湖” (ko, which means lake), “海” (kai, which means sea), and “山” (yama, san or zan, which means mountain) — Suffixes denoting geographical features.
- “主義” (shugi, which means “-ism”) or “教” (kyou, which means religion) — Suffixes for the names of doctrines or principles, or of religions.
- “派” (ha, which means party or school) — A suffix for the name of parties or schools.

- “大統領” (daitoryo, which means president) or “首相” (shusho, which means prime minister) — Suffixes for the name of a president or prime minister.

The words with these suffixes are not put on an axis in the strict sense of “axis.” However, these suffixes show the words belong to a category, i.e., a feature space, or to one of several categories. In addition, the words can be sorted in an order, e.g., a lexicographic order. Thus, if “axis” is interpreted in a wide sense, the suffixes can be regarded to indicate “axes.”

In both quantity and subword searches, the system searches the text for a string that contains a specified substring. In the quantity search, the whole string is a quantity and the substring is a unit. In the subword search, the whole string is a word and the substring is a ending or beginning of the word. These methods have been applied only to Japanese text so far. However, if units, suffixes or prefixes exist for the strings to be extracted, these methods can be equally applied to texts in any language.

The substrings, such as the above units and suffixes, are for general-purpose use. Thus, a specialized subject is specified by a search string, and a method of result ordering is specified by a general-purpose axis.

The FFS problems described in the previous section can be overcome by using the axis-specified search method, if the user can select an appropriate axis. A search result contains an excerpt, which is probably part of a topic. If the topic is not understood from the excerpt, the user can see the whole topic by using the hyperlink. So the problem of text division into units is overcome by eliminating the necessity of division by the system. The user does not have to look at all the results. Because they are ordered, results on a topic are gathered and the user can distinguish results that may be necessary from obviously unnecessary ones by feature values. So the problem of the explosive growth of search results can be overcome. The user can get satisfactory results, which contain many but well-structured excerpts that can be surveyed easily, without specifying a complicated search condition. So the problem of difficulty in search-string selection can be overcome. The user does not have to make the search condition too narrow, but can still survey many but well-structured results. The problem of restricting views can thus also be overcome.

Also, the axis-specified search method does not suffer from the two problems of clustering-based methods. First, this method is not a statistical method. Second, the user’s intention is stated by specifying the axis.

3. IMPLEMENTATION

The outline and several important aspects of implementation are explained here.

3.1 Outline

An axis-specified search system consists of two main parts (**Figure 3**): a set of index generators, and a search engine. Index generators generate axis indices and a full-text index from the text collection. Axis-index generators extract strings that match predefined patterns, normalize the extracted information, and enter it into the index. The method of information extraction is explained in Section 3.2. The purpose of axis-index generation is to drastically reduce the time that a search would take without an axis index. An index is generated for each type of axes because the structure of each type of axes index may be different. The full-text index generator generates an index that has the same structure as one for a conventional full-text search.

The search engine is invoked by the user. The user selects an axis, or selects a feature space, specifies a range of feature values, and specifies search strings. The search engine searches the corresponding axis index for the values within the specified range, filters the search results using the full-text index, and sorts the results. The search engine may use the full-text index first and then use the axis index. However, reversing the order of index use may sometimes improve performance. Because the axis indices are generated before the user’s request, the axes are predefined and the user cannot define a new axis in this method. The search results are scored. If the score of a result is too low, it is dropped from the results list.

3.2 Information extraction and index generation

The axis-index generator inputs all the documents and extracts strings that match predefined string-matching patterns. A set of matching patterns is defined for each axis. The extracted strings are normalized and entered into the index. Most information can be extracted using context-free rules. However, some information, such as abbreviated Christian years (see below), is context-dependent. Matching patterns and actions to be taken when matched, i.e., normalization actions, are dependent on the type of text to be handled. The same strings in a different text might have to be normalized to different strings. The method of information extraction is explained here using two examples.

The first example is a method for a year-table-style search, i.e., an axis-specified search with years as the axis. This search method is used to search the World Encyclopædia [NEC 95] [HDH 98]. In this example, the following forms of years are extracted.

- One to four digits of Christian years followed by “年” (which means “year”), e.g., “1989 年.”
- The last two digits of Christian years followed by “年”, e.g., “89 年”, which means the year 1989.
- One to two digits of Japanese years that are preceded by an era name and followed by “年”, e.g., “平成 10 年” (10th year of the Heisei era).
- “…000 年前” or “… 万年前”, where “年前” means “years ago” and “万年前” means “tens of thousands of years ago.”

