Knowledge and Information Systems (2023) 65:5065–5095
https://doi.org/10.1007/s10115-023-01922-6

REVIEW

Federated search techniques: an overview of the trends and state of the art

Adamu Garba[1] · Shengli Wu[1,2] · Shah Khalid[3]

Received: 18 November 2022 / Revised: 26 April 2023 / Accepted: 21 June 2023 /
Published online: 10 July 2023

Abstract
Conventional search engines, such as Bing, Baidu, and Google, offer a convenient way for users to seek information on the web. However, with all the benefits they provide, one major limitation is that a sizable portion of the information sources on the web may not be available due to commercial or proprietary reasons. Federated search solves this problem by providing a single user interface through which multiple independent resources can be searched and their results are combined for end users. Up to now, federated search has become a well-established research area, with many systems developed and algorithms proposed to deal with three major issues: resource description, resource selection, and results merging. This paper reviews state-of-the-art federated search techniques developed over the past three decades, with more attention to recent achievement. Both resource selection and result merging methods are categorized into three types, heuristic, machine learning-based, and other methods. Apart from the three major issues above-mentioned, we also discuss systems and prototypes developed, and datasets used for federated search experiments. Some other related issues including retrieval evaluation, aggregated search, metasearch, supporting personalization in federated search, are also covered. Finally, we conclude by discussing some directions for future research.

Keywords Federated search · Distributed information retrieval · Resource selection · Result merging · Aggregated search · Metasearch

✉ Adamu Garba
yakasai6@yahoo.com

Shengli Wu
swu@ujs.edu.cn

Shah Khalid
shah.khalid@seecs.edu.pk

1 School of Computer Science and Communication Engineering, Jiangsu University, 301 Xuefu Road, Zhenjiang 212013, China

2 School of Computing, Ulster University, 2-24 York Street, Belfast BT15 1AP, UK

3 Department of Computing, National University of Sciences and Technology, H12, Islamabad 44000, Pakistan

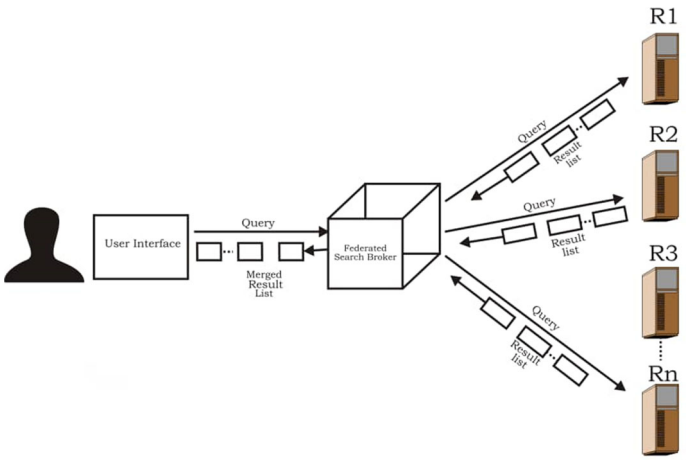

Fig. 1 Processes involved in a federated search system

1 Introduction

Conventional search engines such as Bing,[1] Baidu,[2] Google,[3] and others play a very important role for providing useful information to end users in the today's information age. There is, however, one major limitation to these search engines. A lot of resources are not available for them to access due to proprietary or commercial reasons [1]. Many digital libraries and news blogs are in this category. For example, ACM and IEEE digital libraries are digital libraries of scientific publications on computing and electric & electronics engineering, respectively, while Bloomberg is an authoritative financial news blogs. Full access to them is only granted to legitimate users with login credentials. Some of these information resources, scattered on the web, are either ignored or not known to many web users. Targeting these information sources, federated search makes them accessible to more web users [2, 3]. This is achieved by providing a single search interface that is capable of simultaneously forwarding a user query to multiple independent resources and merging their returned result lists into a single list for end users [4]. In the real-world web, site like Priceline[4] is federated in nature with hundreds of available resources in the back-end.

In federated search, there are three major research problems: First, *"resource description"* concerns the contents of the resource and other information such as the size of the collection and the overlapping rates between two or more collections. Second, *"resource selection"* concerns how to select a group of most useful resources for a given user query. Finally, *"result merging"* is about how to fuse the result lists returned from the multiple resources. These three are also connected in some way. Both tasks of resource selection and result merging need to know information of all component resources (resource description) for a proper decision.

As shown in Fig. 1, the typical processes involved in a federated search system are as follows:

[1] http://www.bing.com/.

[2] http://www.baidu.com/.

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

[4] https://www.priceline.com/.

- The user inputs her query on her machine via a federated search interface for end users.
- The query is transferred to the central broker.
- The broker selects a group of relevant resources and forwards the query to those selected.
- Each selected resource does the retrieval and returns a list of documents to the broker.
- The broker merges results coming from multiple resources to be a single list and send them to the user machine.
- The result list is displayed on the user's machine.

Research on federated search has been conducted in two different scenarios: cooperative and uncooperative environments. In a cooperative environment, resources agree to share some vital information about their contents and corpus statistics with the broker. This is the case for some contexts such as enterprise search. While on the web, many resources are independent and autonomous, and treat the broker as an ordinary user. Apart from that, it may not be possible for such resources to share any extra information with the broker.

STARTS [5] is among the early studies that proposed protocols for a cooperative environment. These protocols define the information and modes of communicating it between all the resources involved and the broker. Similarly, GlOSS [6] is also an early work in the literature that proposed a methodology for identifying the most relevant resources to search for a given user query based on the relevant documents within the resources.

On the other hand, in an uncooperative environment, a key issue is how to obtain accurate information about all the resources. The general idea is to get that through the communication channel as an ordinary user. Query-based sampling [7] and some variants [8, 9] have been proposed in the literature. In this way, the broker can still know some useful information about the resources and then utilize them for resource selection and result merging tasks.

Over the last three decades, considerable progress on federated search has been made. However, to our knowledge, there is only one literature review paper on federated search so far [10]. Although it is very good and comprehensive, it was published over a decade ago. Therefore, we think it is desirable to have a new review paper. The purpose of this paper is to provide a general picture of the major work in federated search over the years, with more attention to recent research work and related activities (such as workshops and evaluation events).

The remaining parts of this paper are organized as follows: Sect. 2 describes survey methodology and selected papers. Sections 3, 4, and 5 explain the methods proposed for resource representation, resource selection, and result merging, along with their benefits and drawbacks, respectively. Section 6 discusses some federated search systems and research prototypes developed. Section 7 discusses the data sets used for performance evaluation of federated search systems or/and their components. Section 8 presents some related research issues including retrieval evaluation, aggregated search, metasearch, and personalizing federated search. Finally, Sect. 9 concludes the paper with some future research directions.

2 Survey methodology and selected papers

2.1 Selection criteria

Shokouhi and Si's review paper [10] was published in 2011, and all the papers reviewed were published in 2010 or before. Therefore, we try to include all relevant papers that were published since 2011. While for those papers published before 2011, we do not try to include all of them, but instead some representative papers considering quality and addressed problems.

We used DBLP and Google scholar to do the search work, in which "distributed information retrieval," "federated search," "federated retrieval" were used as keyword queries. We also checked the reference lists of some selected papers to find more relevant papers. All the papers downloaded were manually checked their relevance for final selection.

However, there are a few exceptions as follows:

1. For those reports submitted to 2013 and 2014 TREC FedWeb tracks, only those of three best-performing runs per year are included.
2. For both topics "aggregated search" and "metasearch," only some representative papers are included.

2.2 Selected papers

Through the selection process, we identified approximately 122 articles that satisfied our set criteria. Among the identified articles, resource selection emerged as the most frequently published topic, with over 55 articles directly related to it. Conversely, security issues had the least number of articles, with only four published pieces. The number of articles addressing the other federated search interconnected problems ranged from 7 to 15.

2.3 Discussion

Despite the fact that conventional search engines are the primary tools used by most web users to locate information on the web, a substantial portion of the web's content is not entirely accessible through these search engines. For instance, Google reported discovering over 30 trillion URLs in 2012,[5] but a study conducted over a period of nine years from 2006 to 2015, as presented in [11], indicates that the Google index's total size was only 45.7 billion documents as of January 2015. Various studies, including those reported in [12] and [13], demonstrate that relying solely on search engines causes web users to miss out on a significant number of relevant documents that are exclusively available from specialized information sources. Federated search provides a solution to this issue by targeting these information sources and linking them directly to web users through a single interface. As a result, web users are able to search multiple independent resources through a single interface, rather than having to search them individually.

To ensure optimal performance of federated search systems, it is crucial for the broker to forward user query to the most relevant resources and merge the results based on their relevance to the query. As such, researchers have identified three interrelated problems that must be addressed for federated search to function properly. These problems are resource representation, resource selection, and results merging.

3 Resource representation

For the broker to function properly, it needs a lot of information for every resource so as to perform resource selection and result merging tasks. There are two typical scenarios. One is that all the resources are cooperative and willing to provide comprehensive information if required by the broker. Then, a special channel between the two parties can be set up for

[5] https://searchengineland.com/google-search-press-129925.

this. The other is some or all the resources are independent and uncooperative. In such a situation, the broker has to collect some useful information through the resource's ordinary communication channel for end users. In the following, let us discuss them one by one.

3.1 Cooperative environment

In a cooperative environment, the resources agreed to exchange the information required by the broker to perform searching and merging accurately via an established protocol. Enterprise search is an example of a federated search that works in a cooperative environment. As both the resources and the interface are owned and maintained by the same entity. Therefore, the resources provide the broker with details of their metadata, such as document frequency, list of stop words, number of terms in each document, and total terms in the collection as a whole [14]. A cooperative resource discovery [15] is another proposed method for the cooperative environment in which each resource provides the broker with the number of terms and the resources in which those terms appear. However, for resources with diverse content, different sets of metadata are required by the broker to function effectively. For this reason, [16] considered previous query logs as metadata to enhance vertical selection. There is, however, a drawback to the cooperative method, which is that it may not be workable in a real-world web environment, where most resources are owned by different entities.

