Foundations and Trends® in
sample
Vol. xx, No xx (xxxx) 1–109

DOI: xxxxxx

now
the essence of knowledge

Federated Search

Milad Shokouhi[1] and Luo Si[2]

[1] *Microsoft Research, 7 JJ Thomson Avenue, Cambridge CB30FB , UK , milads@microsoft.com*
[2] *Purdue University, 250N University Street, West Lafayette IN 47907-2066 , USA , lsi@cs.purdue.edu*

Abstract

Federated search (federated information retrieval or distributed information retrieval) is a technique for searching multiple text collections simultaneously. Queries are submitted to a subset of collections that are most likely to return relevant answers. The results returned by selected collections are integrated and merged into a single list. Federated search is preferred over centralized search alternatives in many environments. For example, commercial search engines such as Google cannot easily index uncrawlable hidden web collections while federated search systems can search the contents of hidden web collections without crawling. In enterprise environments, where each organization maintains an independent search engine, federated search techniques can provide parallel search over multiple collections.

There are three major challenges in federated search. For each query, a subset of collections that are most likely to return relevant documents are selected. This creates the *collection selection* problem. To be able to select suitable collections, federated search systems need to acquire some knowledge about the contents of each collection, creating the *col-*

lection representation problem. The results returned from the selected collections are merged before the final presentation to the user. This final step is the *result merging* problem.
The goal of this work, is to provide a comprehensive summary of the previous research on the federated search challenges described above.

Contents

1

Introduction

Internet search is one of the most popular activities on the web. More than 80% of internet searchers use search engines for finding their information needs [Spink et al., 2006]. In September 1999, Google claimed that it received 3.5 million queries per day.[1] This number increased to 100 million in 2000,[2] and has grown to hundreds of millions since.[3] The rapid increase in the number of users, web documents and web queries shows the necessity of an advanced search system that can satisfy users' information needs both effectively and efficiently.

Since Aliweb [Koster, 1994] was released as the first internet search engine in 1994, searching methods have been an active area of research, and search technology has attracted significant attention from industrial and commercial organizations. Of course, the domain for search is not limited to the internet activities. A person may utilize search systems to find an email in a mail box, to look for an image on a local machine, or to find a text document on a local area network.

[1] `http://www.google.com/press/pressrel/pressrelease4.html`, accessed on 17 Aug 2010.
[2] `http://www.google.com/corporate/history.html`, accessed on 17 Aug 2010.
[3] `http://www.comscore.com/Press_Events/Press_Releases/2010/8/comScore_Releases_July_2010_U.S._Search_Engine_Rankings`, accessed on 17 Aug 2010.

Commercial search engines use programs called crawlers (or spiders) to download web documents. Any document overlooked by crawlers may affect the users perception of what information is available on the web. Unfortunately, search engines cannot easily crawl documents located in what is generally known as the *hidden web* (or *deep web*) [Raghavan and García-Molina, 2001]. There are several factors that make documents uncrawlable. For example, page servers may be too slow, or many pages might be prohibited by the robot exclusion protocol and authorization settings. Another reason might be that some documents are not linked to from any other page on the web. Furthermore, there are many *dynamic pages*—pages whose content is generated on the fly—that are crawlable [Raghavan and García-Molina, 2001] but are not bounded in number, and are therefore often ignored by crawlers.

As the size of the hidden web has been estimated to be many times larger than the number of visible documents on the web [Bergman, 2001], the volume of information being ignored by search engines is significant. Hidden web documents have diverse topics and are written in different languages. For example, PubMed[4]—a service of the US national library of medicine—contains more than 20 million records of life sciences and biomedical articles published since the 1950s. The US census Bureau[5] includes statistics about population, business owners and so on in the USA. There are many patent offices whose portals provide access to patent information, and there are many other websites such for yellow pages and white pages that provide access to hidden web information.

Instead of expending effort to crawl such collections—some of which may not be crawlable at all—*federated search* techniques directly pass the query to the search interface of suitable collections and merge their results. In federated search, queries are submitted directly to a set of searchable collections—such as those mentioned for the hidden web—that are usually distributed across several locations. The final results are often comprised of answers returned from multiple collections.

From the users' perspective, queries should be executed on servers

[4] `http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed`, accessed on 17 Aug 2010.
[5] `http://www.census.gov`, accessed on 17 Aug 2010.

that contain the most relevant information. For example, a government portal may consist of several searchable collections for different organizations and agencies. For a query such as 'Administrative Office of the US Courts', it might not be useful to search all collections. A better alternative may be to search only collections from the `www.uscourts.gov` domain that are likely to contain the relevant answers.

However, federated search techniques are not limited to the web and can be useful for many enterprise search systems. Any organization with multiple searchable collections can apply federated search techniques. For instance, Westlaw[6] provides federated search for legal professionals covering more than 30,000 databases [Conrad et al., 2002a;b; Conrad and Claussen, 2003]. The users can search for case law, court documents, related newspapers and magazines, public records, and in return, receive merged results from heterogeneous sources. FedStats[7] is an online portal of statistical information published by many federal agencies. The crawls for the original centralized search in FedStats could be updated only every three months. Therefore, a federated search solution was requested and this was the main focus of the FedLemur project [Avrahami et al., 2006].[8] FedStats enables citizens, businesses, and government employees to find useful information without separately visiting web sites of individual agencies.

Federated search can be also used for searching multiple catalogs and other information sources. For example, in the Cheshire project,[9] many digital libraries including the UC Berkeley Physical Sciences Libraries, Penn State University, Duke University, Carnegie Mellon University, UNC Chapel Hill, the Hong Kong University of Science and Technology and a few other libraries have become searchable through a single interface at the University of Berkeley. Similarly, The European Library[10] provides a federated search solution to access the resources of 47 national libraries.

---

[6] `http://www.thomsonreuters.com/products_services/legal/legal_products/393832/Westlaw` , accessed on 17 Aug 2010.

[7] `http://search.fedstats.gov`, accessed on 17 Aug 2010.

[8] FedStats search is currently powered by `google.com`.

[9] `http://cheshire.berkeley.edu/`, accessed on 17 Aug 2010.

[10] `http://search.theeuropeanlibrary.org/portal/en/index.html`, accessed on 17 Aug 2010.

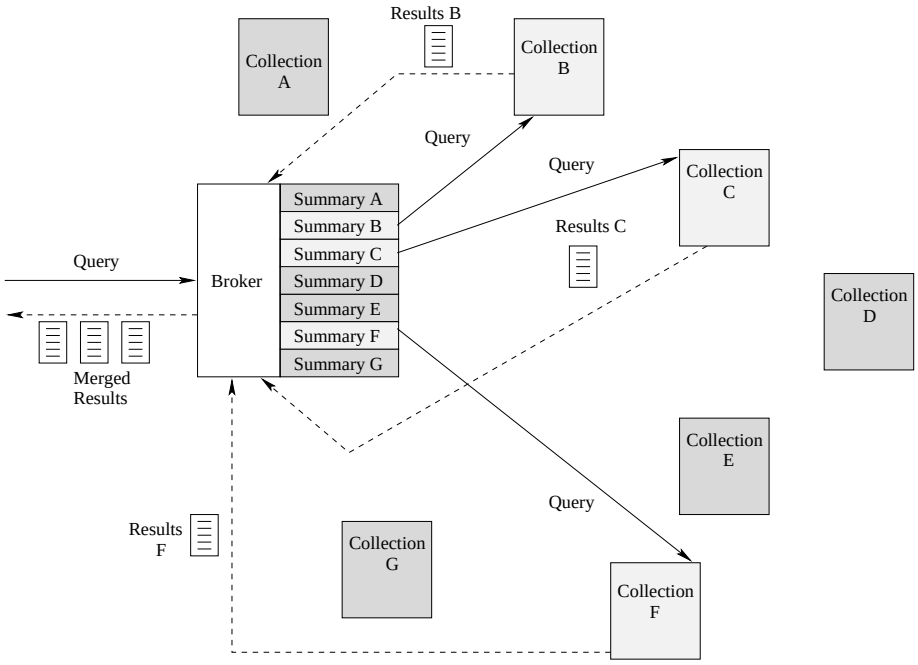

Fig. 1.1 *The architecture of a typical federated search system. The broker stores the representation set (the summary) of each collection, and selects a subset of collections for the query. The selected collections then run the query and return their results to the broker, which merges all results and ranks them in a single list.*

1.1 Federated search

In federated search systems,[11] the task is to search a group of independent collections, and to effectively merge the results they return for queries.

Figure 1.1 shows the architecture of a typical federated search system. A central section (the *broker*) receives queries from the users and sends them to collections that are deemed most likely to contain relevant answers. The highlighted collections in Figure 1.1 are those selected for the query. To route queries to suitable collections, the broker needs to store some important information (summary or representation) about available collections. In a *cooperative* environment, collec-

[11] Also referred to as distributed information retrieval (DIR).

tions inform brokers about their contents by providing information such as their term statistics. This information is often exchanged through a set of shared protocols such as STARTS [Gravano et al., 1997] and may contain term statistics and other metadata such as collection size. In *uncooperative* environments, collections do not provide any information about their contents to brokers. A technique that can be used to obtain information about collections in such environments is to send sampling (*probe*) queries to each collection. Information gathered from the limited number of answer documents that a collection provides in response to such queries is used to construct a *representation set*; this representation set guides the evaluation of user queries and ranking collections. The selected collections receive the query from the broker and evaluate it on their own indexes. In the final step, the broker ranks the results returned by the selected collections and presents them to the user.

Federated search systems therefore need to address three major issues: how to represent the collections, how to select suitable collections for searching; and how to merge the results returned from collections.[12] Brokers typically compare each query to representation sets—also called summaries [Ipeirotis and Gravano, 2004]—of each collection, and estimate the goodness of the collection accordingly. Each representation set may contain statistics about the lexicon of the corresponding collection. If the lexicon of the collections is provided to the central broker—that is, if the collections are cooperative—then complete and accurate information can be used for collection selection. However, in an uncooperative environment such as the hidden web, the collections need to be sampled to establish a summary of their topic coverage. This technique is known as query-based sampling [Callan and Connell, 2001] or query probing [Gravano et al., 2003].

Once the collection summaries are generated, the broker has sufficient knowledge for collection selection. It is usually not feasible to search all collections for a query due to time constraints and bandwidth restrictions. Therefore, the broker selects a few collections that are most likely to return relevant documents based on their summaries.

[12] We briefly describe other common challenges such as *building wrappers* in Chapter 2.

The selected collections receive the query and return their results to the broker.

Result merging is the last step of a federated search session. The results returned by multiple collections are gathered and ranked by the broker before presentation to the user. Since documents are returned from collections with different lexicon statistics and ranking features, their scores or ranks are not comparable. The main goal of result merging techniques is computing comparable scores for documents returned from different collections, and ranking them accordingly.

1.2 Federated search on the web

The most common forms of federated search on the web include *Vertical* search, *Peer-to-Peer* (P2P) networks, and *metasearch* engines. Vertical search—also known as aggregated search—blends the top-ranked answers from search verticals (e.g. images, videos, maps) into the web search results. P2P search connects distributed peers (usually for file sharing), where each peer can be both *server* and *client* . Metasearch engines combine the results of different search engines in single result lists.

1.2.1 Vertical (aggregated) search

Until recently since their first appearance, web search engines used to only show text answers in their results. Users interested in other types of answers (e.g. images, videos, and maps), had to directly submit their queries to the specialized *verticals*.

In 2000, the Korean search engine Naver[13] introduced *comprehensive search* and blended multimedia answers in their default search results. In May 2007, Google launched aggregated search (universal search) "to break down the walls that traditionally separated [their] various search properties and integrate the vast amounts of information available into one simple set of search results".[14] In aggregated search, the top-ranked answers from other information sources (e.g. im-

[13] `http://www.naver.com`, accessed 17 Aug 2010.

[14] `http://googleblog.blogspot.com/2007/05/universal-search-best-answer-is-still.html`, accessed 17 Aug 2010.

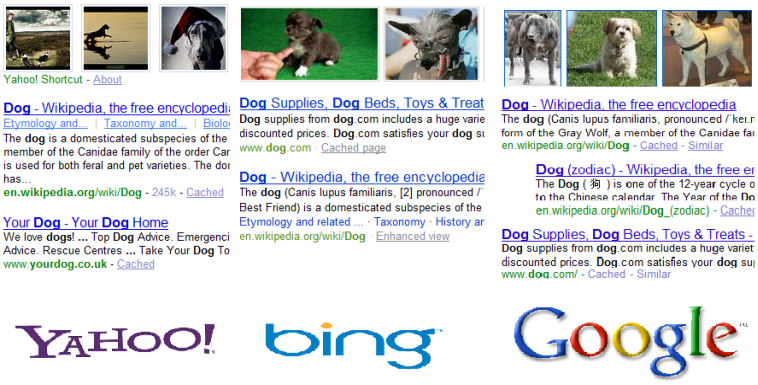

Fig. 1.2 *The outputs of three major search engines for the query "dog". The top-ranked answers from the image vertical are blended in the final results.*

age vertical) are merged with the default text results. Universal search substantially increased the traffic of Google's non-text search verticals. For instance, the traffic of Google Maps increased by more than 20%.[15] Since then, all other major search engines such as Yahoo![16] and Bing have adapted aggregated search techniques. Figure 1.2 shows the results returned by three major search engines for the query "dog". It can be seen that all search engines merge some image answers along with their text results.

An aggregated search interaction consists of two major steps: *vertical selection* and *merging*. In the first step, the verticals relevant to the query are selected. A few examples of common verticals that are utilized by current search engines are: images, videos, news, maps, blogs, groups and books. The answers returned from the selected verticals are integrated with the default web results in the *merging* step.

Aggregated search was discussed in a workshop at SIGIR 2008 [Murdock and Lalmas, 2008] as a promising area of research. Less than a year after, Diaz [2009] proposed a click-based classifier for integration of news answers into web search results—as the first large-scale published study on aggregated search that won the best paper award at

[15] `http://searchengineland.com/070608-091826.php`, accessed 17 Aug 2010.
[16] Yahoo! has recently launched a new website (`http://au.alpha.yahoo.com/`) that applies aggregated search on a greater number of data sources.

WSDM 2009.[17] Arguello et al. [2009b] proposed a classification-based method for vertical selection. The authors trained a classifier with features derived from the query string, previous query logs, and vertical content. They tested their techniques on a framework of 18 verticals, for which they won the best paper award at SIGIR 2009.[18] Diaz and Arguello [2009] showed that integrating users feedback such as clicks can significantly improve the performance of vertical selection methods.

Aggregated search is a new area of research, and has opened several directions for future work; what search verticals shall be selected for a query? How can the results of different verticals be merged into a single list? Do users prefer aggregated search results? How aggregated search changes users' search behaviors?

1.2.2 Peer-to-peer networks

Lu [2007] showed that the search task in a peer-to-peer network is closely related with the research topic of federated search. A peer-to-peer network (P2P) consists of three main types of objects; information providers, information consumers, and a search mechanism that retrieves relevant information from providers for consumers.

The P2P network architectures can be divided into four categories: *broker-based* P2P networks (e.g., the original Napster music file-sharing system[19]) have a single centralized service that also contains document lists shared from peer nodes. The centralized service responds to queries from consumers by returning the pointers of relevant documents. In *Decentralized* P2P architectures such as Gnutella v0.4[20] each peer node can serve as both provider and consumer. *Hierarchical* P2P architectures such as , Gnutella v0.6[21], Gnutella2[22], BearShare[23] and

[17] `http://www.wsdm2009.org`, accessed 17 Aug 2010.
[18] `http://sigir2009.org`, accessed 17 Aug 2010.
[19] `http://www.napster.com`, accessed 17 Aug 2010.
[20] `http://rfc-gnutella.sourceforge.net/developer/stable/index.html`, accessed 17 Aug 2010.
[21] `http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html`, accessed 17 Aug 2010.
[22] `http://g2.trillinux.org/index.php?title=Main_Page`, accessed 17 Aug 2010.
[23] `www.bearshare.com`, accessed 17 Aug 2010.

Swapper.NET[24] utilize local directory services that often work with each other for routing queries and merging search results. *Structured-based* P2P networks such as CAN [Ratnasamy et al., 2001] and Chord [Stoica et al., 2003] often use distributed hash tables for searching and retrieving files.

The search mechanism in P2P network addresses similar research problems of federated search such as representing useful contents of peer nodes and local search directories (collection representation), routing queries to relevant nodes or directories (collection selection), and combining search results (result merging). Early search mechanism in P2P networks focused on named-based or controlled based representation with simple query routing mechanism such as flooding and simple merging methods based on the frequency of term matching or content-independent features. More recent studies [Lu and Callan, 2003a; 2006; Lu, 2007] explored full-text representations with content-based query routing and relevance-based results integration. Therefore, improving collection representation, collection selection and result merging in federated search can have a direct impact on the quality of search in P2P networks.

1.2.3 Metasearch engines

Metasearch engines provide a single search portal for combining the results of multiple search engines. Metasearch engines do not usually retain a document index; they send the query in parallel to multiple search engines, and integrate the returned answers. The architecture details of many metasearch engines such as Dogpile,[25] MetaCrawler [Selberg and Etzioni, 1995; 1997a], AllInOneNews [Liu et al., 2007], ProFusion [Gauch and Wang, 1996; Gauch et al., 1996a], Savvysearch [Dreilinger and Howe, 1997], iXmetafind [Han et al., 2003], Fusion [Smeaton and Crimmins, 1997], and Inquirus [Glover et al., 1999; Lawrence and Giles, 1998] have been published in recent years.

Figure 1.3 shows the answers returned by Metacrawler [Selberg and Etzioni, 1997a] for the query "federated search". It can be seen that

[24] `http://www.revolutionarystuff.com/swapper`, accessed 17 Aug 2010.
[25] `http://www.dogpile.com/dogpile/ws/about?_IceUrl=true`, accessed on 17 Aug 2010.

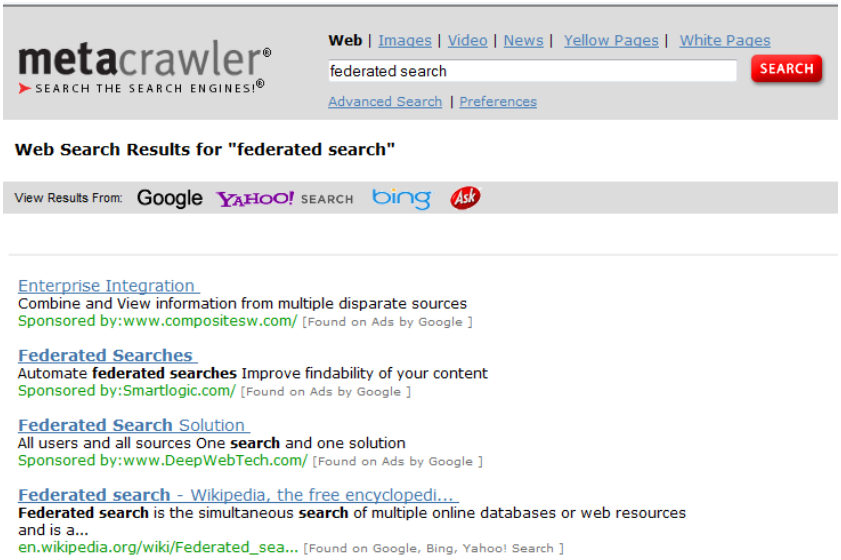

Fig. 1.3 *The results of the query "federated search" returned from Metacrawler [Selberg and Etzioni, 1997a] metasearch engine. It can be seen that the results are merged from different sources such as Google, Yahoo! and Bing search engines.*

the presented results are merged from different search engines such as Yahoo! and Google, Ask and Bing.

Compared to the centralized search engines, metasearch engines have advantages such as broader coverage of the web and better search scalability [Meng et al., 2002]. The index and coverage of commercial search engines are substantially different. Many of the pages that are indexed by one search engine may not be indexed by another search engine. Bar-Yossef and Gurevich [2006] suggested that the amount of overlap between the indexes of Google and Yahoo! is less than 45%.

1.3 Outline

This paper presents a comprehensive summary of federated search techniques. This section provides a road map for the remaining chapters.

In Chapter 2, we compare the collection representation sets (sum-

maries) in cooperative and uncooperative environments. We also discuss several approaches for improving incomplete summaries, including the previous research on estimating the size of collections from sampled documents. We end this chapter by describing *wrappers*, the programs used for interacting with the interfaces of hidden-web collections, and summarizing available techniques for evaluating the quality of collection summaries.

In Chapter 3, we compare different collection selection methods by categorizing the current techniques into two main groups; *lexicon-based*, and *document-surrogates*. The former group mainly consists of techniques that are more suitable for cooperative environments, while the latter group includes collection selection methods based on incomplete sampled documents. We also provide an overview of the previous work on query-classification in the context of federated search. In the last section of this chapter, we discuss the common metrics for evaluating the effectiveness of collection selection methods.

In Chapter 4, we discuss several federated search merging techniques. We also provide a brief summary of commonly used blending techniques in closely related areas of data fusion and metasearch.

In Chapter 5, we discuss common datasets used for evaluating the federated search techniques. This is important because relative performance of federated search methods can vary significantly between different testbeds [D'Souza et al., 2004b; Si and Callan, 2003a].[26]

Finally, in Chapter 6 we present our conclusions and discuss directions for future work.

[26] We use the term *testbed* to refer to a set of collections that are used together for federated search experiments (collection selection and result merging).

2

Collection representation

In order to select suitable collections for a query, the broker needs to know about the contents of each collection as well as other important information (e.g., size). For example the query "basketball" may be passed to sport-related collections, while for the query "Elvis" collections containing articles about music might be more appropriate.

For this purpose, the broker keeps a representation set for each collection. This is illustrated in Figure 2.1. The representation set of each collection contains information about the documents that are indexed by that collection, and can be generated manually on the broker by providing a short description of the indexed documents [Chakravarthy and Haase, 1995; Manber and Bigot, 1997]. However, representation sets created manually are usually brief and cannot capture many terms that occur in a collection. In practice, collection representation sets are therefore usually generated automatically, and their comprehensiveness depends on the level of cooperation in the federated search environment.

In cooperative environments, representation sets may contain the complete lexicon statistics of collections and many other useful metadata [D'Souza et al., 2004a; Gravano et al., 1997; Xu and Callan, 1998;

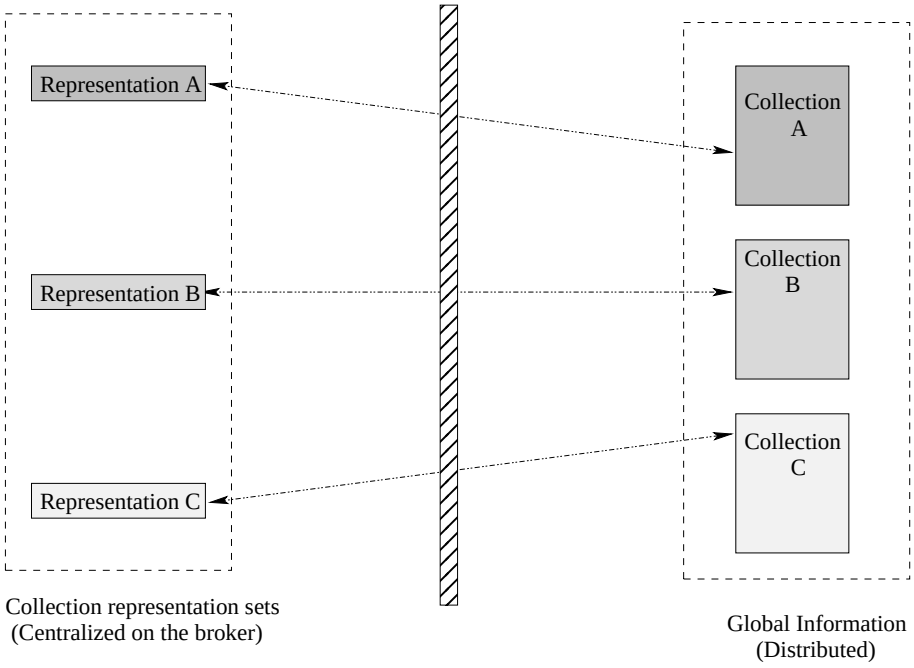

Fig. 2.1 *The representation of distributed collections on a central broker. The broker stores a subset of the global information, available at collections, centrally.*

Zobel, 1997; Yuwono and Lee, 1997]. In such a scenario, the broker has extensive knowledge about each collection and can effectively calculate the score of collections. In uncooperative federated search, collections do not publish their representation sets. Therefore, the broker typically downloads a limited number of documents from each collection and uses these as the representation set [Callan et al., 1999; Callan and Connell, 2001].