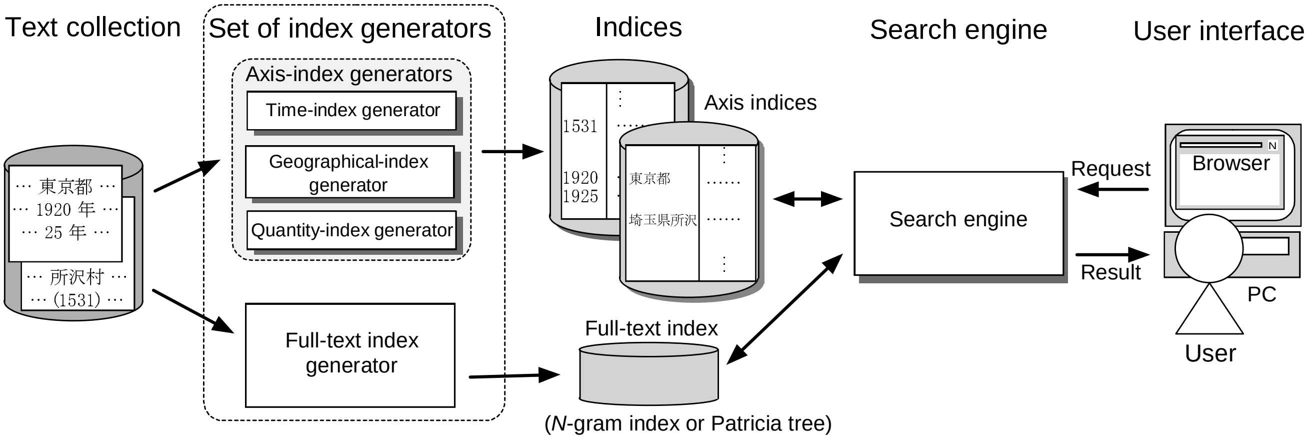


Figure 3. Outline of system structure for axis-specified searches

- A parenthesized year, e.g., “ロシア革命 (1917)” (Russian Revolution (1917))
- “… 世紀” (… Century AD) or “前 … 世紀” (… Century BC).

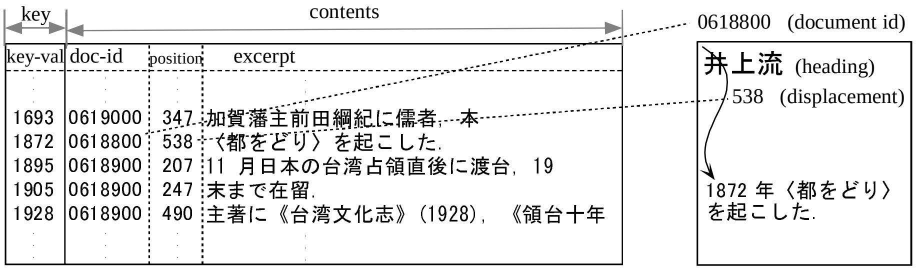
The years are normalized as Christian years. For example, in the second case, the first two digits are supplemented using the context. Both in the World Encyclopædia and in Mainichi newspapers [Mai 95], more than 99% of two-digit year references are normalized easily and correctly. However, the normalization may be more difficult in other contexts.

The second example is an information extraction method for a geographical search: an axis-specified search in which the feature space is the geographical space. This method can be used for a general-purpose search of Japanese text. The following forms of Japanese geographical names are extracted.

- “… 県”, where “県” (ken) means prefecture. Also included are “北海道” (Hokkai-do), “東京都” (Tokyo-to), “大阪府” (Osaka-fu), and “京都府” (Kyoto-fu).
- “… 郡” (gun, which means district), “… 市” (shi, which means city), and “… 区” (ku, which means ward). “区” is extracted only for Tokyo.
- “… 町” (machi or cho, which means town) and “… 村” (mura or son, which means village).

“Ken”, “do”, “to”, and “fu” indicate that the geographical names are at the highest level. “Gun” and “shi” are at the second level. “Ku” is usually at the third level, but it is at the second level in Tokyo. “Machi”, “cho”, “mura” and “son” are usually at the third level. Although there may be four or more levels of geographical names, a rough model that consists of the three levels explained above is currently used. Japanese geographical names are extracted using these three-level patterns in which abbreviation of higher level geographical names is allowed. The geographical names are normalized by supplementing abbreviated parts. For example, if the extracted name is “中野区弥生町”, it is normalized to “東京都 中野区 弥生町” (Tokyo-to Nakano-ku Yayoi-cho).

An example of an axis index is shown in **Figure 4**. The key-value is a normalized feature value, and is used as a key to look up this index. The search key must be normalized because it must be possible to look up the index using keys with interchangeable units or in different form. The location of the original text is specified by the doc-id and the position. The doc-id is an identifier of the document (or the text part), in which the value appears. The position is the position of the text that contains the feature value. The position is represented by the distance from the beginning of the document (the text part). The excerpt is a text fragment extracted from the text at that distance. The excerpt can be stored into the index to improve the search performance. Alternatively, it may be omitted because it can be extracted from the original text when being printed. There may be multiple entries that have the same key-value.[1]



key-val	doc-id	position	excerpt
⋮	⋮	⋮	⋮
1693	0619000	347	加賀藩主前田綱紀に儒者，本
1872	0618800	538	〈都をどり〉を起こした．
1895	0618900	207	11 月日本の台湾占領直後に渡台，19
1905	0618900	247	末まで在留．
1928	0618900	490	主著に《台湾文化志》(1928)，《領台十年
⋮	⋮	⋮	⋮

1872 年〈都をどり〉を起こした．

(a) An axis index

(b) A document (an encyclopaedia article)

Figure 4. The structure of an axis index

[1] This type of index, in which a key consists of a key value, is called a *value-to-position index*. There is another type of index, but an explanation is omitted.

