Information Source Selection for Resource Constrained Environments

Demet Aksoy
Computer Science Dept.
One Shields Ave.
University of California, Davis
530-752-3601
aksoy@cs.ucdavis.edu

ABSTRACT

Distributed information retrieval has pressing scalability concerns due to the growing number of independent sources of on-line data and the emerging applications. A promising solution to distributed retrieval is metasearching, which dispatches a user's query to multiple sources and gathers the results into a single result set. An important component of metasearching is selecting the set of information sources most likely to provide relevant documents. Recent research has focused on how to obtain statistics for the selection task. In this paper we discuss different information source selection approaches and their applicability for resource-constrained sensor network applications.

1. INTRODUCTION

Finding information is increasingly difficult due to the explosion of content that has resulted from advances in computer networking and data storage. Widely distributed information makes it difficult to issue a single query and get globally optimal results. Even in traditional applications of digital libraries such as the one offered by the University of California at Davis [19], there are numerous electronic databases that can be searched. A user must select which databases are most relevant, issue a query to each of those databases, and then review all of the result sets to retrieve the top matching documents. Similar problems arise in the case of information retrieval from a federation of sensor networks.

Some sensor networks make the collected data publicly available on the World Wide Web for general use [6]. In this case, even large general search engines fail to thoroughly analyze the growing content [12]. Therefore there is a new trend towards specialized search engines in particular topics. In addition, beyond the issue of scalability, many sites provide dynamic content that can not be accessed by the hyperlink traversal strategy used by web crawlers. For instance, consider a keyword search page that provides customized results to the user; a web crawler will only consider the text on the search page as a document, but does not have hyperlinks to the documents available beyond this facade. The non-crawlable content becomes part of what is commonly known as the *Hidden Web*. Similar issues arise in sensor network queries.

One approach to querying distributed content is to dispatch the query to each information source that is likely to contain the requested documents and merge the query results before presenting them to the user. This approach, known as metasearching, has received considerable research attention, e.g., [2, 6, 10, 11, 17, 18]. Selecting information sources that are likely to contain the requested document is the first key for a successful search

This paper focuses on *information source selection* for a metasearching system for applications where system resource limitations and restrictions need to be taken into consideration. The paper is organized as follows. Section 2 describes the design of a general metasearch engine and discusses the issues that arise with the introduction of resource constraints. Section 3 presents various approaches for information source selection for metasearching. Finally conclusions and future research directions are presented in section 4.

2. METASEARCH ENGINES

The main challenges in implementing a successful metasearch engine can be categorized as (1) *information source selection*: the process of determining which information sources are likely to contain relevant content, (2) *query execution*: the process of sending the user's query to each of the selected information source, and (3) *result merging*: the process of aggregating the results from each of the selected source and presenting the final result to the user.

The metasearch engine needs to start with a good classification of the information sources it receives results from. Due to the amount of data and the number of hosts available, and the time limitation of power limited devices efficient information source selection needs special attention. The retrieval results usually need to be filtered and narrowed down before presented to the user. Context

and location are two important factors for such filtering. To provide a good filtering criteria, one needs to ensure that the initial set was comphrehensive and accurate.

The simple approach of sending queries to *all* information sources is not scalable to the large number of hosts and large collections. An effective information source selection algorithm must efficiently use resources while maintaining a low probability that relevant documents are missed. Emerging applications of sensor networks have additional challenges due to resource limitations in terms of energy, storage, and processing capabilities. For instance, limited power of mobile devices and their unpredictable environments require that the querying be efficient and information retrieval results arrive fast. Otherwise the clients will need to consume considerable time in idle listening mode which was shown to be significantly high [13]. In general, data generated by a federation of sensors can be quite dynamic and change frequently. This requires an adaptive approach.

Our paper focuses on information source selection for metasearching in a widely distributed environment with heterogeneous sensor network federations. In the following section we present various information source selection algorithms and provide insights for how federated sensor information retrieval applications can be supported.