This chapter provides an overview of the previous work on collection representation.

2.1 Representation sets in cooperative environments

In cooperative environments, collections may provide the broker with comprehensive information about their searchable documents.

In the STARTS protocol [Gravano et al., 1997], the broker stores

several types of *source metadata*, that are used for server selection and other purposes such as query mapping and result merging. Some of the source metadata attributes used by the STARTS protocol are: score range, stopword list, supported fields and sample results. In addition to server selection metadata, the information about the query language of each server is also available in the representation sets. The query language defined by the STARTS protocol consists of two main components: *filter expression*, and *ranking expression*. The filter expression, is used to narrow down search to documents that are more likely to be relevant on each server. Using the filter expressions, the user can specify the fields that the query has to match in each document in the final results (e.g. `title ‘‘harry potter’’, author ‘‘J.K. Rowling’’`). The ranking expression provides information about the importance of different sections of a document for ranking (e.g. `body-of-text ‘‘treasure’’`). Green et al. [2001] later enhanced STARTS with XML features in a more sophisticated protocol called SDARTS. The new protocol is suitable for federated search over multiple XML collections.

The comprehensiveness of information stored in collection representation sets varies depending on the degree of *cooperation* between collections, and the complexity of the search protocol. For instance, Gravano et al. [1994b;a; 1999] stored document frequency and term weight information in the representation sets. In another work by Callan et al. [1995], the broker stores the document frequency information of the terms for each collection. The collection representation sets used by Meng et al. [2001] contain the *adjusted maximum normalized weights* that are computed according to the global inverse document frequency, and the maximum term frequency values in each collection. Similar statistics have been used by Wu et al. [2001] and Yu et al. [1999; 2002]

Yuwono and Lee [1997] stored the number of documents in each collection, and the *collection frequency*[1] of terms in the representation sets of their WISE system. Zobel's *Lexicon Inspection* system [Zobel, 1997] also stored the collection frequency statistics and the number of documents in each collection in representation sets. D'Souza and Thom [1999] proposed *n-term indexing* in which they only store the statistics

[1] The number of collections that contain each term.

about a maximum of n terms per document in representation sets. The *first-n* variant that chooses the first n terms in each document was later adopted by D'Souza et al. [2000].

In heterogenous environments with diverse data types, collection representation sets may contain various metadata to improve other stages of federated search such as collection selection and result merging. For example, Arguello et al. [2009a;b] stored previous vertical-specific query logs in their vertical representation sets. For semi-structured datasets, different fields can be represented by separate metadata. For example, Kim and Croft [2010] stored title, content, date, sender and receiver information separately in the representation set of the email collection they used in their desktop search experiments.

2.2 Representation sets in uncooperative environments

In the absence of cooperation, term statistics are usually approximated by using a number of documents sampled from collections. Next, we discuss different methods for sampling documents from uncooperative collections.

2.2.1 Query-based sampling

Query-based sampling (QBS) [Callan and Connell, 2001; Callan et al., 1999] was proposed for sampling uncooperative environments where the broker does not have access to the complete lexicon statistics of each collection. QBS has been used widely in federated search experiments [Avrahami et al., 2006; Nottelmann and Fuhr, 2003; 2004a; Si and Callan, 2003a; 2004b; 2005b] and can be described as follows:

(1) An initial query is selected and submitted to the collection. The query is usually a single term,[2] selected from words that are likely to return many results.
(2) The top n documents for the query are downloaded. Callan and Connell [2001] have empirically determined that $n = 4$

[2] Craswell et al. [2000], and Shokouhi et al. [2007d] used multi-word queries for sampling.

is an appropriate value for TREC newswire collections.

(3) Sampling continues as long as the stopping criterion has not been met. The stopping criterion is usually defined in terms of the number of documents sampled or the number of sampling queries that have been issued. For example, Callan and Connell [2001] suggested that sampling can stop after downloading 300–500 unique documents. Shokouhi et al. [2006a] later showed that for larger collections, sampling more documents can significantly improve collection selection.

Callan and Connell [2001] reported that the initial query has minimal impact on the quality of final samples. They proposed two major sampling strategies for selecting the sampling queries; *other resource description* (*ord*) and *learned resource description* (*lrd*). The former selects the sampling (probe) queries from a reference dictionary, while the latter selects them from the documents already sampled. Callan et al. [1999] evaluated four strategies for choosing the probe queries (terms) from the sampled documents (based on their document frequencies, collection frequencies, average term frequencies, or randomly).

Overall, the *ord* method produces more representative samples. However, it is not particularly efficient and often chooses many *out of vocabulary* (OOV) terms that do not return any document from the collection. Among the discussed strategies, using average term frequency and *random selection* have been suggested to have the best trade-off between efficiency and effectiveness.

Craswell [2000], and Shokouhi et al. [2007d] employed query-logs for sampling, and showed that samples produced by log-based queries can lead to better collection selection and search effectiveness.

Callan et al. [2000] investigated the effects of query-based sampling on different collection selection algorithms. They compared CORI [Callan et al., 1995], GlOSS [Gravano et al., 1994a], and CVV [Yuwono and Lee, 1997]. It was observed that the performance of GlOSS and CVV decreases dramatically when using incomplete representation sets (sampled documents) while the performance of CORI remained almost unchanged. Monroe et al. [2002] studied the effectiveness of query-based sampling for sampling web collections and showed

that QBS can produce effective representation sets.

Traditional query-based sampling has drawbacks. The sampling queries are selected randomly and thus they may not always return sufficient number of answers, which can make the sampling process inefficient. Furthermore, samples of 300–500 documents may not always be sufficiently representative of the corresponding collections. Hence, *adaptive sampling* techniques have been proposed to address these issues.

Adaptive sampling. The idea of adaptive sampling was first applied by Shokouhi et al. [2006a]. The authors adaptively chose the sample size for each collection according to the rate of visiting new vocabulary in sampled documents. Baillie et al. [2006a] suggested that sampling should stop when the new sampled documents do not download a large number of unvisited terms that are likely to appear in future queries. They divided the sampling process into multiple iterations. At each iteration, n new documents are added to the current samples. The impact of adding new sampled documents for answering a group of queries Q is estimated as:

$$\phi_k = l(\hat{\theta}_k, Q) - l(\hat{\theta}_{k-1}, Q) = \log\left(\frac{P(Q|\hat{\theta}_k)}{P(Q|\hat{\theta}_{k-1})}\right) \tag{2.1}$$

where the likelihood $l(\hat{\theta}_k, Q)$ of generating the terms of training queries Q by the *language model* [Ponte and Croft, 1998] of a collection sample $\hat{\theta}$ is calculated as below:

$$P(Q|\hat{\theta}_k) = \prod_{i=1}^{n}\prod_{j=1}^{m} P(t = q_{ij}|\hat{\theta}_k) \tag{2.2}$$

Here, $t = q_{ij}$ is the jth term of the ith query in a representative query log. $P(t|\hat{\theta}_k)$ is the probability of visiting the term t by picking a random term from the language model ($\hat{\theta}$) of the sampled documents at the kth iteration.[3] The length of the longest training query and the size

[3] The general procedure of estimating language from a document or a collection of documents can be found elsewhere [Lafferty and Zhai, 2001; Ponte and Croft, 1998].

of the query set are respectively specified by m and n. Sampling stops when the value for ϕ_k becomes less than a pre-defined threshold. This approach is reliant on a set of queries or corpus that is representative of the future information needs of the users of the system.

Caverlee et al. [2006] investigated three stopping criteria for adaptive sampling of uncooperative collections:

- *Proportional document ratio (PD):* In this scenario, the number of documents sampled from each collection varies according to the estimated collection sizes.[4] In PD, the same proportion of documents are sampled from each collection.
- *Proportional vocabulary ratio (PV):* In this approach, the broker estimates the vocabulary size of each collection, and downloads the same vocabulary proportion from each collection by sampling.
- *Vocabulary growth (VG):* The vocabulary growth technique aims to download the highest number of distinct terms across all collection representation sets. When there is a maximum limit for the number of documents that can be downloaded by the broker, VG downloads more documents from the collections that return more new terms.

Caverlee et al. [2006] showed that PD and PV produce more representative samples and can significantly improve the effectiveness of collection selection. However, the authors only reported their results for the CORI collection selection method [Callan et al., 1995]. The impact of their suggested methodologies on the performance of more effective collection selection techniques is unclear.

2.2.2 Improving incomplete samples

A few approaches have been suggested for improving the quality of collection samples. Ipeirotis and Gravano [2002] proposed *focused probing* based on the following principle: queries related to a topical category

[4] A summary of collection size estimation techniques is provided in Section 2.3.

are likely to retrieve documents related to that category. Focused probing applies a trained rule-based document classifier such as RIPPER [Cohen, 1996] for sampling. The probe queries for sampling are extracted from the classification rules. For example, if the classifier defines (*Basketball*→*Sport*)—that is documents containing "basketball" are related to sport—and then "basketball" is used as the query and the returned documents are classified as sport-related. As sampling continues, the probe queries are selected according to more specific classification rules. This allows collections to be classified more accurately according to the *specificity* and *coverage* of the documents they return from each class. In addition, the generated samples are argued to be more representative [Ipeirotis and Gravano, 2008] as they can reflect the topicality of collections more effectively.

There are often many terms in collections that occur in only a few documents, and thus these terms often do not appear in the samples downloaded by query-based sampling or focused-probing. The *Shrinkage* technique [Ipeirotis, 2004; Ipeirotis and Gravano, 2004] has been proposed to solve this problem and to improve the comprehensiveness of collection samples. The shrinkage method is based on the assumption that topically related collections share the same terms. Collections are first classified under a set of categories. The vocabulary statistics of each sample are then extended using the samples of other collections in the same category.

In Q-pilot [Sugiura and Etzioni, 2000] the description of each search engine is created by combining the outputs of three methods; front-page, back-link, and query-based sampling. The first method extracts the terms available on the query interface of search engines, while the second method generates a summary from the contents of web pages that have links pointing to the search engine front page. A similar strategy has been used by Lin and Chen [2002] to construct the representation sets of hidden web search engines. In HARP [Hawking and Thomas, 2005], the representation set of each collection consists of the *anchor-text* [Craswell et al., 2001] of URLs available in a crawled repository that are targeting that collection. Hedley et al. [2004a;b;c;d] suggested a two-phrase sampling technique (2PS) to produce more representative samples from the hidden web collections. The 2PS

method is similar to traditional query-based sampling but differs in a few aspects. In 2PS, the initial query is selected from the collection search interface, while in query-based sampling the first query is a frequently-used term or a term extracted from a dictionary. In addition, 2PS detects the templates of web pages and does not select HTML markup terms from the templates for sampling. Instead, it uses the terms that are selected from the text content of web pages. Such terms are more likely to return representative documents from collections.

The size of collection summaries can cause efficiency problems on brokers with space constraints. Lu and Callan [2002], and Shokouhi et al. [2007d] proposed several pruning strategies for reducing the size of collection summaries with minimal impact on final search effectiveness.

A more brief summary of the techniques described above has been provided by Aksoy [2005].

2.3 Estimating the collection size

The size of a collection is used in many collection selection methods, such as ReDDE [Si and Callan, 2003a], KL-Divergence [Si et al., 2002], and UUM [Si and Callan, 2004b], as an important parameter for ranking collections. In an uncooperative environment, information regarding the size of collections is not usually available. Hence a broker must estimate the collection size. This section summarizes current techniques for estimating the size of collections in uncooperative environments.

Capture-recapture. Using estimation as a way to identify a collection's size was initially suggested by Liu et al. [2001], who introduced the *capture-recapture* method for federated search. This approach is based on the number of overlapping documents in two random samples taken from a collection: assuming that the actual size of collection is N, if we sample a random documents from the collection and then sample (after replacing these documents) b documents, the size of collection can be estimated as $\hat{N} = \frac{ab}{c}$, where c is the number of documents common to both samples. However, Liu et al. [2001] did not discuss how random samples can be obtained.

Multiple Capture-recapture. The capture-recapture technique originates from ecology, where a given number of animals is captured, marked, and released. After a suitable time has elapsed, a second set is captured; by inspecting the intersection of the two sets, the population size can be estimated.

This method can be extended to a larger number of samples to give multiple capture-recapture (MCR) [Shokouhi et al., 2006b]. Using T samples of size k, the total number of pairwise duplicate documents D should be:

$$D = \binom{T}{2} E(X) = \frac{T(T-1)}{2} E(X) = \frac{T(T-1)k^2}{2N} \tag{2.3}$$

Here, N is the size of population (collection). By gathering T random samples from the collection and counting duplicates within each sample pair, the expected size of collection is:

$$\hat{N} = \frac{T(T-1)k^2}{2D} \tag{2.4}$$

Although the sample size (k) is fixed for all collections in the above equations, Thomas [2008a] showed that this is not necessary, and MCR can be generalised to use non-uniform sample size values for different collections.

Schumacher-Eschmeyer method (Capture-history). Capture-recapture is one of the oldest methods used in ecology for estimating population size. An alternative, introduced by Schumacher and Eschmeyer [1943], uses T consecutive random samples with replacement, and considers the *capture history* [Shokouhi et al., 2006b]. Here,

$$\hat{N} = \frac{\sum_{i=1}^{T} K_i {M_i}^2}{\sum_{i=1}^{T} R_i M_i} \tag{2.5}$$

where K_i is the total number of documents in sample i, R_i is the number of documents in the sample i that were already marked, and M_i is the number of marked documents gathered so far, prior to the most recent sample. Capture-history has been shown to produce more

accurate size estimates compared to MCR [Shokouhi et al., 2006b; Xu et al., 2007].

Sample-resample. An alternative to the capture-recapture methods is to use the distribution of terms in the sampled documents, as in the *sample-resample* (SRS) method [Si and Callan, 2003a].

Assuming that QBS [Callan and Connell, 2001] produces good random samples, the distribution of terms in the samples should be similar to that in the original collection. For example, if the document frequency of a particular term t in a sample of 300 documents is d_t, and the document frequency of the term in the collection is D_t, the collection size can be estimated by SRS as $\hat{N} = \frac{d_t D_t}{300}$.

This method involves analyzing the terms in the samples and then using these terms as queries to the collection. The approach relies on the assumption that the document frequency of the query terms will be accurately reported by each search engine. Even when collections do provide the document frequency, these statistics are not always reliable [Anagnostopoulos et al., 2005].

Sample-resample and capture-recapture methods assume that documents downloaded by query-based sampling can be regarded as random samples. However, Shokouhi et al. [2006b] showed that the assumption of randomness in QBS is questionable.

Other size estimation methods. Capture-history and other capture-recapture methods assume that all documents have the same probability of being captured. Xu et al. [2007] argued that the probability of capture depends on other parameters such as document length and PageRank [Brin and Page, 1998]. The authors proposed *Heterogeneous capture (HC)* that uses a logistic model to calculate the probability of capture for each document, and can produce better estimations compared to other capture-recapture methods. A similar idea was also suggested by Lu [2008].

The work by Bharat and Broder [1998a] is perhaps the earliest study published on estimating the size of text collections (web in their case) by sampling. The authors tried to obtain random samples from search

engines by submitting random queries and selecting random URLs from the returned results. They used the overlap in samples to estimate the size of the web, and the rate of overlap between the indexes of web search engines.

Bar-Yossef and Gurevich [2006] proposed a *Pool-based sampler* that uses rejection sampling to generate random samples from search engines. To eliminate the search engine ranking bias, the authors reject queries that *underflow* or *overflow*. That is, ignoring queries that return too few or too many documents. As in Bharat and Broder [1998a], the authors leveraged a pool of queries to sample documents from search engines. However, instead of selecting the sampling queries uniformly at random (as in Bharat and Broder [1998a]), the probe queries are selected according to their *cardinality*. The cardinality of a query is defined as the number of answers that it returns from the search engine. Since the cardinality values may not be publicly available, the authors first use uniform sampling and then apply Monte Carlo methods to simulate sampling based on cardinality.

Bar-Yossef and Gurevich [2006] also presented a *Random walk sampler* that performs a random walk on a document-query graph. Two documents are connected in the graph if they both match at least one common query. Given a document d, the next query q is selected from a pool of documents that return d (overflowing and underflowing queries are rejected). The next document is picked randomly from the set of results returned for q. Both Pool-based and Random walk samplers are shown to guarantee producing near-uniform samples, while the latter has been found to be less efficient but more accurate [Thomas and Hawking, 2007].

Proposed by Thomas and Hawking [2007], *Multiple queries sampler* runs several queries with a large cutoff and then selects a random sample from the union of all documents returned for probe queries. The authors also compared the efficiency and accuracy of several size estimation methods on a set of personal metasearch testbeds.

Other work for estimating the size of text collections via queries includes [Bar-Yossef and Gurevich, 2007; Broder et al., 2006; Gulli and Signorini, 2005; Henzinger et al., 2000; Karnatapu et al., 2004; Lu et al., 2008; Lu and Li, 2009].

2.4 Updating collection summaries

The contents of collections may change in different ways. For example, documents can be added, deleted or updated within a collection. Out-of-date results can have a negative impact on how the user perceives the search engine. Therefore, search engines constantly update their indexes by crawling fresh documents to reduce inconsistencies between their index and web documents [Cho and Garcia-Molina, 2003].

One of the advantages of federated search compared to centralized information retrieval is that the problem of fresh data is minimized. The queries are submitted directly to collections that are assumed to contain the latest version of documents. However, collection updates must be reflected in the representation sets, otherwise, collection may be selected based on their old data. Ipeirotis et al. [2005] showed how the vocabulary of collection representation sets can become less representative over time when it is not maintained through periodic updates. Shokouhi et al. [2007a] showed that large collections require more frequent updates.

2.5 Wrappers

An important but often ignored research topic in federated search literature is generating *wrappers* for collections. Wrappers are essential for collection representation, as they define the interaction methods with individual collections. There are at least three major issues in wrapper generation: (1) collection detection, (2) collection connection and query mapping, and (3) extracting records from search result pages.

Most existing federated search systems are given a set of collections to search in a close domain. However, in an open domain such as the web, collections with search engines may dynamically appear or disappear. Therefore, it is an important task to automatically detect collections with independent search interfaces. Cope et al. [2003] showed how search engines interfaces can be identified based on their HTML content. In particular, a decision tree algorithm is trained with a set of features from HTML markup language and some human judgments, which can be used to identify new collections with search engines. More

recently, Barbosa and Freire [2007] utilized a hierarchical identification method that generates accurate identification results by partitioning the space of the features and choosing learning classifiers that best fit in each partition.

Federated search systems need to establish connections with local collections for passing user queries. Previous research mostly focuses on full text search engines, which often utilize HTTP (HyperText Transfer Protocol) for creating connections and receiving queries and sending results. The search engines of different collections often use text search boxes with different HTTP request methods such as GET or POST. It is often not difficult to manually establish search connections with an unstructured full text search engine (e.g., via http link) or use simple rule-based methods.

Extracting result records from the answer page returned by a search engine is relatively difficult due to the diversity in result presentation styles. Some federated search systems such as FedLemur [Avrahami et al., 2006] generate result templates and extract search results with manually compiled rules. RoadRunner [Crescenzi et al., 2001] and EXALG [Arasu and Garcia-Molina, 2003] treat webpages as individual strings. RoadRunner generates a result template by comparing a couple of result pages, while EXALG analyzes a set of webpages. Omini [Buttler et al., 2001] and MDR [Liu et al., 2003] treat webpages as trees of HTML tags. They assume that result records are located in data-rich sub-trees, where a separator (i.e., an HTML tag) is used to segment the records. For example, the MDR approach identifies multiple similar generalized nodes of a tag node and the generalized nodes are further checked for extracting one or multiple data records. The extension of the MDR and ViNTs [Zhao et al., 2005] approaches utilize both HTML tag information and visual information to improve the accuracy of identifying data records. Furthermore, the work in [Zhao et al., 2007] automatically builds a search result template with different fields such as title, snippet and URL from identified search results. While most existing methods assume that results are presented in a single section, Zhao et al. [2006] consider multiple sections in search results (e.g., clustered results). More recently, Liu et al. [2010] proposed

a new vision-based approach for extracting data records that is not specific to any web page programming language.

2.6 Evaluating representation sets

In uncooperative environments where collections do not publish their index statistics, the knowledge of the broker about collections is usually limited to their sampled documents. Since downloaded samples are incomplete, it is important to test whether they are sufficiently *representative* of their original collections.

Collection representation sets usually consist of two types of information: vocabulary and term-frequency statistics. Callan and Connell [2001] proposed two separate metrics that measure the accuracy of collection representation sets in terms of the vocabulary correspondence and frequency correlations, as we now describe.

Measuring the vocabulary correspondence (ctf ratio). The terms available in sampled documents can be considered as a subset of all terms in the original collection. Therefore, the quality of samples can be measured according to their coverage of the terms inside the original collections. Callan and Connell [2001] defined the *ctf ratio* as the proportion of the total terms in a collection that are covered by the terms in its sampled documents. They used this metric for measuring the quality of collection representation sets. For a given collection c, and a set of sampled documents S_c, the ctf ratio can be computed as:

$$\frac{\sum_{t \in S_c} f_{t,c}}{\sum_{t \in c} f_{t,c}} \tag{2.6}$$

where $f_{t,c}$ represents the frequency of term t in collection c. For example, suppose that collection c includes only two documents. Now assume that the first document only contains two occurrences of "computer" and six occurrences of "science"; and the second document consists of two terms: "neural" and "science" each occurring only once. In total, there are three unique terms in collection c, and the cumulative collection frequency value is 10. If only the first document is sampled from the collection, the proportion of the total terms that are present in the

sample is $\frac{9}{10}$ or 90%. Therefore, the impact of downloading a frequent term on the final ctf is greater than the impact of downloading another term that is less frequent, although it may be more representative.

Spearman rank correlation coefficient (SRCC). Callan and Connell [2001] suggested that the downloaded terms can be ranked according to their document frequency values in both the samples and the original collection. The correlation of these two rankings can be computed using a statistical method such as the *Spearman rank correlation coefficient* [Press et al., 1988]. The stronger the correlation is, the more similar are the term distributions in the samples and the original collection. In other words, samples whose terms have a strong correlation with the original index are considered as representative.

SRCC measures the intersection in vocabulary between collection and representation. Therefore, when new terms are added, this often weakens the correlation, and decreases the stability of term rankings. Baillie et al. [2006c;b] showed that SRCC is not always robust and reliable because of this drawback.

df1. Monroe et al. [2000] suggested that the proportion of terms with document frequency of one ($df = 1$) can be used for measuring the completeness of samples. They also suggested that the rate of growth of terms with $df = 1$ in the documents downloaded by query-based sampling can be used to determine the termination point of sampling. That is, downloaded documents are representative enough once the number of $df = 1$ terms in two consequent samples becomes less than a certain threshold.

