Snippet Based Results Merging in Federated Search

Snippet Based Results Merging in Federated Search

Anonymous

Abstract

In federated search, the central broker simultaneously forwards the search query to multiple resources. The returned results from those resources are then merged into a single ranked list. An autonomous resource in a federated search system usually does not provide scores for the retrieved documents; even if some of them do, scores from different resources are incomparable due to the heterogeneity in many aspects of those resource involved such as retrieval models, corpus statistics, etc. Many results merging approaches have been proposed in the literature to deal with this problem. However, to the best of our knowledge, none of them has utilized snippets. This paper proposes a snippet-based weighting scheme for the query terms involved. It quantifies the importance of each query term from two angles: the frequency of the term and the part in which the term occurs inside a snippet. Three parts, which are URL, title, and description, are given different weights. Experiments are conducted with the TREC 2013 FedWeb dataset. The results show that the proposed methods consistently outperform several baseline models. We also find in many instances, a further small slight performance improvement is achievable by an extra measure of weighting each of the resources involved, which can be done in the phase of resource selection.

Keywords: Federated Search, Distributed Information Retrieval, Result Merging, Uncooperative Environment, Term Weighting Scheme, Snippet

1. Introduction

The size of the web and the dynamic nature of some information resources mean it is not feasible for classical search engines, such as Google, Baidu, Bing, etc., to crawl and index all the web documents. Moreover, some information resources can only be accessed by registered users due to proprietary, commercial, or security reasons. These resources are mostly uncrawlable by the crawlers of classical search engines, which crawl web documents through the links between them [1]. This has, therefore, made the contents of these resources mostly inaccessible directly through them [2]. For this reason, the owners of such resources have to provide their search platforms individually. Unfortunately, such platforms might not necessarily be noticed by many users due to the massive size of the web. A possible solution to this problem is to provide a unified search interface capable of concurrent search across multiple distributed resources. Any Information Retrieval (IR) system that searches distributed resources through a unified search interface without a centralized corpus is called a federated search system [2-4].

The consensus in the federated search literature is that the interface (i.e., central broker) mediates between the users of the search system and the multiple resources involved. As such, the

entire retrieval process of a federated search system consists of three separate but intertwined phases. These are:

i. Resource description [5][5] discovers the information of each resource such as its contents, size, and so on and keeps such information in the broker.
ii. Resource selection [6, 7] which, for each user query *q* received by the broker, the broker selects a group of resources that more likely contain documents relevant to the search query and
iii. Result merging [2, 8, 9] merges the results returned by different resources into a single ranked list.

This paper focuses on the result merging problem. The challenge for merging the results lies in which modalities to combine documents returned from different resources. Documents' relevance scores from different resources cannot be compared directly due to:

i. Variation in the sizes of the multiple resources
ii. Heterogeneity exists among different resources. Each of them may have different content coverage from the others.
iii. Different resources may use different retrieval models and various other alternative components to retrieve documents

For these reasons, quite a few algorithms have been explored to merge results. These algorithms investigated the result merging problem in two different environments. The early result merging models [10, 11] fit a cooperative environment. The resources reveal information about their content to the broker, and the broker then uses the information to perform resource selection and result merging. Merging results in such an environment is more effective, but the cooperation level assumed is challenging to achieve in the real-world web.

On the other hand, Shokouhi and Zobel [8] and Hong and Si [2] investigated the result merging problem in an uncooperative environment, where the resources do not reveal their collective information to the broker. Consequently, with a method like query-based sampling (QBS) [5], a pre-determined number of documents are sampled from the resources to build the resources' representative documents, referred to as a centralized sample database, at the broker site. Then, the result merging algorithms work with the sample database as the representative of the original resources. This approach is workable with reduced effectiveness due to the difference between the sampled and original resources.

Other approaches proposed by Pal and Mitra [12] and Di Buccio and Melucci [13] computed the documents merging score by combining their relevance scores and the score of the resources that returned them. The resource scores used in those models are the weight of the resources obtained in the resource selection phase. Giachanou et al. [14] used a sentiment diversification strategy to improve the effectiveness of the merged result list. Thus, the merging score in their model is computed by aggregating each document's relevance and opinion score. Recently, Vo [9] proposed a model that merged the result lists based on the documents excerpt extracted from the

resources list using genetic programming. However, most of the effective approaches need the full-text of documents [15, 16]. This assumption may not hold, as most real-world resources return a ranked list of snippets, rather than the whole documents, in real-time for the query issued to them [17]. Moreover, even if some of the resources provide full-text for the retrieved documents, transferring and processing them at query execution time would be costly and result in increased bandwidth utilization, storage space, and response time. On the other hand, transferring and processing snippets is much simpler and can be done quickly by the broker. It has become a common practice among the general-purpose and focused search engines in returning a ranked list of snippets to the search users for queries issued to them. These observations motivate us to investigate result merging methods in federated search that can fit and benefit from its real environment.