3. COMPARISON OF INFORMATION SOURCE SELECTION APPROACHES

Research in information source selection was originally centered on the assumption that domain administrators export a standardized and accurate content summary for each information source. In traditional database applications, a content summary for a database consists of a *term frequency vector* and a *document frequency vector*. The frequency vector counts the number of times each unique term appears over all documents in the database. The document frequency vector counts the number of unique documents that contain each term. Database selection algorithms that require the export of content summaries are called *cooperative* database algorithms, e.g., CORI [5], GlOSS [9], and CVV [23]. Such algorithms presume that the database administrators *can* export the content summary, that they *want* to export it, and that they *will* do it accurately. In an open system, these assumptions are unrealistic for two main reasons. First, information providers may have an incentive to skew the content summaries in favor of their site. Second, the content summaries are difficult to standardize without detailed communication on semantics such as suffix stemming and stop words [8].

An alternative approach is taken by [2, 4, 11, 21]. These algorithms obtain information by issuing queries to the information source and receiving document counts and/or document samples. Such approaches are geared towards widely distributed collections with multiple administrative domains. In the following, these approaches are described in two main categories: query sampling and topical content summaries.

3.1 Query Sampling

Query based approaches can be categorized between those that obtain information on-demand for each incoming query, and those that maintain persistent information about each information source. On-demand approaches execute an initial probing query on all information sources in such a way as to minimize the computational cost on the servers. Preliminary query results are used to select a subset of the information sources for further execution of the original user query. Persistent approaches gather information from each information source before any user queries are processed.

3.1.1 On-Demand Query Sampling

In Hawking et al's study [12], the idea of "lightweight probes" is introduced. These are queries that are handled differently than normal queries. In response to such probes, the server is expected to reply with the following information: 1) the total number of documents on the server, 2) the number of documents containing a specified number of the query terms within a specified proximity of each other, 3) the number of documents in which a specified number of the query terms co-occur, and 4) the number of documents containing each individual query term *t*.

The collected information is used to estimate the number of relevant documents and to allow the metasearcher to select a subset of the information sources to execute the full query. The main problem with this approach is that it requires the cooperation of servers to process lightweight probes in a different manner than standard queries. This problem is addressed in [2] which takes the approach of executing the full query on each information source. To reduce the costs of query execution, only the first few records are retrieved from each information source. These initial results are evaluated to determine which information sources are most likely to contain relevant documents. Though this approach solves the non-standard lightweight query probe requirements of [12], it introduces a new problem as will be discussed at the end of this section.

3.1.2 Persistent Query Sampling

Voorhees et al. [20] studied the persistence of information source statistics retrieved via query sampling. The approach begins by considering the theoretical optimum performance of a result aggregation algorithm. The problem can be stated as follows: given a query Q, information servers $I_1, I_2, \ldots, I_L$ and N, the total number of documents to be retrieved, find the values of $\bullet_1, \bullet_2, \ldots, \bullet_L$ such that

$$\sum_{i=1}^{L} \bullet_i = N \text{ and}$$

$$\sum_{i=1}^{L} F_Q^I(\bullet_i) \text{ is maximum.}$$

where $F_Q^I(\bullet)$ is the number of relevant documents retrieved as a function of total retrieved documents. The challenge is to find a reasonable approximation for this number so that this optimization can be performed. The authors propose that the relevant document recall characteristics of two queries will be similar so long as the queries themselves are topically similar. Therefore, if recall characteristics are stored for a set of training queries, then the recall characteristics of later queries can be estimated by finding queries similar to the ones issued during the training phase. The topical similarity of queries is determined by clustering the training queries and then comparing the incoming query with the cluster centroid using cosine similarity. The main problem with this approach is that computing the recall characteristics for each training query requires knowing which records are relevant. Such an approach works well in a test environment that includes relevance judgments. It does not, however, scale to an environment with many information sources, many topics, and no prior relevance judgments. The manual effort involved in evaluating document relevance is impractical.

The idea of creating estimated content summaries using random sampling has been investigated in [4]. The algorithm is as follows:

1) Select an initial query term.
2) Run a one-term query on the database.
3) Retrieve the top N documents returned by the database.
4) Update the content summary based on the characteristics of the retrieved documents.
5) If the stopping criterion has not been reached, select a new term and go to step 2.

The initial query term is randomly selected from a general dictionary and two stopping criteria are provided. The first one is to stop sampling when a fixed number of documents are retrieved (e.g., n=300). The second one is a more complex metric that examines the change in the rank of terms during periodic points in the sampling. The stopping criteria are met if the change in ranks drops to below a minimum value.