3.2 Uncooperative environment

In contrast, in an uncooperative environment, which is a typical situation for the Web, no standardization is implemented for resources to provide detailed information about their corpus statistics to the broker [10]. As a result, a widely used query-based sampling strategy (QBS) [7] is used to sample a sufficient number of documents from each resource and index them in what is referred to as a centralized sample database (CSD). In QBS, a single-term query chosen from either a reference dictionary or resource search interface is issued to the resources. The top n documents returned by them are downloaded and indexed in CSD. The next query is selected from the terms of the sampled documents. The sampling process continues until a stoppage criterion is reached, which is mostly about 300 distinct documents. The majority of the uncooperative environment research proposed in the last three decades used this method to obtain representative documents [10, 17]. CSD is used for both resource selection and result merging. Limiting the number of sampled documents to 300 has the drawback of oversampling resources with small content while under-sampling those with a large of content.

3.3 Estimation of resource size

Estimating of resources size at a certain interval yields information about freshness, quality of content, and resources with most diverse contents [22]. Further, the size of the resource is one of the factors used in determining the most relevant resource to search in most of the resource selection algorithms proposed [23, 24]. However, in an uncooperative environment, the resources corpus size is not available to the broker. For this reason, various methodologies were proposed in this regard. Among them, consider the query pool method [18] as an example. This method estimates the size of the resources by randomly selecting a term from a dictionary, issues it as a query to resources and then downloading all the matched

documents to an index. Afterward, extract the terms with the highest document frequency to form the query pool. Next, they select the terms in the query pool one by one and issue it as query to the downloaded index, and then harvested all the documents with distinct ids. Similarly, the sample–resample methods proposed in [24] estimate the size of the resource by selecting a single term query from the centralized sample database and issue it to a resource. The next query is selected from the downloaded documents of the previous query. This process continues until the predefined criteria are met. The resource size is estimated using the following equation:

$$R_{\text{Size}} = \frac{R_{\text{dfqi}} \times R_{\text{sample}}}{R_{\text{dqisample}}} \tag{1}$$

where R_{dfqi} is the number of documents from resource R that contain query qi, R_{Sample} is the number of documents sampled from resource R, and $R_{\text{dqisample}}$ is the number documents sampled from R that contain qi.

Other methods proposed in the literature include: random sampling [20], uncorrelated terms query [21], and the capture–recapture method [19].

However, most of the aforementioned methods [9, 19, 24] were based on random samples. Nguyen et al. [25] argued that these approaches proposed based on random samples in most cases contain noisy data. Therefore, they proposed the reference corpus method of resource size estimation. This method used the ClueWeb09 dataset as the reference corpus. For the given queries Q, Eq. 2 estimates resource size:

$$S_{\text{size}} = \frac{1}{|Q|} \sum_{q \in Q} \frac{R_s}{\text{df}_q} \times |ClueWeb_{\text{size}}| \tag{2}$$

where $|Q|$ is the query size, R_S is the number of documents resource R return for a particular query q, df_q is the ClueWeb documents frequency for query q, and $|ClueWeb\_\text{size}|$ is the total size of the ClueWeb collection.

A summary of some selected studies on resource description and corpus size estimation is presented in Table 1.

3.4 Estimation of resource overlapping rates

Resource overlap rates refer to the extent to which two or more resources share the same or similar documents. In a federated environment, it is essential to estimate the extent to which the contents of the resources overlap. This is because searching different resources that return similar documents not only wastes the search user time but also degrades search effectiveness [26]. As such, Bernstein et al. [27] proposed a method based on hash vectors that detects and discards similar and near-similar documents from the merged result list in a cooperative environment setting. For an uncooperative environment, Shokouhi and Zobel [26] used the sampled documents in the centralized sample index to estimate the overlapping rate between two resources. That is, the number of similar documents within two different resources can be estimated using the following equation:

$$K = \frac{|R_1||R_2| \times D}{|\text{Sr}_1||\text{Sr}_2|} \tag{3}$$

This equation estimates the number of similar documents between two resources using overlap documents K, sampled documents Sr_1 and Sr_2 from resources R_1 and R_2, and expected similar documents D.

Table 1 Summary of some selected published studies for the resource description and corpus size estimation

S/N	Paper	Year	Dataset	Method overview
1	Query-based sampling of text databases [7]	2001	TREC-123, WSJ88, and CACM	This method discovers the content of the resources by issuing a single-term query and downloading the same number of documents from the resources, and then index them in the centralized sample database
2	Adaptive query-based sampling of distributed collections [8]	2006	TREC WT10g and Aquaint	This method uses predictive likelihood method to estimate the size and the number of documents to be sampled from each resource
3	Capturing collection size for distributed non-cooperative retrieval [9]	2006	TREC WT10g, TREC.GOV, and Dateline 509	This method used the ecological method of multiple capture–recapture technique to estimate the resource size
4	Estimating the size of hidden data sources by queries [18]	2014	TREC GOV2, Newswire, and Newsgroup	In this method, a query pool is constructed from the downloaded sample documents and then uses query weights to estimate the resource size
5	Estimating deep web data source size by capture–recapture method [19]	2010	TREC GOV2, Newswire, and Wikipedia	Based on the capture-recapture method, this method estimates the resource size using duplicates between successive samples of documents
6	Efficient estimation of text deep web data source [20]	2008	Newswire and Newsgroup	A random sampling of documents from each resource is used to estimate the size of each resource based on the duplicate document IDs
7	Estimating corpus size via queries [21]	2006	TREC GOV7	Based on random sampling from the resources corpus, this method estimates the resource size by using uncorrelated terms as queries

4 Resource selection

In federated search, it is not a good policy for the broker to forward the received query to all the participating resources, as some may not be relevant to that given query. As such, resource selection is necessary for the broker to select only those with a high probability of returning relevant documents. This section reviews and categorizes the well-known resource selection methods proposed in the literature.

4.1 Heuristic methods

The heuristic methods rely on the lexicon statistics either obtained or provided by the resources. Most of the early studies consider each resource as big document. That is, the boundaries of the documents in each information resource are collapsed to form a single big document that contains only a bag of words. Upon receiving the user's query, the broker computes the query similarity with the lexicon statistics of each information resource and ranks them based on their relevance score. The big document approach includes CORI [28], GlOSS [6], and CVV [15]. The studies in [24, 44] reported that CORI [28] is the most effective and straightforward resource selection method in the literature. A Bayesian inference network is used in CORI to calculate the relevance score of each information resource for a given query. However, the limitation of the big document approach is that, by removing the boundaries of the separate documents, the relevance of the individual document cannot be ascertained; instead, only the resource's overall relevance to the given query can be estimated.

On the other hand, the models proposed by [23, 24, 45, 46] move away from collapsing the document boundaries as applied in the big document approach. Rather, they consider each resource as a collection of documents, and the relevance of a resource is estimated based on the relevance of its constituent documents. CRCS [23] is a resource selection algorithm based on the small document approach. In CRCS, the broker issues the user query to CSD. The number of documents and their ranking positions in the top k of the generated ranking are used to determine the relevance of a resource to that given query. As such, the relevance score of resource R_i is computed using Eq. 4, described as follows:

$$S\left(R_i\right) = \frac{S_i}{S_{\max} \times S_s} \times \sum_{d \in S_s} S(d) \tag{4}$$

where S_i is the estimated size of resource i, $S_{\max}$ is the estimated size of the largest resource, S_s is the number of documents sampled from resource i during the sampling phase, and S(d) is the contribution of document d to the weight of the resource that returned it. The S(d) value is computed either linearly or exponentially as shown in Eqs. 5 and 6.

$$S\left(d\right) = \begin{cases} k - l, \; if \; l < k \\ 0, \qquad \text{Otherwise} \end{cases} \tag{5}$$

$$S\left(d\right) = \beta \; exp(-\gamma \; \bullet \; l) \tag{6}$$

where k is the top documents in the CSD ranked list, which is set to 50, l is the rank of the document r_j in the CSD ranked list, and β and γ are the constant parameters whose values are set to 1.2 and 0.28. ReDDE [24] is considered the most common resource selection algorithm based on the small-document approach [47]. In ReDDE, the relevance of a resource to a given query is estimated based on the number of documents that particular resource has in the top k results when the query is run on the CSD.

The Text Retrieval Conference (TREC) recently built large collections of documents gathered from real-world search engines in order to facilitate research on federated search using a dataset similar to the real-world federated environment. Numerous approaches were proposed for the resource selection task in both 2013 and 2014 TREC FedWeb tracks [48, 49].

In the approach proposed by [31], a term-weighted frequency scheme was used to select the relevant resources for the given queries. Their approach considered each search engine as a collection of document descriptors, e.g., terms, and the relevance score of a search engine (resource) for a given query is obtained based on the number of documents and the number

of query terms that appear in such documents. The approach proposed in [50] ranked the resources based on their relevance as well as the opinion of the given query. Furthermore, in [51], they use the Google search API in computing the relevance score of the resources. The search engine impact factor (SEIF) method was proposed in [34]. In this method, the sources are ranked based on their popularity or market share. They assumed that the most popular search engines (Google, Bing, Baidu, etc.) would contain more relevant documents than the non-popular ones. Although their method is independent of a user query, it is the best performing method in the 2014 TREC FedWeb track [48]. In the model proposed in [35], all the documents for each search engine (resource) provided in the dataset are concatenated into a single big document. Then, the topic model of each resource and the given queries is obtained using latent Dirichlet allocation (LDA). The resources are ranked based on the number of topics they share with each given query. The methodology proposed in [32] employs the Tally statistical method [52] for resource selection. According to their submission, keeping representative documents in CSD is expensive. It is easier to handle if you preserve the term-related features. Thus, they extracted each resource's terms' features and computed the relevance score of the sources based on these terms' features. Recently, Urak et al. [41] argued that using the SEIF model [34] to select the relevant resources would repeatedly choose the same resources since the search engine market is dominated by giants such as Google, Baidu, and Bing. Based on this observation, they proposed a method that includes the long tail resources among the selected resources for a given query. With long tail resources, the user who issued the query can explore documents from other smaller relevant resources. Thus, Eq. 7 is used to select the final resources to search for the given query q.

$$S_f\,(q,\ s) = (1-\delta)\,S_{\text{best}}\,(q,\ s) + \ \delta S_{\text{tail}}(q,\ s) \tag{7}$$

where δ is the control parameter with a range value between 0 and 1. This parameter is used to ensure that the final selected resources are balanced.

