A Method for Information Source Selection using Thesaurus for Distributed Information Retrieval

Shoji Goto, Tadachika Ozono, and Toramatsu Shintani

Dept. of Intelligence and Computer Science, Nagoya Institute of Technology
Gokiso, Showa-ku, Nagoya 466-8555, JAPAN.
Tel: +81-52-744-3153, Fax:+81-52-735-5584, E-mail:{shoji,ozono,tora}@ics.nitech.ac.jp

Abstract

In this paper, we describe a new method for selecting information sources in a distributed environment. Recently, there has been much research on distributed information retrieval, that is information retrieval (IR) based on a multi-database model in which the existence of multiple sources is modeled explicitly. In distributed IR, a method is needed that would enable selecting appropriate sources for users' queries. Most existing methods use statistical data such as document frequency. These methods may select inappropriate sources if a query contains polysemous words. In this paper, we describe an information-source selection method using two types of thesaurus. One is a thesaurus automatically constructed from documents in a source. The other is a hand-crafted general-purpose thesaurus(e.g. WordNet). Terms used in documents in a source differ from one another and the meanings of a term differ depending on the situation in which the term is used. The difference is a characteristic of the source. In our method, the meanings of a term are distinguished between by the relationship between the term and other terms, and the relationship appear in the co-occurrence-based thesaurus. In this paper, we describe an algorithm for evaluating a usefulness of a source for a query based on a thesaurus. For a practical application of our method, we have developed Papits, a multi-agent-based information sharing system. An experiment of selection shows that our method is effective for selecting appropriate sources.

Keyword:
Information Source Selection; Distributed Information Retrieval; Thesaurus; Multi-Agent System

1 Introduction

The increasing amount of information requires information retrieval (IR) systems in order for users to access information effectively. The problem of locating relevant information in distributed information sources is partially solved by large-scale centralized retrieval systems such as Altavista[1] and Google[2]. In centralized systems, documents from around the network are copied to a centralized database, where they are indexed and made searchable. This centralized system suffers from a number of limitations, including coverage limitation, outdated data, and unavailable documents due to limited access.

Distributed IR is a new research area to overcome these problems [4, 19]. On distributed IR, each source has a search function and a user's query is processed in it. The querying user receives results from each source. If a user sends query to all sources, the precision of the search becomes lower due to results from sources with no relation to the query. So, selecting sources appropriate for a user's query is very important part in distributed IR.

In this paper, we focus on selecting sources appropriate for a user's query and describe a new selection method using two types of thesauruses. One is a co-occurrence-based thesaurus, which is as a source description automatically constructed from documents in each source. The other is WordNet [14], a hand-crafted general-purpose thesaurus. We use our method to implement Papits, a multi-agent-based information sharing system developed in our laboratory. The users of Papits share various research information, such as PDF files of research papers, with the aid of agents. Our method is used for information retrieval and "Know Who" search in Papits.

The remainder of this paper is organized as follows: in Section 2, we describe distributed information retrieval. In Section 3, we describe our method with regard to source description and automatical thesaurus construction and we present of our selection method using thesauruses. In Section 4, we provide an outline of Papits as an application of our method. In Section 5, we describe experimental results of our method and shows the usefulness of our method. In Section 6, we show related works. Finally, we conclude with a brief summary and future research directions.

2 Distributed Information Retrieval

Distributed IR is based on multi-database model in which the existence of multiple sources is modeled explicitly. The comparison between centralized IR and distributed IR is shown

[1] http://www.altavista.com/
[2] http://www.google.com/

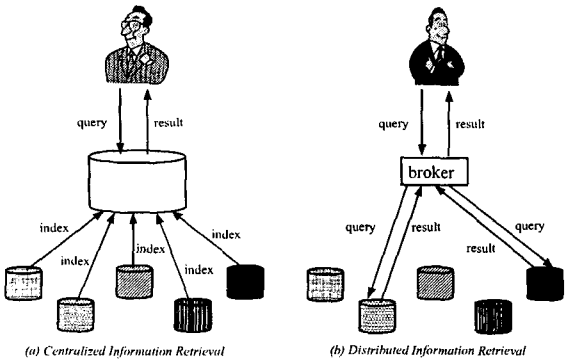

Figure 1. Centralized IR and Distributed IR

in Figure 1. On distributed IR, a broker receives a user's query and sends the query to appropriate sources. The query is processed in each sources. Distributed IR enables IR in environments where source contents are proprietary or carefully controlled, or where access is limited.