The estimated content summaries can be used with standard cooperative database selection algorithms. Callan et al. [4] analyze the effects of estimated summaries on the performance of CORI [5]. Powell et al. [16] continued this work by examining the effects on GlOSS [9] and CVV [23]. These experiments show that the estimated content summaries are accurate enough for use with CORI, though there is performance degradation with GlOSS and CVV.

Si et al. [18] investigated the use of random sampling for a database selection algorithm that uses a language modeling approach. Language modeling considers information retrieval from a natural language processing perspective. Rather than computing the probability that a document matches a query, it computes the probability that a query is generated from the language model of a document [15]. That is, given the distribution of terms in a document, it computes the chance that randomly selecting terms from this distribution will result in the generation of the given query. [18] shows that this language modeling approach when used with random query sampling yields better performance than CORI.

Gravano et al. [11] propose that query sampling can be improved by creating a hierarchical classification tree to focus the queries on increasingly specific topics. The classification tree is generated during a training phase by a document classifier that creates rules for each topic using sample documents. The particular classification tree chosen in [11] was the first few levels of the Yahoo! hierarchy. The terms for each rule are sent as queries to the database and the number of matching documents is recorded. Two thresholds are set to decide whether a database should be assigned to a topic. The first one, *Coverage*(t_i), is the absolute number of documents in the database that match the query terms for topic t_i. The second one, *Specificity*(t_i), is the fraction of documents that match the query. As the classification algorithm issues the queries, the content summaries for each database are built up by scanning the terms in the retrieved documents and adding their frequency. The actual document count for single term queries are used as anchors to estimate the document count for those words that were not explicitly queried with a single term query. The content summaries of the databases are summed together

based on their position within the classification tree. This turns each node in the tree into a super information source that contains all of the information sources underneath it. When a user issues a query, the query terms are first checked against the content summaries at the top of the tree. The process then compares the query with the topics on the next level of the classification. This continues until the desired number of databases is identified, at which point the queries are issued directly to those databases. The details of the document classifier are presented in [10].

The experiments in [10] suggest an improvement in the content summary accuracy over random document sampling for a fixed number of samples, as well as an improvement in overall retrieval performance. However, the need for supervised machine learning (the document classifier) is a major weakness because obtaining high quality training documents for a non-trivial topic hierarchy is unrealistic. Another disadvantage is the reliance on accurate document counts from the server, which the administrator may skew to make the database more likely to be selected.

A similar approach to [11] is taken by Wang et al. [21] that does not rely on accurate document counts. However, it still requires a manually constructed topic tree, as well as a description for each topic, rather than document samples for each topic. Gravano et al. [10] show that the classification performance of [21] is worse than the earlier approach.

3.2 Topical Content Summaries

The results of [20] and [11] suggest that estimating the quantity of topically related sets of documents in an information source can improve selection performance. In addition to [20] and [11], two other studies using term distributions to model topics are [22] and [17].

In [22], an unsupervised clustering algorithm is run against all of the documents in a distributed collection. Rather than distilling the contents of the database into a single content summary, a term vector is computed for each topical cluster. Incoming queries are compared to the term vectors, and the closest matching clusters are queried for matching documents. The results from the experiments show that the retrieval performance approaches the same level as centralized retrieval. The main weaknesses in this algorithm is the need for each database to export a set of topical content summaries, and the need to limit the queries to documents within a selected set of topics. Both of these requirements entail a substantial amount of cooperation from the database. Shen et al. [17] take a similar approach as [22], where the main difference is a more complex method of clustering documents and comparing incoming queries to the clusters.

For such topical approaches to be practical outside of a test environment, it is necessary to generate the topical content summaries without the need for a cooperative environment. The existing sampling solutions are inadequate for this task. The random sampling approach of Callan et al. [4] fails to provide data on topics when the sample size is small, the database is large, and the topics are not uniformly distributed. A simple example best illustrates this. Assume we have a database D_1 containing 10,000 records, where 100 documents $d_{1,T}$ are related to topic T. The probability of randomly selecting a document related to topic T is given by:

$$p_{T,1} = \frac{|d_{T,1}|}{|D_1|} = \frac{100}{10{,}000} = 0.01$$

Further assume that we have a second database D_2 containing one million records where 100 documents are related to topic T. The probability of randomly selecting a document $d_{2,T}$ related to topic T is given by:

$$p_{T,2} = \frac{|d_{T,2}|}{|D_2|} = \frac{100}{1{,}000{,}000} = 0.0001$$

The number of documents, n, that must be randomly sampled to give a 95% probability p_c of detecting topic T in database D_1 is given by:

$$(1 - p_c) = (1 - p_T)^n$$
$$\log(1 - p_c) = n \log(1 - p_T)$$
$$n = \frac{\log(1 - p_c)}{\log(1 - p_T)}$$