Kullback-Leibler divergence (KL). Another approach for evaluating the accuracy of collection representation sets is to compare their language models with that of the original collections [Baillie et al., 2006c;b; Ipeirotis and Gravano, 2004; Ipeirotis et al., 2005]. Therefore, a KL-Divergence method [Kullback, 1959] can be used for comparing the term distribution (language model) of a collection with that of its sampled documents:

$$KL(\hat{\theta}_{S_c}|\hat{\theta}_c) = \sum_{t \in c} P(t|\hat{\theta}_{S_c}) \log \frac{P(t|\hat{\theta}_{S_c})}{P(t|\hat{\theta}_c)} \tag{2.7}$$

Here, $\hat{\theta}_{S_c}$ and $\hat{\theta}_c$ respectively represent the language models of sampled documents and the original collection, and $P(t|\hat{\theta})$ is the probability of visiting the term t, if it is randomly picked from a language model $\hat{\theta}$. The KL values can range from 0 to infinity, where $KL = 0$ indicates that the two language models are identical. Compared to the metrics discussed previously, KL has been shown to be more stable and precise [Baillie et al., 2006c;b].

Topical KL. Baillie et al. [2009] proposed a topic-based measure for evaluating the quality of collection representation sets (sampled documents). For each collection c, the authors utilized latent Dirichlet allocation [Blei et al., 2003] techniques to estimate the set of k term distributions that represent the major *themes* covered by the collection. For each generated topic T, the authors compared its term distribution $\hat{\theta}_T$ with the language model of collection $\hat{\theta}_c$ and its sampled documents $\hat{\theta}_{S_c}$. That is:

$$p(\hat{\theta}_T|\hat{\theta}_c) = \frac{1}{|c|} \sum_{d \in c} p(\hat{\theta}_T|d) \tag{2.8}$$

$$p(\hat{\theta}_T|\hat{\theta}_{S_c}) = \frac{1}{|S_c|} \sum_{d \in S_c} p(\hat{\theta}_T|d) \tag{2.9}$$

where, S_c is the set of sampled documents from collection c. In the final stage, the topical KL divergence between the a collection and its sampled documents is used as a measure of quality for the representation set, and is computed as follows:

$$TopicalKL(\hat{\theta}_c|\hat{\theta}_{S_c}) = \sum_{T \in K} p(\hat{\theta}_T|\hat{\theta}_c) \log \frac{p(\hat{\theta}_T|\hat{\theta}_c)}{p(\hat{\theta}_T|\hat{\theta}_{S_c})} \tag{2.10}$$

Here, K denotes the set of topics (term distributions) generated by LDA for collection c. In summary, Equation 2.10 measures the quality of sampled documents in terms of covering the major themes (topics) in the collection.

Predictive likelihood (PL). Baillie et al. [2006a] argued that the *predictive likelihood* [DeGroot, 2004] of user information needs can be used as a metric for evaluating the quality of collection samples. The PL value of sampled documents verifies how representative it is with respect to the information needs of users. In contrast to the previous methods that measure the completeness of samples compared to the original index, PL measures the quality of samples for answering queries. Collection representation sets are compared against a set of user queries. Representation sets that have high coverage of query-log terms produce large PL values, and are more likely to satisfy user information needs by routing queries to suitable collections. For a query log described as set of n queries $Q = \{q_{i,j} : 1, ..., n; 1, ..., m\}$, where $q_{i,j}$ represents the jth term in the ith query, the predictive likelihood of the language model of a sample S can be computed as follows:

$$PL(Q|\hat{\theta}_S) = \prod_{i=1}^{n} \prod_{j=1}^{m} P(t = q_{i,j}|\hat{\theta}_S) \tag{2.11}$$

where $P(t = q_{i,j}|\hat{\theta}_S)$ is the probability of visiting the term t from the query log Q in the language model of sample S.

Precision. The evaluation techniques described so far all measure the representation quality in isolation of the core retrieval task. Alternatively, the quality of collection representation sets can be measured by their impact on collection selection [Caverlee et al., 2006; Ipeirotis and Gravano, 2008] or final downstream performance [Shokouhi et al., 2006a; 2007d].

2.7 Summary

Collection representation sets are the main information source used by the broker for collection selection and result merging. The degree of comprehensiveness for collection representation sets often depends on the level of cooperation between collections and the broker.

Early work in collection representation mainly focused on cooperative environments, and utilized manually-generated metadata. The STARTS [Gravano et al., 1997] protocol and its variants require each

collection to provide the query language, ranking method, and important corpus statistics. This type of solutions work well with full cooperation from available collections. However, they are not appropriate for uncooperative environments.

Query-based sampling methods [Callan and Connell, 2001] have been proposed to obtain information such as term frequency statistics in uncooperative federated search environments. These methods send probe queries to get sample documents from individual collections in order to build approximated collection representation sets. Different variants of query-based sampling techniques use different types of probe queries and stopping criteria for generating the representation sets.

The sizes of available collections have been used in many collection selection algorithms as an important feature for ranking collections. Different techniques have been proposed to automatically estimate the collection size in uncooperative environments. The majority of these methods analyze a small number of sampled documents from collections to estimate their size.

It is important to keep collection representations up to date in order to make accurate selection decisions. Therefore, updating policies should obtain reasonably accurate collection representation given limited communication and computing resources. We provided an overview of prior research on updating policies for collection representation sets.

An important but often ignored issue in federated search is building wrappers to automatically send queries and extract result records from individual collections. It is often not difficult to identify the method of sending queries for searching full text search engines. However, the task of extracting result records is more complicated, and several methods have been proposed to utilize features such as HTML tags and visual contents to achieve this goal.

Several metrics have been proposed to evaluate the quality of collection representations. The *ctf ratio* [Callan and Connell, 2001] evaluates vocabulary coverage by sampled documents. The *SRCC* metric [Callan and Connell, 2001] measures the consistency of term rankings with respect to document frequency in sampled documents and the original collection. The *KL divergence* metric [Baillie et al., 2009; Ipeirotis and Gravano, 2004; Ipeirotis et al., 2005] treats the distributions of terms

(or topics) in the sampled documents and the original collection as two probabilistic language models, and considers the the distance between the language models as a sample quality measure. The *PL* technique [Baillie et al., 2006b] takes a further step by measuring the quality of sampled documents for answering user queries. Collection representation sets can be also evaluated according to their impact on collection selection and final performance [Caverlee et al., 2006; Ipeirotis and Gravano, 2008; Shokouhi et al., 2006a; 2007d].

In the next chapter, we discuss the previous work on collection selection.

3

Collection selection

The first step after receiving a query in federated search is to select suitable collections. Once a query is entered, the broker ranks collections and decides which collections to select and search (Figure 3.1). Due to resource constraints such as bandwidth limits, it is usually not feasible to search all collections. Therefore, the broker often selects only a subset of available collections that are likely to return relevant documents.

This chapter provides an overview of previous work in the area of collection selection.

3.1 Lexicon-based collection selection

Early collection selection strategies treat collections as a big bag of words and rank them according to their lexicon similarity with the query [Baumgarten, 1997; 1999; Callan et al., 1995; de Kretser et al., 1998; D'Souza and Thom, 1999; D'Souza et al., 2004a;b; Gravano, 1997; Gravano et al., 1997; 1999; Yuwono and Lee, 1997; Xu and Callan, 1998; Zobel, 1997]. In these techniques, the broker calculates the similarity of the query with the representation sets by using the detailed lexicon

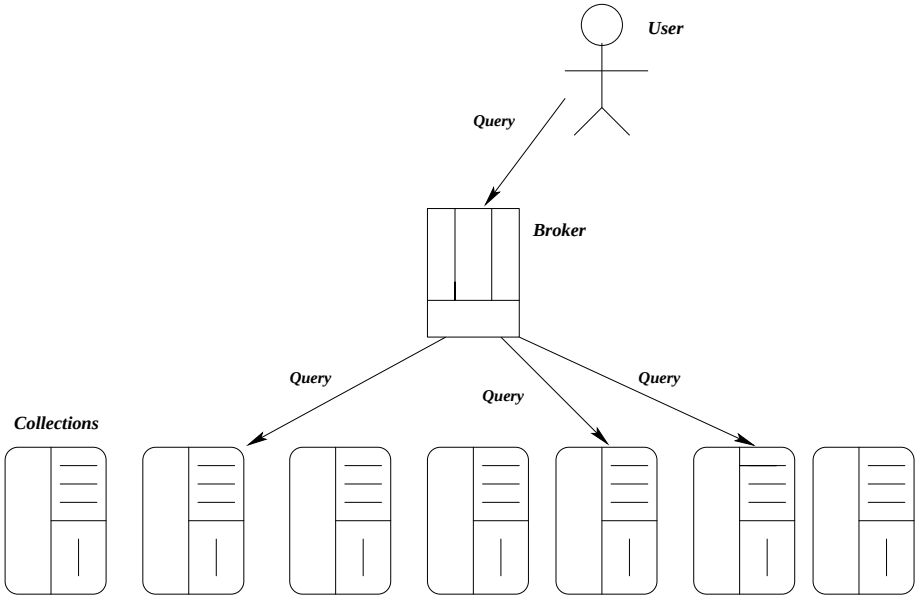

Fig. 3.1 *The collection selection process. The broker receives the user query and selects the subset of available collections that it considers most likely to return relevant documents.*

statistics of collections. In uncooperative environments where collections do not share their lexicon information, these statistics can be approximated based on their sampled documents (See Chapter 2 for an overview of sampling techniques).

GlOSS. The initial version of GlOSS [Gravano et al., 1994a]—also known as bGlOSS—only supports Boolean queries. In bGlOSS, collections are ranked based on their estimated number of documents that satisfy the query. The bGlOSS method was designed for cooperative environments and thus, the collection size values and term frequency information were assumed to be available for the broker. Overall, bGlOSS estimates the number of documents containing all the m query terms as:

$$\frac{\prod_{j=1}^{m} f_{t_j,c}}{|c|^{m-1}} \tag{3.1}$$

Collections are ranked according to their estimated number of answers

for the query. In the vector-space version of GlOSS (vGlOSS) [Gravano et al., 1999], collections are sorted according to their *goodness* values, defined as:

$$Goodness(q, l, c) = \sum_{d \in Rank(q,l,c)} sim(q, d) \tag{3.2}$$

where $sim(q, d)$ is the Cosine similarity [Salton and McGill, 1986; Salton et al., 1983] of the vectors for document d and query q. In other words, the goodness value of a collection for a query is calculated by summing the similarity values of the documents in the collection. To avoid possible noise produced by low-similarity documents, vGlOSS uses a similarity threshold l.

CORI. The CORI collection selection algorithm [Callan, 2000; Callan et al., 1995] calculates belief scores of individual collections by utilizing a Bayesian inference network model with an adapted Okapi term frequency normalization formula [Robertson and Walker, 1994]. CORI is related to the INQUERY ad-hoc retrieval algorithm [Turtle, 1991; Turtle and Croft, 1990]. In CORI, the belief of the ith collection associated with the word t, is calculated as:

$$T = \frac{df_{t,i}}{df_{t,i} + 50 + 150 \times cw_i / avg\_cw} \tag{3.3}$$

$$I = \frac{log(\frac{N_c + 0.5}{cf_t})}{log(N_c + 1.0)} \tag{3.4}$$

$$P(t|c_i) = b + (1 - b) \times T \times I \tag{3.5}$$

where $df_{t,i}$ is the number of documents in the ith collection that contain t; cf_t is the number of collections that contain t; N_c is the total number of available collections; cw_i is the total number of words in the ith collection, and avg_cw is the average cw of all collections. Finally, b is the default belief, which is usually set to 0.4. The belief $P(Q|c_i)$ is used by the CORI algorithm to rank collections. The most common way to

calculate the belief $P(Q|c_i)$ is to use the average value of the beliefs of all query terms, while a set of more complex query operators are also available for handling structured queries [Callan, 2000].

CVV. *Cue-validity variance* (CVV) was proposed by Yuwono and Lee [1997] for collection selection as a part of the WISE index server [Yuwono and Lee, 1996]. The CVV broker only stores the document frequency information of collections, and defines the goodness of a given collection c for an m-term query q as below:

$$Goodness(c, q) = \sum_{j=1}^{m} CVV_j \cdot df_{j,c} \tag{3.6}$$

where $df_{j,c}$ represents the document frequency of the jth query term in collection c and CVV_j is the variance of cue-validity (CV_j) [Goldberg, 1995] of that term. $CV_{c,j}$ shows the degree that the jth term in the query can distinguish collection c from other collections and is computed as:

$$CV_{c_i,j} = \frac{\frac{df_{j,c_i}}{|c_i|}}{\frac{df_{j,c_i}}{|c_i|} + \frac{\sum_{k\neq i}^{N_c} df_{j,c_k}}{\sum_{k\neq i}^{N_c} |c_k|}} \tag{3.7}$$

Here, $|c_k|$ is the number of documents in collection c_k and N_c is the total number of collections. The variance of cue-validity CVV_j can be calculated as:

$$CVV_j = \frac{\sum_{i=1}^{N_c} (CV_{c_i,j} - \overline{CV_j})^2}{N_c} \tag{3.8}$$

where $\overline{CV_j}$ is the average $CV_{c_i,j}$ over all collections and is defined as below:

$$\overline{CV_j} = \frac{\sum_{i=1}^{N_c} CV_{c_i,j}}{N_c} \tag{3.9}$$

Other lexicon-based methods. Several other lexicon-base collection selection strategies have been proposed; Zobel [1997] tested

four lexicon-based methods for collection selection. Overall, his *Inner-product* ranking function was found to produce better results than the other functions such as the Cosine formula [Baeza-Yates, 1992; Salton and McGill, 1986]. CSams [Yu et al., 1999; 2002; Wu et al., 2001] uses the global frequency, and maximum normalized weights of query terms to compute the ranking scores of collections.

Si et al. [2002] proposed a collection selection method that builds language models from the representation sets of available collections and ranks collections by calculating the Kullback-Leibler divergence between the query model and the collection models.

D'Souza and Thom [1999] proposed a n-term indexing method, in which a subset of terms from each document is indexed by the broker. For each document, a subset of terms should be provided by collections to the broker. Thus, a high level of cooperation is needed. A comparison between the lexicon-based methods of Zobel [1997], CORI [Callan et al., 1995], and n-term indexing strategies has been presented by D'Souza et al. [2004a], showing that the performance of collection selection methods varies on different testbeds, and reporting that no approach constantly produces the best results.

Baumgarten [1997; 1999] proposed a probabilistic model [Robertson, 1976; 1997] for ranking documents in federated search environments. Sogrine et al. [2005] combined a group of collection selection methods such as CORI and CVV with a latent semantic indexing (LSI) strategy [Deerwester et al., 1990]. In their approach, instead of the term frequency information of query terms, elements of an LSI matrix are used in collection selection equations.

Lexicon-based collection selection techniques are analogous to centralized IR models, but documents are now collections. In these approaches, the document boundaries within collections are removed, which may potentially affect the overall performance of such models [Si and Callan, 2003a].

3.2 Document-surrogate methods

Document-surrogate methods are typically designed for uncooperative environments where the complete lexicon information of collections is

not available. However, these techniques could be also applied in cooperative environments. Document-surrogate methods do not rank collections solely based on the computed similarities of queries and representation sets, but they also use the ranking of sampled documents for collection selection. This is a step away from treating collections as large single documents or vocabulary distributions (as in lexicon-based methods), and somewhat retains document boundaries.[1]

ReDDE. The relevant document distribution estimation (ReDDE) collection selection algorithm [Si and Callan, 2003a] was designed to select a small number of collections with the largest number of relevant documents. To achieve this goal, ReDDE explicitly estimates the distribution of relevant documents across all the collections and ranks collections accordingly.

In particular, the number of documents relevant to a query q in a collection c is estimated as follows:

$$\mathcal{R}(c,q) = \sum_{d \in c} P(\mathcal{R}|d) P(d|c) |c| \tag{3.10}$$

where $|c|$ denotes the number of documents in collection c, and the probability $P(d|c)$ is the generation probability of a particular document d in this collection. In uncooperative federated search environments, ReDDE can utilize different methods described in Chapter 2 to obtain the size estimates. $P(\mathcal{R}|d)$ is the estimated probability of relevance for document d. In uncooperative federated search environments, it is not practical to access all individual documents in available collections. Therefore, ReDDE regards sampled documents as representative, in which case Equation 3.10 can be approximated as:

$$\mathcal{R}(c,q) \approx \sum_{d \in S_c} P(\mathcal{R}|d) \frac{|c|}{|S_c|} \tag{3.11}$$

[1] Unless specified otherwise, we assume that collections and their representation sets only contain text documents. In vertical search environments, it is common to have collections with different media types, and some of the described techniques may not be applicable without modification.

where S_c is the set of sampled documents from collection c. The idea behind this equation is that when one sampled document from a collection is relevant to a query, it is expected that there are about $|c|/|S_c|$ similar documents in the original collections that are also relevant. $P(\mathcal{R}|d)$ represents the probability of relevance of an arbitrary sampled document with respect to q. Calculating the probability of relevance given a query-document pair is a fundamental problem in information retrieval and despite various studies [Ponte and Croft, 1998; Lafferty and Zhai, 2001], it is still an open problem. In ReDDE, the probability of relevance of a document is approximated according to its position in the ranked list of all sampled documents. For this purpose, the broker in ReDDE creates an index of all sampled documents from all collections (CSI). For each query, the broker ranks all sampled documents, and assumes that this ranking approximates the centralized ranking of all documents indexed by all collections (CCI).

ReDDE considers a constant positive probability of relevance (α) for the top-ranked documents in CCI. Formally, this can be represented as below:

$$p(\mathcal{R}|d) = \begin{cases} \alpha & \text{if } r_{CCI}(d) < \beta \sum_i |c_i| \\ 0 & \text{Otherwise} \end{cases} \tag{3.12}$$

Here, $|c_i|$ denotes the number of documents in collection c_i, and β is a percentage threshold, which separates relevant and irrelevant documents. While the optimal value of the threshold may vary from collection to collection, the prior research [Si and Callan, 2003a] set it to 0.003 and obtained robust performance on several datasets. $r_{CCI}(d)$ represents the position of document d in the centralized ranking of all documents from all collections. In federated search environments, the knowledge of the broker about the documents indexed by collections is often very limited and hence, obtaining the CCI ranking may not be practical. Therefore, the broker approximates the CCI ranking by running the query on a centralized index of all sampled documents (CSI), as follows:

$$r_{CCI}(d) = \sum_{d_j : r_{CSI}(d_j) < r_{CSI}(d)} |c|/|S_c| \tag{3.13}$$

where, c represents the collection from which d is sampled, and $|S_c|$ is the number of sampled documents from that collection. Using Equations 3.12 and 3.13, the number of relevant documents in a collection ($\mathcal{R}(c, q)$) can be estimated. The ReDDE algorithm utilizes the estimated distribution of relevant documents to rank all collections, and to select a subset containing the largest number of relevant documents.

$$Goodness(c, q) = \frac{\mathcal{R}(c, q)}{\sum_i \mathcal{R}(c_i, q)} \tag{3.14}$$

ReDDE makes collection selection decisions by analyzing the top-ranked sampled documents and estimating the distribution of relevant documents in collections. Different variants of the ReDDE algorithms have emerged, which weight top-ranked documents and estimate the probability of relevance in different ways. We cover four of these variants (UUM [Si and Callan, 2004b], RUM [Si and Callan, 2005b], CRCS [Shokouhi, 2007a] and SUSHI [Thomas and Shokouhi, 2009]) in more details later in this section. Furthermore, similar algorithms have been developed in the TREC Blog Track,[2] where the task to is select a small number of most relevant blogs for a user query. For instance, Elsas et al. [2008], and Seo and Croft [2008] proposed blog selection algorithms inspired by the same principle to select blogs that contain the largest number of relevant postings.

CRCS. As in ReDDE [Si and Callan, 2003a], the broker in the *centralized-rank collection selection* method (CRCS) [Shokouhi, 2007a] runs the query on a centralized index of all sampled documents (CSI), and ranks collections accordingly. However, in contrast to ReDDE,

[2] The Text REtrieval Conference (TREC) is an international collaboration that provides large datasets to participants for large-scale evaluation of information retrieval systems. More information about the TREC datasets can be found at: `http://trec.nist.gov/data.html`.

CRCS considers different importance for sampled documents according to their ranks. In CRCS, the impact of a sampled document d on the weight of its original collection c is computed according to the position of d in the CSI ranking. In the simplest form (CRCS-LIN), this can be computed linearly as:

$$R(d) = \begin{cases} \gamma - r_{CSI}(d) & \text{if } r_{CSI}(d) < \gamma \\ 0 & \text{otherwise} \end{cases} \tag{3.15}$$

where $r_{CSI}(d)$ represents the impact of document d at the j^{th} position of the results returned by the centralized index of all sampled documents. Parameter γ specifies the number of top-ranked documents in CSI that are considered as relevant, and was set to 50 by Shokouhi [2007a]. The impact of documents decreases linearly according to their ranks. Shokouhi [2007a] also proposed CRCS-EXP, a variant in which the importance of documents drops exponentially as follows:

$$R(d) = \alpha \exp(-\beta \times r_{CSI}(d)) \tag{3.16}$$

Parameters α and β were suggested to be set to 1.2 and 0.28 respectively [Thomas and Shokouhi, 2009]. The remaining is similar to ReDDE; for a given query q, CRCS calculates the goodness (weight) of each collection as:

$$Goodness(c, q) = \frac{|c_i|}{|c^{\max}| \times |S_c|} \times \sum_{d \in S_c} r_{CSI}(d) \tag{3.17}$$

where, $|c|$ is the—estimated—size of collection c. Shokouhi [2007a] normalized the collection sizes by dividing the size of each collection by the size of the largest collection involved ($|c^{\max}|$). The number of documents sampled from collection c is represented by $|S_c|$. Overall, the final score of each collection is calculated by summing up the impact values for its sampled documents.

The exponential version of CRCS (CRCS-EXP) has been reported to produce slightly better results compared to the linear form (CRCS-LIN) [Shokouhi, 2007a; Thomas, 2008b].

SUSHI. Most of the collection selection techniques described so far assume fixed cutoff values. That is, the number of collections that are selected for all queries is the same. Thomas and Shokouhi [2009] relaxed this assumption in SUSHI. The authors fitted several curves to the score distribution of sampled documents in order to verify the number of collections that should be selected for a query. The authors showed that SUSHI can achieve comparable performance to ReDDE and CRCS, while selecting fewer collections.

UUM, RUM. The ReDDE algorithm follows a high-recall goal to select a small number of collections with the largest number of relevant documents. However, the high-recall goal may not be preferred in all federated search applications. For example, for a federated document retrieval application, the main goal might be high precision and maximizing the number of relevant documents in the top part of the final merged ranked lists. The *unified utility maximization framework* (i.e., UUM) was proposed by Si and Callan [2004b] to adjust the goal of collection selection to maximize the utility of different types of applications (e.g., high recall or high precision).

Compared to ReDDE, the unified utility maximization framework provides a more formal method for estimating the probabilities of relevance of documents in distributed collections with the cost of requiring training information. UUM first builds a logistic transformation model using a small number of training queries that maps the centralized document scores from CSI to the corresponding probabilities of relevance. In the second stage, UUM estimates the probabilities of relevance of all (mostly unseen) documents in collections using the sampled document scores and deploying the trained mapping function. Finally, based on these probabilities, collections are ranked by solving different utility maximization problems according to the high-precision goal or the high-recall goal depending on the application.

Similar to all collection selection techniques described so far, UUM makes a strong assumption that all the collections are using effective retrieval models. However, collections may be associated with ineffective retrieval models in many real world applications (e.g., [Avrahami et al., 2006]). Hence, ignoring the search engine effectiveness factor can seri-

ously degrade the performance of collection selection in practice. The *returned utility maximization* (i.e, RUM) method was proposed by Si and Callan [2005b] to address this issue. The RUM method measures the effectiveness of collection retrieval models by first sending a small number of sample queries and retrieving their top-ranked documents. RUM learns rank mapping models by investigating the consistency between the ranked lists of individual collections, and the corresponding lists generated by an effective centralized retrieval algorithm on the same set of documents. In the collection selection stage, collections are ranked according to the number of relevant documents that they are expected to return.