4.2 Machine learning-based methods

All of the above-mentioned resource selection methodologies use traditional document query similarity in selecting the relevant resources to search. Machine learning techniques have recently proven to be a viable alternative to traditional methods of computing document query similarity. To this effect, various machine learning methods have been proposed for resource selection in the literature. Arguello et al. [29] extracted three types of features, namely collection features, query topic features, and click-through features, and trained a classifier for resource selection. A joint probabilistic model that estimates a source relevance based on its similarity with the already selected resource was proposed by [30]. Xu and Li [53] postulated that using more features can improve the performance of the collection selection algorithms. As such, they proposed a method that used two separate sets of features, query-dependent and query-independent, and then combined them to form the query-collection features vector. SVMrank [54] was used to learn a ranking function of all the resources. Similarly, in [38], three different sets of features, query-independent, term-based, and sampled-documents, were used for resource ranking in selective search. Wu et al., [42] proposed the LTRRS algorithm, which combined all the features proposed in [38] in addition to the topic relevance feature introduced in their paper. They used LambdaMART [55] to train the function for ranking the resources.

As previously stated, the broker must search the CSD for each query received in order to determine the most relevant resources for that given query. But this process is considered

repetitive and bandwidth-consuming [46]. Therefore, Garba et al. [43] recently proposed an embedding base model for resource selection that utilizes past queries. In particular, for each current user query received by the broker, its similarity with the past queries that reside in the query log is obtained. Then, resources selected for the past queries that are similar to the current query are reselected for search. Specifically, let $S_k = \{s_1, s_2, ..., s_m\}$ be the set of resources with the indexed documents in the CSD. Let also $Q_p = \{q_1, q_2, ..., q_n\}$ be the set of the past queries. The similarity between the current query and each past query stored in the query log sim (q, q_l) is estimated by computing the cosine similarity of their term vectors using a word embedding technique as explicated in Eq. 8:

$$\text{sim}\,(q, q_l) = \frac{|V_q|\,|V_{ql}|}{||V_q||\,||V_{ql}||} \tag{8}$$

where V_q is the vector of terms of the current query and $V_q l$ is the vector of terms of the past query. In their paper, they considered the current and past queries similar if their sim(q,q_l) score is greater or equal to 0.65. Finally, the current query relevance to resource s_k is estimated using Eq. 9 described as follows:

$$\text{rel}\,(q,\ s_k) = \sum_{k=1}^{m} \text{rel}\,(s_k|q_l)\,\text{sim}(q,\ \ q_l) \tag{9}$$

where rel(s_k | q_l) is the relevance score of resource s_k given the past query q_l which is obtained using ReDDE algorithm and sim(q,q_l) is the current and past queries similarity score. Zhu et al. [56] used k-means and latent semantic index (LSI) for resource selection. In their approach, the content of each resource is partitioned into a number of clusters with the help of the k-means clustering algorithm. After that, the semantic structure of each cluster is captured using LSI, which measures the relationship between them and then estimates the cluster relevance to the given query.

Other recent approaches in the literature are proposed by [57, 58]. Calì and Straccia, 2017 argue that since most of the content from federated resources is accessed by filling out an online form, this can be equated to querying relational database tables. Based on this notion, they proposed a novel approach that uses a mediated schema to integrate the resources into a single interface. On that interface, their approach automated all of the building blocks of federated search (document sampling, size estimation, resource selection, and result merging). But in [58], an approach that detects the unlawful alteration, manipulation, and reuse of copyrighted works via distributed information retrieval was proposed.

Almost all of the above-mentioned resource selection models [23, 24, 28, 42, 43] select a group of most relevant resources to search for the given query by considering relevance alone. However, it was established that many users' queries issued to the search systems are either ambiguous or multifaceted [59, 60]. Therefore, the LDA-RS resource selection algorithm was proposed in [39] to balance both relevance and diversity in selecting the resources to search for the given query. To generate the diversity rank list, each document in the initial ranked list is considered as a vector of terms $d_i = \{t_1, t_2, ..., t_n\}$ in which LDA is applied on each document to compute the probability of the query topics it covers. The goodness (i.e., relevance and diversity) of each document is obtained using the following expression:

$$G\,(d_i, q, R_{\text{div}}) = \lambda\Gamma\,(d_i, q) - (1-\lambda)\max_{r_j \in R_{\text{div}}} \text{sim}(d_i, r_j) \tag{10}$$

where λ is the relevant and diversity control parameter, $\gamma(d_i$, q) is the document relevant score obtained in the initial ranked list, and sim(d_i, d_j) is the similarity score obtained using

cosine similarity of the documents d_i and d_j vectors. In the LDA-RS paper, they used the Indri search engine to obtain the $\gamma(d_i, \mathrm{q})$ and KL-divergence retrieval model for the $\mathrm{sim}(d_i, d_j)$.

Similarly, a mean-variance method of search result diversification was proposed by Ghansah and Wu [36]. In their approach, the query received by the broker is executed on the CSD to generate the initial ranking. An Indri retrieval system is used as a retrieval model, and the resources with the highest number of documents are selected as the most relevant for the query. Afterward, they reranked the initially generated ranking using the portfolio algorithm proposed in [61]. A constant score is assigned to the selected resources for each of their documents in the reranked list. The resources with the highest scores are considered the most relevant and diverse.

Tables 2, 3, and 4summarize some selected studies on resource selection proposed in the literature.

4.3 Other methods

In most organizations, information is stored on multiple servers due to location or technical issues and mostly is available in unstructured files. To facilitate access to this information, most companies create an enterprise search system [62]. This system is designed to save employees time, improve the decision-making process, and find information regardless of its format or the server on which it is stored. In [63], an advanced resource selection model for enterprise search that utilizes semantic middleware schemas was proposed.

5 Result merging

Result merging is the last lap of the federated search interrelated problems. The goal of the result merging models is to collate all the results returned by those resources through calculated scores. The scores generated should be comparable across multiple resources. Consequently, an effective result-merging approach is critical to the success of federated search systems. This is because, even with the most relevant resources chosen, proper result merging is a necessity to guarantee the effectiveness of the final result list.

Nevertheless, merging the multiple result lists is a challenging task due to discrepancies among all the resources in terms of content, as well as the use of different retrieval models to retrieve the documents and, in most cases, the non-availability of the documents' full text at the merging time. These reasons make the result merging problem the least research area in federated search, especially for an uncooperative environment. Similar to resource selection, result merging can be subdivided into the heuristic method and machine learning method.

Table 2 Summary of some selected published studies for the resource selection problem (records from 1 to 9)

S/N	Paper	Year	Dataset	Method overview
1	The effectiveness of GlOSS for the text database discovery problem [6]	1994	Newsgroup dataset	This method selects the most relevant resource to search based on the number of all the relevant documents that particular resource has for the given query
2	Distributed information retrieval [28]	2000	TREC 1,2,3 CD's, TREC VLC	This method selects the most relevant resource to search based on the belief that given query can be generated from that particular resource
3	Relevant documents estimation method for resource selection [24]	2003	Trec123-100col, Trec4-Kmeans, Trec123-10col	Select a few relevant resources to search for a given query based on the estimated number of relevant documents the resources have in the sampled database index
4	Central-rank-based collection selection in uncooperative distributed information retrieval [23]	2006	Trec123-100col, Trec4-Kmeans, 100col-TREC-GOV2	The resources to search are selected based on the number of documents and their ranks when the given query is executed in the sampled database index
5	Classification-based resource selection [29]	2009	TREC GOV2	This method used multiple source of evidences such as relevance documents resource have, click-through rate, and query topic in selecting the most relevant resource to search
6	A joint probabilistic classification model for resource selection [30]	2010	Trec123-100 col-bysource	This method selects the resources to search based on joint relationship that exists between individuals relevant resources, which predicts the probability of their relevance to the given query
7	University of Padua at TREC 2013: Federated Web Search Track [31]	2013	TREC FedWeb 2013	This method proposed a term weighted frequency which selects the most relevant to search based on aggregation of query terms in a particular resource
8	Mirex and Taily at TREC 2013 [32]	2013	TREC FedWeb 2013	This approach uses a vocabulary-based method in which resources to search are selected based on the frequency of query terms in the resource
9	The University of Stavanger at the TREC 2013 Federated Web Search Track [33]	2013	TREC FedWeb 2013	A relevance resource is selected based on the likelihood that the resource contains relevant documents

Table 3 Summary of some selected published studies for the resource selection problem (records from 10 to 16)

S/N	Paper	Year	Dataset	Method overview
10	Simple May Be Best—A Simple and Effective Method for Federated Web Search via Search Engine Impact Factor Estimation [34]	2014	TREC FedWeb 2014	This method selects the most relevant resource to search based on both resource relevance to the user query as well as its market dominance
11	Drexel at TREC 2014 Federated Web Search Track [17]	2014	TREC FedWeb 2014	This method tested the effectiveness of various small document approaches such as REDDE, CRCS, SUSHI, etc., on the FedWeb 2014 dataset
12	RUC at TREC 2014: Select Resources Using Topic Models [35]	2014	TREC FedWeb 2014	In this method, sample database documents are trained using LDA. The most relevant resources to search are selected based on their topic distribution for the given query
13	University of Padua at TREC 2014: Federated Web Search Track [31]	2014	TREC FedWeb 2014	This method selects the most relevant resource to search based on how frequently query terms appear in that particular resource compared to other resources
14	Resource Selection for Federated Search on the Web [25]	2016	TREC ClueWeb09 dataset	This method performed comparative analysis of various resource selection approaches such as CORI, REDDE, and CRCS on the TREC Clueweb09 dataset
15	A Mean-Variance Analysis-Based Approach for Search Result Diversification in Federated Search [36]	2016	TREC Clue Web09 Cat-B	This method uses portfolio theory to select resources that are both relevant and with diversity content for the given query
16	Contextual source selection for federated search in mobile environment [37]	2016	Not provided	In addition to the query given, this method considers the user's situation, such as request time, type of device, and previous clicks, to select the most relevant resource for search