Our main objective of this paper is to merge the multiple results list returned by the resources using only the snippets in the ranked list without any assumption about the size of their corpus statistics or retrieval models. In achieving this objective, we make the following three contributions:

1. We propose a term weighting scheme for query terms to match the terms that appear in different parts of snippets. The importance of those different parts are differentiated when calculating scores for merging documents from multiple resources.
2. We also investigate the effect of including resource weights in the performance improvement of the merged result list. In particular, we consider the two resource weighting approaches mostly used in the literature: first, based on the resource relevant score obtained in the resource selection phase, and, second, based on the number of documents the resource returns in their ranked list and performance analysis to see how this affects the effectiveness of the merged result list.
3. We test the effectiveness of the proposed methods with the TREC 2013 FedWeb dataset. The evaluated results show the competitiveness of the proposed methods as they significantly outperformed the baselines on all fronts.

The rest of the paper is organized as follows; a summary of the previous works on federated search result merging is discussed in Section 2. Section 3 proposes the algorithm of results merging from multiple resources. Some explanations and discussion are also given. The dataset, experimental setup, and evaluation metric, are discussed in Section 4. Experimental results and related discussion are presented in Section 5, and the paper is concluded in Section 6.

2. Related Work

For federated search, there are two main research issues: resource selection and results merging. Resource selection has been investigated by [6, 7, 18, 19] . Han et al. [19] argued that most of the existing methods for collection selection use the flat meaning of query terms in computing query-collection relevance score. To overcome this limitation, they proposed a method

that considers a collection as a weighted entity and computes its semantic relationship with query terms to select the most relevant ones for a given query. Wu et al. [18] exploited the learning to rank technique for resource selection. An LDA-based resource selection for result diversification in a federated search was proposed by [7]. Similarly, Garba et al. [6] proposed embedding base learning for collection selection. Their method first exploited the query log to obtain the similarity between past and current user queries using a word2vec technique. Then, those collections selected for the past queries with high similarity with the current one are selected for the current query.

In the following we focus on results merging because it is the research issue of this paper. Result merging in federated search has received researchers' attention for a long time, and many approaches have been proposed to merge the multiple results list returned by multiple resources. CORI [20] is a simple, yet effective algorithm [2], which merges results through some degree of cooperation between the broker and resources. The basic assumption of computing the documents' merging scores is that the relevance score of each document of the selected resources held in the centralized sample database are first computed by the broker. Based on the ranked list returned by the resources, the final relevance score for each document is calculated by taking the weighted average between the two scores. The semi-supervised learning (SSL) result merging algorithm was proposed in [21]. This algorithm uses the linear regression method to estimate the merging score. That is, the user query is forwarded both to the relevant resources selected and to the centralized database. Since the documents in the centralized database are sampled from resources, there is a high likelihood that the documents returned by resources are also in the central database ranked list. By mapping the overlapping documents in both the ranked list, their ranking scores could be estimated. Assuming that the documents ranking generated by the central database is a sub-ranking of the resources where the documents were sampled, the Sample Agglomerate Fitting Estimate (SAFE) algorithm [8] estimates the documents merging scores using the curve fitting method.

Bellogín, et al. [22] merged the documents in a round-robin manner. That is, the first document in the merged result list is the first document of the most relevant resource selected in the resource selection phase; the second document is the first document in the second most relevant resource, and so on. Palakodety and Callan [16] explored query reformulation techniques to improve the effectiveness of the merged result list. In Guan et al.'s [23], the latent semantic indexing (LSI) based method was used to calculate the documents' relevance scores for a given query. In such a process, documents that appear in multiple resources' result lists receive an extra boost their final scoring and ranking. The models proposed in [24] merged the results by converting the resources' documents rank into a ranking score. First, a modified reciprocal rank fusion (RRF) [25] converts rank into a score for all the documents involved. Then a frequency score for each document involved is calculated by considering its frequency of occurrences in a result list. Finally, each document's merging score is calculated as a product of its ranking score and frequency score in the multiple results list. Some previous studies on genetic programming [26] suggests that it is more effective than expert systems for certain tasks. Vo [9] exploited genetic programming by proposing a model to compute 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 can be used. A machine learning model that merges the multiple result lists based on either multiple or global models for both cooperative and uncooperative environments in the patent domain was proposed [27].