DTF. The *decision-theoretic framework* (DTF) [Fuhr, 1996; 1999a;b] aims to minimize the typical costs of collection selection such as time and cost, while maximizing the number of relevant documents retrieved. As in UUM, the search effectiveness of collections can be learned by using a set of training queries in advance.

DTF was initially suggested as a promising method for selecting suitable collections. However the method had not been tested in federated search environments until Nottelmann and Fuhr [2003] showed that the effectiveness of DTF can be competitive with that of CORI for long queries. However, for short queries, DTF is usually worse than CORI. DTF and CORI were later combined in a single framework [Nottelmann and Fuhr, 2004a]. The hybrid model still produced poorer results than CORI for shorter queries, but competitive results for longer queries.

DTF requires a large number of training queries, but has one of the most solid theoretical models among available collection selection techniques. It combines costs (monetary, network) along with relevance into a decision-theoretic framework, and has been used in a few real-world federated retrieval applications such as the MIND project [Berretti et al., 2003; Nottelmann and Fuhr, 2004b;c].

The document-surrogate methods discussed in this section assume that sampled documents from available collections are comparable (e.g., through document retrieval scores). This assumption may be problematic when collections contain information in different media

(e.g., image or video). This was a key motivation for introducing more sophisticated supervised techniques for collection selection that we will cover in the next section.

3.3 Classification (or clustering)-based collection selection

The *query clustering* techniques often identify a set (or a cluster) of most similar training queries with respect to a testing query, and model the distribution of relevant documents by analyzing the information learned from the training queries. Voorhees et al. [1995] proposed two techniques for learning the number of documents that should be retrieved from each collection for a query. In their first approach for *modeling relevant document distribution* (MRDD), the authors learn the topical relevance of collections by sending them a number of training queries and analyzing the number of relevant documents returned by each collection. In the testing phase, each query is compared to all the queries in the training set. The set of k most similar training queries (according to the vector space model [Salton and McGill, 1986]) are extracted and then used to predict the performance of each collection for the test query.

In their second approach known as the *query clustering* (QC) [Voorhees et al., 1995], training queries are clustered based on the number of common documents they return from collections. A centroid vector is generated for each cluster and the testing queries are compared against all available centroid vectors. The final weight of each collection is computed according to its performance on the past training queries for the top-ranked clusters.

Similarly, Cetinta et al. [2009] learn from the performance of past training queries to rank collections for unvisited queries. Each query is compared against the set of all past queries. Collections are ranked according to the weighted average of their performance for the most similar past queries. The similarity value for each query pair is computed with respect to a centralized index of sampled documents. Queries that return similar ranked lists are regarded as similar. In addition, the performance of collections for past queries is approximated according to the positions of their documents in a centralized ranking of sampled

documents. Hence, the suggested approach does not rely on relevance judgements for training queries.

In a series of papers, Ipeirotis and Gravano [2002; 2004; 2008] proposed a *classification-aware* technique for collection selection. The authors assign collections to the branches of a hierarchical classification tree according to the terms in their sampled documents. Each branch represents a topical category and may be related to several collections. The term statistics of collection representation sets are propagated to generate the *category summaries*. For collection selection, the broker compares the query against the category summaries, and sends the query to the collections of the categories with the highest scores.

Collection selection can be regarded as a classification (or clustering) problem, in which the goal is to classify (cluster) collections that should be selected for a query. Several techniques have been proposed based on this analogy recently [Arguello et al., 2009b; Diaz and Arguello, 2009]. Arguello et al. [2009b] proposed a classification-based approach for vertical selection trained based on three types of features: (1) query string features, (2) corpus features, and (3) query-log features. They showed that their classification-based collection selection can outperform standard federated search baselines on an aggregated search testbed with 18 verticals. A similar approach was taken for collection selection over three simulated federated search testbeds [Arguello et al., 2009a]. Diaz and Arguello [2009] showed how vertical selection can be tuned in the presence of user feedback. The same authors explored *domain adaptation* techniques for improving classification-based vertical selection in the presence of unlabeled data in a more recent work [Diaz and Arguello, 2010].

While most existing collection selection algorithms (e.g., CORI, ReDDE or other classification-based methods) focus on the evidence of individual collections to determine the relevance of available collections, the work by Hong et al. [2010] considers a joint probabilistic classification model that estimates the probabilities of relevance in a joint manner by considering the relationship among collections.

3.4 Overlap-aware collection selection

Management of duplication across collections can be done at either or both of the collection selection and result merging stages. At the collection selection stage, an *overlap-aware* algorithm can select a set of collections that contain a large number of unique relevant documents. For such an approach to be effective, the rate of overlap between the underlying pairs of collections must be accurately estimated in advance; small estimation errors may lead to the loss of many relevant documents located in the ignored collections.

Hernandez and Kambhampati [2005] introduced COSCO for management of duplicates at selection time. The system estimates the overlap between different bibliographic collections and avoids selecting pairs of servers that appear to have high overlap for a query. They use CORI [Callan et al., 1995] as a benchmark and show that COSCO finds more unique relevant documents for a given selected number of collections.

Shokouhi and Zobel [2007] estimated the rate of overlap between collections based on the intersection of their sampled documents. They showed that the estimated values can be used to prevent ReDDE from selecting collections with high overlap at the same time (F-ReDDE). They also proposed Relax, an overlap-aware method that selects collections that are expected to maximize the number of unique relevant documents in the final results.

3.5 Other collection selection approaches.

Rasolofo et al. [2001] ranked collections according to the quality of the top-ranked documents they return. The approach suggested by the authors does not require collection representation sets. Instead, the query is sent to all collections and the top-ranked documents returned by collections are indexed by the broker. In the final step, the broker computes the similarity of these documents to the query and ranks collections accordingly.

Abbaci et al. [2002] proposed a collection selection method that can be described in two steps. First, the query is passed to all collections.

Then, using the approach suggested by Lawrence and Giles [1998], the snippets of the top n answers returned by each collection are downloaded. In the second step, the broker measures the similarity of the query with the top n downloaded documents. Collections whose corresponding downloaded documents have the highest similarities with the query are selected.

Similar to Si and Callan [2005b], Craswell et al. [2000] considered the search effectiveness of collections for collection selection. In their approach, the broker sends a number of training multi-term probe queries to collections. The top results from each collection are downloaded and are gathered in a single index. The broker then applies an effective retrieval model to rank the downloaded documents for the initial training queries. The search effectiveness of collections are computed according to their contribution to the top n (they suggested $n = 20$) results when the query is compared against the downloaded documents. Experiments showed that adding the effectiveness factor to CORI can significantly improve its final search precision. Estimating the search effectiveness of online search engines has been also considered by Rasolofo et al. [2003]. They have used the approach suggested by Craswell et al. [2000] to approximate the effectiveness of a set of news search engines for their metasearch experiments.

Larson [2002; 2003] introduced a logistic regression approach for collection selection. His proposed method has been reported to be as effective as CORI.

Xu and Croft [1999] suggested a collection selection technique based on document clustering and language modeling. They used the *k-means* clustering algorithm [Jain and Dubes, 1988] for clustering documents based on their topics, and utilized the KL-Divergence equation [Lafferty and Zhai, 2001] for comparing the queries with representation sets and ranking collections. They showed that when collections are clustered and generated based on their topicality, federated search systems can outperform centralized indexes in terms of search effectiveness. However, Larkey et al. [2000] showed that in heterogeneous environments, where collections are not clustered based on their topicality, the performance of the suggested collection selection algorithm decreases

and becomes worse than CORI. Similar observations have been reported by Shen and Lee [2001]. The major difference between their work and the approach reported by Xu and Croft [1999] is that Shen and Lee [2001] used a form of *TF-IDF* for computing the text similarities [Salton and McGill, 1986], while Xu and Croft [1999] utilized the KL-Divergence instead. In addition, Xu and Croft [1999] divided the global information into clustered collections, while Shen and Lee [2001] clustered the content of each collection.

A two-stage language modeling approach is proposed by Yang and Zhang [2005; 2006] for collection selection. First, collections are clustered in a hierarchical structure. The query is then compared against available clusters. Once the suitable clusters for a query are found, the most relevant collections in those clusters are selected by a language modeling technique.

King et al. [2006] proposed an ontology-based method for collection selection. In their approach, queries are initially mapped to an ontology tree. The queries are then expanded by the associated terms in the ontology-based classification tree. The expanded queries are found to be more effective than the original queries for collection selection.

In a series of papers [Meng et al., 2001; Yu et al., 1999; 2002; Wu et al., 2001] a collection selection method has been proposed that ranks collections according to the estimated *global similarity* of their most similar documents.

Hawking and Thistlewaite [1999] suggested using lightweight probe queries to rank collections. The broker sends a number of n-term probe queries to each collection ($n = 2$ was suggested by the authors). Collections return small packets of term frequency information to the broker. The broker then ranks collections according to the term frequency information provided in packets. Probe queries are picked from the query terms according to their document frequency factors in a reference collection. Once the promising collections are recognized—by comparing the answers returned for the probe queries—the original query is passed to the top-ranked collections.

Wu and Crestani [2002; 2003] proposed a multi-objective collection selection strategy. Similar to the approach suggested by Fuhr [1999a], they used a utility function that can be optimized according to dif-

ferent factors such as document relevance, query time, query cost and duplication among collections. However, Wu and Crestani [2002; 2003] have not provided evaluation results for their method in terms of the final search effectiveness.

Finally, Thomas and Hawking [2009] provided a comparative empirical study of collection selection techniques for personal metasearch.

3.6 Evaluating collection selection

Metrics for evaluating collection selection methods are usually *recall-based*. That is, collection selection techniques are compared according to the number of relevant documents available in selected collections [D'Souza et al., 2004b;a; Gravano et al., 1994b; Si and Callan, 2003a].

Binary precision-recall. Gravano et al. [1994b] assumed that any collection with at least one matching document for a query q is a *right* collection for that query. They defined $Right(q)$ as the set of all collections that contain at least one matching answer for the query q. Assuming that the number of matching documents in the k selected collections is represented by δ_k, the precision and recall values for collection selection can be computed as in Equations 3.18 and 3.19 [Gravano et al., 1994b; Gravano, 1997]:

$$P_k = Precision = \frac{\delta_k \cap Right(q)}{\delta_k} \quad if \quad \delta_k > 0 \tag{3.18}$$

$$R_k = Recall = \frac{\delta_k \cap Right(q)}{Right(q)} \quad if \quad Right(q) > 0 \tag{3.19}$$

Precision (P_k) is the proportion of selected collections that contain at least one matching document, and recall (R_k) is the fraction of right collections that are selected. These binary metrics may be suitable for evaluating collection selection techniques in relational databases. However, for unstructured text retrieval, where Boolean matching is a poor indicator of relevance, more sophisticated metrics are required. Therefore, a modified version of Equation 3.19 was suggested [Gravano and García-Molina, 1995; Gravano et al., 1999]. In this version, the optimal baseline $Right(q)$ consists of collections whose approximated goodness

values are higher than a pre-defined threshold. Further information about how goodness values are approximated, can be found in Section 3.1.

The recall metric for collection selection was later formalized in a more general form [French and Powell, 2000; French et al., 2001; Powell and French, 2003]:

$$Recall = R_k = \frac{\sum_{i=1}^{k} \Omega_i}{\sum_{i=1}^{k} O_i} \tag{3.20}$$

where $\sum_{i=1}^{k} \Omega_i$ and $\sum_{i=1}^{k} O_i$ are respectively the total number of relevant documents available in the top k collections selected by a collection selection method, and an optimal baseline. We describe current baselines for collection selection later in this section. Sogrine et al. [2005] combined the precision P_k and recall R_k values in a single metric called maxF_k as:

$$\max F_k = \max_k \frac{2}{\frac{1}{R_k} + \frac{1}{P_k}} \tag{3.21}$$

The authors compared collection selection methods according to their maxF_k values for all possible values of k. They also compared the *discounted cumulative gain* [Järvelin and Kekäläinen, 2000] of collection selection rankings with an optimal baseline.

French and Powell [2000] introduced $\widehat{R}_k$, a modified version of R_k [Gravano et al., 1994a], in which only collections with non-zero weights are considered. The modified recall metric is defined as:

$$\widehat{R}_k = \frac{\sum_{i=1}^{k} \Omega_i}{\sum_{i=1}^{k^*} O_i} \tag{3.22}$$

where k^* is the number of collections with non-zero weights, and k is the number of collections that are selected. In a similar methodology, Zobel [1997] suggested the use of the number of relevant documents in selected collections for comparing collection selection methods. Thomas and Shokouhi [2009] showed that $\widehat{R}_k$ does not always correlate with other metrics such as precision.

Mean square error (MSE). Callan et al. [1995] measured the *mean square error* (MSE) of collection selection methods against an optimal baseline. For a given query q, the effectiveness of a collection selection ranking Ω can be computed as follows:

$$\frac{1}{N_c} \cdot \sum_{i \in C} (O_i - \Omega_i)^2 \tag{3.23}$$

Here, N_c shows the total number of collections; while Ω_i and O_i represent the positions of the i^{th} collection respectively in the rankings of a collection selection method and an optimal baseline. In the optimal ranking—as will be discussed later in Section 3.6.1—collections are ranked according to the number of relevant documents they contain. Rankings with low MSE values are considered to be effective.

Spearman rank correlation coefficient (SRCC). The application of SRCC for measuring the quality of collection samples was previously discussed in Section 2.6. A simplified version of the Spearman rank correlation coefficient has been suggested for comparing the rankings produced by collection selection methods with that of an optimal baseline [French and Powell, 2000; French et al., 1999; Powell, 2001; Powell and French, 2003]:

$$SRCC = 1 - \frac{6 \sum_{i=1}^{N_c} (O_i - \Omega_i)^2}{N_c({N_c}^2 - 1)} \tag{3.24}$$

Here, N_c is the total number of collections, while Ω_i and O_i are respectively the positions of the i^{th} collection in the rankings of a collection selection technique, and a baseline method.

3.6.1 Collection selection baselines

Collection selection has a wide variety of baselines. French and Powell [2000] suggested a random collection selection baseline for analyzing the worst-case behavior of selection methods. The random selection baseline has been also used by Craswell et al. [2000] as a worst-case baseline for collection selection. *Count-based ranking* (CBR) [French

and Powell, 2000] is a Boolean baseline that ranks collections according to their numbers of matching answers for queries. Since containing the query terms is not usually enough for a document to be relevant, CBR does not necessarily rank collections according to their number of relevant documents. Gravano and García-Molina [1995] defined an *ideal* ranking baseline *Ideal(l)* for collection selection. In *Ideal(l)*, first the similarity values of a query q with documents in all collections are computed. Collections are then ranked according to the number of documents with similarity values greater than l, where l is a predefined threshold. *Ideal(l)* has been also used as a baseline for collection selection [Gravano et al., 1999; French and Powell, 2000; French et al., 1998; Yuwono and Lee, 1997].

Relevance-based ranking (RBR) is the most common baseline for evaluating collection selection methods [Callan, 2000; D'Souza, 2005; D'Souza et al., 2004a;b; Gravano et al., 1999; French and Powell, 2000; French et al., 1998; Powell, 2001; Powell and French, 2003; Powell et al., 2000; Si and Callan, 2003a; 2004b;a]. In RBR, collections are ranked according to the number of relevant documents that they contain for queries.

Zobel [1997] introduced a baseline that sorts collections according to their number of *highly ranked* documents. For a given query, he considered highly ranked documents as the top answers returned by a centralized monolithic index of all collections.

Another common baseline for collection selection methods is the ranking of collections according to the number of documents they contain [D'Souza, 2005; D'Souza et al., 2004a;b; French et al., 1999; Powell, 2001; Powell and French, 2003; Zobel, 1997]. Compared to other collections, larger collections are more likely to contain relevant documents due to their greater size. This is known as the *size-based ranking* (SBR). The SBR baseline is query independent and does not take specific needs of each query is into consideration.

3.7 Summary

The goal of collection selection techniques is to select a subset of collections that are more likely to return relevant documents. Early collection

selection methods rank collections by calculating the lexical similarity of a query with collection representation sets. Most lexicon-based methods such as GlOSS [Gravano et al., 1994a] and CORI [Callan et al., 1995] treat collection representation sets as bags of words. These methods ignore the document boundaries, which limits their performance particularly in uncooperative environments. Document-surrogate collection selection methods such as ReDDE [Si and Callan, 2003a] step away from treating each collection as a single big document. Most of these approaches create an index of all sampled documents from different collections. They rank collections according to the ranking of their sampled documents for the query. More recently, classification-based collection selection methods have been proposed to directly estimate the probabilities of relevance of collections for a user query based on supervised learning.

Several metrics have been proposed for evaluating the performance of collection selection methods. Most metrics are recall-oriented. That is, they compare the performance of any collection selection algorithm with that of an oracle baseline that ranks collections according to their number of relevant documents.

We provide an overview of result merging techniques in the next chapter.

4

Result merging

The last step in a typical federated search session is result merging (Figure 4.1). In result merging, the broker receives the top-ranked answers of selected collections and orders them in a single list for presentation to the user.

This chapter describes the previous work on result merging and briefly covers some of the related areas such as data fusion and metasearch merging.

4.1 Federated search merging

In a federated search environment, collections may use different retrieval models and have different ranking features. Thus, the document scores or ranks returned by multiple collections are not directly comparable and are not reliable for merging. The goal of result merging algorithms is to calculate a global score for each document that is comparable to the scores of documents returned by other collections.

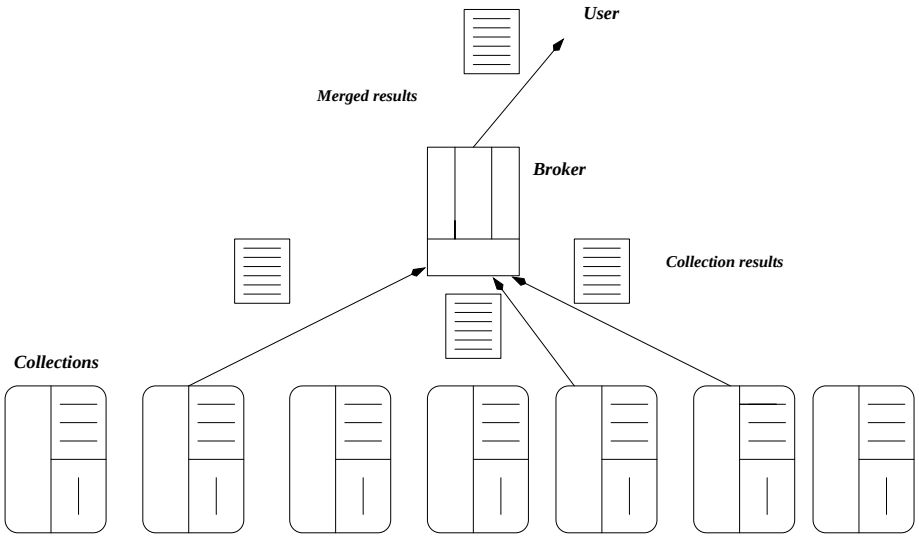

Fig. 4.1 *The result merging process; selected collections return their top-ranked answers to the broker. The broker then merges those documents and returns them to the user.*

4.2 Terminology

Federated search merging, metasearch merging (collection fusion), and data fusion are similar but not identical concepts.

In *federated search merging*, the top-ranked results returned for the query by different collections are blended into a single list. Most federated search merging techniques assume that the rate of overlap among collections is either none or negligible.

In *data fusion*, the query is sent to a single collection but is ranked by multiple retrieval models. The rankings generated by different retrieval models are then merged to produce the final result list [Aslam and Montague, 2001; Aslam et al., 2003; Croft, 2000; Fox and Shaw, 1993; Lee, 1997; Lillis et al., 2006; Ng, 1998; Oztekin et al., 2002; Wu and McClean, 2006; Vogt and Cottrell, 1999; Vogt, 1999].

Metasearch and federated search have been often used interchangeably. However, we only use metasearch when referring to metasearch engines described in Section 1.2.3.

This chapter summarizes the previous work on federated search

merging. We also provide a brief overview of data fusion and metasearch techniques.

4.3 Federated search merging

The main task in result merging is to compute comparable scores for documents returned by different collections. When available, the document scores reported by collections can be used by the broker to compute the merging scores. In environments where document scores are not reported by collections, merging methods assign *pseudoscores* to the returned answers. For example, when 1,000 documents are returned from a collection, the scores of the first-ranked document is set to 1, the next is set to 0.999, and so on [Rasolofo et al., 2003; Si and Callan, 2003b].

CORI merging. The CORI result merging formula [Callan, 2000; Callan et al., 1995] is a linear combination of the collection selection scores and the document scores returned by collections. CORI uses a simple heuristic formula to normalize collection-specific document scores. First, associated with the CORI collection selection algorithm, the collection scores are normalized as:

$$C' = \frac{C - C_{\min}}{C_{\max} - C_{\min}} \tag{4.1}$$

where, C is the collection selection score of collection c, computed by the CORI collection selection algorithm [Callan et al., 1995; Callan, 2000] (more detailed information of CORI collection selection can be found in Section 3.1). C' denotes the normalized score of C ranging between $[0, 1]$. $C_{\min}$ and $C_{\max}$ are calculated by setting the T component in Equation 3.5 to 0 and 1 respectively. The collection-specific document scores are normalized in a similar manner.

For a document returned with score D from a collection with normalized collection selection score of C', CORI computes the final merging score as:

$$D' = \frac{D + 0.4 \times D \times C'}{1.4} \tag{4.2}$$

CORI merging formula uses heuristic weighting schemes such as weight 1 for normalized document score and weight 0.4 for normalized collection selection score in Equation 4.2. The heuristic weighting scheme strongly limits the performance of CORI merging as it may not adapt to different types of queries and collections.

SSL. Si and Callan [2002; 2003b] proposed a semi-supervised learning (SSL) method for result merging. SSL trains a regression model for each collection that maps document scores into their global (merging) scores. For this purpose, SSL creates a central index of all sampled documents downloaded from collections (CSI). For a given query, some of the documents that are returned from the collections may already be available in the central sample index. SSL runs the query against CSI and compares the centralized scores of such *overlap* sampled documents with the scores (or pseudoscores) reported by collections to compute the merging scores.

When collections use an identical retrieval model, SSL can use all overlap documents to train a single model that converts the collection-specific scores into global scores. In such a scenario for the jth overlap document $d_{i,j}$ returned from a selected collection c_i, SSL uses two scores: the score reported by the original collection ($D_{i,j}$) and the score computed using CSI ($E_{i,j}$).

$$\begin{bmatrix} D_{1,1} & C_1 D_{1,1} \\ D_{1,2} & C_1 D_{1,2} \\ \cdots & \cdots\cdots \\ D_{n,m} & C_n D_{n,m} \end{bmatrix} \times [a \quad b] = \begin{bmatrix} E_{1,1} \\ E_{1,2} \\ \cdots \\ E_{n,m} \end{bmatrix} \tag{4.3}$$

Using the $D_{i,j}$ and $E_{i,j}$ values of the overlap documents, SSL trains a single regression model as:

$$D'_{i,j} = a \times E_{i,j} + b \times E_{i,j} \times C_i \tag{4.4}$$

where C_i is the selection score of collection c_i that has returned document $d_{i,j}$. The combining parameters a and b can be estimated using

a sufficient number of overlap documents. Si and Callan [2003b] suggested that at least three overlap documents are required for training the SSL models.