Distributed IR consists of three major steps [11] as follows:

Source Selection
Given a set of information sources, the system determines which sources are most likely to contain relevant documents for a given query.

Query Translation
The users' query is translated into the query language of the respective sources and processed at the selected sources, producing lists with a set of individual results.

Result Merging
These lists of results are merged into a single list of documents to be presented to the user.

In these three step, the *Source Selection* is the most important one for search effectiveness because selection correctness affects the effectiveness of the search. Powell et al. [17] have shown that good source selection can result in higher retrieval effectiveness than that achieved in a centralized system,

In the *Source Selection*, the first task is to represent what each source contains. This representation is called the *Source Description*. A simple source description is represented by the words that occur in the source and their frequencies of occurrence[3, 20, 10]. These representations are not effective if a users' query or a document contains polysemous words. This is because the same word is used to describe different things in polysemous words, both in queries and in documents. So, a semantic knowledge about source is needed for a source description. Next, a method is required for selecting sources based on a user's query and a source description.

3 Thesaurus based Source Selection Algorithm

In our method, we use two types of thesaurus. One is thesaurus automatically constructed from documents in a source, as a source description (here in after called SDT). The other is WordNet [14]; it is used as a complement of the SDT. Our source description requires no manpower and all the data for selection are automatically constructed or already prepared. Thus, our method is more scalable than other existing methods.

With co-occurrence-based thesauruses, the form of a particular thesaurus depends on the documents used for its construction. Terms used in documents in a source differ from one another and meanings of a term differ, depending on the situation in which the term is used. The difference is a characteristic of the source and is used for selection. Terms used in each source are distinguished by the words that occur in a source and their frequencies of occurrence. However, methods using only statistical data face the problems caused by polysemous words. In our method, the meanings of a term are distinguished between by the relationship between the term and other terms. The relationship appear in the co-occurrence-based thesaurus. By using a co-occurrence-based thesaurus, our method overcomes the problems caused by polysemous words.

In our algorithm, we assume that terms in a given query are related to each other. These related terms are adjacent to each other or linked closely in a thesaurus. If terms in a query are adjacent to each other or linked closely in the SDT of a source, the query is related to the source. Our method evaluate the usefulness of each source for a given query based on this assumption and a broker selects appropriate sources by ranking the usefulness value receiving from each source,

3.1 Source Description

A recent study on automatic thesaurus construction uses term co-occurrence data [13]. The general idea underlying the use of these data for thesaurus construction is that terms that tend to occur together in documents are likely to have similar, or related, meanings [18].

Our source description forms a graph whose node contains a term and the value of its term frequency occurrence in documents in the source. The edges have term-term similarity. Several similarity measures have been suggested for co-occurrence-based thesauruses [12].

We use average conditional probability(ACP) as similarity measure which was used in collocation map[16] for automatic thesaurus construction. ACP is the average value of two conditional probabilities $P(x|y)$ and $P(y|x)$ defined as Equation (1).

$$\begin{aligned} \text{similarity}(x,y) &= \frac{P(x|y)+P(y|x)}{2} \\ &= \frac{1}{2}\left(\frac{P(x,y)}{P(y)}+\frac{P(x,y)}{P(x)}\right) \end{aligned} \quad (1)$$

For example, suppose that co-occurrence terms are defined as terms occurred in a same document. Then, let N the number of documents, df_x the term x of document frequency and df_{xy} is the co-occurrence frequency of two terms,

Algorithm 1 evaluating a source for a query

$Terms \leftarrow \phi$, $eval \leftarrow 0$,
α and β are given($0 < \alpha < 1, 0 < \beta < 1$)

function EvalSource(q:set):double
  foreach term t_i **in** query q **do**
    TermMapping($t_i, tf_{t_i}, 1.0$)
  end
  foreach $\langle t_1, tf_{t_1}, w_{t_1} \rangle, \langle t_2, tf_{t_2}, w_{t_2} \rangle \in Terms$ **and** $t_1 \neq t_2$ **do**
    $eval \leftarrow eval +$ relation(t_1, t_2)
  end
  return $\frac{eval}{|Terms|C_2}$
end

procedure TermMapping(t:string, tf_t:integer, w:double)
  if $w < \beta$ **then return**
  if a term t is in the source description **then do**
    $Terms \leftarrow Terms \cup \langle t, tf_t, w \rangle$
  else if a term t is in WordNet **then do**
    $Neighbors_t \leftarrow \{t_i | \text{distance}_w(t, t_i) = 1\}$
    foreach t_i **in** $Neighbors_t$ **do**
      TermMapping($t_i, tf_{t_i}, w * \alpha$)
    end
  end if
end