5.1 Heuristic methods

In the literature, one of the early result merging model [64] assumed that the resources should return their ranked results with their collection index terms statistics. However, it was argued in [10] that this assumption is not entirely achievable in a realistic web environment. This is because most of the resources are not cooperative. Because of the uncooperative nature of

Table 4 Summary of some selected published studies for the resource selection problem (records from 17 to 22)

S/N	Paper	Year	Dataset	Method overview
17	Learning to rank resources	2017 [38]	TREC Clue Web09 Cat-B	This method ranks the resources using SVMrank by combining query likelihood features, Central index features, and term-based feature
18	LDA-based resource selection for results diversification in federated search [39]	2018	TREC Clue Web09 Cat-B	This method uses LDA to discover the underlying topics in each resource by its sampled documents in the centralized sample index. It then selects the resources to search based on their relevance and diversity to the given query
19	Knowledge-based collection selection for distributed information retrieval [40]	2018	TREC Clue Web09 Cat-B	In this method, the semantic distance between a resource entity and the query entity is used to select relevant resources for search
20	Source selection of long tail sources for federated search in an uncooperative setting [41]	2018	TREC FedWeb 2014	This method proposes the strategy to include less relevant resources among the ones selected to search in order to give the search user the ability to explore more documents from diverse sources
21	LTRRS: A Learning to Rank-Based Algorithm for Resource Selection in Distributed Information Retrieval [42]	2019	Sogou-QCL	This method ranks the resources by training a LambdaMART model using different matching features. These include term matching, topic matching, and a centralized sample database documents
22	Embedding-based learning for collection selection in federated search [43]	2020	Trec123-100col, Trec4-Kmeans, 100col-TREC-GOV2	This method utilizes the query log in which the similarity between the past queries and the current query is computed. Resources selected for past queries similar to the current query are reselected for search

most resources, the approaches proposed by [44, 47, 65, 66], with different methodologies, utilized the representative documents in CSD to compute the merging score. That is, when the broker receives a user query, it forwards the query to the most relevant resources and runs it on the CSD. The merging score of a document is estimated by mapping its rank in a resource result list to its relevance score obtained from the CSD ranking. One disadvantage of these approaches is that their effectiveness depends on the high number of overlapped documents between resource results and CSD-ranked lists.

For the result merging tasks, a few runs were submitted in both the 2013 and 2014 TREC FedWeb tracks. In [67], they used some data fusion techniques to merge the results. Specifically, they converted the document ranks returned by the resources into a ranking score using the rank fusion technique [68]. That is, each document's relevance score was calculated by adding its ranks and frequency of appearance across multiple resource lists. However, the merged effectiveness of this approach suffers in the absence of many similar documents across the different resources result list. Similarly, the approaches proposed in [31, 69] computed the documents' merging scores by first converting their ranks into relevance scores and then multiplying them by the resource relevance score obtained in the resource selection phase. Specifically, Pal and Mitra [69] obtained the document score by taking the reciprocal of the log of document ranks. The effectiveness of these approaches depends on the effectiveness of the resource selection algorithm.

In [70], sentiment diversification was used to improve the effectiveness of the merged result list. Specifically, they converted the document ranks returned by the resources into a ranking score using the following equation:

$$s\,(d) = \frac{r(d)}{n} \times s(S_i) \tag{11}$$

where s(d) is the document relevance score, r(d) is the document rank in the resource ranked list, n is the number of documents the resource returned in its ranked list, and s(S_i) is the source relevance score obtained in the resources selection phase. The sentiment diversification is obtained using the SentWordNet lexicon approach [71]. That is, for each document, its sentiment toward the given query is obtained based on the sentiment of the terms that appear in it, which is obtained using the following equation:

$$\text{sent}\,(d) = \sum_{t \in d} \text{sent}\,(t)\,\frac{tf\,(t,d)}{|d|} \tag{12}$$

where sent(t) is the sentiment of the term t obtained from the SentWordNet, tf(t,d) is the frequency of term t in document d, and | d | is the total number of terms in document d. The final merging score for each document s_m (d) is obtained by iterative adding a document to the final ranking list using the following equation:

$$s_m\,(d) = \text{argmax}(s_{\text{norm}}\,(d) \times sent\,(d)) \tag{13}$$

Unfortunately, no significant difference was observed for this method compared to the non-diversified result methods proposed in the TREC 2014 FedWeb track. Recently, a snippet-based result merging model was proposed in [72]. In merging the results, they only used the snippets provided by search engines at query time to estimate the merging score for each document, making no assumptions about the resources' corpus size or retrieval models.

Table 5 Summary of some selected published studies for the result merging problem

S.NO	Paper	Year	Dataset	Method overview
1	Learning to merge search results for efficient distributed information retrieval [73]	2010	TREC WT10g	This method used SVMrank to learn a ranking function to merge the result based on the documents summaries such as title, ranking position, and description provided by the resources
2	Novasearch at TREC 2013 federated web search track [67]	2013	TREC FedWeb 2013	In this method, a reciprocal rank fusion technique is used to convert the document ranking into a ranking score and then sum it with the frequency of document appearance across different resources
3	ISI at the TREC 2013 federated task [69]	2013	TREC FedWeb 2013	This method obtained the documents' merging scores by first converting their ranks into relevance scores and then multiplying them by the resource relevance score obtained in the resource selection phase

5.2 Machine learning-based methods

Although many machine learning models have been applied in various tasks of information retrieval, only a few have been used for the result merging in federated search. Tjin–Kam–Jet and Hiemstra [73] treated the result merging problem as a classification problem. Based on the readily available information in the resource result list, they extracted some relevant features, such as the number of documents in each resource result list, the presence or absence of a URL for a document, query terms occurrences in the title, etc. They utilized SVMrank to train a ranking function that merged the multiple results lists into a single ranking list. A similar approach was proposed in [74] with additional features such as the resource ranking score obtained in the resource selection phase and then employing a boosting algorithm [75] to learn the ranking function. Furthermore, Ponnuswami et al. [76] used a gradient boost algorithm to learn the composition of the final merged result list when different verticals returned the result list. Recently, Vo [77] used genetic programming to propose a methodology for calculating the scores for all the documents to be merged. Either full text or excerpts such as ranking position, title, and description of the documents in question, BM25 scores of both title and description are also used. In their study, they used 45 attributes and 4 parameters in computing the merging score. Similarly, a reranker for a multilingual metasearch engine was proposed in [78]. This reranker is proposed for a multi-stage metasearch engine: The first stage is retrieving candidate documents for a given query from conventional search engines. In the second stage, the retrieved documents are reranked with a neural model. Each document is then scored according to its relevance to a given query. A final step is to format the documents and return them to the user.

Table 5 summarizes some selected studies on result merging proposed in the literature. Furthermore, based on Tables 2, 3, 4, and 5, it is evident that most of the approaches proposed over the last three decades have been focused on solving the resource selection problem, while

Table 5 continued

S.NO	Paper	Year	Dataset	Method overview
4	ICTNET at Federated Web Search Track 2014 [51]	2013	TREC FedWeb 2013	In this method, the merging score for each document is computed first, the LSI model is used to estimate the relevance of each query document and then the resulting score is combined by the reciprocal of the document rank
5	Opinions in federated search: University of Lugano at TREC 2014 federated web search track [70]	2014	TREC FedWeb 2014	This method uses the sentiment diversification strategy to merge the result list
6	Query Transformations for Result Merging [79]	2014	TREC FedWeb 2014	This method used query expansion strategy to improve the effectiveness of the merge result list
7	Simple May Be Best—A Simple and Effective Method for Federated Web Search via Search Engine Impact Factor Estimation [34]	2014	TREC FedWeb 2014	In this method, the merging score is obtained by first converting the documents rank into a ranking score and then combining it with the resource relevant score
8	Rankboost-based result merging [74]	2015	TREC WT10g	This method used rank boost method to learn a ranking function to merge the result based on the documents summaries provided by the resources
9	New re-ranking approach in merging search result	2019	OHSUMED	This method merges the results by extracting features from either full text or excerpt such as title, description, and ranking position, then using genetic programming to construct the merged result list
10	Snippet-based result merging in federated search [72]	2023	TREC FedWeb 2013	In this method, the results are merged based on information extracted from the resources' snippets provided at query time

a few have been focused on solving the result merging problem. In addition, very few of them use machine learning methods, as most of them employ heuristic methods.

5.3 Other methods

Several search engines get most of their revenue from sponsored search, where advertisers bid on slots to display targeted sponsored ads alongside the search results. For conventional search engines, the process of displaying ads alongside results is straightforward; however, for federated search, it is not. This is because, for a federated search system to know which ads to show, the documents returned must have relevance scores, as it is only the scores that determine likelihood of whether or not an ad will be clicked [80]. For this reason, a mechanism that incentivizes the inclusion of documents relevance scores in the returned resource result list was proposed in [80]. In [81], a revenue sharing mechanism between the search interface

provider and the information sources that provide the contents in the federated setting was proposed.

Due to the autonomous nature of the resources, there is no uniformity in which programming language each resource presents its results to the broker. As such, each resource presents its results in its generic language even though some of them provide an application programming interface (API) for easy extraction of their results by the broker [82]. As a result, a standardized protocol for the exchange of search results between the resources and a broker was proposed [82]. Furthermore, a model that predicts web-page relevance to a given query based on the web-page snippet provided by the resource was proposed in [83].

During the past decade, LinkedIn has evolved into a site that contains information about professionals, their profiles, job postings, and professional groups. Usually, people visit the site to search for jobs, hire people, join professional groups, and download content. To enhance user experience, [84] proposed a personalized federated search that utilizes users search history to aggregate the search results into a single list for LinkedIn users.

6 Systems and project prototypes

In the mid-1990s, researchers began exploring the potential benefits of federated search technology to enhance information retrieval systems' efficiency and accessibility. Studies have been conducted over the years to investigate the effectiveness of federated search systems, including the impact of resource selection and result merging algorithms on meeting users' information needs. The findings of this research aided in the development of federated search systems that are now utilized in digital libraries, government databases, and corporate enterprises.