When the retrieval models used in collections are not identical, SSL cannot train a single model that converts the outputs of all collections into global scores. The scores returned by collections may have different ranges. For example, KL-Divergence language modeling [Lafferty and Zhai, 2001] produces negative weights (likelihood values), while INQUERY [Callan et al., 1992; Allan et al., 2000] produces positive weights between zero and one (probabilities of relevance). Therefore, for each collection a separate model is trained that maps the scores returned from different collections to global values. That is,

$$D'_{i,j} = a_i \times E_{i,j} + b_i \tag{4.5}$$

For a given document $d_{i,j}$ from collection c_i, $D'_{i,j}$ is the estimated global score and $E_{i,j}$ is the score of $d_{i,j}$ reported by collection c_i. The values for a_i and b_i can be obtained by training a regression matrix for each collection as follows:

$$\begin{bmatrix} D_{1,1} & 1 \\ D_{1,2} & 1 \\ \dots & 1 \\ D_{n,m} & 1 \end{bmatrix} \times [a_i \quad b_i] = \begin{bmatrix} E_{1,1} \\ E_{1,2} \\ \dots \\ E_{n,m} \end{bmatrix} \tag{4.6}$$

Since a separate model is trained for each collection according to its returned answers, the likelihood of visiting an overlap document in the downloaded samples (training data) is lower than under the SSL single-model. Therefore, the broker may need to receive longer result lists from collections or download some documents on the fly [Si and Callan, 2003b].

SAFE. *SAFE* (sample-agglomerate fitting estimate) [Shokouhi and Zobel, 2009] is designed to work with minimum cooperation between the broker and collections. SAFE uses the scores of all documents in agglomeration of all the collection samples, and generates a statistical fit to estimate scores. SAFE does not rely on the presence of overlap

documents and is based on the following principle: For a given query, the results of the sampled documents is a subranking of the original collection, so curve fitting to the subranking can be used to estimate the original scores.

Similar to SSL, SAFE also utilizes a centralized index of sampled documents from all collections to calculate the merging scores. SAFE merging can be summarized in two steps: First, the broker ranks the documents available in the centralized sample index (CSI) for the query. Second, for each collection, the sampled documents that received non-zero scores in the first step are used to estimate the merging scores. SAFE employs collection size estimations (see Section 2.3) to adjust the scores of sampled documents. Each sampled document is assumed to be representative for $|c|/|S_c|$ documents in the collection, where $|S_c|$ and $|c|$ respectively denote the number of documents in the sample, and collection. That is, the sampled documents are assumed to be uniformly selected from the collection. Although previous studies suggested that the documents downloaded by query-based sampling are not uniformly sampled [Bar-Yossef and Gurevich, 2006; Bharat and Broder, 1998b; Garcia et al., 2004; Shokouhi et al., 2006b; Thomas and Hawking, 2007], Shokouhi and Zobel [2009] empirically suggested that the performance of SAFE is not significantly affected by that assumption.

In the final step, SAFE uses the regression techniques [Gross, 2003] to fit a curve to the adjusted scores, and to predict the scores of the top-ranked—unseen—documents returned by each collection. Since the estimated scores for all documents are computed with reference to the same corpus (CSI), they are comparable across different collections.

In contrast to SSL, SAFE does not rely on overlap documents between CSI and the results returned by collection. Therefore, it is suitable for environments in which downloading documents on the fly is restricted.

4.4 Multilingual result merging

Most existing federated search research methods focus on the environments where all documents in collections are in the same language. However, in some federated search applications collections may contain

documents in different languages (e.g. patent databases). Therefore, it is important to extend monolingual result merging techniques for multilingual environments.

The majority of previous work on merging multilingual ranked lists have been conducted in the Cross-Language Evaluation Form (CLEF).[1] The problem of merging multilingual ranked lists is similar to federated search result merging. Simple score normalization methods are not effective, and some methods download all retrieved documents and translate the documents into a single language for ranking.

Si et al. proposed an approach similar to SSL for merging multilingual result lists [Si and Callan, 2005a; Si et al., 2008]. Their method downloads a subset of top-ranked documents from each ranked list and utilizes a multilingual centralized retrieval algorithm for calculating comparable scores for the small set of downloaded documents. The multilingual centralized retrieval method in their approach performs both query translation and document translation for computing comparable merging scores. The query translation method converts queries into different languages and applies monolingual retrieval methods to documents in individual languages. The document translation method is complementary to query translation method and translates all the documents into a single language (e.g., English). The final comparable document scores are obtained by combining scores from the query translation and the document translation methods.

The standard SSL method uses a linear regression model to map scores from individual collections to global scores. The multilingual result merging approach [Si and Callan, 2005a; Si et al., 2008] was tested with both logistic regression and linear regression models, and the logistic model was found to produce more robust results.

4.5 Merge-time duplicate management for federated search

Management of within-collection redundancy has been a subject of active research, with a range of techniques having been proposed [Bernstein and Zobel, 2004; Brin et al., 1995; Broder et al., 1997; Fetterly

[1] `http://www.clef-campaign.org/`, accessed 17 Aug 2010.

et al., 2003; Manber, 1994]. However, management of redundancy between collections as in the case of federated search is subject to additional constraints. In particular, since collections are not centrally managed, it may not be practical to use a pre-processing approach to redundancy management; rather, it must occur at query time based on additional document information transmitted to the broker. Thus, management of near-duplicate documents is highly sensitive to both time (because it must be done on the fly) and bandwidth.

ProFusion [Gauch et al., 1996b], MetaCrawler [Selberg and Etzioni, 1997b], and Grouper [Zamir and Etzioni, 1999] attempt to eliminate duplicate documents from the final results, by aggregating results that point to the same location according to their URLs. However, the elimination of near-duplicate documents has not been addressed by these techniques.

Bernstein et al. [2006] proposed using the *grainy hash vector* (GHV) for detecting duplicate and near-duplicate documents during merging. GHV is a derivation of the minimal-chunk sampling techniques [Fetterly et al., 2003], that operate by parsing documents into strings of contiguous text, known as *chunks*, and comparing the number of identical chunks shared by a pair of documents.

Shokouhi et al. [2007b] tested GHV on three federated search testbeds with overlapped collections, and showed that GHV can be effectively used for detecting and removing duplicate documents from the merged results. In uncooperative environments where GHV vectors may not be provided, other duplicate management techniques may be used instead (See Section 3.4).

4.6 Other papers on result merging

In the STARTS protocol [Gravano et al., 1997], collections return the term frequency, document frequency, term weight, and document weight information of each returned answer to the broker. Kirsch [2003] suggested that each collection return the term frequencies, document frequencies, and the total number of indexed documents to the broker. In such methods, documents are merged according to their calculated similarities based on the statistics received by the broker.

As in CORI result merging, Rasolofo et al. [2001] calculated the final score of a document by multiplying the document weight and collection score parameters. In their approach, document scores are reported by collections, and collection scores are calculated according to the number of documents that are returned by each collection for queries. This is based on the assumption that collections returning a greater number of results for a query are more likely to contain relevant documents. The same approach has been used by Abbaci et al. [2002] for merging.

CVV merging [Yuwono and Lee, 1997], calculates the merging scores according to the goodness values of collections, and the positions of documents in collection ranked lists. The authors assume that the difference in relevance scores between two consecutive documents in the ranked list returned by a collection is inversely proportional to the normalized goodness value of that collection.

Craswell et al. [1999] partially downloaded the top returned documents (say the first 4 KB of each document) and used a reference index of term statistics for reranking and merging the downloaded documents. They showed that the effectiveness of their approach is comparable to that of a merging scenario where documents are downloaded completely and the actual term statistics are used.

Xu and Croft [1999] utilized a version of INQUERY [Allan et al., 2000; Callan et al., 1992] that uses the global inverse document frequency values to calculate the final score of documents for merging. The basic requirement for this approach, is that collections provide the broker with the document frequency information of their terms.

Wang and DeWitt [2004] used the *PageRank* [Brin and Page, 1998] of returned documents for merging. In their approach, the final PageRank of a page d returned by a selected collection c is computed according to the estimated *ServerRank* of c and the computed *LocalRank* of d inside c. For calculating the rank values for d and c, the link information of all pages in collections is required.

Shou and Sanderson [2002] proposed two result merging methods without downloading the full-text information of returned documents. The first method re-ranks merged results by using a centralized search engine on text fragments (e.g., titles and snippets) returned from in-

dividual collections. The second method examines how similar a returned document is to other returned documents. In particular, returned text fragments have been used for calculating similarity scores and returned documents are finally fused into a single ranking by the similarity scores.

Voorhees et al. [1995] suggested two *collection fusion* methods based on previous training data, where their goal was to determine the number of documents that have to be fetched from selected collections. In their first approach, the number of relevant documents returned by each collection for the training queries is investigated; The similarities of testing queries with the previous queries are measured, and the k most similar training queries are selected to compute the average probabilities of relevance for different collections. The number of documents fetched from each collection for merging varies according to their probabilities of relevance.

In their second approach, Voorhees et al. [1995] clustered the training queries based on the number of common documents they return from collections. A centroid vector is calculated for each cluster and the testing queries are compared with all available centroid vectors. For a given query, the weights of collections are computed according to their performance for previous training queries in the same cluster. The number of documents that are fetched from collections is proportional to their weights.

In aggregated search environments with different data types, result merging is relatively more challenging and less explored. The merging score computed for a document of a given type (say video), not only should be comparable to the scores of other documents with the same type, but also to the scores of documents with other types (e.g. image). Click-through rate has been suggested and used as a suitable measure for this task [Diaz, 2009; König et al., 2009]. However, due to various sources of presentation bias the focus has been mostly devoted on blending vertical results at fixed positions. In a recent study, Shushmita et al. [2010] have investigated the impact of presenting the vertical results at different positions on the page.

4.6.1 Data fusion and metasearch merging

In data fusion, documents in a single collection are ranked according to different ranking functions or features. Therefore, metasearch engines can be regarded as data fusion systems where the collection being ranked contains the entire web.[2]

Several algorithms have been used commonly in both areas. The simplest merging algorithm is the round-robin strategy [Savoy et al., 1996]. In round-robin, it is assumed that collections have similar search effectiveness with similar numbers of relevant documents. The results returned by multiple collections (or retrieval models) are merged according to their ranks. That is, the top-ranked documents of all collections are merged first, followed by the second-ranked documents and so forth.

When a document is returned by more than one collection, several combination methods, including CombMNZ, CombSum, CombMax, and CombMin, have been proposed for calculating the final scores [Fox and Shaw, 1993; 1994]. In CombMax, the maximum score reported for a duplicate document is used as its final score for merging. CombMin uses the minimum score of a duplicate document for merging. CombSum adds all the reported scores for a duplicate document, while CombMNZ adds all the reported scores and then multiplies the total sum by the number of collections that have returned that document. In most of these methods, therefore, documents that are returned by multiple collections are ranked higher than the other documents. These methods have been used widely in both data fusion and collection fusion (metasearch merging) experiments and thus we do not classify them specifically under any of these categories.

Data fusion. In data fusion methods, documents in a single collection are ranked with different search systems. The goal in data fusion is to generate a single accurate ranking list from the ranking lists of different retrieval models. There are no collection representation sets and no collection selection.

[2] Note that this is a strong assumption given that the documents indexed by one engine might be missed by another.

Data fusion methods are based on a voting principle, where, for a given query, a document returned by many search systems should be ranked higher than the other documents. In addition, data fusion strategies should take the rank of documents into account. A document that has been returned on top of three ranking lists is intuitively more likely to be relevant than a document that has appeared at low positions in four ranking lists.

Aslam and Montague [2001] divided data fusion methods into four categories according to their training and scoring functions (training versus no training, and relevance scores versus ranks only). They showed that, when training data is available, the effectiveness of data fusion methods using only ranks can be comparable to those that use document scores reported by the individual systems.

A comparison between score-based and rank-based methods is provided by Renda and Straccia [2002; 2003] suggesting that rank-based methods are generally less effective. Lillis et al. [2006] divided each ranking into segments with different scores. The final score of a document is calculated according to its rank and segment number. Shokouhi [2007b] and Lillis et al. [2008] suggested different ranked list segmentation strategies for more effective fusion.

Metasearch merging. In metasearch merging, the results returned by multiple search engines—with overlapping indexes—are combined in a single ranked list.

The D-WISE system [Yuwono and Lee, 1996] uses the ranks of retrieved documents for merging. The Inquirus system [Glover et al., 1999; Lawrence and Giles, 1998] computes the merging scores after the full contents of the retrieved results are fetched. A similar approach has been suggested by Yu et al. [1999].

Rasolofo et al. [2003] described a metasearch merging method for combining the results returned from multiple news search engines. They suggested that the title, date, and summary of the results returned by search engines can be used effectively for merging. Snippet information is also used by the Mearf metasearch engine [Oztekin et al., 2002], Lu et al. [2005] and Tsikrika and Lalmas [2001] for merging the results returned by different sources.

Glover and Lawrence [2001] proposed a method for calculating the confidence values of relevance predictions for the returned snippets. When the returned snippets are found to be not sufficiently informative, additional information such as link statistics or the contents of documents are used for merging. Savoy et al. [1996] and Calvé and Savoy [2000] applied logistic regression [Hosmer and Lemeshow, 1989] to convert the ranks of documents returned by search engines into probabilities of relevance. Documents are then merged according to their estimated probabilities of relevance.

In *shadow document* methods for result merging [Wu and Crestani, 2004], the document scores returned by multiple search engines are normalized by a regression function that compares the scores of overlapped documents between the returned ranked lists. In the Savvy-Search metasearch engine [Dreilinger and Howe, 1997], document scores returned by each search engine are normalized into a value between zero and one. The normalized scores of overlapped documents are summed for computing the final score.

In metasearch merging, voting plays an important role for calculating the final rank of a document. Documents that are returned by many search engines are likely to rank highly in the final merged list. In the absence of overlap between the results, most metasearch merging techniques become ineffective. For example, methods such as CombMNZ and CombSum [Fox and Shaw, 1993; 1994; Lee, 1997] that are used in metasearch engines such as SavvySearch [Dreilinger and Howe, 1997] degrade to a simple round-robin approach [Savoy et al., 1996].

4.7 Evaluating result merging

Result merging techniques are usually compared according to the number of relevant documents in the final merged results [Callan, 2000; Callan et al., 1995; Chakravarthy and Haase, 1995; Craswell et al., 1999; Rasolofo et al., 2001; 2003; Si and Callan, 2003b].

Counting correct matches. Chakravarthy and Haase [1995] used the total number of queries that return at least one relevant answer in the top n results for comparing result merging methods.

Precision. Precision is the most commonly used metric for evaluating the effectiveness of federated search merging. It has been used in different forms such as mean average precision [Craswell et al., 1999; Rasolofo et al., 2001; 2003], and precision at different cutoff ranks ($P@n$) [Callan, 2000; Callan et al., 1995; Rasolofo et al., 2003; Si and Callan, 2003b].

The application of precision for evaluating federated search systems is not only limited to the result merging stage. Collection selection and representation methods can be also evaluated according to their impact on precision. The precision-oriented methods discussed in this section have been also used for evaluating the performance of collection selection and collection representation methods [Callan, 2000; Craswell et al., 2000; Nottelmann and Fuhr, 2003; 2004a; Hawking and Thomas, 2005; Rasolofo et al., 2001; Ogilvie and Callan, 2001; Si and Callan, 2003a; 2004b;a; 2005b; Xu and Callan, 1998; Xu and Croft, 1999].

Result merging (search effectiveness) baselines. Federated search techniques, particularly in uncooperative environments, cannot access the complete term statistics of collections. Therefore, an effective centralized search engine that has indexed all available documents in collections (using complete term statistics) is often used as an *oracle* baseline for federated search systems [Craswell, 2000; Craswell et al., 2000; Lu and Callan, 2002; 2003b; Ogilvie and Callan, 2001; Towell et al., 1995; Voorhees and Tong, 1997; Voorhees et al., 1995; Xu and Callan, 1998; Xu and Croft, 1999]. Hawking and Thistlewaite [1999] referred to the rankings of documents returned by the oracle index as *correct merging*. They also defined *perfect merging* as an unrealistic ranked list that contains all relevant documents before all irrelevant documents.

In the majority of published related work, the effectiveness of the oracle centralized baseline has been reported to be higher than that of federated search alternatives. However, there are some exceptional cases in which federated search systems have been reported to outperform centralized baselines. For example, Xu and Croft [1999] suggested that, if documents are partitioned into homogeneous collections by clustering and individual collections use the same retrieval mod-

els with identical lexicon statistics, then federated search methods can produce better precision values compared to the centralized baselines. Similarly, Craswell et al. [2000] suggested that merging the results from a few collections that contain the highest number of relevant documents for a query, can be more effective than running the query on the oracle centralized index. However, finding collections with the highest number of relevant documents is still an open question.

4.8 Summary

The goal of result merging in federated search is to combine the ranking lists from multiple collections into a single list. This is a challenging task due to differences in retrieval models and lexicon statistics of individual collections that make the document scores reported by different collections less comparable. Result merging algorithms try to map the scores/pseudoscores from collections into comparable scores for merging.

Result merging algorithms can rely on the document scores and other important information reported by collections to merge the results. For example, the CORI algorithm [Callan, 2000; Callan et al., 1995] calculates the normalized document scores with the cooperation of individual collections. SSL [Si and Callan, 2002; 2003b] utilizes regression techniques to build models that transform scores from individual collections to comparable scores. The SAFE merging method [Shokouhi and Zobel, 2009] goes a step further by relaxing the requirements of overlapped documents in the SSL algorithm.

Result merging in federated search is closely related to the areas of data fusion and metasearch merging. The majority of data fusion and metasearch merging techniques favor documents that are returned by multiple retrieval models or collections.

Federated search merging methods have been often evaluated by precision-oriented techniques. Furthermore, the centralized retrieval results of all available documents in collections have been commonly considered as an oracle baseline for merging algorithms.

In the next chapter, we describe the commonly used federated search testbeds.

5

Federated search testbeds

The relative effectiveness of federated search methods tends to vary between different testbeds [D'Souza et al., 2004b; Si and Callan, 2003a]. Therefore, it is important to describe detailed information of experimental testbeds for reliable analysis of current federated search techniques. This section is devoted to the discussion of testbeds that have been proposed for federated search experiments.

In typical federated search testbeds, collections are disjoint and do not overlap. The descriptions of a few commonly used testbeds are provided below:

SYM236 & UDC236. *SYM236* [French et al., 1998; 1999; Powell, 2001; Powell and French, 2003] includes 236 collections of varying sizes, and is generated from documents on TREC disks 1–4 [Harman, 1994; 1995]. UDC236 [French et al., 1999; Powell and French, 2003], also contains 236 collections, and is generated from the same set of documents (i.e., TREC disks 1–4). The difference is only in the methodology used for assigning documents to collections. In UDC236, each collection contains almost the same number of documents; in SYM236, documents are distributed between collections according to their pub-

lication date, generating collections with different sizes.[1] SYM236 and UDC236 are both created from 691,058 documents—an average of 2,928 documents per collection—which is significantly smaller than many federated search testbeds developed more recently. Therefore, they are no longer suitable for simulating large-scale federated search environments with today's standards. More details about the attributes of SYM236 and UDC236 can be found elsewhere [D'Souza, 2005; Powell, 2001; Powell and French, 2003].

trec123-100col-bysource (uniform). Documents on TREC disks 1, 2, and 3 [Harman, 1994] are assigned to 100 collections by publication source and date [Callan, 2000; Powell and French, 2003; Si and Callan, 2003a;b]. The TREC topics 51–150 and their corresponding relevance judgements are available for this testbed. The `<title>` fields of TREC queries have been more commonly used for federated search experiments on this testbed, although description and narrative fields are also available.

trec4-kmeans. A k-means clustering algorithm [Jain and Dubes, 1988] has been applied on the TREC4 data [Harman, 1995] to partition the documents into 100 homogeneous collections [Xu and Croft, 1999].[2] The TREC topics 201–250 and their corresponding relevance judgements are available for the testbed. These queries do not contain the `<title>` fields, and the `<description>` fields have been mainly used instead.

trec123-AP-WSJ-60col (relevant). This and the next two testbeds have been generated from the trec123-100col-bysource (uniform) collections. Documents in the 24 Associated Press and 16 Wall Street Journal collections in the uniform testbed are collapsed into two separate large collections. The other collections in the uniform testbed are as before. The two largest collections in the testbed have

[1] SYM236 and UDC236 testbeds can be downloaded from: `http://www.cs.virginia.edu/~cyberia/testbed.html`, accessed 17 Aug 2010.

[2] The definitions of uniform and trec4 testbeds are available at: `http://boston.lti.cs.cmu.edu/callan/Data/`, accessed 17 Aug 2010.

a higher density of relevant documents for the corresponding TREC queries compared to the other collections.

trec123-2ldb-60col (representative). Collections in the uniform testbed are sorted by their names. Every fifth collection starting with the first collection is merged into a large collection. Every fifth collection starting from the second collection is merged into another large collection. The other 60 collections in the uniform testbed are unchanged.

trec123-FR-DOE-81col (nonrelevant). Documents in the 13 Federal Register and 6 Department of Energy collections from the uniform testbed are merged into two separate large collections. The remaining collections remain unchanged. The two largest collections in the testbed have lower density of relevant documents for the TREC topics compared to the other collections.

The effectiveness of federated search methods may vary when the distribution of collection sizes is skewed or when the density of relevant documents varies across different collections [Si and Callan, 2003a]. The latter three testbeds can be used to evaluate the effectiveness of federated search methods for such scenarios. More details about the trec4-kmeans, uniform, and the last three testbeds can be found in previous publications [Powell, 2001; Powell and French, 2003; Si, 2006; Si and Callan, 2003b;a; Xu and Croft, 1999].

Among the disjoint data collections described so far, the uniform testbed (and its derivatives: relevant, nonrelevant, representative), and the trec4-kmeans testbed are the most commonly used [Callan, 2000; Lu and Callan, 2002; Ogilvie and Callan, 2001; Powell and French, 2003; Si and Callan, 2003b;a; 2004b; 2005b; Si et al., 2002; Shokouhi, 2007a; Shokouhi and Zobel, 2009; Shokouhi et al., 2006a; 2009; Thomas and Shokouhi, 2009].

GOV2 testbeds. In recent years, larger datasets have become publicly available to account for the growth in the size of real-life collections. The GOV2 dataset [Clarke et al., 2005] is a crawl of about 25 million "`.gov`" webpages. Several federated search testbeds have been

produced based on the GOV2 data. Shokouhi [2007a] split the documents from the largest 100 crawled hosts—in terms of the number of crawled documents— into one hundred separate collections. Similarly, Arguello et al. [2009a] generated their *gov2.1000* testbed based on the largest 1000 hosts in GOV2. Arguello et al. [2009a] created *gov2.250* and *gov2.30* by sampling documents from the hosts in GOV2 and clustering the hosts accordingly in respectively 250 and 30 collections.

Overall, the GOV2 testbeds are many times larger than the previously discussed alternatives, and are more realistic for simulating large-scale federated search experiments.

Other testbeds. Several other federated search testbeds with disjoint collections have been generated based on the TREC newswire documents [Callan et al., 1995; D'Souza et al., 2004a; Hawking and Thistlewaite, 1999; Moffat and Zobel, 1994; Xu and Callan, 1998; Xu and Croft, 1999; Zobel, 1997]. In most of these datasets, the partitioned collections are either similar in size or the document publication source/date.

The first federated search testbed generated from the crawled web documents was proposed by French et al. [1999]. They divided the TREC6 VLC dataset [Hawking and Thistlewaite, 1997] into 921 collections according to the document domain addresses. Similarly, Craswell et al. [2000] divided the TREC WT2G dataset [Hawking et al., 2000] into 956 collections according to the domain addresses of documents. Rasolofo et al. [2001] proposed two testbeds created from the TREC8 and TREC9 (WT10G) [Bailey et al., 2003] datasets, respectively containing four and nine collections. In a similar study [Abbaci et al., 2002], documents available in the WT10G dataset were divided into eight collections for evaluating collection selection experiments.

Hawking and Thomas [2005] created a hybrid testbed based on documents available in the TREC GOV dataset [Craswell and Hawking, 2002]. Using a document classifier, the authors managed to find the Homepages of 6,294 servers in the TREC GOV dataset, from which 1,971 (31%) had a search interface. The authors allocated the documents from each of these servers into separate collections. They gath-

ered all non-searchable servers into a large crawled collection. Therefore, in total, their hybrid testbed is comprised of 1,972 collections.