3.3 Search

When the search engine is invoked with both an axis and search strings, intermediate search results are obtained from both the full-text index and an axis index. These results are merged and sorted by the following method (**Figure 5**). The location of the search string, denoted position_F, is obtained from the full-text index. The location of the feature value, denoted position_A, is obtained from the axis index. Here, *d* is defined as the distance between the search string and the feature value, and is positive regardless of the order in which the search string and the feature value occur. The scoring function contains a monotonically decreasing function of *d* as a term. The function currently used in our prototype is $1 - 10^{-5}d^2$, where *d* is measured by the number of characters. If the score is too low, the search result is discarded. If multiple search strings occur in a document, the nearest one is used. The scoring function also contains terms for evaluating the number of string occurrences in the document (i.e., the term frequency) and in the collection.

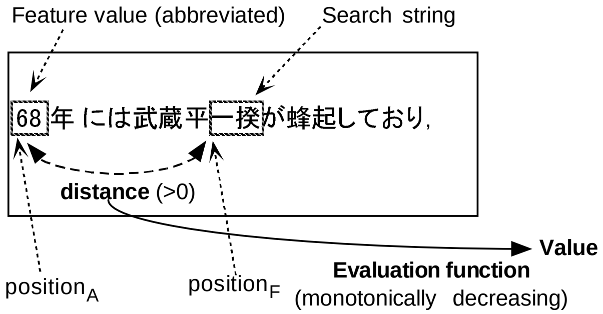

Feature value (abbreviated)
Search string
68年 には武蔵平一揆が蜂起しており,
distance (>0)
Value
Evaluation function
(monotonically decreasing)
position_A
position_F

Figure 5. Search-result evaluation

The scored search results are sorted using multiple keys in the following manner. The feature value is used as the primary key, and the score is used as the secondary key. So the results are ordered by the feature value, and results with the same feature value are ordered by the evaluation value. (See the results of the search for the world's largest river basins in Section 5.)

4. APPLICATIONS

Two applications of the axis-specified search method are discussed here.

4.1 Encyclopedia search

A prototype for searching the whole text of the World Encyclopædia [NEC 95] [HDH 98] using the axis-specified search method has been developed. This encyclopedia contains about 83,000 articles and 160 MB of text including SGML tags.

The menu of this prototype offers three axis-specified search methods: a year-table-style search, a more general quantity search, and a geographical-name search. The geographical-name search is slightly different from the geographical axis search shown in Figure 1 (a) in that search results are sorted by the lexicographic order of the geographical names. The year-table-style search is separate from other quantity searches because it requires both several techniques for specific information extraction and several specific input fields for the user. The user can specify year range by Christian years or Japanese years, and can select the unit of time to be searched.

The unit of time can be chosen from "年" (year) or "世紀" (century). Quantities followed by "年" and "世紀" are collected to the time index. However, other units, such as "月" (month), "日" (day) and "時" (hour), are not collected, because in an encyclopedia they are less important for an axis than the year or century. All the units listed in Section 2 are in the menu of this prototype, and there are some additional ones.

The interface and the result of a geographical-name search for "一揆" (riot) are shown in **Figure 6**. The user specifies an axis and search strings in the input frame.[1] Results are then displayed in the results-list frame. If the user clicks a hyperlink in this frame, the article is displayed in the article frame.

The search system prototype currently works on a Pentium PC with a Linux operating system. Both the index generator and the search engine are written in JPerl5 (the Japanese language version of Perl5). Web browsers, such as Netscape Navigator or Internet Explorer, are used to browse and search the text. Thus, the search engine is called through the common gateway interface (CGI). A GNU database manager (GDBM), which is a non-volatile hash-table manager, is used to generate and search indices.

There are two benefits of using Perl to build this system. One is that the pattern match for extracting information from the text can be written easily in Perl. The other is that a GDBM or several other hash-table managers can be easily accessed through the associative arrays of Perl. Axis and full-text indices can be very easily implemented using a GDBM. However, because Perl programs are executed by an interpreter, the search engine is slow. They are, therefore, suitable for a prototype but not suitable for the final product.

The index sizes and the number of entries in each index are summarized in **Table 1**. The indices include text excerpts. The full-text index is also managed by the GDBM. Uni- and bi-gram indices are used. The size of the full-text index is 420 MB in the current implementation.