6.1 First-generation systems

The first generation of federated search systems were developed using various protocols proposed to work in a cooperative environment, allowing multiple independent resources to be searched simultaneously through a single interface. Some of these protocols allow users to specify which resource their query should be routed to. Early systems such as MetaCrawler and some digital libraries utilized these protocols. Research such as STARTS [5] and SDLIP [85] proposed these protocols

The STARTS project aimed to create search protocols that allow each participating resource to share information with the broker to enable simultaneous searching across multiple independent resources. Meanwhile, SDLIP proposed middleware for search interfaces that facilitate cross-searching and information sharing among various digital libraries. The protocol was used to connect the digital libraries of the universities of California at Berkeley, San Diego, Santa Barbara, and the California Digital Library (CDL) through a single interface. However, in SDLIP, it is the search users who determine which resource receives their query

These two early researches were conducted to propose protocols for a cooperative environment in which resources involved disclose their corpus information to the broker through the agreed channel of communication. Despite the fact that the resources provide the broker with full information about their corpus, the systems developed using these protocols had limitations. First, merging documents from different resources was difficult due to varying corpus management. Second, the broker needed to periodically check for free and charged

information from the resources. Third, these protocols were designed for textual data only. Additionally, in SDLIP, the interface controlled the time allocated for a search session, which could result in session closure before the user was done. Lastly, the assumed level of information disclosure may not be realistic in a web environment

6.2 Second-generation systems

Combining the advantages of STARTS and SDLIP protocols led to the development of advanced search systems that enable automatic resource selection in a cooperative environment. SDARTS study [86] falls into this category. As it combines the SDLIP and STARTS protocols in order to develop an advanced search system capable of performing cross-searching on both local and internet resources. The SDART model used the combined protocols to develop three sets of wrappers: text documents, XML documents, and web documents. A wrapper is a piece of software that defines the interaction between resources that participate in a federated setting. These developed wrappers were integrated to create a sophisticated search interface that can access information on local resources and those on the internet. However, because SDARTS combined the protocols of STARTS and SDLIP, all their limitations are inherited by SDARTS.

6.3 Third-generation systems

In response to the wide acceptance of federated search technology among organizations and government institutions, researchers turned their attention to the development of a variety of wrappers. With these wrappers, hundreds of resources with different content can be accessed in an uncooperative environment setting.

The FedStats portal is a federated search portal that provides statistical information published by more than 100 federal agencies in the USA. With this portal, individuals and businesses can search for information without having to know which agency provides it. This portal was developed by Carnegie Mellon University researchers and the Federal statistics team under the FedLemur project proposed in [87]. The project aimed to create a wrapper for each of the target agencies' websites. By using the wrapper created, user queries can be translated into the programming language of the target agency. Then, forward the query, receive the results, and merge them into a single list. For each of these processes, a separate wrapper was developed. A limitation of this project, when it was developed, was the use of SSL and CORI algorithms for merging the results. These algorithms were found to be less effective in the literature as discussed in Sect. 5.1.

6.4 Fourth-generation systems

Federated searches are extensively employed across various sectors, primarily in academics, enterprises, and the tourism industry. Such searches cater to the needs of users seeking relevant information from multiple sources, thereby necessitating the development of sophisticated systems. These systems operate in an uncooperative environment, leveraging advanced resource selection and result merging methods.

Jayakody et al. [88] highlighted the challenges faced by the European Connected Factory Platforms for Agile Manufacturing (EFPF) project,[6] which aims to connect participating

[6] https://efpf-portal.ascora.eu/.

resources such as NIMBLE,[7] COMPOSITION,[8] vf-OS,[9], and DIGICOR[10] to offer seamless access to users. Due to different content in the repositories, the project faces significant challenges in content acquisition and interoperability. The Zenedo[11] is an open-access federated search system developed to enable researchers to share their findings and promote collaboration, while the EEXCESS eu-project [89] aimed to create a federated search system with access to different third-party search engines. Additionally, Tanium Reveal [90] a federated search engine for unstructured file systems managing sensitive data in enterprise networks, is designed such that each endpoint controls its index documents, and the central interface does not interfere with the resources' indexed content or keep a sample of it in its local database. Thus, when the broker receives a query, it is forwarded to all resources, and they perform three tiers of processing to generate a result list that is returned to the broker.

Collarana et al. [91] proposed a Federated Hybrid Search Engine (FuhSen).They also identified resource content variation as a major barrier to the interoperability of both searching and merging results. Nonetheless, they addresses the challenges of resource content variation using an on-demand knowledge graph to estimate semantic similarity and relatedness of resources to a given query. Damas et al. [4] developed a federated search system for sports-related websites, where four separate indexes are created for competitions, teams, managers, and players. The query is divided into terms and sent to the respective indexes, and the results are merged using the approach proposed in [92].

In summary, result merging and query optimization are the major challenges faced in the development of sophisticated federated search systems. The reason for this is that the resources involved in the federated setting use different methods of indexing, processing, and retrieving documents. Optimizing a query that generalizes across all the resources such that each resource retrieves its best result for a given query and then the broker merges those results returned by the resources into a single list is a challenging task.

7 Datasets

In information retrieval research, document corpora or testbeds serve as real-world search engine simulations for users to submit their information needs and receive a ranked list of documents. These testbeds contain a document corpus, a set of test queries to simulate user information needs, and relevance judgments for each document. These testbeds enable researchers to test the effectiveness of retrieval systems and develop improved approaches to meet user needs.

In the domain of federated search, information sources are considered autonomous, containing diverse content with some overlap between them [5]. Consequently, there is a need to develop testbeds that can simulate real-world federated search systems. One common method of creating such testbeds is to partition TREC datasets into smaller corpora that can serve as information sources. For instance, Xu et al. [93] used a K-means clustering algorithm to divide the TREC4 dataset into 100 information sources, while Powell et al. [94] used TREC 1–4 disks to create TREC-123-100col. Nevertheless, the primary drawbacks of these testbeds

[7] https://www.nimble-project.org.

[8] https://www.composition-project.eu.

[9] https://www.vf-os.eu.

[10] https://www.digicor-project.eu.

[11] CERN and OpenAIREplus launch new European research repository (sciencenode.org)

are their limited size compared to actual real-world information sources, and a nearly uniform distribution of documents across the created information sources.

According to [10, 95], the performance of federated search approaches is heavily influenced by the datasets used to evaluate them. In other words, models that perform well on smaller testbeds may not perform equally well on larger ones. To address this issue, 100col-GOV2 testbeds were created from the TREC GOV2 dataset [23], and wikipedia-100col-Kmeans was created from the Wikipedia Clueweb dataset [44].

The aforementioned testbeds are artificially created by dividing TREC datasets and assigning retrieval models, which may not reflect real-world federated search environments. Additionally, these testbeds primarily consist of text documents and may not account for the diverse range of content provided by some resources.

In an effort to address the limitations of existing testbeds and enable research that simulates real-world federated search environments, TREC has created the FedWeb datasets. FedWeb datasets are extensive collections of documents obtained from real-world search engines where search engines retrieve the documents using their proprietary retrieval models. This is contrary to previous federated search datasets mentioned earlier, the TREC FedWeb 2013 dataset was sampled from 157 real-world search engines in 24 vertical categories such as academic journals, blogs, news, videos, images, entertainment, shopping, and kids [48]. The 2014 TREC FedWeb dataset, on the other hand, was drawn from 149 real-world search engines across 24 vertical categories [48]. Another dataset created for federated search research was the one proposed in [96]. This dataset was crawled from 109 real-world conventional search engines and specialized databases.

The 2013 TREC FedWeb dataset was created using 2000 queries, of which the first 1000 were single-term queries sampled from the ClueWeb09 Cat-A collection. The remaining 1000 queries were search engine dependent, selected from the vocabulary of the snippets returned by the search engines for the first 1000 queries.

In contrast to the TREC 2013 FedWeb dataset, the TREC 2014 FedWeb dataset was created by issuing 4000 queries to search engines. The first 2000 queries were single-term queries from ClueWeb09 Cat-A, while the remaining 2000 queries were search engine-specific.

Real-world search engines have significant overlap among their return results, and to account for this, the TREC FedWeb datasets include a list of duplicate documents that must be removed before evaluating model-generated rankings that utilized the datasets. These datasets have features that resemble those of real-world federated search systems, making them ideal for testing federated search approaches.

8 More related issues

In the last three sections, we have reviewed work on three major aspects of federated search: resource description, resource selection, and result merging. In this section, we review work on some other issues than these three.

8.1 Evaluation

Result evaluation is an important aspect in information retrieval. However, it is more complicated for a federated search system than for a centralized search system. Of course, it might be desirable to evaluate the three major components (resource representation, resource

selection and result merging) separately. There are also some effort that tries to evaluate the whole system in different ways.

Probably [97] is the first that addressed this problem. A flexible simulation model is defined to analyze performance issues of a distributed information retrieval system. Response time, throughput, and resource utilization are measured in the condition of different settings of parameters including the number of users and text collections, average query length, I/O and CPU workloads network latency, the time to merge results from different IR servers, and so on.

[98] proposed a new measure, average ranked relative recall, to evaluate the results of a distributed information retrieval system. Considering that the result from a distributed information retrieval system is almost always worse than that of a centralized retrieval system, the results from a distributed retrieval system can be evaluated using the results from a centralized system as baseline.

Both [99] and [100] concerned the performance of component retrieval servers and corresponding estimation methods were proposed. They can be useful for tasks in federated search including resource selection and result merging, or may be useful for the evaluation of the whole federated search system as well.

A user study was presented in [101] to evaluate a federated medical search engine, MedSocket, in an established clinical setting. The Human, Organization, and Technology (HOT-fit) evaluation framework was applied. [102] carried out another user study to an interactive patent search system PerFedPat. a Prototype Web-Based Federated Search Engine for Art and Cultural Heritage was evaluated in [103]. In these studies, both efficiency and effectiveness were evaluated.

8.2 Aggregated search

In the context of web search, information seeking users are becoming more adept at identifying documents that are relevant to their queries. Some users are looking for more than just textual documents. Therefore, most search engines nowadays display multiple types of content such as images, maps, videos, and other media in search engine result page (SERP). Aggregation of diverse content on SERPs is referred to as aggregated search. Aggregated search can be regarded as an instance of federated search; it needs to deal with three key problems for a given user query. The first problem is to determine which verticals (resources) are relevant. The second problem is to determine which documents from the chosen vertical should appear in the SERP. Finally, there is the vertical presentation problem. It concerns how to display all the selected contents in the SERP.