Different from the above approaches, which are proposed for general-purpose web search,[28] proposed a method that can combine results from web search e-commerce search systems. Their approach decomposes the merging process into two phases using hierarchical reinforcement learning. At each ranking position of the final merged list, the first phase selects the most relevant resource for the given user query. While in the second phase, the top ranked item of the selected resource is placed at that particular ranking position of the final merged list. In other words, their approach ranked the items not based on their relevance score but the relevancy of the resource that returned them.

Genealogy search systems crawl their data from different vital records sources like census records, birth, and death certificate records, etc. The crawled data is indexed and made available to the public that wants to trace their ancestry. A stochastic search that fused the different record types returned by the records server was proposed in [29]. In their approach, the initial user query is augmented by adding terms such as place and date of birth, family member name, etc. Then, the augmented query is routed across all the records servers, and the results returned by them are fused into a single ranking list.

Based on the above discussion, we can summarize that mainly there are two different ways of computing scores for merged documents: (1) it converts a document's ranking position into a score. (2) It download the entire documents returned by the resources to the broker server and compute scores for all the documents at query time. The drawback of the former is its effectiveness is usually low, because the usefulness of selected resources may vary significantly in support of the given query. On the other hand, the latter may not be practical in many cases. Even it does, it is very costly because it needs to process full-text of documents at query execution time. To avoid these problems, the proposed methods work with snippets which are commonly available in the real-world web environment.

3. Proposed Result Merging Algorithm

3.1. Preliminaries

Assuming a user issued a query q to the broker, the broker selects N most relevant resources in the resource selection phase, forwards the query to those selected resources, and each resource returns a ranked list of documents. Let's further assume $R_i = \{r_{i1}, r_{i2}, \dots, r_{il}\}$ for $i = \{1, 2, \dots, N\}$ is the results list returned by all N resources. Our goal is to use the results list to estimate the relevant score of each document in R_i, and use the scores to merge all of them into a single ranking list $\mathrm{R} = \{r_1, r_2, \dots, r_l\}$, which is similar to what would have been obtained, had the query q been issued on a centralized search engine.

Fig. 1 shows the major processes involved for the query processing in a federated search system. Upon receiving the user's query, the broker selects a group of sources and forwards the query to those selected. Each resource processes the query and retrieval a list of documents and returns them to the broker. Then the broker merges those documents from multiple resources,

generates a single list of documents, and presents them to the user. This study investigates how to merge documents from multiple resources, which is the final step in federated search. In particular, snippets of web documents are used for this purpose.