4.2 Newspaper search

A prototype for searching the text of Mainichi Newspaper issues published in 1995 by using the axis-specified search method has also been developed. The number of articles is about 111,000 and the amount of text is 120 MB including tags.

[1] The menu items shown in Figure 6 are written in English. However, interface that we usually use is written in Japanese.

Table 1. The sizes of axis indices for the encyclopedia search

Index type	Index size (MB)	Number of entries
Time index	12.0[*1] (4.5[*2])	220,536 (years) 6,579 (centuries)
Quantity index	32.3[*1] (14.5[*2])	611,889
Geographical index	1.9 (1.3[*2])	17,224

[*1] The structure of these indices, which is called *document-value-pair-to-position index*, is slightly different from that shown in Figure 4.

[*2] Index sizes without excerpts.

The programs used in this system are shared or similar to those used in the encyclopedia search. The axis menu is similar too; year-table-style, quantity, and geographical-name searches are implemented. In the year-table-style search, months are collected in addition to years because information on a shorter time scale than in an encyclopedia is important in a newspaper. Days are not currently collected.

In the general quantity search, the range of collected units is similar to that in the encyclopedia search. However, the expressions of units in newspapers are different from those in the encyclopedia. Many units in the encyclopedia are written in symbolic form. They are written in Japanese Kanji characters in newspapers mainly because characters are written downwards, so symbolic forms are not suitable. Several examples are shown:

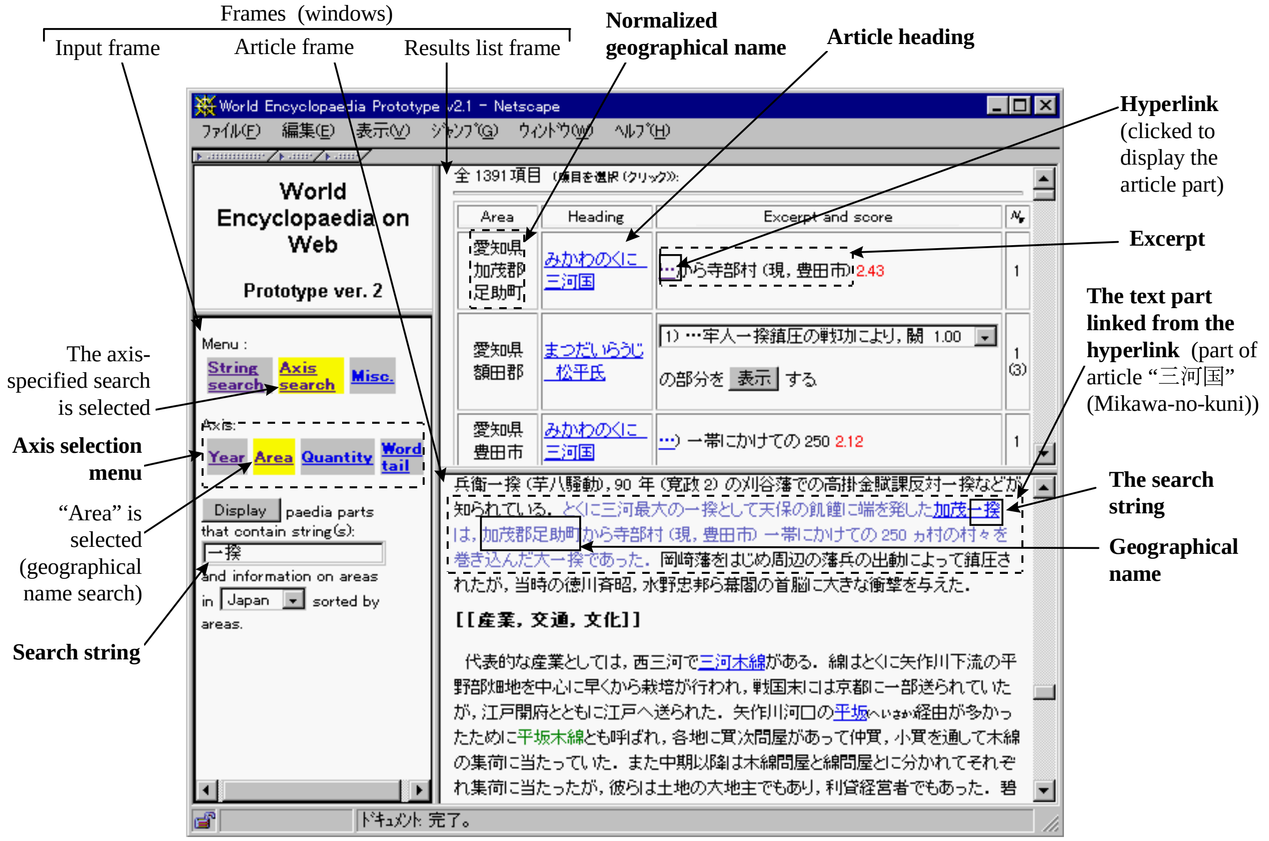
- Square meter is written as “平方メーター” in the newspaper but written as “m^2” in the encyclopedia.
- Kilogram is written as “キログラム” in the newspaper but written as “kg” in the encyclopedia.