Although federated search and aggregated search have some similarities, they also differ in some aspects, as highlighted in [104]: First, most of the recent studies on federated search were carried out in the uncooperative environment in which no cooperation exists between the broker and the resources. In aggregated search, on the other hand, there is full cooperation and the verticals are maintained centrally. Second, the goal of federated search resource selection is to select as few resources as possible for a given query. The premise is that selecting a few resources to search may lead to an improvement in retrieval performance. But in vertical selection, the goal is to determine which verticals are relevant to the query and which are not. Third, the same scoring formula is used to evaluate the relevance of the resources for a given query in federated search resource selection. Vertical selection, on the other hand, scores each vertical relevance to a query separately. For the last decade, different approaches

[105–107] on vertical selection and presentation are proposed in the literature. In a nutshell, aggregated search is a research area that focuses on the composition of the SERP. Its primary goals are as follows: (i) determining which verticals to include and where in the SERP; (ii) determining the users' behavior on the presented result; and (iii) determining what factors influence that behavior.

8.3 Metasearch

Metasearch engines [108] try to combine results from a given number of component search engines. It can work as general-purpose or specialized search engines depending on the type of search engines underneath. Metasearch and federated search look very similar, but many Metasearch papers assume that the collections in those component search systems are the same or overlap significantly. Therefore, a major objective of the research on Metasearch is how to improve retrieval performance by combing results from different retrieval systems with identical collection. In some cases, metasearch is referred to as data fusion [109].

In order to achieve better retrieval performance, a variety of techniques have been tried to obtain good weighting schemes for merging results. Borda count and Bayesian inference-based approaches were investigated in [110], Condorcet fusion was investigated in [111, 112], a multiple linear regression-based methods was proposed in [113], linear programming was investigated in [114], a method that using fuzzy analytical hierarchy process and modified extended ordered weighted averaging operator was investigated in [115], and an ant colony-based search was investigated in [116].

As an alternative to fusion, another type of approaches is re-ranking all the results from multiple search engines with all the duplicates removed. A re-ranking method was proposed in [117] that considered text-based, factor-based, rank-based, semantic-based, and classifier-based features extracted from the web pages retrieved by component search engines.

As another alternative to fusion, one policy is to estimate the effectiveness of all component search engines and choose the best per query. In [118], five heuristic measures were proposed for evaluating the relative relevance of all result lists from multiple search engines. All of them take into account the redundancy and ranking of documents across the lists.

The design and implementation of some metasearch systems were presented in [115, 117, 119].

8.4 Personalizing federated search

With the advancement of communication technologies and the latest generation of mobile devices (i.e., smartphones, tablets, etc.), people can now access the internet at any time, from any location using any mobile gadget. This internet penetration gave birth to different types of large-scale social media networks, such as Facebook, WeChat, Twitter, and WhatsApp. These social networks are now widely recognized as important tools for disseminating information and exchange of ideas [120]. The social media network allows users to tag a post or document and subsequently used the tags to label them by topics [121]. Several bookmarking sites for tagging such as Pinterest[12] and Flickr[13] are available on the internet. These set of tags can be used to build a user preference profile [122]. Several approaches [37, 122, 123] have exploited these tags to personalize resource selection and result merging.

[12] https://www.pinterest.com.

[13] https://www.flickr.com.

Kechid and Drias [123] argued that most result merging approaches proposed before only considered document-query relevance for the ranking of the final results, while the user's preferences and interest were not taken into account. To deal with this problem, they proposed a personalized approach that takes into account document relevance to: (i) user query; (ii) user profile; and (iii) user preferences. The documents are then ranked based on the sum of the three scores. A similar approach was proposed in [122]. The difference is that instead of using personal data and preferences to create a user profile as proposed in [123], a set of tags is used to build the user profile in [122]. Similarly, Hamid and Samir [37] posited that in order to meet user information needs, user profiles need to be considered apart from the documents' relevance to the query. As such, they proposed a resource selection algorithm that considers the user's profile. In their approach, a set of local and global user profiles are created. Local profiles include document preferences and interests, whereas global profiles include the user's device preferences and situation. Then, they used a collaborative scoring schema to compute the relevance score for the resources.

8.5 Security issues

Security is a very important issue for a distributed information system because it can be accessed by many different people in many different end points. When developing a federated search system, security should be considered at different levels.

Reveal [90] can evaluate compliance with security standards for data protection, such as those mandated by government regulations and laws. Some such examples include PCI standards for protecting personal credit card payment information [51]], HIPAA standards for secure patient health data [17], and GDPR standards for protecting personally identifiable information [23, 81]. Reveal can detect patterns of sensitive text, thereby identifying regulatory noncompliance.

9 Conclusion and future research directions

The ubiquity of conventional search engines as vital tools in the present-day information age is undeniable. Although they cater to the information needs of numerous individuals seeking information on the web, they are insufficient in providing complete access to a substantial proportion of information sources available on the web.

Federated search targets those information sources by acting as an intermediary between them and information seekers, enabling the forwarding queries to multiple resources through a single search interface.

Researchers have made significant progress in addressing the interrelated issues in federated search, including resource description, resource selection, and result merging. This paper reviews various state-of-the-art models, with a particular emphasis on resource selection and result merging, and highlights their methodology and some limitations, providing insights into potential areas for future research. Furthermore, the available testbeds used for evaluating federated search models are discussed, and some federated search systems and prototype were also discussed.

Although numerous approaches have been proposed to tackle federated search challenges, most of them utilize partitioned datasets that is not realistic reflection of real-world web federated search systems. To address this dataset gap, TREC created the 2013 and 2014 TREC FedWeb datasets, which replicate actual federated search systems. Despite this, few

new models have been proposed using these datasets. Therefore, a potential research direction using these datasets and the development of additional ones is proposed.

Search Results Diversification: In the field of information retrieval, it is commonly reported that many search users' queries are ambiguous or multi-faceted. Result diversification has been proposed as a solution to disambiguate search queries. However, there are few proposed approaches to diversifying search results in the federated search result merging problem. Thus, there is a need for an approach that can use only the FedWeb dataset snippets for result merging in federated search.

Query Expansion: In the field of information retrieval, previous research has established that query expansion can significantly enhance retrieval performance for short queries in centralized search systems. However, the same level of success has not been reported in the federated search literature. This is due to the difficulty of finding suitable sources to select expansion terms from. As such, there is a need for an approach that explores alternative sources for selecting expansion terms beyond traditional feedback documents or external dictionaries, such as WordNet.

Multimedia Data Sampling: In the context of obtaining resources corpus information in uncooperative environments, sampling methods have been proposed in the literature, primarily for textual data. However, it is becoming increasingly apparent that multimedia data, such as images and videos, are prevalent in resources indexes. As a result, there is a need for novel approaches that can effectively sample multimedia data based on their features to cater to the needs of federated search research.

Image Retrieval: In recent times, image retrieval has garnered significant attention from researchers due to the exponential growth in the volume of images generated in various domains such as medical images, satellite images, and social media. While several approaches have been proposed for real-time retrieval in centralized systems, there is a notable gap in the literature concerning federated search approaches for image retrieval. Hence, creating an image dataset that simulates a real-world federated search environment and proposing models for resource selection and result merging is a promising direction to explore. Such models could be useful for effectively retrieving images in federated search systems, which will enhance their performance and utility for various applications.

Author Contributions AG conceptualized the idea for the article, performed the literature search and data analysis, prepared the figures, and wrote the first draft. SW manually read and selected the papers to be included in the manuscript, restructured the manuscript, critically revised the work for important intellectual content, and proofread it. SK performed the literature search, proofread the manuscript, prepared the tables, and formatted it in Latex for journal submission.

Funding This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Declarations

Conflict of interest The authors of this manuscript have no potential conflicts of interest to disclose.

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent to publish The manuscript was read and approved for submission by all authors.