Figure 2. Thesaurus-query similarity evaluation algorithm

$P(x) = \frac{df_x}{N}$ and $P(x, y) = \frac{df_{xy}}{N}$. So the similarity of term x and y are calculated as Equation (2).

$$\text{similarity}(x, y) = \frac{df_{xy}}{2}\left(\frac{1}{df_x} + \frac{1}{df_y}\right) \quad (2)$$

3.2 Source Evaluation Algorithm

Here, we describe an algorithm for evaluating the usefulness of a source based on a source description and a query. The algorithm is shown in Figure 2. The query q is a set of terms, $q = \{t_1, t_2, \cdots, t_n | t_i \text{ is a search term in } q\}$. The thesauruses, SDT and WordNet, are graphs T, $T = \langle V, E \rangle$, where V is a set of nodes and E is a set of edges in T. Here, $T_s = \langle V_s, E_s \rangle$ and $T_w = \langle V_w, E_w \rangle$ represent SDT and WordNet, respectively. A node is a tuple n, $n = \langle t, tf \rangle$, where t is the term and tf is the term frequency of t. An edge is a tuple e, $e = \langle t_x, t_y, w \rangle$, where t_x and t_y are the terms and w is the similarity between t_x and t_y. Here, w is found by using the similarity measure described in Section 3.1. In the following, we describe each step of our algorithm.

(step 0) For each term in a given query, (step 1) is processed.

(step 1) Check whether each term is in SDT or not. If it is, keep the term, and use its *tf* value and the weight of the term. If the term is in query, its weight is 1.0.

(step 2) If the term is not in SDT, check whether the term is in WordNet or not. If it is, select the terms adjacent to the term in WordNet, such as its *synonymy* and *hyponymy / hypernymy*.

(step 3) (step 1) and (step 2) are executed by using the term selected from WordNet instead of the query term. The weight of the term is $\alpha (\alpha < 1)$ times greater than the weight of the query term adjacent to the term in WordNet. This is because the relationship between the term in WordNet and query term becomes lower as the distance between the terms in WordNet is increases.

(step 4) Calculate the average of the relationship between the kept terms. We assume that the relationship between two terms is inversely proportional to the distance between terms in SDT. We define the relationship between term t_1 and t_2 ($t_1 \neq t_2$) as follows:

$$\text{relation}(t_1, t_2) = \frac{w_{t_1} * tf_{t_1} * w_{t_2} * tf_{t_2}}{\text{distance}(t_1, t_2)} * \text{similarity}(t_1, t_2) \quad (3)$$

where similarity(t_1, t_2) is a similarity between t_1 and t_2 and it can be calculated by the similarity measure described in Section 3.1. distance(t_1, t_2) is the shortest path from node t_1 to t_2 in SDT. if node t_1 and t_2 are not adjacent each other, then the similarity of these two terms are the product of the similarity of each edge in shortest path. However, if the graph SDT is non-connected, the path from t_1 to t_2 may not exist. Then, we define the distance and similarity as Equation (4) and (5).

$$\text{distance}(t_1, t_2) = |T_s| \quad (4)$$

$$\text{similarity}(t_1, t_2) = \frac{1}{|T_s|} \quad (5)$$

where $|T_s|$ is the size of source description SDT of the source. This average is the usefulness of the source for a given query, and this value is returned to the broker.

4 Papits: an application of our method

In this section, we present the outline of Papits, a system we used to test our method. Papits is a information and knowledge sharing system developed in our laboratory. Papits consists of personal agents associated with each user as shown in Figure 3. Each agent maintains its user's research information, such as PDF files of research papers or bookmarks for Web browsing and acts on behalf of its user in sharing the files and searching data appropriate for its user's needs by communicating with other agents.