Numbers that precede the units are usually written in Kanji characters in newspapers, but are sometimes written in Arabic digits and are sometimes written in a mixture of Kanji and Arabic digits. They are converted to normal numbers. Two examples are shown:

- “一億九千万人”, which means 190,000,000 people and is written fully in Kanji characters.
- “3 万アンペア”, which means 30,000 amperes and is written in a mixture of Kanji and Arabic characters.

For example, the user interface and input/output values of a year-table-style search on semiconductors are shown in **Figure 7**. Here, “半導体” (semiconductor) is specified as a search string, and the range of years is not specified.

The size of the axis indices and the number of entries in each index are summarized in **Table 2**. The time index, which is used for the year-table-style search, are much smaller than those for the encyclopedia search.


Frames (windows)
Input frame
Article frame
Results list frame
Normalized geographical name
Article heading
Hyperlink (clicked to display the article part)
Excerpt
The text part linked from the hyperlink (part of article “三河国” (Mikawa-no-kuni))
The search string
Geographical name
The axis-specified search is selected
Axis selection menu
“Area” is selected (geographical name search)
Search string
World Encyclopaedia Prototype v2.1 - Netscape
ファイル(F) 編集(E) 表示(V) ジャンプ(G) ウィンドウ(W) ヘルプ(H)
World Encyclopaedia on Web
Prototype ver. 2
Menu :
String search
Axis search
Misc.
Axis:
Year
Area
Quantity
Word tail
Display
ドキュメント: 完了。

paedia parts that contain string(s): 一揆 and information on areas in Japan sorted by areas.

全 1391 項目 (項目を選択 (クリック)):

Area	Heading	Excerpt and score	№
愛知県 加茂郡 足助町	みかわのくに 三河国	…から寺部村 (現, 豊田市) 2.43	1
愛知県 額田郡	まつだいらうじ 松平氏	1) …牢人一揆鎮圧の戦功により, 關 1.00 の部分を 表示 する	1 (3)
愛知県 豊田市	みかわのくに 三河国	…) 一帯にかけての 250 2.12	1

兵衛一揆 (芋八騒動), 90 年 (寛政 2) の刈谷藩での高掛金賦課反対一揆などが知られている. とくに三河最大の一揆として天保の飢饉に端を発した加茂一揆は, 加茂郡足助町から寺部村 (現, 豊田市) 一帯にかけての 250 ヵ村の村々を巻き込んだ大一揆であった. 岡崎藩をはじめ周辺の藩兵の出動によって鎮圧されたが, 当時の徳川斉昭, 水野忠邦ら幕閣の首脳に大きな衝撃を与えた.

[[産業, 交通, 文化]]

代表的な産業としては, 西三河で三河木綿がある. 綿はとくに矢作川下流の平野部畑地を中心に早くから栽培が行われ, 戦国末には京都に一部送られていたが, 江戸開府とともに江戸へ送られた. 矢作川河口の平坂へいさか経由が多かったために平坂木綿とも呼ばれ, 各地に買次問屋があって仲買, 小買を通して木綿の集荷に当たっていた. また中期以降は木綿問屋と綿問屋とに分かれてそれぞれ集荷に当たったが, 彼らは土地の大地主でもあり, 利貸経営者でもあった. 碧

Figure 6. The interface and a search result from a World Encyclopædia search (on Netscape Navigator 4)

Table 2. The sizes of axis indices for the newspaper search

Index type	Index size (MB)	Number of entries
Time index	4.9 (4.4[*1])	64,092
Quantity index	94.3 (73.1[*1])	2,034,330
Geographical index	5.6 (4.0[*1])	184,419

[*1] Index sizes without excerpts.