References

1. Sreeja SR, Chaudhari S (2014) Review of web crawlers. Int J Knowl Web Intell 5(1):49–61
2. Nguyen D, Demeester T, Trieschnigg D, Hiemstra D (2012) Federated search in the wild: the combined power of over a hundred search engines. In Chen X, Lebanon G, Wang H, Zaki MJ (eds) 21st ACM international conference on information and knowledge management, CIKM'12, Maui, HI, USA, October 29–November 02, 2012, pp. 1874–1878. https://doi.org/10.1145/2396761.2398535
3. Li X (2022) Federated search to merge the results of the extracted functional requirements. PhD thesis, University of Cincinnati
4. Damas J, Devezas J, Nunes S (2022) Federated search using query log evidence. In: Progress in artificial intelligence: Proceedings of 21st EPIA conference on artificial intelligence, EPIA 2022, Lisbon, Portugal, August 31–September 2, 2022, pp 794–805. Springer. https://doi.org/10.1007/978-3-031-16474-3_64.
5. Gravano L, Chang C-CK, Garcia-Molina H, Paepcke A (1997) STARTS: stanford proposal for internet meta-searching. In: Proceedings of the 1997 ACM SIGMOD international conference on management of data, pp 207–218. https://doi.org/10.1145/253262.253299
6. Gravano L, Garcia-Molina H, Tomasic A (1994) The effectiveness of GlOSS for the text database discovery problem. In: Proceedings of the 1994 ACM SIGMOD international conference on management of data, pp 126–137
7. Callan J, Connell M (2001) Query-based sampling of text databases. ACM Trans Inf Syst 19(2):97–130. https://doi.org/10.1145/382979.383040
8. Baillie M, Azzopardi L, Crestani F (2006) Adaptive query-based sampling of distributed collections. In: International symposium on string processing and information retrieval, pp 316–328. Springer
9. Shokouhi M, Zobel J, Scholer F, Tahaghoghi SM (2006) Capturing collection size for distributed non-cooperative retrieval. In: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval, pp 316–323
10. Shokouhi M, Si L (2011) Federated search. Found Trends Inf Retriev 5(1):1–102
11. Van den Bosch A, Bogers T, De Kunder M (2016) Estimating search engine index size variability: a 9-year longitudinal study. Scientometrics 107(2):839–856
12. Khelghati M, Hiemstra D, Van Keulen M (2013) Deep web entity monitoring. In: Proceedings of the 22Nd international conference on world wide web, pp 377–382
13. Bergman MK (2001) White paper: the deep web: surfacing hidden value. J Electron 7(1)
14. Craswell N (2000) Methods for distributed information retrieval
15. Yuwono B, Lee DL (1997) Server ranking for distributed text retrieval systems on the internet. In: 5th International conference on database systems for advanced applications database systems for advanced applications' 97 (Melbourne, Australia), pp 41–49
16. Arguello J, Diaz F, Callan J, Crespo J-F (2009) Sources of evidence for vertical selection. In: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval, pp 315–322
17. Zhao H, Hu X (2014) Drexel at trec 2014 federated web search track. Technical report, Drexel univ Philadelphia pa coll of computing and informatics
18. Wang Y, Liang J, Lu J (2014) Estimating the size of hidden data sources by queries. In: 2014 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM 2014), pp 712–719. IEEE
19. Lu J, Li D (2010) Estimating deep web data source size by capture-recapture method. Inf Retriev 13:70–95
20. Lu J (2008) Efficient estimation of the size of text deep web data source. In: Proceedings of the 17th ACM conference on information and knowledge management, pp 1485–1486
21. Broder A, Fontura M, Josifovski V, Kumar R, Motwani R, Nabar S, Panigrahy R, Tomkins A, Xu Y (2006) Estimating corpus size via queries. In: Proceedings of the 15th ACM international conference on information and knowledge management, pp 594–603
22. Dasgupta A, Jin X, Jewell B, Zhang N, Das G (2010) Unbiased estimation of size and other aggregates over hidden web databases. In: Proceedings of the 2010 ACM SIGMOD international conference on management of data, pp 855–866
23. Shokouhi M (2007) Central-rank-based collection selection in uncooperative distributed information retrieval. In: European conference on information retrieval, pp 160–172. Springer
24. Si L, Callan J (2003) Relevant document distribution estimation method for resource selection. In: Proceedings of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval, pp 298–305
25. Nguyen D, Demeester T, Trieschnigg D, Hiemstra D (2016) Resource selection for federated search on the web. arXiv preprint arXiv:1609.04556

26. Shokouhi M, Zobel J (2007) Federated text retrieval from uncooperative overlapped collections. In: Proceedings of the 30th annual international acm sigir conference on research and development in information retrieval, pp 495–502
27. Bernstein Y, Shokouhi M, Zobel J (2006) Compact features for detection of near-duplicates in distributed retrieval. In: Proceedings of string processing and information retrieval: 13th international conference, SPIRE 2006, Glasgow, UK, October 11-13, 2006, pp 110–121. Springer
28. Callan J (2000) Distributed information retrieval. Adv Inf Retriev, pp 127–150
29. Arguello J, Callan J, Diaz F (2009) Classification-based resource selection. In: Proceedings of the 18th ACM conference on information and knowledge management, pp 1277–1286
30. Hong D, Si L, Bracke P, Witt M, Juchcinski T (2010) A joint probabilistic classification model for resource selection. In: Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval, pp 98–105
31. Di Buccio E, Melucci M (2014) University of padua at TREC 2014: Federated web search track. Technical report, Padua Univ (Italy)
32. Hiemstra D, Trieschnigg D, Demeester T (2013) Mirex and taily at trec 2013
33. Balog K (2013) The university of stavanger at the trec 2013 federated web search track
34. Jin S, Lan M (2014) Simple may be best-a simple and effective method for federated web search via search engine impact factor estimation. In: TREC
35. Wang Q, Shi S, Cao W (2014) Ruc at TREC 2014: select resources using topic models. Technical report, Renmin Univ Beijing (China)
36. Ghansah B, Wu S (2016) A mean-variance analysis based approach for search result diversification in federated search. Int J Uncert Fuzziness Knowl-Based Syst 24(02):195–211
37. Hamid B, Samir K (2016) Contextual source selection for federated search in mobile environment. In: 2016 30th international conference on advanced information networking and applications workshops (WAINA), pp 883–888. https://ieeexplore.ieee.org/document/7471315/. IEEE
38. Dai Z, Kim Y, Callan J (2017) Learning to rank resources. In: Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval, pp 837–840
39. Li L, Zhang Z, Wu S (2018) LDA-based resource selection for results diversification in federated search. In: Proceedings of web information systems and applications: 15th international conference, WISA 2018, Taiyuan, China, September 14–15, pp 147–156. Springer
40. Han B, Chen L, Tian X (2018) Knowledge based collection selection for distributed information retrieval. Inf Process Manage 54(1):116–128
41. Urak G, Ziak H, Kern R (2018) Source selection of long tail sources for federated search in an uncooperative setting. In: Proceedings of the 33rd annual ACM symposium on applied computing, pp 720–727
42. Wu T, Liu X, Dong S (2019) Ltrrs: A learning to rank based algorithm for resource selection in distributed information retrieval. In: China conference on information retrieval, pp 52–63. Springer
43. Garba A, Khalid S, Ullah I, Khusro S, Mumin D (2020) Embedding based learning for collection selection in federated search. Data Technologies and Applications
44. Hong D, Si L (2012) Mixture model with multiple centralized retrieval algorithms for result merging in federated search. In: Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval, pp 821–830
45. Hong D, Si L (2013) Search result diversification in resource selection for federated search. In: Proceedings of the 36th international ACM SIGIR Conference on research and development in information retrieval, pp 613–622
46. Cetintas S, Si L, Yuan H (2009) Learning from past queries for resource selection. In: Proceedings of the 18th ACM conference on information and knowledge management, pp 1867–1870
47. Shokouhi M, Zobel J (2009) Robust result merging using sample-based score estimates. ACM Trans Inf Syst 27(3):1–29
48. Demeester T, Trieschnigg D, Nguyen D, Zhou K, Hiemstra D (2014) Overview of the TREC 2014 federated web search track. Technical report, Ghent Univ (Belgium)
49. Demeester T, Trieschnigg D, Nguyen D, Hiemstra D, Zhou K (2015) Fedweb greatest hits: presenting the new test collection for federated web search. In: Proceedings of the 24th international conference on world wide web, pp 27–28
50. Bellogín A, Gebremeskel GG, He J, Said A, Samar T, de Vries AP, Lin J, Vuurens JB (2013) Cwi and tu delft notebook TREC 2013: contextual suggestion, federated web search, kba, and web tracks. In: TREC. Citeseer
51. Guan F, Xue Y, Yu X, Liu Y, Cheng X (2014) Ictnet at federated web search track 2013. In: TREC

52. Aly R, Hiemstra D, Demeester T (2013) Taily: shard selection using the tail of score distributions. In: Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval, pp 673–682
53. Xu J, Li X (2007) Learning to rank collections. In: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval, pp 765–766
54. Joachims T (2006) Training linear SVMS in linear time. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, pp 217–226
55. Wu Q, Burges CJ, Svore KM, Gao J (2010) Adapting boosting for information retrieval measures. Inf Retrieval 13(3):254–270
56. Zhu Q, Li D, Lee DL (2018) C-dlsi: an extended lsi tailored for federated text retrieval. arXiv preprint arXiv:1810.02579
57. Calì A, Straccia U (2017) Integration of deep web sources: A distributed information retrieval approach. In: Proceedings of the 7th international conference on web intelligence, mining and semantics, pp 1–4
58. Benbelgacem S, Guezouli L, Seghir R (2020) A distributed information retrieval approach for copyright protection. In: Proceedings of the 3rd international conference on networking, information systems and security, pp 1–6
59. Xia L, Xu J, Lan Y, Guo J, Zeng W, Cheng X (2017) Adapting markov decision process for search result diversification. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, pp 535–544
60. Yigit-Sert S, Altingovde IS, Macdonald C, Ounis I, Ulusoy Ö (2020) Supervised approaches for explicit search result diversification. Inf Process Manage 57(6):102356
61. Wang J, Zhu J (2009) Portfolio theory of information retrieval. In: Proceedings of the 32nd International ACM SIGIR conference on research and development in information retrieval, pp 115–122
62. Cleverley PH, Burnett S (2019) Enterprise search: a state of the art. Bus Inf Rev 36(2):60–69
63. Wauer M, Schuster D, Schill A (2011) Advanced resource selection for federated enterprise search. In: Business information systems workshops: BIS 2011 international workshops and BPSC international conference, Poznań, Poland, June 15-17, 2011. Revised Papers 14, pp. 154–159. Springer
64. Rasolofo Y, Hawking D, Savoy J (2003) Result merging strategies for a current news metasearcher. Inf Process Manage 39(4):581–609
65. Si L, Callan J (2003) A semisupervised learning method to merge search engine results. ACM Trans Inf Syst 21(4):457–491
66. He C, Hong D, Si L (2011) A weighted curve fitting method for result merging in federated search. In: Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval, pp 1177–1178
67. Mourao A, Martins F, Magalhaes J (2013) Novasearch at trec 2013 federated web search track: experiments with rank fusion. In: TREC
68. Cormack GV, Clarke CL, Buettcher S (2009) Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval, pp. 758–759
69. Pal D, Mitra M (2013) Isi at the trec 2013 federated task. In: TREC
70. Giachanou A, Markov I, Crestani F (2014) Opinions in federated search: University of lugano at trec 2014 federated web search track. Technical report, Lugano Univ (Switzerland)
71. Esuli A, Sebastiani F (2006) Sentiwordnet: a publicly available lexical resource for opinion mining. In: Proceedings of the fifth international conference on language resources and evaluation (LREC'06)
72. Garba A, Wu S (2023) Snippet-based result merging in federated search. J Inf Sci
73. Tjin-Kam-Jet K, Hiemstra D (2010) Learning to merge search results for efficient distributed information retrieval
74. Ghansah B, Wu S, Ghansah N (2015) Rankboost-based result merging. In: 2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing, pp 907–914. IEEE
75. Freund Y, Iyer R, Schapire RE, Singer Y (2003) An efficient boosting algorithm for combining preferences. J Mach Learn Res 4(Nov):933–969
76. Ponnuswami AK, Pattabiraman K, Wu Q, Gilad-Bachrach R, Kanungo T (2011) On composition of a federated web search result page: using online users to provide pairwise preference for heterogeneous verticals. In: Proceedings of the fourth ACM international conference on web search and data mining, pp 715–724
77. Vo HT (2019) New re-ranking approach in merging search results. Informatic 43(2)
78. Almeida TS, Laitz T, Seródio J, Bonifacio LH, Lotufo R, Nogueira R (2022) Neuralsearchx: serving a multi-billion-parameter reranker for multilingual metasearch at a low cost. arXiv preprint arXiv:2210.14837