Based on this, we would need to retrieve approximately n = 300 documents to have a 95% chance of detecting topic T in database D_1. On the other hand, 30,000 documents are required to have a 95% chance of detecting topic T in database D_2 because the probability $p_{T,2}$ is much smaller than $p_{T,1}$. To ensure a high probability p_c that all topics are represented in the content summaries, the number of documents required by random sampling is two orders of magnitude greater for database D_2 than D_1. In fact, the number of document samples required grows linearly with the size of the collection. It is therefore not a scalable solution unless one also assumes that the number of documents per topic also grows linearly.

Table 1. A comparison for different information source selection approaches

Technique	*Advantages*	*Disadvantages*
On-Demand Sampling	Can capture the latest information stored based on a particular query	High overhead, initial query time can not be masked
Persistent Query Sampling	Relatively low overhead	Difficult to produce the framework that can provide a good sampling
Topical Content Summaries	Provides detailed information based on individual topics.	Requires a good categorization of topics

The approaches described in [20], [21], and [11] do not require a linear increase in the number of samples to detect topics, but they introduce the problem of manual training. [20] requires a prior relevance judgment for each training document. Wang et al. [21] require a pre-constructed topic hierarchy with a textual description of each topic. Gravano et al. [11] also require a pre-constructed topic hierarchy, as well as document samples for each topic so that the document classifier can generate query terms. All of these approaches are not practical outside of a test environment due to the manual effort involved.

Lawrence and Giles [14] estimate that the largest search engines cover less than 20% of the total number of documents on the World Wide Web. However, when the documents managed by all of the search engines are combined, the overall coverage increases dramatically. The effect of this increased coverage in conjunction with an accurate database selection algorithm is investigated by Craswell et al. [6]. In addition to these inherent problems the associated increase in multimedia content in databases, broadcasts, streaming media etc. has generated further requirements for more effective access to giant global information repositories for mobile clients.

The definition of a topic however is ambiguous. Different users may choose to classify documents into different topics. In addition, the same user may choose to assign a document to different categories depending on the current information need. Despite these problems, simple approaches that model topics based on term distributions appear to be successful.

3.3 Comparison

In Table 1 we present a comparison of various approaches in three major categories. As suggested Persistent Query Sampling approaches are not appropriate for dynamically changing contents in the information source. On-Demand Sampling, on the other hand, results in a high latency and resource usage since the generation of the first records result in the most significant latency.

An interesting question is how these approaches can be applied in a sensor network environment where different administrations make their data publicly available. If the base stations of sensor networks are directly connected to the Internet, we result in a similar setting to traditional information retrieval within the Internet. The main difference would be the dynamically changing content with updates from the sensor nodes even though the type of the information that is published stays static in most cases. In this regard, topical Content Summaries when combined with approximations of On-Demand Sampling can provide a good trade-off for federated sensors.

Retrieving information directly from the sensor nodes deployed in the field, on the other hand, requires additional power, space and storage optimizations. Due to the resource constraints of sensor networks, none of these approaches can be directly applied in this emerging field. It is important to adapt information source selection approaches to provide a trade-off between efficiency and accuracy [1]. A key requirement for information source selection algorithms is to obtain topical content summaries in non-cooperative environments using a scalable number of document samples and without the need for manual training.

4. CONCLUDING REMARKS

Emerging applications such as sensor networks have limited resources. In this paper we provided a survey of query sampling based methods for obtaining information source selection statistics. The on-demand approaches either require a cooperative environment or introduce latency and excessive resource usage. Of the persistent approaches, only random sampling is practical due to the manual training requirements of the other solutions. Though random sampling provides accurate estimated content summaries for selection algorithms, it is not a scalable solution for algorithms that require the identification of topics. On the other hand topical selection algorithms outperform single content summary approaches, and therefore represents an avenue for further

research for research constrained environments of sensor networks.

5. REFERENCES

[1] D. Aksoy, M.S. Leung. *Pull vs Push: A Quantitative Comparison for Data Broadcast*. In Proc. of IEEE GLOBECOM, 2004.