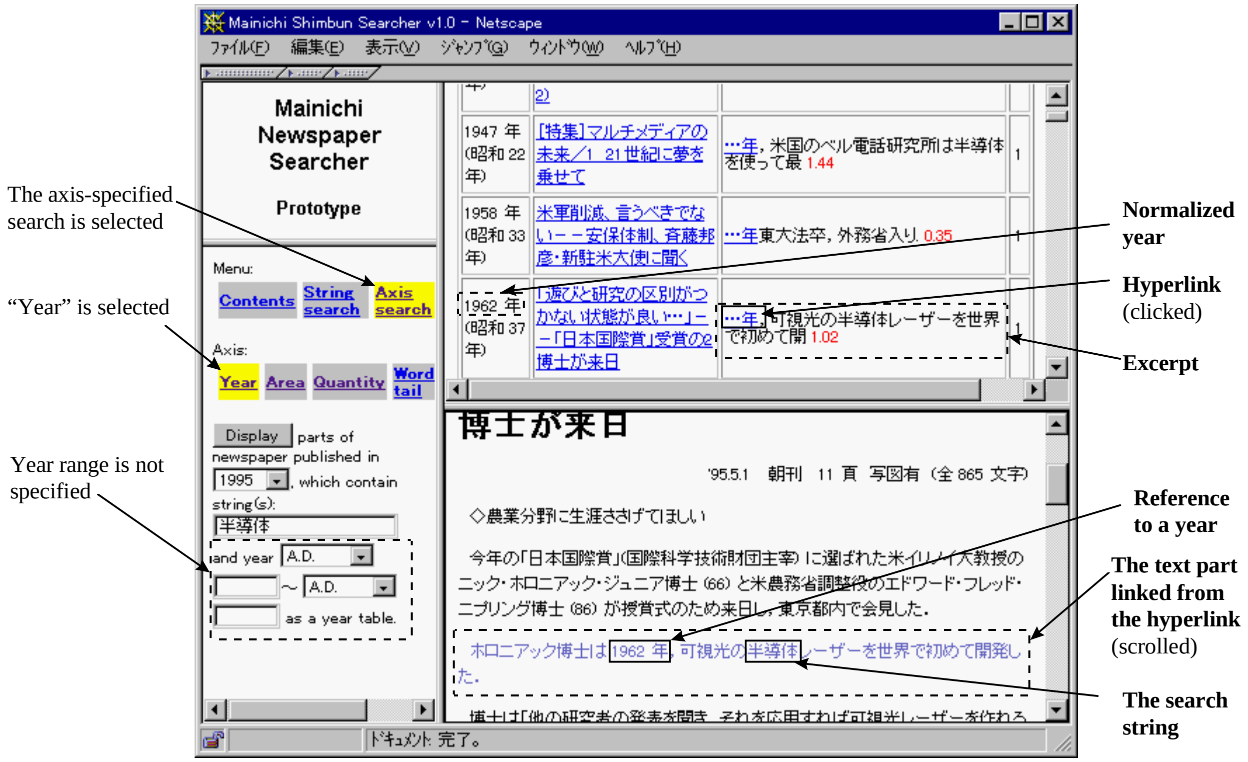
However, the quantity index, which is used for the general quantity search, are much larger. This shows the relative importance of year descriptions in the encyclopedia. The size of the full-text index was 360 MB.

5. CASE STUDIES

Hundreds of experimental search sessions on encyclopedia and newspaper search prototypes have been performed. Some results of the case studies are shown and analyzed here.

The first case study is for the newspaper database and involves a year-table-style search on semiconductors. It was asserted that the user wanted to know the history of semiconductors. Although a historical book would be better than a year of newspaper issues for this purpose, it was assumed that the user wanted to try this search. The user may have received much more information than expected from this search. The interface and search results are shown in Figure 7, and the search results can be summarized as follows.

1. The number of articles that contained "半導体" (semiconductor) was 275. (The number of occurrences was counted using a conventional full-text search function on the prototype.)
2. The number of listed items was 453. This exceeded the number of articles because the articles contained references to two or more years on average. If the articles contained less than one reference to years on average, the number of listed items would be less than the number of articles.
3. The range of years that was extracted from the database was from 1930 to 2002, since 1930 marks the decade when theoretical research on semiconductors began and 2002 is the planned completion date for the construction of a space station for which new semiconductors have been developed.
4. The most noticeable topic in this result is the relationship, especially conflicts, between Japan and the United States concerning semiconductors. The first article on this topic is dated 1980 (as a decade), and it is the seventh item overall. (See the next item.)
5. Of the 453 items, which dated from 1930 to 1993, 75 are examined in detail. The results were as follows.
   - The number of items (excerpts) directly related to semiconductors (as judged by the author) was 38, so the precision was 51%. In other items, a reference to a year was also near the string "semiconductor", but the referred year had no relation to semiconductors.



Mainichi Shimbun Searcher v1.0 - Netscape
ﾌｧｲﾙ(F) 編集(E) 表示(V) ｼﾞｬﾝﾌﾟ(G) ｳｨﾝﾄﾞｳ(W) ﾍﾙﾌﾟ(H)
Mainichi Newspaper Searcher
Prototype
Menu:
Contents
String search
Axis search
Axis:
Year
Area
Quantity
Word tail
Display
ﾄﾞｷｭﾒﾝﾄ: 完了。
The axis-specified search is selected
"Year" is selected
Year range is not specified
Normalized year
Hyperlink (clicked)
Excerpt
Reference to a year
The text part linked from the hyperlink (scrolled)
The search string

Display parts of newspaper published in 1995, which contain string(s): 半導体

and year A.D. ～ A.D. as a year table.