Thomas and Hawking [2009] created an artificial testbed for personal metasearch. Their testbed included collections generated from a public mailing list, a personal mailbox and calendar, plus text collections generated from the TREC data. More details about their testbed can be found elsewhere [Thomas, 2008b].

In standard federated search testbeds, there is often no overlap among collections [Powell and French, 2003; Si and Callan, 2003b]. However, in practice, a significant proportion of documents may overlap between collections. Shokouhi et al. [Shokouhi et al., 2007c; Shokouhi and Zobel, 2007] created five new testbeds with overlapping collections based on documents available in the TREC GOV dataset.

5.1 Summary

It is important to investigate the effectiveness of different federated search methods on a variety of testbeds. Most existing testbeds contain disjoint collections, while some recent testbeds share overlapped documents among their collections. A common strategy to create federated search testbeds is to partition different TREC corpora into many collections. This approach has several advantages; many queries and corresponding relevant judgements have been provided for these testbeds; it is possible to create testbeds with many collections of various sizes, the experimental results are reproducible by other researchers. The main disadvantage of such testbeds is that they may not represent real-life federated search environments.

Realistic testbeds such as those used in the FedLemur project [Avrahami et al., 2006] are more suitable for investigating the performance of federated techniques in practice. However, access to such testbeds is often restricted to a small number of groups or organizations.

The next chapter concludes the paper and suggests directions for future research.

6

Conclusion and Future Research Challenges

Web search has significantly evolved in recent years. For many years, web search engines such as Google, Yahoo! were only providing search service over text documents. Aggregated search was one of the first steps to go beyond text search, and was the beginning of a new era for information seeking and retrieval. These days, web search engines support aggregated search over a number of verticals, and blend different types of documents (e.g. images, videos) in their search results. Moreover, web search engines have started to crawl and search the hidden web [Madhavan et al., 2008].

Federated search (a.k.a federated information retrieval), has played a key role in providing the technology for aggregated search and crawling the hidden web.

The application of federated search is not limited to the web search engines. There are many scenarios in which information is distributed across different sources/servers. Peer-to-peer networks and personalized search are two examples in which federated search has been successfully used for searching multiple independent collections (e.g.,[Lu, 2007; Thomas, 2008b]).

In this work, we provided a review of previous research on federated

search. This chapter summarizes the materials we covered, and points out a few directions for future research.

6.1 The state-of-the-art in federated search

Research on federated search can be dated back to the 1980s [Mazur, 1984]. Since then, substantial progress has been made in different subproblems of federates search.

Collection representation. The representation set of each collection, may contain information regarding its size, contents, query language as well as other key features that can be used by the broker during collection selection and result merging.

Early research demanded human-generated metadata for collection representation sets. More robust approaches rely on statistical metadata. In cooperative environments, collections are required to provide their vocabularies and corpus statistics upon request. In the absence of cooperation between collections and the broker, query-based sampling [Callan and Connell, 2001] is used to generate collection representation sets. In query-based sampling, several probe queries are submitted to the collection and the returned results are collected to generate the collection representation set.

Different variants of query-based sampling methods have been proposed to acquire accurate collection content representation efficiently. Adaptive sampling techniques [Baillie et al., 2006a; Caverlee et al., 2006; Shokouhi et al., 2006a] choose sample size for each collection with respect to vocabulary growth in sampled documents, or the predicted ratio of collection documents that are sampled. In focused probing [Ipeirotis and Gravano, 2002], the sampling queries are selected from the categories of a hierarchical classification tree, and collections can be classified according to the number of results they return for each category. The shrinkage technique [Ipeirotis and Gravano, 2004] improves the comprehensiveness of collection representation by assuming that topically related collections share many terms. Since out-of-date representation sets may no longer be representative of their corresponding collections, Ipeirotis et al. [2005] and Shokouhi et al. [2007a] proposed

several techniques for modeling content changes and updating collection representation sets in federated search environments.

The collection size statistics have been used in many collection selection algorithms as important parameters (e.g. ReDDE [Si and Callan, 2003a]). In the sample-resample method [Si and Callan, 2003a], the collection size is estimated by comparing the term frequencies of the sampled documents with the entire collection. Capture-recapture methods such as CH and MCR [Shokouhi et al., 2006b] estimate the size of collections by sampling. Alternatives such as random-walk sampling [Bar-Yossef and Gurevich, 2006], and multiple-queries sampling [Thomas and Hawking, 2007] can provide better estimations at the cost of running more sampling queries.

Collection selection. For each query, the broker often selects a subset of collections that are more likely to return relevant documents. Selecting more collections not only causes extra efficiency costs, but also may not even improve the performance [Thomas and Shokouhi, 2009]. Early collection selection methods treated each collection as a big document or a lexicon distribution, and used different variations of traditional document ranking algorithms to rank them with respect to the query (e.g.,[Callan et al., 1995; Yuwono and Lee, 1997]). However, recent research has demonstrated that ignoring the document boundaries in the big document approach may lead to low effectiveness, particularly in environments that have skewed distribution of collection size [Si and Callan, 2003a]. Motivated by this observation, a new family of *document-surrogate* collection selection methods have been proposed that explicitly estimate the goodness/usefulness of individual documents in collections [Si and Callan, 2003a; Shokouhi, 2007a]. These methods have been shown to obtain more robust results on a wide range of testbeds.

The *utility-based* collection selection techniques are another group of selection methods that can be used in the presence of training data [Si and Callan, 2004b; 2005b]. Such techniques can model the collection search effectiveness, and can be optimized for high precision or recall.

Result merging. Once the selected collections return their top-ranked results, the broker compares them and ranks them in a single list for presentation to the user. Result merging is a difficult task; different collections may use different retrieval algorithms and have different lexicon statistics. Therefore, the document scores reported by different collections are often not directly comparable. Early result merging methods [Callan, 2000] either used simple heuristics to rank returned documents, or downloaded all returned documents for calculating comparable scores [Craswell et al., 1999]. Recent methods tried to approximate comparable document scores in more accurate and efficient way. For example, SSL [Si and Callan, 2003b] uses the overlap between the top-ranked results returned by collections and their sample documents to compute comparable scores for merging. The SAFE algorithm [Shokouhi and Zobel, 2009] assumes that the ranking of sampled documents is a sub-ranking of the original collection. Therefore, SAFE applies curve fitting to the subranking to estimate the merging scores, and does not rely on the overlapped documents between sample documents and collection results.

Federated search testbeds. Construction of valuable testbeds is one of the most important contributions of previous research on federated search. These testbeds serve the purpose for evaluating the relative effectiveness of different federated search algorithms. The trend is to construct testbeds with more collections, larger amount of data and more heterogeneous collection statistics, that better simulate large-scale real world federated search environments.

Most of the current testbeds have been constructed by splitting TREC newswire or TREC web collections based on different criteria. Many early testbeds are constructed with uniform or moderately skewed collection statistics (e.g., similar number of documents or similar amount of relevant documents in each collection), while recent testbeds are more diverse.

6.2 Future Research Challenges

Despite recent advancements in all aspects of federated search, there are many opportunities for further improvements.

Beyond bag of words. The majority of previous work on federated search use only basic bag of words features. Utilizing the power of clicks, anchor-text and link-graph features is a promising next step for federated search.

Hawking and Thomas [2005] showed that the anchor-text can be a useful feature for ranking distributed collections. Arguello et al. [2009a] used clicks as a features in their *classification-based* collection selection method.

Yu et al. [2001] combined text similarity and linkage information for collection selection. Wang and DeWitt [2004] described how PageRank can be computed over distributed collections.

Query expansion for federated search. Query expansion techniques have been widely used to improve the retrieval effectiveness of ad-hoc information retrieval with centralized search engines [Diaz and Metzler, 2006; Metzler and Croft, 2007; Xu and Croft, 1996]. In the context of federated search however, query expansion techniques have made little success [Ogilvie and Callan, 2001; Shokouhi et al., 2009]. Global query expansion techniques send the same expanded query to all collections. Alternatively, the expansion terms can be generated specifically to a collection (or a cluster of collections). Collection-specific expansion terms can be less vulnerable to topic drift, but are generated based on smaller feedback collections that may affect their quality.

Classifying the queries for local/global expansion, or expand/not-expand are potential directions for future work.

Classification-based collection selection. The problem of selecting suitable collections for a query can be regarded as a classification task. Given the query, the output of the classifier indicates the selection decisions for individual collections.

Classification-based collection/vertical selection techniques are the

latest generation of collection selection methods. The best paper awards at WSDM09[1] and SIGIR09[2] conferences were given to papers on classification-based vertical selection [Arguello et al., 2009b; Diaz, 2009]. Arguello et al. [2009a] have recently proposed and tested the first classification-based collection selection framework on three typical federated search testbeds generated from the TREC GOV2 documents. The authors showed that the classification-based selection techniques can outperform the state-of-the-art methods such as ReDDE [Si and Callan, 2003a].

Classification-based federated search is still a new area of research and can be extended and explored in many ways. For example, it may be worthwhile to investigate the application of such frameworks for result merging. In addition, it would be interesting to combine the earlier work of Ipeirotis and Gravano [2008] on topically classifying collections with existing classification-based collection selection techniques in a hybrid framework.

Optimized merging. The common goal between existing federated search merging techniques is to compute comparable scores for the documents returned for selected collections. Early techniques [Callan, 2000] were using the normalized document and collection scores to compute the final score of a document. More recent techniques such as SSL [Si and Callan, 2003b] and SAFE [Shokouhi and Zobel, 2009] use linear regression and curve fitting over the score distribution of sampled documents to compute the merging scores. The common neglected fact is that accurate comparable scores do not necessarily optimize precision or any other metric that is used for evaluating the final retrieval. Developing merging techniques that can be optimized for different evaluation metrics can be considered as a direction for future investigation.

Merging becomes more challenging in scenarios such as aggregated search in which different types of results are blended into a single list. Although vertical selection has been recently discussed in the literature [Arguello et al., 2009b; Diaz, 2009; Diaz and Arguello, 2009], studies

[1] `http://wsdm2009.org`, accessed 17 Aug 2010.
[2] `http://sigir2009.org`, accessed 17 Aug 2010.

such as [Shushmita et al., 2010] on merging results from different verticals are fairly rare.

Evaluating federated search. Different stages of federated search such as collection selection and collection representation are generally evaluated by different metrics. Collection representation techniques are often evaluated according to the comprehensiveness of their representations sets, and collection selection and result merging methods can be evaluated based on the quality of their final merged results. However, the effectiveness of each stage also depends on the performance of previous stages. For example, it is not possible to compare the effectiveness of different merging methods, when the selected collections do not contain relevant documents. Therefore, modeling the relationship between different stages of federated search for evaluation is an important direction for future research.

Very large scale federated search. Most existing federated search systems deal with a relative small number of collections that ranges from a few dozen to a few thousand. However, in 2007 it was estimated that there were about 25 millions of text data collections on the web [Madhavan et al., 2007]. To design federated search systems in such environments, it is important to design extremely scalable solutions for collection selection and result merging. Furthermore, it is also important to build fully automatic collection detection and result extraction solutions that can deal with dynamic environments, where independent collections are often subject to change.

Federated search in other contexts. Federated search techniques have been successfully utilized in different areas.

Lu and Callan [2005; 2006] applied federated search collection representation and collection selection techniques in peer-to-peer full text search applications.

Thomas and Hawking [2008; 2009] pioneered the application of federated search techniques in personal metasearch. Carman and Crestani [2008] proposed some preliminary ideas for personalized federated

search. For example, a personalized QBS approach can sample more documents from the specific areas that a user is interested in. In a similar project,[3] individual users may have personal bias for information from different collections. This type of information can be useful for both collection selection and result merging.

Elsas et al. [2008] and Seo and Croft [2008] used federated search collection selection techniques for blog site search. In federated search the goal is to select collections with relevant documents, while in blog site search the goal is to identify blogs with relevant posts.

There are more and more web collections that contain multimedia data such as image and video. Most existing research in federated search works only with textual features. It is important to design collection selection and result merging algorithms for media types other than text. The work by Berretti et al. [2003] selects image databases with abstract data that reflects the representative visual features of each visual database. The authors merge retrieved images from distributed image collections with a learning approach that maps image retrieval scores assigned by different collections into normalized scores for merging. The learning approach is similar to the SSL result merging for text data except that Berretti et al. [2003] use a set of sample queries for creating training data in learning the score mappings.

The improvement of federated search solutions will directly impact the above and many other applications.

6.3 Acknowledgements

The authors are grateful to Jamie Callan and reviewers for their insightful comments. Some parts of this work have appeared in previous publications [Bernstein et al., 2006; Si et al., 2002; Si and Callan, 2002; 2004a; 2003a;b; 2004b; 2005b;a; Shokouhi, 2007a; Shokouhi and Zobel, 2007; 2009; Shokouhi et al., 2006b; 2007d;a;b; 2009; Thomas and Shokouhi, 2009].

[3] `http://www.cs.purdue.edu/homes/lsi/Federated_Search_Career_Award.html`, accessed 17 Aug 2010.

References

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*, pages 529–535, Washington, DC, 2002. IEEE Computer Society. ISBN 0-7695-1503-1.

D. Aksoy. Information source selection for resource constrained environments. *SIGMOD Record*, 34(4):15–20, 2005. ISSN 0163-5808.

J. Allan, J. Aslam, M. Sanderson, C. Zhai, and J. Zobel, editors. *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Boston, MA, 2009. ISBN 978-1-60558-483-6.

J. Allan, M. Connell, and B. Croft. INQUERY and TREC-9. In E. Voorhees and D. Harman, editors, *Proceedings of the Ninth Text REtrieval Conference*, pages 551–563, Gaithersburg, MD, 2000. NIST Special Publication.

A. Anagnostopoulos, A. Broder, and D. Carmel. Sampling search-engine results. In Ellis and Hagino [2005], pages 245–256. ISBN 1-59593-046-9.

P. Apers, P. Atzeni, S. Ceri, S. Paraboschi, K. Ramamohanarao, and

R. Snodgrass, editors. *Proceedings of the 27th International Conference on Very Large Data Bases*, Roma, Italy, 2001. Morgan Kaufmann. ISBN 1-55860-804-4.

A. Arasu and H. Garcia-Molina. Extracting structured data from web pages. In Y. Papakonstantinou and A. Halevy Z. Ives, editors, *Proceedings ACM SIGMOD International Conference on Management of Data*, pages 337–348, San Diego, CA, 2003. ISBN 1-58113-634-X.

J. Arguello, J. Callan, and F. Diaz. Classification-based resource selection. In Cheung et al. [2009], pages 1277–1286. ISBN 978-1-60558-512-3.

J. Arguello, F. Diaz, J. Callan, and J. Crespo. Sources of evidence for vertical selection. In Allan et al. [2009], pages 315–322. ISBN 978-1-60558-483-6.

H. Ashman and P. Thistlewaite, editors. *Proceedings of the Seventh International Conference on World Wide Web*, Brisbane, Australia, 1998. Elsevier. ISBN 0169-7552.

J. Aslam and Mark Montague. Models for metasearch. In Croft et al. [2001], pages 276–284. ISBN 1-58113-331-6.

J. Aslam, V. Pavlu, and R. Savell. A unified model for metasearch, pooling, and system evaluation. In Kraft et al. [2003], pages 484–491. ISBN 1-58113-723-0.

T. Avrahami, L. Yau, L. Si, and J. Callan. The FedLemur: federated search in the real world. *Journal of the American Society for Information Science and Technology*, 57(3):347–358, 2006. ISSN 1532-2882.

R. Baeza-Yates. *Information retrieval: data structures and algorithms*. Prentice-Hall, Upper Saddle River, NJ, 1992. ISBN 0-13-463837-9.

P. Bailey, N. Craswell, and D. Hawking. Engineering a multi-purpose test collection for web retrieval experiments. *Information Processing and Management*, 39(6):853–871, 2003. ISSN 0306-4573.

M. Baillie, L. Azzopardi, and F. Crestani. Adaptive query-based sampling of distributed collections. In Crestani et al. [2006], pages 316–328. ISBN 3-540-45774-7.

M. Baillie, L. Azzopardi, and F. Crestani. An evaluation of resource description quality measures. In H. Haddad, editor, *Proceedings of the ACM symposium on Applied computing*, pages 1110–1111, Dijon,

France, 2006b. ISBN 1-59593-108-2.

M. Baillie, L. Azzopardi, and F. Crestani. Towards better measures: evaluation of estimated resource description quality for distributed IR. In X. Jia, editor, *Proceedings of the First International Conference on Scalable Information systems*, page 41, Hong Kong, 2006c. ACM. ISBN 1-59593-428-6.

M. Baillie, M. Carman, and F. Crestani. A topic-based measure of resource description quality for distributed information retrieval. In M. Boughanem, C. Berrut, J. Mothe, and C. Soulé-Dupuy, editors, *Proceedings of the 31st European Conference on Information Retrieval Research*, volume 5478 of *Lecture Notes in Computer Science*, pages 485–496, Toulouse, France, 2009. Springer.

Z. Bar-Yossef and M. Gurevich. Random sampling from a search engine's index. In L. Carr, D. Roure, A. Iyengar, C. Goble, and M. Dahlin, editors, *Proceedings of the 15th International Conference on World Wide Web*, pages 367–376, Edinburgh, UK, 2006. ACM. ISBN 1-59593-323-9.

Z. Bar-Yossef and M. Gurevich. Efficient search engine measurements. In Williamson et al. [2007], pages 401–410. ISBN 978-1-59593-654-7.

L. Barbosa and J. Freire. Combining classifiers to identify online databases. In Williamson et al. [2007]. ISBN 978-1-59593-654-7.

C. Baumgarten. A probabilistic model for distributed information retrieval. In Belkin et al. [1997], pages 258–266. ISBN 0-89791-836-3.

C. Baumgarten. A probabilistic solution to the selection and fusion problem in distributed information retrieval. In Gey et al. [1999], pages 246–253. ISBN 1-58113-096-1.

N. Belkin, P. Ingwersen, and M. Leong, editors. *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Athens, Greece, 2000. ISBN 1-58113-226-3.

N. J. Belkin, A. D. Narasimhalu, and P. Willett, editors. *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Philadelphia, PL, 1997. ISBN 0-89791-836-3.

M. Bergman. The deep web: Surfacing hidden value. *Journal of Electronic Publishing*, 7(1), 2001. ISSN 1080-2711.

Phil Bernstein, Yannis Ioannidis, and Raghu Ramakrishnan, editors. *Proceedings of the 28th International Conference on Very Large Data Bases*, Hong Kong, China, 2002. Morgan Kaufmann.

Y. Bernstein, M. Shokouhi, and J. Zobel. Compact features for detection of near-duplicates in distributed retrieval. In Crestani et al. [2006], pages 110–121. ISBN 3-540-45774-7.

Y. Bernstein and J. Zobel. A scalable system for identifying coderivative documents. In A. Apostolico and M. Melucci, editors, *Proceedings of the 11th International String Processing and Information Retrieval Conference*, volume 3246 of *Lecture Notes in Computer Science*, pages 55–67, Padova, Italy, 2004. Springer. ISBN 3-540-23210-9.

S. Berretti, J. Callan, H. Nottelmann, X. M. Shou, and S. Wu. MIND: resource selection and data fusion in multimedia distributed digital libraries. In Clarke et al. [2003], pages 465–465. ISBN 1-58113-646-3.

K. Bharat and A. Broder. A technique for measuring the relative size and overlap of public web search engines. In Ashman and Thistlewaite [1998], pages 379–388. ISBN 0169-7552.

K. Bharat and A. Broder. A technique for measuring the relative size and overlap of public web search engines. *Computer Networks and ISDN Systems*, 30(1–7):379–388, 1998b. ISSN 0169-7552.

D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. ISSN 1533-7928.

S. Brin, J. Davis, and H. García-Molina. Copy detection mechanisms for digital documents. In M. Carey and D. Schneider, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 398–409, San Jose, CA, 1995. ISBN 0-89791-731-6.

S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In Ashman and Thistlewaite [1998], pages 107–117. ISBN 0169-7552.

A. Broder, M. Fontura, V. Josifovski, R. Kumar, R. Motwani, S. Nabar, R. Panigrahy, An. Tomkins, and Y. Xu. Estimating corpus size via queries. In P. Yu, V. Tsotras, E. Fox, and B. Liu, editors, *Proceedings of the ACM CIKM International Conference on Information and Knowledge Management*, pages 594–603, Arlington, VA, 2006. ISBN

1-59593-433-2.

A. Broder, S. Glassman, M. Manasse, and G. Zweig. Syntactic clustering of the web. *Computer Networks and ISDN System*, 29(8-13): 1157–1166, 1997. ISSN 0169-7552.

D. Buttler, L. Liu, and C. Pu. A fully automated object extraction system for the World Wide Web. In Shen et al. [2001], pages 361–370. ISBN 1-58113-348-0.

J. Callan. Distributed information retrieval. In B. Croft, editor, *Advances in information retrieval, Chapter 5*, volume 7 of *The Information Retrieval Series*, pages 127–150. Kluwer Academic Publishers, 2000. ISBN 978-0-7923-7812-9.

J. Callan and M. Connell. Query-based sampling of text databases. *ACM Transactions on Information Systems*, 19(2):97–130, 2001. ISSN 1046-8188.

J. Callan, M. Connell, and A. Du. Automatic discovery of language models for text databases. In A. Delis, C. Faloutsos, and S. Ghandeharizadeh, editors, *Proceedings ACM SIGMOD International Conference on Management of Data*, pages 479–490, Philadelphia, PA, 1999. ISBN 1-58113-084-8.

J. Callan, F. Crestani, and M. Sanderson, editors. *Distributed Multimedia Information Retrieval, SIGIR 2003 Workshop on Distributed Information Retrieval, Revised Selected and Invited Papers*, volume 2924 of *Lecture Notes in Computer Science*, Toronto, Canada, 2004. Springer. ISBN 3-540-20875-5.

J. Callan, B. Croft, and S. Harding. The INQUERY retrieval system. In A. Tjoa and I. Ramos, editors, *Proceedings of Third International Conference on Database and Expert Systems Applications*, pages 78–83, Valencia, Spain, 1992. ISBN 3-211-82400-6.

J. Callan, Z. Lu, and B. Croft. Searching distributed collections with inference networks. In Fox et al. [1995], pages 21–28. ISBN 0-89791-714-6.

J. Callan, A. Powell, J. French, and M. Connell. The effects of query-based sampling on automatic database selection algorithms. Technical report, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 2000.

A. Calvé and J. Savoy. Database merging strategy based on logistic

regression. *Information Processing and Management*, 36(3):341–359, 2000. ISSN 0306-4573.

M. Carman and F. Crestani. Towards personalized distributed information retrieval. In Myaeng et al. [2008], pages 719–720. ISBN 978-1-60558-164-4.

J. Caverlee, L. Liu, and J. Bae. Distributed query sampling: a quality-conscious approach. In Efthimiadis et al. [2006], pages 340–347. ISBN 1-59593-369-7.

S. Cetinta, L. Si, and H. Yuan. Learning from past queries for resource selection. In Cheung et al. [2009], pages 1867–1870. ISBN 978-1-60558-512-3.

A. Chakravarthy and K. Haase. NetSerf: using semantic knowledge to find internet information archives. In Fox et al. [1995], pages 4–11. ISBN 0-89791-714-6.

D. Cheung, I Song, W. Chu, X. Hu, and J. Lin, editors. *Proceedings of the ACM CIKM International Conference on Information and Knowledge Management*, Hong Kong, China, 2009. ISBN 978-1-60558-512-3.

J. Cho and H. Garcia-Molina. Effective page refresh policies for web crawlers. *ACM Transactions on Database Systems*, 28(4):390–426, 2003. ISSN 0362-5915.

C. Clarke, G. Cormack, J. Callan, D. Hawking, and A. Smeaton, editors. *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Toronto, Canada, 2003. ISBN 1-58113-646-3.