[2] F.Abbaci, J. Savoy, and M. Beigbeder. A Methodology for Collection Selection in Heterogeneous Contexts. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC)*, Las Vegas, NV, pages 529-535, 2002.

[3] J. Callan and M. Connell. Query-based sampling of text databases. *ACM Transactions on Information Systems (TOIS)*, 19(2), pages 97-130, April 2001.

[4] J. Callan, M. Connell, and A. Du. Automatic discovery of language models for text databases. In *Proceedings of the International ACM Conference on Management of Data (SIGMOD)*, Philadelphia, PA, pages 479-490, 1999.

[5] J. Callan, Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In *Proceedings of the Eighteenth International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, Seattle, WA, pages 21-29, July 1995.

[6] N. Craswell, P. Bailey, and D. Hawking. Server selection on the World Wide Web. In *Proceedings of the Fifth ACM Conference on Digital Libraries (DL)*, San Antonio, TX, pages 37-46, 2000.

[7] K. A. Delin, S. P. Jackson, S. C. Burleigh, D. W. Johnson, R. R. Woodrow, and J. T. Britton, The JPL Sensor Webs Project: Fielded Technology , In *Proceedings of Space Mission Challenges for Information Technology*, Pasadena, CA, July 2003.

[8] L. Gravano, K. C. Chang, H. Garcia-Molina, and A. Paepcke. STARTS: Stanford proposal for Internet Meta-Searching. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, Tucson, AZ, pages 207-218, May 1997.

[9] L. Gravano, H. Garcia-Molina, and A. Tomasic. GlOSS: text-source discovery over the Internet. *ACM Transactions on Information Systems (TOIS)*, 24(2), pages 229-264, June 1999.

[10] L. Gravano, P. Ipeirotis, and M. Sahami. QProber: A system for automatic classification of hidden-web databases. *ACM Transactions on Information Systems (TOIS)*, 21(1), pages 1-41, January 2003.

[11] L. Gravano, and P. Ipeirotis. Distributed Search over the Hidden Web: Hierarchical Database Sampling and Selection. In *Proceedings of the Twenty-Eighth VLDB Conference*, Hong Kong, 2002.

[12] D. Hawking, and P. Thistlewaite. Methods for information server selection. *ACM Transactions on Information Systems (TOIS)*, 17(1), pages 40-76, January 1999.

[13] B. Hohlt, L. Doherty, and E. Brewer. *Flexible Power Scheduling for Sensor Networks*. IPSN 2004.

[14] S. Lawrence and C. L. Giles. Accessibility of information on the web. *Nature*, 400, pages 107-109, 1999.

[15] J. Ponte and W. Croft. A language modeling approach to information retrieval. In *Proceedings of the International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 275-281, 1998.

[16] A.L. Powell, J.C. French, J. Callan, M. Connell, and C.L. Viles. The impact of database selection on distributed searching. In *Proceedings of the Twenty-Third International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, Athens, Greece, pages 232-239, 2000.

[17] Y. Shen, and D. Lee. A Meta-Search Method Reinforced by Descriptors of Clusters. In *Proceedings of the Second International Conference on Web Information Systems Engineering*, Kyoto, Japan, pages 129-136, Dec 2001.

[18] L. Si, R. Jin, J. Callan, and P. Ogilvie. Language modeling framework for resource selection and results merging. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM)*, McLean, VA, pages 391-397, 2002.

[19] University of California Davis Digital Library, 2005.

[20] E. Voorhees, N. Gupta, B. Johnson-Laird. *Learning collection fusion strategies*, In Proceedings of the Eighteenth International ACM Conference on Research and Development in Information Retrieval (SIGIR), Seattle, Washington, pages 172-179, July 1995.

[21] W. Wang, W. Meng, and C. Yu. Concept hierarchy based text database categorization in a metasearch engine environment. In *Proceedings of the First International Conference on Web Information Systems Engineering*, Hong Kong, pages 283-290, June 2000.

[22] J. Xu and W. B. Croft. Cluster-based language models for distributed retrieval. In *Proceedings of the Twenty-Second International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, Berkeley, CA, pages 254-261, 1999.

[23] B. Yuwono and D. Lee. Server ranking for distributed text retrieval systems on the internet. In *Proceedings of the Fifth International Conference on Database Systems for Advanced Applications (DASFAA)*, pages 41–49, Melbourne, April 1997.