年)	2)		
1947 年 (昭和 22 年)	[特集]マルチメディアの未来／1 21世紀に夢を乗せて	…年，米国のベル電話研究所は半導体を使って最 1.44	1
1958 年 (昭和 33 年)	米軍削減、言うべきでない－－安保体制、斉藤邦彦・新駐米大使に聞く	…年東大法卒，外務省入り. 0.35	1
1962 年 (昭和 37 年)	「遊びと研究の区別がつかない状態が良い…」－－「日本国際賞」受賞の2博士が来日	…年，可視光の半導体レーザーを世界で初めて開 1.02	1

博士が来日

'95.5.1 朝刊 11 頁 写図有 (全 865 文字)

◇農業分野に生涯ささげてほしい

今年の「日本国際賞」(国際科学技術財団主宰) に選ばれた米イリノイ大教授のニック・ホロニアック・ジュニア博士 (66) と米農務省調整役のエドワード・フレッド・ニプリング博士 (86) が授賞式のため来日し，東京都内で会見した．

ホロニアック博士は1962 年，可視光の半導体レーザーを世界で初めて開発した．

博士は「他の研究者の発表を聞き，それを応用すれば可視光レーザーを作れる

Figure 7. The interface and search results from a Mainichi Newspaper search (on Netscape Navigator 4)

- The number of items that mentioned the relationship between Japan and the US concerning semiconductor devices was 24 (32%).

From points 3 and 4, it can be concluded that historical information on one or more specific topics over several decades can be gathered from newspaper issues published over a year. From point 5, it can be concluded that if the user's interest is not very specific, the precision is good.

The second case study was on a quantity search for information on the basins of the world's largest rivers. It was asserted that the user wanted to find information about the rivers that have a large basin. The user could specify the following inputs: “流域” (basin) as a search string and units of area (square meter, etc.) for an axis. The range of feature values was not specified. The user could then see the desired results. The first five items are translated into English here. The order of items coincides with the ranking of basin area except for the Río de la Plata. The basin area of the Río de la Plata, which is written in the encyclopedia, follows that of the Rio Amazônas, but it is described as the fourth largest. The user can see the result without following the hyperlinks, but if the user wants to confirm the ranking, this can be easily done by following the underlined links.

- 6,500,000 km^2 (6.50e+12 m^2)
  1. Rio Amazônas 0.66 (keyword count: 2)
     around … km^2 and it is the largest in the world …
  2. Amazonia 0.39 (keyword count: 6)
     occupies an area as large as … km^2 …
- 4,350,000 km^2 (4.35e+12 m^2)
  1. Río de la Plata 1.02 (keyword count: 2)
     … km^2 and it is the fourth largest in the world …
- 3,690,000 km^2 (3.69e+12 m^2)
  1. Congo River 1.02 (keyword count: 5)
     as large as … km^2. It is next to Rio Amazônas, and is the second largest in the world …
- 3,248,000 km^2 (3.25e+12 m^2)
  1. Mississippi River 1.34 (keyword count: 11)
     as large as … km^2. It is next to Rio Amazônas and the Congo River, and it is the third …

In this result, there are two descriptions of the basin area of the Rio Amazônas, and only one description of the others. The quantities are shown using both the original and normalized units, i.e., m^2 in this case, because the normalized quantities help the user compare the original quantities in different units. “Keyword” means the search string above.

The above result can be regarded as a two-column table or relational database as shown in **Table 3**. This demonstrates that a ranking of certain type of quantities, such as the basin area of the world's largest rivers, can be gathered from an encyclopedia. This is an example of gathering distributed information by using an axis-specified search.

The third case study was on a quantity search for the melting points of materials. It was asserted that the user wanted to know which materials had very high melting points. The user specified the inputs: “融点” (melting point) as a search string and “℃” as a unit. The range of feature values was not specified. The results contained information on many materials. Some of them contained information on tungsten, which has the highest melting point of all metals. Three contiguous items from the search results are translated here.

Table 3. Result of quantity search for m^2 with “流域”

Name of river	Basin area
Rio Amazônas	6,500,000 km^2
Río de la Plata	4,350,000 km^2
Congo River	3,690,000 km^2
Mississippi River	3,248,000 km^2
…	…

- 3407 ℃ (3.41e+03 ℃)
  1. Heat resistant material 0.73 (keyword count: 19)
     … ℃, tantalum 2985 ℃, hafnium 2…
- 3400 ℃ (3.40e+03 ℃)
  1. Tungsten wolfram 0.93 (keyword count: 2)
     have … ℃ …
- 3387 ℃ (3.39e+03 ℃)
  1. Refractory metal 0.80 (keyword count: 12)
     … ℃), rhenium (3180 ℃), tantalum …

All the above items describe the melting point of tungsten. It is not clear why the melting points are different, but it is probably because the measurement conditions differed. The quantity search made these variations in the melting points clear. This is another example of gathering distributed information using an axis-specified search.