C. Clarke, N. Craswell, and I. Soboroff. The TREC terabyte retrieval track. *SIGIR Forum*, 39(1):31–47, 2005. ISSN 0163-5840.

W. Cohen. Learning trees and rules with set-valued features. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI)*, pages 709–716, Portland, OR, 1996. ISBN 0-262-51091-X.

J. Conrad and J. Claussen. Early user—system interaction for database selection in massive domain-specific online environments. *ACM Transactions on Information Systems*, 21(1):94–131, 2003. ISSN 1046-8188.

J. Conrad, X. Guo, P. Jackson, and M. Meziou. Database selection

using actual physical and acquired logical collection resources in a massive domain-specific operational environment. In Bernstein et al. [2002], pages 71–82.

J. Conrad, C. Yang, and J. Claussen. Effective collection metasearch in a hierarchical environment: global vs. localized retrieval performance. In Järvelin et al. [2002]. ISBN 1-58113-561-0.

J. Cope, N. Craswell, and D. Hawking. Automated discovery of search interfaces on the web. In *Proceedings of the 14th Australasian database conference-Volume 17*, page 189. Australian Computer Society, Inc., 2003.

N. Craswell. *Methods for Distributed Information Retrieval*. PhD thesis, Australian National University, 2000.

N. Craswell, P. Bailey, and D. Hawking. Server selection on the World Wide Web. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, pages 37–46, San Antonio, TX, 2000. ISBN 1-58113-231-X.

N. Craswell and D. Hawking. Overview of the TREC-2002 web track. In E. Voorhees, editor, *Proceedings of the 11th Text REtrieval Conference*, pages 86–95, Gaithersburg, MD, 2002. NIST Special Publication.

N. Craswell, D. Hawking, and S. Robertson. Effective site finding using link anchor information. In Croft et al. [2001], pages 250–257. ISBN 1-58113-331-6.

N. Craswell, D. Hawking, and P. Thistlewaite. Merging results from isolated search engines. In *Proceedings of the 10th Australasian Database Conference*, pages 189–200, Auckland, New Zealand, 1999. Springer. ISBN 981-4021-55-5.

Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. Roadrunner: Towards automatic data extraction from large web sites. In Apers et al. [2001], pages 109–118. ISBN 1-55860-804-4.

F. Crestani, P. Ferragina, and M. Sanderson, editors. *Proceedings of the 13th International String Processing and Information Retrieval Conference*, volume 4209 of *Lecture Notes in Computer Science*, Glasgow, UK, 2006. Springer. ISBN 3-540-45774-7.

F. Crestani, S. Marchand-Maillet, H. Chen, E. Efthimiadis, and J. Savoy, editors. *Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Re-*

trieval, Geneva, Switzerland, 2010. ISBN 978-1-4503-0153-4.

B. Croft. Combining approaches to information retrieval. In B. Croft, editor, *Advances in information retrieval, Chapter 1*, volume 7 of *The Information Retrieval Series*, pages 1–36. Kluwer Academic Publishers, 2000. ISBN 978-0-7923-7812-9.

B. Croft, D. Harper, D. H. Kraft, and J. Zobel, editors. *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New Orleans, LA, 2001. ISBN 1-58113-331-6.

B. Croft, A. Moffat, K. Rijsbergen, R. Wilkinson, and J. Zobel, editors. *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, 1998. ISBN 1-58113-015-5.

O. de Kretser, A. Moffat, T. Shimmin, and J. Zobel. Methodologies for distributed information retrieval. In M. Papazoglou, M. Takizawa, B. Kramer, and S. Chanson, editors, *Proceedings of the Eighteenth International Conference on Distributed Computing Systems*, pages 66–73, Amsterdam, The Netherlands, 1998. ISBN 0-8186-8292-2.

S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Sciences*, 41(6):391–407, 1990.

M. DeGroot. *Optimal Statistical Decisions (Wiley Classics Library)*. Wiley interscience, 2004. ISBN 978-0-471-72614-2.

F. Diaz. Integration of news content into web results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 182–191, Barcelona, Spain, 2009. ACM. ISBN 978-1-60558-390-7.

F. Diaz and J. Arguello. Adaptation of offline vertical selection predictions in the presence of user feedback. In Allan et al. [2009], pages 323–330. ISBN 978-1-60558-483-6.

F. Diaz and J. Arguello. Adaptation of offline vertical selection predictions in the presence of user feedback. In Crestani et al. [2010], pages 323–330. ISBN 978-1-4503-0153-4.

F. Diaz and D. Metzler. Improving the estimation of relevance models using large external corpora. In Efthimiadis et al. [2006], pages 154–161. ISBN 1-59593-369-7.

D. Dreilinger and A. Howe. Experiences with selecting search engines using metasearch. *ACM Transaction on Information Systems*, 15(3): 195–222, 1997. ISSN 1046-8188.

D. D'Souza. *Document Retrieval in Managed Document Collections*. PhD thesis, RMIT University, Melbourne, Australia, 2005.

D. D'Souza and J. Thom. Collection selection using n-term indexing. In Y. Zhang, M. Rusinkiewicz, and Y. Kambayashi, editors, *Proceedings of the second International symposium on cooperative database systems for advanced applications (CODAS'99)*, pages 52–63, Wollongong, NSW, Australia, 1999. Springer. ISBN 9814021644.

D. D'Souza, J. Thom, and J. Zobel. A comparison of techniques for selecting text collections. In *Proceedings of the Australasian Database Conference*, page 28, Canberra, Australia, 2000. IEEE Computer Society. ISBN 0-7695-0528-7.

D. D'Souza, J. Thom, and J. Zobel. Collection selection for managed distributed document databases. *Information Processing and Management*, 40(3):527–546, 2004a. ISSN 0306-4573.

D. D'Souza, J. Zobel, and J. Thom. Is CORI effective for collection selection? an exploration of parameters, queries, and data. In P. Bruza, A. Moffat, and A. Turpin, editors, *Proceedings of the Australaian Document Computing Symposium*, pages 41–46, Melbourne, Australia, 2004b. Melbourne, Australia. ISBN 0-9757172-0-0.

E. Efthimiadis, S. Dumais, D. Hawking, and K. Järvelin, editors. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, WA, 2006. ISBN 1-59593-369-7.

A. Ellis and T. Hagino, editors. *Proceedings of the 14th International Conference on World Wide Web*, Chiba, Japan, 2005. ACM. ISBN 1-59593-046-9.

J. Elsas, J. Arguello, J. Callan, and J. Carbonell. Retrieval and feedback models for blog feed search. In Myaeng et al. [2008], pages 347–354. ISBN 978-1-60558-164-4.

D. Fetterly, M. Manasse, and M. Najork. On the evolution of clusters of near-duplicate web pages. In *Proceedings of the First Conference on Latin American Web Congress*, page 37, Washington, DC, 2003. IEEE Computer Society. ISBN 0-7695-2058-8.

E. Fox, P. Ingwersen, and R. Fidel, editors. *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, WA, 1995. ISBN 0-89791-714-6.

E. Fox and J. Shaw. Combination of multiple searches. In D. Harman, editor, *Proceedings of the Second Text REtrieval Conference*, pages 243–252, Gaithersburg, MD, 1993. NIST Special Publication.

E. Fox and J. Shaw. Combination of multiple searches. In D. Harman, editor, *Proceedings of the Third Text REtrieval Conference*, pages 105–108, Gaithersburg, MD, 1994. NIST Special Publication.

J. French and A. Powell. Metrics for evaluating database selection techniques. *World Wide Web*, 3(3):153–163, 2000. ISSN 1386-145X.

J. French, A. Powell, J. Callan, C. Viles, T. Emmitt, K. Prey, and Y. Mou. Comparing the performance of database selection algorithms. In Gey et al. [1999], pages 238–245. ISBN 1-58113-096-1.

J. French, A. Powell, F. Gey, and N. Perelman. Exploiting a controlled vocabulary to improve collection selection and retrieval effectiveness. In Paques et al. [2001], pages 199–206. ISBN 1-58113-436-3.

J. French, A. Powell, C. Viles, T. Emmitt, and K. Prey. Evaluating database selection techniques: a testbed and experiment. In Croft et al. [1998], pages 121–129. ISBN 1-58113-015-5.

N. Fuhr. Optimum database selection in networked IR. In J. Callan and N. Fuhr, editors, *Proceedings of the SIGIR'96 Workshop on Networked Information Retrieval (NIR'96)*, Zurich, Switzerland, 1996.

N. Fuhr. A decision-theoretic approach to database selection in networked IR. *ACM Transactions on Information Systems*, 17(3):229–249, 1999a. ISSN 1046-8188.

N. Fuhr. Resource discovery in distributed digital libraries. In *Proceedings of Digital Libraries Advanced Methods and Technologies, Digital Collections*, pages 35–45, Petersburg, Russia, 1999b.

S. Garcia, H. Williams, and A. Cannane. Access-ordered indexes. In V. Estivill-Castro, editor, *Proceedings of the 27th Australasian Computer Science Conference*, pages 7–14, Darlinghurst, Australia, 2004. Australian Computer Society. ISBN 1-920682-05-8.

S. Gauch, editor. *Proceedings of the ACM CIKM International Conference on Information and Knowledge Management*, Kansas, MO,

1999. ISBN 1-58113-1461.

S. Gauch, G. Wang, , and M. Gomez. ProFusion: Intelligent fusion from multiple distributed search engines. *Journal of Universal Computer Science*, 2(9):637–649, 1996a. ISSN 0948-695X.

S. Gauch and G. Wang. Information fusion with ProFusion. In *Proceedings of the First World Conference of the Web Society*, pages 174–179, San Francisco, CA, 1996.

S. Gauch, G. Wang, and M. Gomez. ProFusion: Intelligent fusion from multiple, distributed search engines. *Journal of Universal Computer Science*, 2(9):637–649, 1996b. ISSN 1041-4347.

F. Gey, M. Hearst, and R. Tong, editors. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, 1999. ISBN 1-58113-096-1.

E. Glover and S. Lawrence. Selective retrieval metasearch engine (United States Patent 2002/0165860 a1), 2001.

E. Glover, S. Lawrence, W. Birmingham, and C. Giles. Architecture of a metasearch engine that supports user information needs. In Gauch [1999], pages 210–216. ISBN 1-58113-1461.

J. Goldberg. CDM: an approach to learning in text categorization. In *Proceedings of the Seventh International Tools with Artificial Intelligence*, pages 258–265, Herndon, VA, 1995. IEEE Computer Society. ISBN 0-8186-7312-5.

L. Gravano. *Querying multiple document collections across the internet*. PhD thesis, Stanford University, 1997.

L. Gravano, C. Chang, H. García-Molina, and A. Paepcke. STARTS: Stanford proposal for internet meta-searching. In J. Peckham, editor, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 207–218, Tucson, AZ, 1997. ISBN 0-89791-911-4.

L. Gravano and H. García-Molina. Generalizing GlOSS to vector-space databases and broker hierarchies. In U. Dayal, P. Gray, and S. Nishio, editors, *Proceedings of the 21st International Conference on Very Large Data Bases*, pages 78–89, Zurich, Switzerland, 1995. Morgan Kaufmann. ISBN 1-55860-379-4.

L. Gravano, H. García-Molina, and A. Tomasic. The effectiveness of

GlOSS for the text database discovery problem. In R. Snodgrass and M. Winslett, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 126–137, Minneapolis, MN, 1994a. ISBN 0-89791-639-5.

L. Gravano, H. García-Molina, and A. Tomasic. Precision and recall of GlOSS estimators for database discovery. In *Proceedings of the Third International Conference on Parallel and Distributed Information Systems*, pages 103–106, Austin, TX, 1994b. IEEE Computer Society. ISBN 0-8186-6400-2.

L. Gravano, H. García-Molina, and A. Tomasic. GlOSS: text-source discovery over the internet. *ACM Transactions on Database Systems*, 24(2):229–264, 1999. ISSN 0362-5915.

L. Gravano, P. Ipeirotis, and M. Sahami. Qprober: A system for automatic classification of hidden web databases. *ACM Transactions on Information Systems*, 21(1):1–41, 2003. ISSN 1046-8188.

N. Green, P. Ipeirotis, and L. Gravano. SDLIP + STARTS = SDARTS a protocol and toolkit for metasearching. In E. Fox and C. Borgman, editors, *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, pages 207–214, Roanoke, VA, 2001. ACM. ISBN 1-58113-345-6.

J. Gross. *Linear regression*. Springer, 2003. ISBN 3540401784.

A. Gulli and A. Signorini. The indexable web is more than 11.5 billion pages. In Ellis and Hagino [2005], pages 902–903. ISBN 1-59593-046-9.

E. Han, G. Karypis, D. Mewhort, and K. Hatchard. Intelligent metasearch engine for knowledge management. In Kraft et al. [2003], pages 492–495. ISBN 1-58113-723-0.

D. Harman. Overview of the Third Text REtrieval Conference (TREC-3). In D. Harman, editor, *Proceedings of the Third Text REtrieval Conference*, pages 1–19, Gaithersburg, MD, 1994. NIST Special Publication.

D. Harman. Overview of the fourth Text REtrieval Conference (TREC-4). In D. Harman, editor, *Proceedings of the Fourth Text REtrieval Conference*, pages 1–24, Gaithersburg, MD, 1995. NIST Special Publication.

D. Hawking and P. Thistlewaite. Overview of TREC-6 very large col-

lection track. In E. Voorhees and D. Harman, editors, *Proceedings of the Seventh Text REtrieval Conference*, pages 93–106, Gaithersburg, MD, 1997. NIST Special Publication.

D. Hawking and P. Thistlewaite. Methods for information server selection. *ACM Transactions on Information Systems*, 17(1):40–76, 1999. ISSN 1046-8188.

D. Hawking and P. Thomas. Server selection methods in hybrid portal search. In Marchionini et al. [2005], pages 75–82. ISBN 1-59593-034-5.

D. Hawking, E. Voorhees, N. Craswell, and P. Bailey. Overview of the TREC-8 web track. In E. Voorhees and D. Harman, editors, *Proceedings of the Eight Text REtrieval Conference*, pages 131–150, Gaithersburg, MD, 2000. NIST Special Publication.

Y. Hedley, M. Younas, A. James, and M. Sanderson. Information extraction from template-generated hidden web documents. In P. Isaías, N. Karmakar, L. Rodrigues, and P. Barbosa, editors, *Proceedings of the IADIS International Conference WWW/Internet*, pages 627–634, Madrid, Spain, 2004a. ISBN 972-99353-0-0.

Y. Hedley, M. Younas, A. James, and M. Sanderson. Query-related data extraction of hidden web documents. In M. Sanderson, K. Järvelin, J. Allan, and P. Bruza, editors, *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 558–559, Sheffield, UK, 2004b. ISBN 1-58113-881-4.

Y. Hedley, M. Younas, A. James, and M. Sanderson. A two-phase sampling technique for information extraction from hidden web databases. In A. Laender and D. Lee, editors, *Proceedings of the Sixth Annual ACM International Workshop on Web information and data management*, pages 1–8, Washington DC, 2004c. ISBN 1-58113-978-0.

Y. Hedley, M. Younas, A. James, and M. Sanderson. A two-phase sampling technique to improve the accuracy of text similarities in the categorisation of hidden web databases. In X. Zhou, S. Su, M. Papazoglou, M. Orlowska, and K. Jeffery, editors, *Proceedings of the Fifth International Conference on Web Information Systems Engineering*, volume 3306 of *Lecture Notes in Computer Science*, pages 516–527,

Brisbane, Australia, 2004d. Springer. ISBN 3-540-23894-8.

M. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork. On near-uniform url sampling. In Herman and Vezza [2000], pages 295–308. ISBN 1-930792-01-8.

I. Herman and A. Vezza, editors. *Proceedings of the Ninth International Conference on World Wide Web*, Amsterdam, The Netherlands, 2000. Elsevier. ISBN 1-930792-01-8.

T. Hernandez and S. Kambhampati. Improving text collection selection with coverage and overlap statistics. In Ellis and Hagino [2005], pages 1128–1129. ISBN 1-59593-046-9.

D. Hong, L. Si, P. Bracke, M. Witt Michael, and T. Juchcinski. A joint probabilistic classification model for resource selection. In Crestani et al. [2010], pages 98–105. ISBN 978-1-4503-0153-4.

D. Hosmer and S. Lemeshow. *Applied logistic regression*. John Wiley & Sons, New York, NY, 1989. ISBN 0-471-35632-8.

P. Ipeirotis. *Classifying and searching hidden-web text databases*. PhD thesis, Columbia University, 2004.

P. Ipeirotis and L. Gravano. Distributed search over the hidden web: Hierarchical database sampling and selection. In Bernstein et al. [2002], pages 394–405.

P. Ipeirotis and L. Gravano. When one sample is not enough: improving text database selection using shrinkage. In G. Weikum, A. König, and S. Deßloch, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 767–778, Paris, France, 2004. ISBN 1-58113-859-8.

P. Ipeirotis and L. Gravano. Classification-aware hidden-web text database selection. *ACM Transactions on Information Systems*, 26 (2):1–66, 2008. ISSN 1046-8188.

P. Ipeirotis, A. Ntoulas, J. Cho, and L. Gravano. Modeling and managing content changes in text databases. In *Proceedings of the 21st International Conference on Data Engineering*, pages 606–617, Tokyo, Japan, 2005. IEEE. ISBN 0-7695-2285-8.

A. Jain and R. Dubes. *Algorithms for clustering data*. Prentice-Hall, Upper Saddle River, NJ, 1988. ISBN 0-13-022278-X.

K. Järvelin, M. Beaulieu, R. Baeza-Yates, and S. Myaeng, editors. *Proceedings of the 25th Annual International ACM SIGIR Conference*

on Research and Development in Information Retrieval, Tampere, Finland, 2002. ISBN 1-58113-561-0.

K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In Belkin et al. [2000], pages 41–48. ISBN 1-58113-226-3.

K. Kalpakis, N. Goharian, and D. Grossman, editors. *Proceedings of the ACM CIKM International Conference on Information and Knowledge Management*, McLean, VA, 2002. ISBN 1-58113-492-4.

S. Karnatapu, K. Ramachandran, Z. Wu, B. Shah, V. Raghavan, and R. Benton. Estimating size of search engines in an uncooperative environment. In Jingtao Yao, V. Raghavan, and G. Wang, editors, *Proceedings of the Second International Workshop on Web-based Support Systems*, pages 81–87, Beijing, China, 2004. Saint Mary's University, Canada. ISBN 0-9734039-6-9.

J. Kim and B. Croft. Ranking using multiple document types in desktop search. In Crestani et al. [2010], pages 50–57. ISBN 978-1-4503-0153-4.

J. King, P. Bruza, and R. Nayak. Preliminary investigations into ontology-based collection selection. In P. Bruza, A. Spink, and R. Wilkinson, editors, *Proceedings of the 11th Australasian Document Computing Symposium*, pages 33–40, Brisbane, Australia, 2006. ISBN 1-74107-140-2.

T. Kirsch. Document retrieval over networks wherein ranking and relevance scores are computed at the client for multiple database documents (United States Patent 5,659,732), 2003.

A. König, M. Gamon, and Q. Wu. Click-through prediction for news queries. In Allan et al. [2009], pages 347–354. ISBN 978-1-60558-483-6.

M. Koster. ALIWEB, Archie-like indexing in the web. *Computer Networks and ISDN Systems*, 27(2):175–182, 1994. ISSN 1389-1286.

W. Kraaij, A. de Vries, C. Clarke, N. Fuhr, and N. Kando, editors. *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Amsterdam, The Netherlands, 2007. ISBN 978-1-59593-597-7.

D. Kraft, O. Frieder, J. Hammer, S. Qureshi, and L. Seligman, editors. *Proceedings of the ACM CIKM International Conference on Infor-*

mation and Knowledge Management, New Orleans, LA, 2003. ISBN 1-58113-723-0.

S. Kullback. *Information theory and statistics*. John Wiley & Sons, New York, NY, 1959. ISBN 0486696847.

J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In Croft et al. [2001], pages 111–119. ISBN 1-58113-331-6.

L. Larkey, M. Connell, and J. Callan. Collection selection and results merging with topically organized U.S. patents and TREC data. In A. Agah, J. Callan, E. Rundensteiner, and S. Gauch, editors, *Proceedings of the ACM CIKM International Conference on Information and Knowledge Management*, pages 282–289, McLean, VA, 2000. ISBN 1-58113-320-0.

R. Larson. A logistic regression approach to distributed IR. In Järvelin et al. [2002], pages 399–400. ISBN 1-58113-561-0.

R. Larson. Distributed IR for digital libraries. In T. Koch and I. Sølvberg, editors, *Research and Advanced Technology for Digital Libraries, Seventh European Conference*, volume 2769 of *Lecture Notes in Computer Science*, pages 487–498, Trondheim, Norway, 2003. Springer. ISBN 3-540-40726-X.

S. Lawrence and C. Giles. Inquirus, the NECi meta search engine. In Ashman and Thistlewaite [1998], pages 95–105. ISBN 0169-7552.

J. Lee. Analyses of multiple evidence combination. In Belkin et al. [1997], pages 267–276. ISBN 0-89791-836-3.

D. Lillis, F. Toolan, R. Collier, and J. Dunnion. ProbFuse: a probabilistic approach to data fusion. In Efthimiadis et al. [2006], pages 139–146. ISBN 1-59593-369-7.

D. Lillis, F. Toolan, R. Collier, and J. Dunnion. Extending probabilistic data fusion using sliding windows. In C. Macdonald, I. Ounis, V. Plachouras, I. Ruthven, and R. White, editors, *Proceedings of the 30th European Conference on Information Retrieval Research*, volume 4956 of *Lecture Notes in Computer Science*, pages 358–369, Glasgow, UK, 2008. Springer.

K. Lin and H. Chen. Automatic information discovery from the invisible web. In *Proceedings of the International Conference on Information Technology: Coding and Computing*, pages 332–337, Washing-

ton, DC, 2002. IEEE Computer Society. ISBN 0-7695-1503-1.

B. Liu, R. Grossman, and Y. Zhai. Mining data records in web pages. In Lise Getoor, Ted E. Senator, Pedro Domingos, and Christos Faloutsos, editors, *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, Washington, DC, USA, 2003. ISBN 1-58113-737-0.

K. Liu, W. Meng, J. Qiu, C. Yu, V. Raghavan, Z. Wu, Y. Lu, H. He, and H. Zhao. Allinonenews: development and evaluation of a large-scale news metasearch engine. In Chee Yong Chan, Beng Chin Ooi, and Aoying Zhou, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1017–1028, Beijing, China, 2007. ISBN 978-1-59593-686-8.

K. Liu, C. Yu, and W. Meng. Discovering the representative of a search engine. In Paques et al. [2001], pages 652–654. ISBN 1-58113-436-3.

W. Liu, X. Meng, and W. Meng. Vide: A vision-based approach for deep web data extraction. *IEEE Transactions on Knowledge and Data Engineering*, 22(3), 2010. ISSN 1041-4347.

J. Lu. *Full-text federated search in peer-to-peer networks.* PhD thesis, Carnegie Mellon University, 2007.

J. Lu. Efficient estimation of the size of text deep web data source. In Shanahan et al. [2008], pages 1485–1486. ISBN 978-1-59593-991-3.

J. Lu and J. Callan. Pruning long documents for distributed information retrieval. In Kalpakis et al. [2002], pages 332–339. ISBN 1-58113-492-4.

J. Lu and J. Callan. Content-based retrieval in hybrid peer-to-peer networks. In Kraft et al. [2003], pages 199–206. ISBN 1-58113-723-0.

J. Lu and J. Callan. Reducing storage costs for federated search of text databases. In *Proceedings of the 2003 Annual national Conference on Digital government research*, pages 1–6, Boston, MA, 2003b. Digital Government Research Center.

J. Lu and J. Callan. Federated search of text-based digital libraries in hierarchical peer-to-peer networks. In D. Losada and J. Fernández-Luna, editors, *Proceedings of the 27th European Conference on IR Research*, pages 52–66, Santiago de Compostela, Spain, 2005. Springer. ISBN 3-540-25295-9.