The research paper search and the "Know Who" search functions are provided in Papits. "Know Who" search is a function for searching persons who knows the user's question such as "who has the paper related to *'intelligent agent'*?" or "who knows the words *'reinforcement learning'*?". Each user have different data respectively. When a agent searches

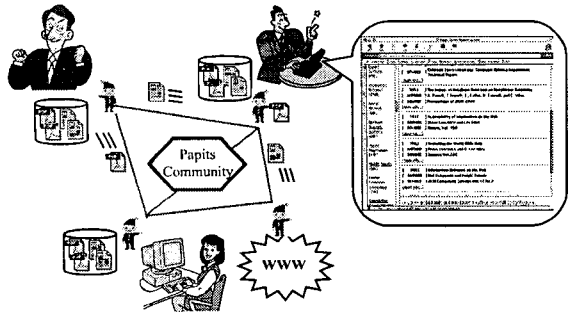

Figure 3. Overview of Papits

```
<?xml version="1.0"?>
<MESSAGE>
  <PERFORMATIVE>query</PERFORMATIVE>
  <CONTENT>
    <SEARCH>
      <PATTERN>
        <TITLE>
          <KEY>agent</KEY>
        </TITLE>
        <YEAR>
          <AFTER>2000</AFTER>
        </YEAR>
      </PATTERN>
    </SEARCH>
  </CONTENT>
</MESSAGE>
```

Figure 4. XML message to search paper

required data or process "Know Who" search, a querying agent becomes a broker and the other agents become sources. The querying agent must select the appropriate agents that have appropriate data for user's query.

Papits is implemented using Java and XML-based inter-agent communication library Xgent implemented in our laboratory. Figure 4 shows a example of a XML message used when an agent searches papers with two conditions; containing "agent" in title of a paper and published after the year 2000.

Our method is used in this situation as follows. First, personal agents construct a thesaurus from PDF files of papers that the corresponding user manages. This thesaurus is the SDT in our method. When an agent receives a query from an user, the agent broadcasts the query to all agents in the community. Each agent that receives the query calculates the usefulness of the own source by using our method and returns the usefulness value. Finally, the querying agent selects appropriate agents and sends the query to these agents. It is considered the user corresponding to the selected agent is persons who knows the question of querying user. So the querying user is notified of the persons on the "Know Who" search

5 Evaluation Experiments

In this section, we describe experiments to check the usefulness of our method. We measured the usefulness of our method in two ways: time needed for constructing source descriptions and precision of selection correctness. We evaluate the usefulness by comparing our method and other existing methods, CORI [3] and CVV [20]. These methods are based on terms' frequencies of occurrence. CORI measures usefulness of each source as the Equation (6) and CVV as the Equation (7)

$$G_{cori}(s|q) = \frac{\sum_{t\in q} p(t|s)}{|q|} \tag{6}$$

$$p(t|s) = d_b + (1 - d_b) * T(t|s) * I(t|s)$$

$$T(t|s) = \frac{\log(\frac{|s|+0.5}{SF_t})}{\log(|s| + 1.0)}$$

$$I(t|s) = d_t + (1 - d_t) * \frac{\log(DF_{t,s} + 0.5)}{\log(DF_t^{max} + 1.0)}$$

$$G_{cvv}(s|q) = \sum_{t\in q} CVV_t * DF_{t,s} \tag{7}$$

$$CVV_t = \frac{\sum_{s\in S}(CV_{t,s} - \overline{CV_t})}{|S|}$$

$$CV_{t,s} = \frac{\frac{DF_{t,s}}{|s|}}{\frac{DF_{t,s}}{|s|} + \frac{\sum_{k\neq s}^{|S|} DF_{t,k}}{\sum_{k\neq s}^{|S|} |k|}}$$

where $G(s|q)$ is the usefulness of a source s for a query q, $DF_{t,s}$ is a document frequency of term t in source s, S is a set of sources, DF_t^{max} is maximum number of document frequency of term t in all sources, SF_t is source frequency of term t, $\overline{CV_t}$ is population mean of $CV_{t,s}$ over all sources, and d_b and d_t are constant numbers. In this experiments, we use 0.4 as d_b and d_t.

We collected web documents from some domain-specific search engines and removed HTML tags from each HTML documents. Each domain-specific search engines has the documents in a same domain. We regards the set of documents collected from it as a virtual information source. We virtually constructed 20 information sources. We found these domain-specific search engines from InvisibleWeb[3]. Each source has about 500 documents. We evaluate the usefulness of our method on this testbed. To evaluate the usefulness, we measures time needed for constructing source descriptions and precision of selection for these three methods. To evaluate the precision, we prepare the 100 queries and appropriate source for each query. These pair of a query and an answer is checked by persons. Both our method and other method(CORI and CVV) are implemented in Java.