6. RELATED WORK

Hearst and Plaunt [Hea 93] proposed a method for retrieving documents that contain a specific subtopic that has a specific relation to a main topic. Their method is based on passage retrieval. However, their purpose is not to extract passages or subtopics themselves, but to retrieve whole documents.

Takeda, Morohashi, and Nomiyama [Mor 95] [Tak 97] proposed a method of showing documents by “information outlining.” In their method, documents can be viewed in chronologically, geographically or in several other ways. The chronological and geographical views are similar to the views shown in Figure 1. However, the purpose of this method is to categorize whole documents and to show the outline of the document collection. Thus, only the main topics of the documents are categorized. Also, only statistical information, such as a histogram, is shown in the view.

Nowell, et al. [Now 96] developed a visual interface for a digital library system called Envision. Envision visualizes search results using two axes. The axes show bibliographic information, such as author or date, and the unit of the search is the document.

Zhu, et al. [Zhu 97] proposed a method for searching for parts and services on the Web. They extracted the types of parts and companies related to these parts from WWW pages and summarized the results as histograms.

Their purpose was not just outlining, but to mine useful information from the results. However, their method has not yet been fully automated, and useful individual information may be discarded because they used statistical methods.

7. CONCLUSION

The axis-specified search method, which is a fine-grained full-text search method has been proposed. The function of axis-specified searching is summarized below.

- The user specifies an axis and search strings, and the search results are ordered along this axis.
- The search results contain excerpts from the text and hyperlinks into the corresponding part of the original text.
- The user can see each topic in the results, which is related to the query, even when a document contains multiple topics.

The axis-specified search was implemented and the feasibility has been demonstrated;

- Indices for feature values were generated during preprocessing, and they are used with a full-text index.
- The method has been applied to an encyclopedia and a newspaper database.

The usefulness of the axis-specified search has been shown through several case studies;

- Distributed information, such as a ranking of a certain quantities or descriptions on a topic, can be effectively gathered.
- A history of one or more specific topics over several decades can be extracted from newspaper issues published over a year.

The main focuses of future work will be as follows:

- The encyclopedia and the news database search systems should be evaluated by users.
- The axis-specified search should be applied to other types of texts, such from Internet newsgroups or from the WWW, and should be evaluated.

Error corrections and updates of this paper will be available at http://www.st.rim.or.jp/~kanada/Papers/-search-papers.html#Axis-DL.

ACKNOWLEDGMENTS

I am grateful to the Hitachi Digital Heibonsha Company for allowing use of the text of the World Encyclopædia. I also thank Mainichi Newspaper Company for allowing use of the newspaper text through the Association for Natural Language Processing.

REFERENCES

[Cut 92] Cutting, D. R., Karger, D. R., Pedersen, J. O., and Tukey, J. W.: Scatter/Gather: a cluster-based approach to browsing large document collections, *15th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*, 318–329, 1992.

[Cut 93] Cutting, D. R., Karger, D. R., Pedersen, J. O.: Constant interaction-time scatter/gather browsing of very large document collections, *16th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*, 126-134, 1993.

[Fra 92] Frakes, W., and Baeza-Yates, R., ed.: *Information Retrieval: Data Structures & Algorithms*, Prentice Hall, 1992.

[HDH 98] *CD-ROM World Encyclopædia*, Hitachi Digital Heibonsha, 1998 (in Japanese).

[Hea 93] Hearst, M. A., and Plaunt, C.: Subtopic Structuring for Full-Length Document Access, *16th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*, 59–68, 1993.

[Mai 95] *CD Mainichi Newspapers 1995*, Mainichi Newspaper Company, 1996.

[Mor 68] Morrison, D. R.: PATRICIA — Practical Algorithm to Retrieve Information Coded in Alphanumeric, *Journal of the ACM*, 15:4, 514–534, 1968.

[Mor 95] Morohashi, M., and Takeda, K.: Information Outlining — Filling the Gap between Visualization and Navigation in Digital Libraries, *Int'l Symp. on Research, Development and Practice in Digital Libraries 1995*, pp. 151–158, Univ. of Library and Information Science, 1995.

[NEC 95] *World Encyclopædia*, NEC Home Electronics Ltd., 1993.

[Now 96] Nowell, L. T., France, R. K., Hix, D., Heath, L. S., and Fox, E. A.: Visualizing Search Results: Some Alternatives to Query-Document Similarity, *19th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*, 67-75, 1996.

[Tak 97] Takeda, K., and Nomiyama, H.: Information Outlining and Site Outlining, *Int'l Symp. on Research, Development and Practice in Digital Libraries 1997*, 99–106, Univ. of Library and Information Science, 1997.

[Zhu 97] Zhu, Q., Hu, F., Yao, K., and Will, P.: Searching for Parts and Services on the Web, *Int'l Symp. on Research, Development and Practice in Digital Libraries 1997*, 123–130, Univ. of Library and Information Science, 1997.