79. Palakodety S, Callan J (2014) Query transformations for result merging. Technical report, Carnegie-Mellon Univ Pittsburgh, PA School of Computer Science
80. Ceppi S, Gatti N, Gerding E (2011) Mechanism design for federated sponsored search auctions. Proc AAAI Confer Artific Intell 25:608–613
81. Bonetti LE, Ceppi S, Gatti N, et al (2011) Designing a revenue mechanism for federated search engines. In: VLDS, pp 46–51. Citeseer
82. Trieschnigg D, Tjin-Kam-Jet K, Hiemstra D (2013) Searchresultfinder: Federated search made easy. In: Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval, pp 1113–1114
83. Demeester T, Nguyen D, Trieschnigg D, Develder C, Hiemstra D (2013) Snippet-based relevance predictions for federated web search. In: Advances in information retrieval: 35th European conference on IR research, ECIR 2013, Moscow, Russia, March 24-27. Proceedings 35, pp 697–700. Springer
84. Arya D, Ha-Thuc V, Sinha S (2015) Personalized federated search at linkedin. In: Proceedings of the 24th ACM international on conference on information and knowledge management, pp 1699–1702
85. Paepcke A, Brandriff R, Janee G, Larson R, Ludaescher B, Melnik S, Raghavan S (2000) Search middleware and the simple digital library interoperability protocol. DLIB Magazine 6(3)
86. Green N, Ipeirotis PG, Gravano L (2001) SDLIP+ STARTS= SDARTS a protocol and toolkit for metasearching. In: Proceedings of the 1st ACM/IEEE-CS joint conference on digital libraries, pp 207–214
87. Avrahami TT, Yau L, Si L, Callan J (2006) The fedlemur project: Federated search in the real world. J Am Soc Inform Sci Technol 57(3):347–358
88. Jayakody D, Selvanathan N, Damjanovic-Behrendt V (2020) Federated search and recommendation. In: I-ESA Workshops
89. Dragoni M, Rexha A, Ziak H, Kern R (2017) A semantic federated search engine for domain-specific document retrieval. In: Proceedings of the symposium on applied computing, pp 303–308
90. Stoddard J, Mustafa A, Goela N (2021) Tanium reveal: a federated search engine for querying unstructured file data on large enterprise networks. Proc VLDB Endow 14(12):3096–3109
91. Collarana D, Galkin M, Lange C, Grangel-González I, Vidal M-E, Auer S (2016) Fuhsen: A federated hybrid search engine for building a knowledge graph on-demand (short paper). In: OTM confederated international conferences on the move to meaningful internet systems, pp 752–761. Springer
92. Rasolofo Y, Abbaci F, Savoy J (2001) Approaches to collection selection and results merging for distributed information retrieval. In: Proceedings of the tenth international conference on information and knowledge management, pp. 91–198
93. Xu J, Croft WB (1999) Cluster-based language models for distributed retrieval. In: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval, pp 54–261
94. Powell AL, French JC (2003) Comparing the performance of collection selection algorithms. ACM Trans Inf Syst 21(4):412–456
95. D'Souza DJ, Zobel J, Thom JA (2004) Is cori effective for collection selection? An exploration of parameters, queries, and data. In: ADCS, pp 41–46
96. Nguyen D, Demeester T, Trieschnigg D, Hiemstra D (2012) Federated search in the wild: the combined power of over a hundred search engines. In: Proceedings of the 21st ACM international conference on information and knowledge management, pp 1874–1878
97. Cahoon B, McKinley KS (1996) Performance evaluation of a distributed architecture for information retrieval. In: Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR'96, August 18-22, 1996, Zurich, Switzerland (Special Issue of the SIGIR Forum), pp 110–118. ACM
98. Witschel HF, Holz F, Heinrich G, Teresniak S (2008) An evaluation measure for distributed information retrieval systems. In: Proceedings 30th European conference on IR research, advances in information retrieval, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Lecture Notes in Computer Science, vol vol 4956, pp 607–611. https://doi.org/10.1007/978-3-540-78646-7_64
99. Losee RM LC Jr (2004) Information retrieval with distributed databases: analytic models of performance. IEEE Tran. Parall Distribut Syst 15(1):18–27
100. Jung JJ (2009) Consensus-based evaluation framework for distributed information retrieval systems. Knowl Inf Syst 18(2):199–211
101. Williams J, Kochendorfer KM (2012) Evaluation of a federated medical search engine during third-year medical clerkship. In: AMIA 2012, American medical informatics association annual symposium, Chicago, Illinois, USA, November 3-7, 2012

102. Buccio ED, Masiero I, Melucci M (2014) Evaluation of a recursive weighting scheme for federated web search. In: Basili R, Crestani F, Pennacchiotti M (eds) Proceedings of the 5th Italian information retrieval workshop, Roma, Italy, January 20-21, 2014. CEUR workshop, vol 1127, pp 1–10
103. Pergantis M, Varlamis I, Giannakoulopoulos A (2022) User evaluation and metrics analysis of a prototype web-based federated search engine for art and cultural heritage. Information 13(6):285
104. Arguello J (2017) Aggregated search. Found Trends Inf Retriev 10(5):365–502
105. Arguello J, Diaz F, Callan J (2011) Learning to aggregate vertical results into web search results. In: Proceedings of the 20th ACM international conference on information and knowledge management, pp 201–210
106. Ma X (2020) A new aggregated search method. J Intell Fuzzy Syst 38(1):55–63
107. Rashid U, Saleem K, Ahmed A (2021) Mirre approach: nonlinear and multimodal exploration of mir aggregated search results. Multimed Tools Appl 80(13):20217–20253
108. Meng W, Yu CT (2010) Advanced metasearch engine technology. Synth Lect Data Manage 2(1):1–129
109. Wu S (2012) Data fusion in information retrieval. Adapt Learn Optim 13:1–228. https://doi.org/10.1007/978-3-642-28866-1
110. Aslam JA, Montague MH (2001) Models for metasearch. In: Croft WB, Harper DJ, Kraft DH, Zobel J (eds) SIGIR 2001: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval, September 9-13, 2001, New Orleans, Louisiana, USA, pp 275–284
111. Montague MH, Aslam JA (2002) Condorcet fusion for improved retrieval. In: Proceedings of the 2002 ACM CIKM international conference on information and knowledge management, McLean, VA, USA, November 4-9, 2002, pp 538–548
112. Wu S (2013) The weighted condorcet fusion in information retrieval. Inf Process Manage 49(1):108–122
113. Wu S (2012) Linear combination of component results in information retrieval. Data Knowl Eng 71(1):114–126
114. Amin GR, Emrouznejad A, Sadeghi H (2012) Metasearch information fusion using linear programming. RAIRO Oper Res 46(4):289–303
115. Tayal DK, Jain A, Dimri N, Gupta S (2015) Metasurfer: a new metasearch engine based on FAHP and modified EOWA operator. Int J Syst Assur Eng Manag 6(4):487–499
116. Kaur P, Singh M, Josan GS, Dhillon SS (2018) Rank aggregation using ant colony approach for metasearch. Soft Comput 22(13):4477–4492
117. Vijaya P, Chander S (2018) Lionrank: lion algorithm-based metasearch engines for re-ranking of webpages. Sci China Inf Sci 61(12):122102–112210216
118. Liu W, Han C, Lian F (2009) An alternative derivation of a bayes tracking filter based on finite mixture models. In: 12th international conference on information fusion, FUSION '09, Seattle, Washington, USA, July 6-9, pp 842–849
119. Smalheiser NR, Lin C, Jia L, Jiang Y, Cohen AM, Yu CT, Davis JM, Adams CE, McDonagh MS, Meng W (2014) Design and implementation of metta, a metasearch engine for biomedical literature retrieval intended for systematic reviewers. Health Inf Sci Syst 2(1):1
120. Saito K, Kimura M, Ohara K, Motoda H (2010) Selecting information diffusion models over social networks for behavioral analysis. In: Joint European conference on machine learning and knowledge discovery in databases, pp 180–195. Springer
121. Chelmis C, Prasanna VK (2013) Social link prediction in online social tagging systems. ACM Trans Inf Syst 31(4):1–27
122. Saoud Z, Kechid S (2016) Integrating social profile to improve the source selection and the result merging process in distributed information retrieval. Inf Sci 336:115–128
123. Kechid S, Drias H (2009) Personalizing the source selection and the result merging process. Int J Artif Intell Tools 18(02):331–354

Adamu Garba received a B.E. degree in Electrical Engineering from Kano University of Science and Technology Wudil, Kano, Nigeria, in 2010 and an M.Tech. degree in Computer Networking from Sharda University, India, in 2015. He is currently pursuing his Ph.D. degree in the School of Computer Science and Communication Engineering, Jiangsu University, China. His research interests include distributed information retrieval, federated search, web search engines, digital libraries, data mining, recommender systems, and information retrieval.

Shengli Wu received the Ph.D. degree in Computer Science from Southeast University, China. He is currently a lecturer in the School of Computer Science and Communication Engineering, Jiangsu University, China and School of Computing, Ulster University, UK. His current research interests include database and information retrieval, scientific metrics and citation analysis, and machine learning.

Shah Khalid received the M.S. degree from the University of Peshawar, Pakistan, and the Ph.D. degree from Jiangsu University, China. He is currently an Assistant Professor at the School of Electrical Engineering and Computer Science, National University of Science and Technology (NUST-SEECS), Islamabad, Pakistan. His research interests include information retrieval, web search engines, scholarly retrieval systems, recommender systems, knowledge graphs, social web, real-time sentiment analysis, web engineering, text summarization, federated search, and digital libraries.