J. Lu and J. Callan. User modeling for full-text federated search in

peer-to-peer networks. In Efthimiadis et al. [2006], pages 332–339. ISBN 1-59593-369-7.

J. Lu and D. Li. Estimating deep web data source size by capture-recapture method. *Information Retrieval*, page to appear, 2009. ISSN 1386-4564.

J. Lu, Y. Wang, J. Liang, J. Chen, and J. Liu. An approach to deep web crawling by sampling. *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 1:718–724, 2008.

Y. Lu, W. Meng, L. Shu, C. Yu, and K. Liu. Evaluation of result merging strategies for metasearch engines. In A. Ngu, M. Kitsuregawa, E. Neuhold, J. Chung, and Q. Sheng, editors, *Proceedings of the Sixth International Conference on Web Information Systems Engineering*, volume 3806 of *Lecture Notes in Computer Science*, pages 53–66, New York, NY, 2005. Springer. ISBN 3-540-30017-1.

J. Madhavan, S. Jeffery, S. Cohen, X. Dong, D. Ko, C. Yu, and A. Halevy. Web-scale data integration: You can only afford to pay as you go. In *Proceedings of Conference on Innovative Data Systems Research*, pages 342–350, 2007.

J. Madhavan, D. Ko, Ł. Kot, V. Ganapathy, A. Rasmussen, and A. Halevy. Google's Deep Web crawl. *Proceedings of VLDB*, 1(2): 1241–1252, 2008.

U. Manber. Finding similar files in a large file system. In *Proceedings of the USENIX Winter Technical Conference*, pages 1–10, San Fransisco, CA, 1994. ISBN 1-880446-58-8.

U. Manber and P. Bigot. The search broker. In *USENIX Symposium on Internet Technologies and Systems*, Monterey, CA, 1997.

G. Marchionini, A. Moffat, J. Tait, R. Baeza-Yates, and N. Ziviani, editors. *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Salvador, Brazil, 2005. ISBN 1-59593-034-5.

Z. Mazur. On a model of distributed information retrieval systems based on thesauri. *Information Processing and Management*, 20(4): 499–505, 1984. ISSN 0306-4573.

W. Meng, Z. Wu, C. Yu, and Z. Li. A highly scalable and effective

method for metasearch. *ACM Transactions on Information Systems*, 19(3):310–335, 2001. ISSN 1046-8188.

W. Meng, C. Yu, and K. Liu. Building efficient and effective metasearch engines. *ACM Computing Surveys*, 34(1):48–89, 2002. ISSN 0360-0300.

D. Metzler and B. Croft. Latent concept expansion using markov random fields. In Kraaij et al. [2007], pages 311–318. ISBN 978-1-59593-597-7.

A. Moffat and J. Zobel. Information retrieval systems for large document collections. In D. Harman, editor, *Proceedings of the Third Text REtrieval Conference*, pages 85–94, Gaithersburg, MD, 1994. NIST Special Publication.

G. Monroe, J. French, and A. Powell. Obtaining language models of web collections using query-based sampling techniques. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 3*, pages 1241–1247, Honolulu, HI, 2002. IEEE Computer Society. ISBN 0-7695-1435-9.

G. Monroe, D. Mikesell, and J. French. Determining stopping criteria in the generation of web-derived language models. Technical report, University of Viginia, 2000.

V. Murdock and M. Lalmas. Workshop on aggregated search. *SIGIR Forum*, 42(2):80–83, 2008. ISSN 0163-5840.

W. Myaeng, D. Oard, F. Sebastiani, T. Chua, and M. Leong, editors. *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Singapore, 2008. ISBN 978-1-60558-164-4.

K. Ng. *An investigation of the conditions for effective data fusion in information retrieval*. PhD thesis, Rutgers University, 1998.

H. Nottelmann and N. Fuhr. Evaluating different methods of estimating retrieval quality for resource selection. In Clarke et al. [2003], pages 290–297. ISBN 1-58113-646-3.

H. Nottelmann and N. Fuhr. Combining CORI and the decision-theoretic approach for advanced resource selection. In Sharon McDonald and John Tait, editors, *Proceedings of the 26th European Conference on IR Research*, volume 2997 of *Lecture Notes in Computer Science*, pages 138–153, Sunderland, UK, 2004a. Springer.

ISBN 3-540-21382-1.

H. Nottelmann and N. Fuhr. Decision-theoretic resource selection for different data types in MIND. In Callan et al. [2004], pages 43–57. ISBN 3-540-20875-5.

H. Nottelmann and N. Fuhr. The MIND architecture for heterogeneous multimedia federated digital libraries. In Callan et al. [2004], pages 112–125. ISBN 3-540-20875-5.

P. Ogilvie and J. Callan. The effectiveness of query expansion for distributed information retrieval. In Paques et al. [2001], pages 183–190. ISBN 1-58113-436-3.

B. Oztekin, G. Karypis, and V. Kumar. Expert agreement and content based reranking in a meta search environment using mearf. In D. Lassner, D. Roure, and A. Iyengar, editors, *Proceedings of the 11th International Conference on World Wide Web*, pages 333–344, Honolulu, HI, 2002. ACM. ISBN 1-58113-449-5.

H. Paques, L. Liu, and D. Grossman, editors. *Proceedings of the ACM CIKM International Conference on Information and Knowledge Management*, Atlanta, GA, 2001. ISBN 1-58113-436-3.

J. Ponte and B. Croft. A language modeling approach to information retrieval. In Croft et al. [1998], pages 275–281. ISBN 1-58113-015-5.

A. Powell. *Database selection in Distributed Information Retrieval: A Study of Multi-Collection Information Retrieval.* PhD thesis, University of Virginia, 2001.

A. Powell and J. French. Comparing the performance of collection selection algorithms. *ACM Transactions on Information Systems*, 21(4):412–456, 2003. ISSN 1046-8188.

A. Powell, J. French, J. Callan, M. Connell, and C. Viles. The impact of database selection on distributed searching. In Belkin et al. [2000], pages 232–239. ISBN 1-58113-226-3.

W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical recipes in C: the art of scientific computing.* Cambridge University Press, New York, NY, 1988. ISBN 0-521-35465-X.

S. Raghavan and H. García-Molina. Crawling the hidden web. In Apers et al. [2001], pages 129–138. ISBN 1-55860-804-4.

Y. Rasolofo, F. Abbaci, and J. Savoy. Approaches to collection selection and results merging for distributed information retrieval. In Paques

et al. [2001], pages 191–198. ISBN 1-58113-436-3.

Y. Rasolofo, D. Hawking, and J. Savoy. Result merging strategies for a current news metasearcher. *Information Processing and Management*, 39(4):581–609, 2003. ISSN 0306-4573.

Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Schenker. A scalable content-addressable network. In *Proceedings of the Conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172, San Diego, CA, 2001. ACM.

M. Renda and U. Straccia. Metasearch: rank vs. score based rank list fusion methods (without training data). Technical report, Instituto di Elaborazione della Informazione - C.N.R., Pisa, Italy, 2002.

M. Renda and Umberto Straccia. Web metasearch: rank vs. score based rank aggregation methods. In G. Lamont, H. Haddad, G. Papadopoulos, and B. Panda, editors, *Proceedings of the ACM Symposium on Applied computing*, pages 841–846, Melbourne, FL, 2003. ISBN 1-58113-624-2.

S. Robertson. Relevance weighting of search terms. *Journal of the American Society for Information Sciences*, 27(3):129–146, 1976.

S. Robertson. The probability ranking principle in IR. In *Readings in information retrieval*, pages 281–286. Morgan Kaufmann, 1997. ISBN 1-55860-454-5.

S. Robertson and S. Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In B. Croft and K. Rijsbergen, editors, *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 232–241, Dublin, Ireland, 1994. ACM/Springer. ISBN 3-540-19889-X.

G. Salton, E. Fox, and E. Voorhees. A comparison of two methods for boolean query relevance feedback. In *Technical report, Cornell University*, Ithaca, NY, 1983.

G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY, 1986. ISBN 0070544840.

J. Savoy, A. Calvé, and D. Vrajitoru. Information retrieval systems for large document collections. In E. Voorhees and D. Harman, editors,

Proceedings of the Fifth Text REtrieval Conference, pages 489–502, Gaithersburg, MD, 1996. NIST Special Publication.

F. Schumacher and R. Eschmeyer. The estimation of fish populations in lakes and ponds. *Journal of the Tennesse Academy of Science*, 18: 228–249, 1943.

E. Selberg and O. Etzioni. Multi-service search and comparison using the metacrawler. In *Proceedings of the Fourth International Conference on World Wide Web*, Boston, MA, 1995. Oreilly. ISBN 978-1-56592-169-6.

E. Selberg and O. Etzioni. The MetaCrawler architecture for resource aggregation on the web. *IEEE Expert*, 12(1):8–14, 1997a. ISSN 0885-9000.

E. Selberg and O. Etzioni. The MetaCrawler architecture for resource aggregation on the web. *IEEE Expert, January–February*, pages 11–14, 1997b. ISSN 0885-9000.

J. Seo and B. Croft. Blog site search using resource selection. In Shanahan et al. [2008], pages 1053–1062. ISBN 978-1-59593-991-3.

J. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D. Evans, A. Kolcz, K. Choi, and A. Chowdhury, editors. *Proceedings of the ACM CIKM International Conference on Information and Knowledge Management*, Napa Valley, CA, 2008. ISBN 978-1-59593-991-3.

V. Shen, C. Saito, C. Lyu, and M. Zurko, editors. *Proceedings of the 10th International Conference on World Wide Web*, Hong Kong, China, 2001. ACM. ISBN 1-58113-348-0.

Y. Shen and D. Lee. A meta-search method reinforced by cluster descriptors. In M. Özsu, H. Schek, K. Tanaka, Y. Zhang, and Y. Kambayashi, editors, *Proceedings of the Second International Conference on Web Information Systems Engineering*, pages 125–132, Kyoto, Japan, 2001. IEEE Computer Society. ISBN 0-7695-1393-X.

M. Shokouhi. Central-rank-based collection selection in uncooperative distributed information retrieval. In G. Amati, C. Carpineto, and G. Romano, editors, *Proceedings of the 29th European Conference on Information Retrieval Research*, volume 4425 of *Lecture Notes in Computer Science*, pages 160–172, Rome, Italy, 2007a. Springer.

M. Shokouhi. Segmentation of search engine results for effective data-fusion. In G. Amati, C. Carpineto, and G. Romano, editors, *Pro-*

ceedings of the 29th European Conference on Information Retrieval Research, volume 4425 of *Lecture Notes in Computer Science*, pages 185–197, Rome, Italy, 2007b. Springer.

M. Shokouhi, M. Baillie, and L. Azzopardi. Updating collection representations for federated search. In Kraaij et al. [2007], pages 511–518. ISBN 978-1-59593-597-7.

M. Shokouhi, F. Scholer, and J. Zobel. Sample sizes for query probing in uncooperative distributed information retrieval. In X. Zhou, J. Li, H. Shen, M. Kitsuregawa, and Y. Zhang, editors, *Proceedings of Eighth Asia Pacific Web Conference*, pages 63–75, Harbin, China, 2006a. ISBN 3-540-31142-4.

M. Shokouhi, P. Thomas, and L. Azzopardi. Effective query expansion for federated search. In Allan et al. [2009], pages 427–434. ISBN 978-1-60558-483-6.

M. Shokouhi and J. Zobel. Federated text retrieval from uncooperative overlapped collections. In Kraaij et al. [2007], pages 495–502. ISBN 978-1-59593-597-7.

M. Shokouhi and J. Zobel. Robust result merging using sample-based score estimates. *ACM Transactions on Information Systems*, 27(3): 1–29, 2009. ISSN 1046-8188.

M. Shokouhi, J. Zobel, and Y. Bernstein. Distributed text retrieval from overlapping collections. In J. Bailey and A. Fekete, editors, *Proceedings of the 18th Australasian Database Conference*, volume 63 of *CRPIT*, pages 141–150, Ballarat, Australia, 2007b. ACS.

M. Shokouhi, J. Zobel, and Y. Bernstein. Distributed text retrieval from overlapping collections. In J. Bailey and A. Fekete, editors, *Proceedings of the Australasian Database Conference*, pages 141–150, Ballarat, Australia, 2007c.

M. Shokouhi, J. Zobel, F. Scholer, and S. Tahaghoghi. Capturing collection size for distributed non-cooperative retrieval. In Efthimiadis et al. [2006], pages 316 – 323. ISBN 1-59593-369-7.

M. Shokouhi, J. Zobel, S. Tahaghoghi, and F. Scholer. Using query logs to establish vocabularies in distributed information retrieval. *Information Processing and Management*, 43(1):169–180, 2007d. ISSN 0306-4573.

X.M. Shou and M. Sanderson. Experiments on data fusion using head-

line information. In Järvelin et al. [2002], pages 413–414. ISBN 1-58113-561-0.

S. Shushmita, H. Joho, M. Lalmas, and R. Villa, editors. *Proceedings of the ACM CIKM International Conference on Information and Knowledge Management*, Toronto, Canada, 2010.

L. Si. *Federated search of text search engines in uncooperative environments*. PhD thesis, Carnegie Mellon University, 2006.

L. Si and J. Callan. Using sampled data and regression to merge search engine results. In Järvelin et al. [2002], pages 19–26. ISBN 1-58113-561-0.

L. Si and J. Callan. Relevant document distribution estimation method for resource selection. In Clarke et al. [2003], pages 298–305. ISBN 1-58113-646-3.

L. Si and J. Callan. A semisupervised learning method to merge search engine results. *ACM Transactions on Information Systems*, 21(4): 457–491, 2003b. ISSN 1046-8188.

L. Si and J. Callan. The effect of database size distribution on resource selection algorithms. In Callan et al. [2004], pages 31–42. ISBN 3-540-20875-5.

L. Si and J. Callan. Unified utility maximization framework for resource selection. In D. Grossman, L. Gravano, C. Zhai, O. Herzog, and D. Evans, editors, *Proceedings of the ACM CIKM International Conference on Information and Knowledge Management*, pages 32–41, Washington, DC, 2004b. ISBN 1-58113-874-1.

L. Si and J. Callan. CLEF2005: multilingual retrieval by combining multiple multilingual ranked lists. In *the Sixth Workshop of the Cross-Language Evaluation Forum*, Vienna, Austria, 2005a. URL `http://www.cs.purdue.edu/homes/lsi/publications.htm`.

L. Si and J. Callan. Modeling search engine effectiveness for federated search. In Marchionini et al. [2005], pages 83–90. ISBN 1-59593-034-5.

L. Si, J. Callan, S. Cetintas, and H. Yuan. An effective and efficient results merging strategy for multilingual information retrieval in federated search environments. *Information Retrieval*, 11(1):1–24, 2008.

L. Si, R. Jin, J. Callan, and P. Ogilvie. A language modeling framework

for resource selection and results merging. In Kalpakis et al. [2002], pages 391–397. ISBN 1-58113-492-4.

A. Smeaton and F. Crimmins. Using a data fusion agent for searching the WWW. In P. Enslow, M. Genesereth, and A. Patterson, editors, *Selected papers from the Sixth International Conference on World Wide Web*, Santa Clara, CA, 1997. Elsevier. Poster Session.

M. Sogrine, T. Kechadi, and N. Kushmerick. Latent semantic indexing for text database selection. In *Proceedings of the SIGIR 2005 Workshop on Heterogeneous and Distributed Information Retrieval*, pages 12–19, 2005. URL `http://hdir2005.isti.cnr.it/index.html`.

A. Spink, B. Jansen, C. Blakely, and S. Koshman. A study of results overlap and uniqueness among major web search engines. *Information Processing and Management*, 42(5):1379–1391, 2006. ISSN 0306-4573.

I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking*, 11(1):17–32, 2003. ISSN 1063-6692.

A. Sugiura and O. Etzioni. Query routing for web search engines: architectures and experiments. In Herman and Vezza [2000], pages 417–429. ISBN 1-930792-01-8.

P. Thomas. Generalising multiple capture-recapture to non-uniform sample sizes. In Myaeng et al. [2008], pages 839–840. ISBN 978-1-60558-164-4.

P. Thomas. *Server characterisation and selection for personal metasearch*. PhD thesis, Australian National University, 2008b.

P. Thomas and D. Hawking. Evaluating sampling methods for uncooperative collections. In Kraaij et al. [2007], pages 503–510. ISBN 978-1-59593-597-7.

P. Thomas and D. Hawking. Experiences evaluating personal metasearch. In *Proceedings of the second international Symposium on Information Interaction in Context*, pages 136–138, London, UK, 2008. ACM. ISBN 978-1-60558-310-5.

P. Thomas and D. Hawking. Server selection methods in personal metasearch: a comparative empirical study. *Information Retrieval*,

12(5):581–604, 2009. ISSN 1386-4564.

P. Thomas and M. Shokouhi. SUSHI: scoring scaled samples for server selection. In Allan et al. [2009], pages 419–426. ISBN 978-1-60558-483-6.

G. Towell, E. Voorhees, K. Narendra, and B. Johnson-Laird. Learning collection fusion strategies for information retrieval. In A. Prieditis and S. Russell, editors, *Proceedings of The 12th International Conference on Machine Learning*, pages 540–548, Lake Tahoe, CA, 1995. ISBN 1-55860-377-8.

T. Tsikrika and M. Lalmas. Merging techniques for performing data fusion on the web. In Paques et al. [2001], pages 127–134. ISBN 1-58113-436-3.

H. Turtle. *Inference networks for Document Retrieval*. PhD thesis, University of Massachusetts, 1991.

H. Turtle and B. Croft. Inference networks for document retrieval. In J. Vidick, editor, *Proceedings of the 13th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1–24, Brussels, Belgium, 1990. ISBN 0-89791-408-2.

C. Vogt. *Adaptive combination of evidence for information retrieval*. PhD thesis, University of California, San Diego, 1999.

C. Vogt and G. Cottrell. Fusion via a linear combination of scores. *Information Retrieval*, 1(3):151–173, 1999. ISSN 1386-4564.

E. Voorhees, N. Gupta, and B. Johnson-Laird. Learning collection fusion strategies. In Fox et al. [1995], pages 172–179. ISBN 0-89791-714-6.

E. Voorhees and R. Tong. Multiple search engines in database merging. In R. Allen and Edie Rasmussen, editors, *Proceedings of the Second ACM Conference on Digital Libraries*, pages 93–102, Philadelphia, PA, 1997. ISBN 0-89791-868-1.

Y. Wang and D. DeWitt. Computing PageRank in a distributed internet search engine system. In M. Nascimento, M. Özsu, D. Kossmann, R. Miller, J. Blakeley, and K. Schiefer, editors, *Proceedings of the 30th International Conference on Very Large Data Bases*, pages 420–431, Toronto, Canada, 2004. Morgan Kaufmann. ISBN 0-12-088469-0.

C. Williamson, M. Zurko, P. Patel-Schneider, and P. Shenoy, editors.

Proceedings of the 16th International Conference on World Wide Web, Alberta, Canada, 2007. ACM. ISBN 978-1-59593-654-7.

S. Wu and F. Crestani. Multi-objective resource selection in distributed information retrieval. In Kalpakis et al. [2002], pages 1171–1178. ISBN 1-58113-492-4.

S. Wu and F. Crestani. Distributed information retrieval: a multi-objective resource selection approach. *International Journal of Uncertainty, Fuzziness Knowledge-Based Systems*, 11(supp01):83–99, 2003. ISSN 0218-4885.

S. Wu and F. Crestani. Shadow document methods of resutls merging. In H. Haddad, A. Omicini, R. Wainwright, and L. Liebrock, editors, *Proceedings of the ACM symposium on Applied computing*, pages 1067–1072, Nicosia, Cyprus, 2004. ISBN 1-58113-812-1.

S. Wu and S. McClean. Performance prediction of data fusion for information retrieval. *Information Processing and Management*, 42 (4):899–915, 2006. ISSN 0306-4573.

Z. Wu, W. Meng, C. Yu, and Z. Li. Towards a highly-scalable and effective metasearch engine. In Shen et al. [2001], pages 386–395. ISBN 1-58113-348-0.

J. Xu and J. Callan. Effective retrieval with distributed collections. In Croft et al. [1998], pages 112–120. ISBN 1-58113-015-5.

J. Xu and B. Croft. Query expansion using local and global document analysis. In H. Frei, D. Harman, P. Schäuble, and R. Wilkinson, editors, *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11, Zurich, Switzerland, 1996. ISBN 0-89791-792-8.

J. Xu and B. Croft. Cluster-based language models for distributed retrieval. In Gey et al. [1999], pages 254–261. ISBN 1-58113-096-1.

J. Xu, S. Wu, and X. Li. Estimating collection size with logistic regression. In Kraaij et al. [2007], pages 789–790. ISBN 978-1-59593-597-7.

H. Yang and M. Zhang. Ontology-based resource descriptions for distributed information sources. In X. He, T. Hintza, M. Piccardi, Q. Wu, M. Huang, and D. Tien, editors, *Proceedings of the Third International Conference on Information Technology and Applications*, volume I, pages 143–148, Sydney, Australia, 2005. IEEE Computer Society. ISBN 0-7695-2316-1.

H. Yang and M. Zhang. Two-stage statistical language models for text database selection. *Information Retrieval*, 9(1):5–31, 2006. ISSN 1386-4564.

C. Yu, K. Liu, W. Meng, Z. Wu, and N. Rishe. A methodology to retrieve text documents from multiple databases. *IEEE Transactions on Knowledge and Data Engineering*, 14(6):1347–1361, 2002. ISSN 1041-4347.

C. Yu, W. Meng, K. Liu, W. Wu, and N. Rishe. Efficient and effective metasearch for a large number of text databases. In Gauch [1999], pages 217–224. ISBN 1-58113-1461.

C. Yu, W. Meng, W. Wu, and K. Liu. Efficient and effective metasearch for text databases incorporating linkages among documents. *SIGMOD Records*, 30(2):187–198, 2001. ISSN 0163-5808.

B. Yuwono and D. Lee. WISE: A world wide web resource database system. *IEEE Transactions on Knowledge and Data Engineering*, 8 (4):548–554, 1996. ISSN 1041-4347.

B. Yuwono and D. Lee. Server ranking for distributed text retrieval systems on the internet. In R. Topor and K. Tanaka, editors, *Proceedings of the Fifth International Conference on Database Systems for Advanced Applications*, volume 6 of *Advanced Database Research and Development Series*, pages 41–50, Melbourne, Australia, 1997. World Scientific. ISBN 981-02-3107-5.

O. Zamir and O. Etzioni. Grouper: a dynamic clustering interface to web search results. *Computer Networks and ISDN Systems*, 31(11–16):1361–1374, 1999. ISSN 1389-1286.

H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu. Fully automatic wrapper generation for search engines. In Ellis and Hagino [2005], pages 66–75. ISBN 1-59593-046-9.

H. Zhao, W. Meng, and C. Yu. Automatic extraction of dynamic record sections from search engine result pages. In Umeshwar Dayal, Kyu-Young Whang, David B. Lomet, Gustavo Alonso, Guy M. Lohman, Martin L. Kersten, Sang Kyun Cha, and Young-Kuk Kim, editors, *Proceedings of the 30th International Conference on Very Large Data Bases*, pages 989–1000, Seoul, Korea, 2006. Morgan Kaufmann. ISBN 1-59593-385-9.

Hongkun Zhao, Weiyi Meng, and Clement Yu. Mining templates from

search result records of search engines. In Pavel Berkhin, Rich Caruana, and Xindong Wu, editors, *Proceedings of the 13th Annual International ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 884–893, San Jose, California, USA, 2007. ISBN 978-1-59593-609-7.

J. Zobel. Collection selection via lexicon inspection. In P. Bruza, editor, *Proceedings of the Australian Document Computing Symposium*, pages 74–80, Melbourne, Australia, 1997.