5.1 Experimental Results

Figure 5. shows the experimental results for the amount of time needed to construct a source description for each method. A horizontal axis of Figure 5 shows the number of documents to construct source description and the vertical axis shows the time needed for construction(milli-second). As compared with our method and CORI or CVV, our method takes 10 times as much time as other methods. Because

[3] http://www.invisibleweb.com/

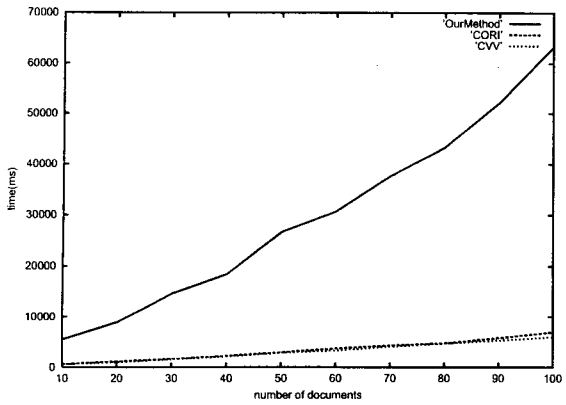

Figure 5. The amount of time to construct a source description

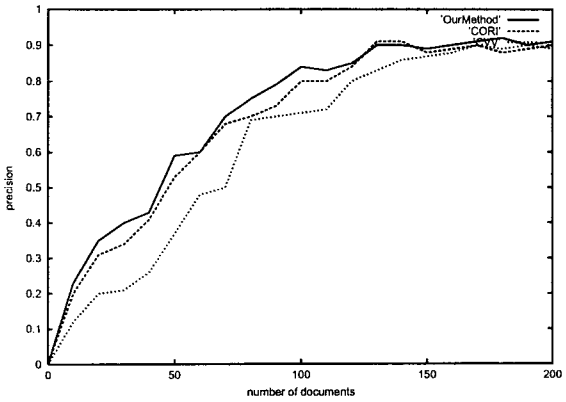

Figure 6. The precision of selection

CORI and CVV only calculate DF or SF value but our method additionally collects term co-occurrence data.

Figure 6 shows the precision of selection correctness for each method. A horizontal axis of Figure 6 shows the number of documents to construct source description and the vertical axis shows the precision of selection for 100 query. All method improve the precision as the number of documents used for source description construction increase. And our method method is higher or same precision than CORI and CVV in all scene.

These experimental results shows that our method needs more time to construct source description than other method based on terms' frequencies of occurrence. But the correctness of source selection of our method becomes higher precision than these method.

6 Related Works

Several approaches to the *Source Selection* have been reported [1, 6, 17, 11, 15, 8]. Some use statistical data such as document frequency. The CORI [3] method uses collection retrieval inference nets whose edges are weighed by the term document frequency and inverse source frequency. The CORI method is based on a modification of INQUERY [2], a document-ranking algorithm, and uses document frequency instead of term frequency and inverse source frequency instead of inverse document frequency. The CVV [20] server-ranking method is based on the cue-validity variance(CVV) of query terms. The CVV method has a centralized broker architecture in which the broker maintains document frequency tables for all sources. The variance of document frequency values across all sources is used to calculate CVV values. All sources must transmit changes in the document frequency values when they occur and therefore need to know the address of the broker. The GlOSS [9] method ranks available vector-space sources according to their usefulness. It uses document frequency and the sum of the weights of each term for all documents in a source, as determined by the vector-space retrieval algorithm used by the source. In [10], Gravano et al. describe GlOSS for Boolean retrievals. These method described above are based on statistical data and are not effective if users' query contains polysemous words. This is because the same word is used to describe different things in polysemous words, both in queries and in documents.

A number of other methods are based on the use of source descriptions. The NetSerf [5] method constructs representations of information sources manually by using semantic knowledge from WordNet [14]. For each query, a semantic distance based on the WordNet hyponymy trees is computed for each source and the source with the smallest distance is chosen. The Pharos [7] method uses decentralized, hierarchical source descriptions to select appropriate sources. The Pharos generates a source description automatically by performing cluster analysis of the source data and classifying the clusters within a manually defined meta-data taxonomy. These method using source description is manpower intensive and the cost of performing a search or constructing a source description with these method is high.

7 Conclusions

In this paper, we described a new information-source selection method for distributed information retrieval. Our algorithm uses an automatically constructed thesaurus and WordNet. The sources are represented by automatically constructed thesaurus. Our method can distinguish between the meanings of a term by using the thesaurus and enables overcoming the problem of the precision degradation caused by polysemous words. Our method does not have a centralized broker maintaining data, such as document frequency for all sources, and all the data needed for selection are automatically constructed or already prepared. As a result, it is easy to add a new information source, and IR systems become more scalable with our method compared to other existing methods. We used our method on Papits, a multi-agent-based information sharing system developed in our laboratory. In our experiment on Papits, we found that our method is effective for information retrieval on multi-agent systems such as

Papits. Through experiments, we evaluate the effectiveness of our methods and compare our method with methods proposed by other researchers. Experimental results shows that our method needs more time than that of existing method, but the precision of selection correctness is higher than that of existing methods.

References

[1] Brendon, C.; Kathryn, S. M.; and Zhihong, L. 2000. Evaluating the Performance of Distributed Architectures for Information Retrieval Using a Variety of Workloads. *ACM Transactions on Information Systems*18(1):1–43.

[2] James, P. C.; W. Bruce, C.; and Stephen, M. H. 1992. The INQUERY Retrieval System. *Proc. of the 3rd International Conference on Database and Expert Systems Applications*: 78–83.

[3] James, P. C.; Zhihong, L.; and W. Bruce, C. 1995. Searching Distributed Collections With Inference Networks. *Proc. of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*: 21–28.

[4] Jamie, C. 2000. Distributed information retrieval. *In W.B. Croft, editor, Advances in Information Retrieval, Kluwer Academic Publishers*: 127–150.

[5] Anil, S. C.; and Kenneth, B. H. 1995. NetSerf: Using Semantic Knowledge to Find Internet Information Archives. *Proc. of the 18th Annual International ACM SIGIR Conference*: 4–11.

[6] Nick, C.; Peter, B.; and David, H. 2000. Server Selection on the World Wide Web. *Proc. of the 5th ACM Conference on Digital Libraries*: 37–46.

[7] R. Dolin; D. Agrawal; and A. El Abbbadi. 1999. Scalable Collection Summarization and Selection. *Proc. of the 4th ACM Conference on Digital Libraries*: 49–58.

[8] Norbert, F. 1999. A Decision-Theoretic Approach to Database Selection in Networked IR. *ACM Transactions on Information Systems* 17(3): 229–249.

[9] Luis, G.; and Hector, G. M. 1995. Generalizing *GlOSS* to Vector-Space Databases and Broker Hierarchies. *Proc. of the 21st International Conference on Very Large Data Bases(VLDB'95)*.

[10] Luis, G.; Hector, G. M.; and Anthony, T. 1999. GlOSS: Text-Source Discovery over the Internet. *ACM Transactions on Database Systems* 24(2): 265–318.

[11] David, H.; and Paul, T. 1999. Methods for Information Server Selection. *ACM Transactions on Information Systems* 17(1): 40–76.

[12] Myoung, C. K.; and Key, S. C. 1998. A comparison of collocation-based similarity measures in query expansion. *Information Processing and Management* 35(1): 19–30.

[13] Rila, M.; Takenobu T.; and Hozumi T. 1999. Combining General Hand-Made and Automatically Constructed Thesauri for Information Retrieval. *16th Proc. of International Joint Conference on Artificial Intelligence(IJICAI99)*: 920–925.

[14] George, A. M. 1995. WordNet: A Lexical Database for English. *Communications of the ACM* 38(11): 39–41.

[15] Alistair, M.; and Justin Z. 1995. Information Retrieval Systems for Large Document Collections. *Proc. of the 3rd Text Retrieval Conference(TREC-3)*: 85–94.

[16] Young, C. P.; Young, S. H.; and Key S. C. 1995 Automatic thesaurus construction using Bayesian networks. *Proc. of the International Conference on information and knowledge management*: 212–217.

[17] Allison L. P.; James, C. F.; Jamie, C.; Margaret, C.; and Charles L. V. 2000. The Impact of Database Selection on Distributed Searching, *Proc. of the International ACM SIGIR Conference*: 232–239.

[18] Qiu, Y.; and Frei H. P. 1993. Concept Based Query Expansion. *Proc. of the 16th International ACM SIGIR Conference*: 160–169.

[19] Baeza Y. 1999. *Modern Information Retrieval.*: ACM Press.

[20] Budi, Y.; and Dik L. L. 1997. Server Ranking for Distributed Text Retrieval Systems on the Internet. *Proc. of the 5th International Conference on Database Systems for Advanced Applications*: 41–49.