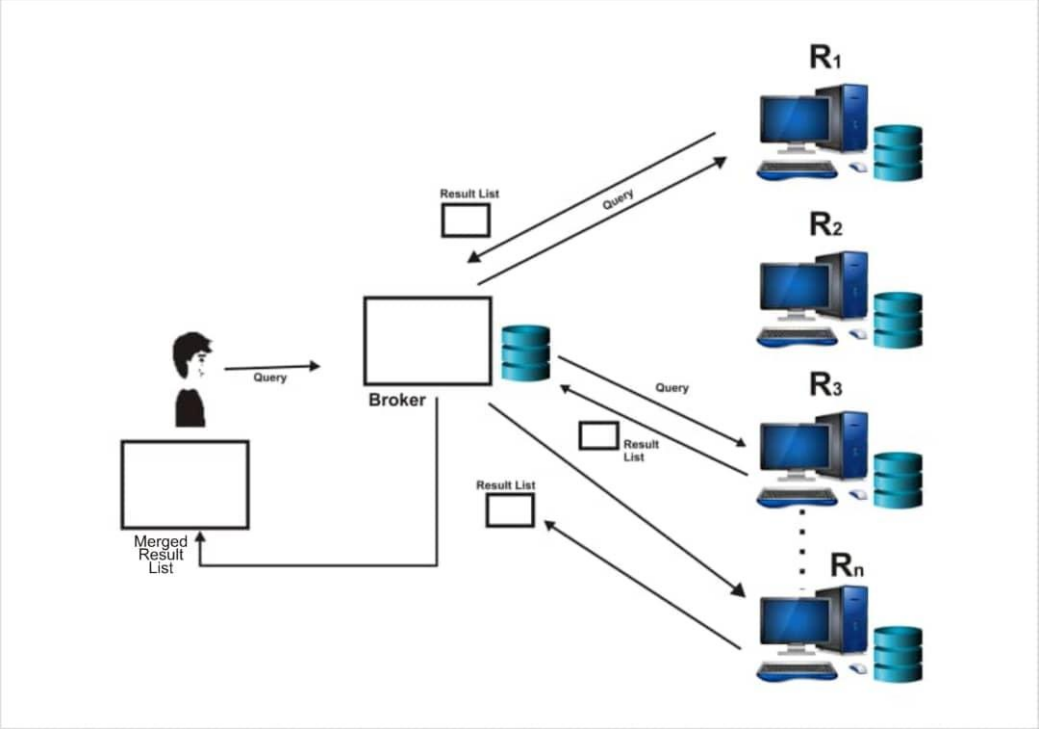


Figure 1: Query processing in a federated search system

3.2.Snippet and its Structure

Snippets have become an integral part of search engines, as most provide them to assist users in quickly identifying relevant documents returned in a search engine result page (SERP). Snippets summarize approximately 130 – 150 characters or two-three lines the search engine extracts from the web page and structures it into three parts: title, description, and URL [30]. In order to generate snippets, search engines need to analyze documents, identify keywords and important sentences from each of them, and then summarize each document as a snippet. Various studies [30-32] have shown the importance of the snippets in assisting users in clicking on relevant documents for their query.

As described earlier, on the web, most real-world resources return a ranked list of snippets for any query received. However, how to use them for results merging has not been explored. Some traditional retrieval models, such as vector space and BM25 may not work well because they mainly work with longer text documents, rather than very short ones. Especially, Term Frequency (TF) and Inverse Document Frequency (IDF) are two most important features for almost all retrieval models. But TF may not be as helpful as in long text, because in a snippet most of the terms do not occur repeatedly. IDF cannot be used at all because there is no way to calculate it. A

recent study [33] shows that most of these retrieval models show higher retrieval performance on full text documents than short ones. As such, we posit that, for an accurate estimate of a term's importance in a snippet, apart from term frequency, we should exploit the snippet's structure and differentiate terms that appear in different parts. Considering both aspects can lead to a better estimating the importance of each term in a snippet.

In the real-world web, most of the resources are autonomous; therefore, the snippets they provide may vary in structure as well. After analyzing the snippets provided in the TREC 2013 FedWeb dataset, we found some inside about the snippets' structure. We summarize our findings by categorizing the snippets in the resources result list using query *No 7090 Eurovision 2012* as follows:

Category 1: The snippets in this category contain all the three parts (i.e., title, description, and URL). Almost all the resources returned this category of snippet. As shown in Fig. 1, the documents snippets ids' are created by the FedWeb search track organizers to conceal the identity of the search engines to avoid infringement. For example, the first snippet id in Fig. 2 (i.e., FW13-e102-7090-01) is from the search engine 102 for query 7090 at ranking position 1.

```
b'<snippet id="FW13-e102-7090-01"><link>http://www.forbes.com/sites/matthewhulbert/2012/06/16/latin-america-the-enron-of-oil/</link>
<title>Latin America: The Enron Of Oil?</title>
<description>As any energy watcher will tell you, one of permanent dates in the diary is the release of BP Statistical Review, providing a detailed overview
of how the numbers do or don&#8217;t stack up. The 2012 edition was unveiled in London this week, and the findings were as interesting as [...] read &#
b'<snippet id="FW13-e102-7090-02"><link>http://www.forbes.com/sites/hilarykramer/2012/11/07/america-meet-azerbaijan/</link>
<title>America, Meet Azerbaijan</title>
<description>America, Meet Azerbaijan By Hilary Kramer   When the Soviet Union disintegrated in 1991, Americans were suddenly forced to add a host of n
independent countries to their global awareness. Some countries became a part of our lexicon more quickly than others (Ukraine and Georgia, for example).
But few of these post-Soviet countries [...] read &#187;</description></snippet>'
b'<snippet id="FW13-e185-7090-08"><link>http://www.guardian.co.uk/tv-and-radio/eurovision-2012</link>
<title>Eurovision 2012 | Television &amp;amp; radio | The Guardian</title>
<description>Latest news and comment on Eurovision 2012 from guardian.co.uk</description></snippet>
```

Figure 2: Sample of the snippets that contain title, description, and URL returned by a search engine

Category 2: In this category, the snippets contain only the title and URL. No resource returned this category of snippet in its entire result list. We made this observation for a few queries in the resources result list.

```
b'<snippet id="FW13-e002-7090-02"><link>http://liinwww.ira.uka.de/cgi-bin/bibshow?e=Njtd0ECMQ03117/fyqboefe%7d9:158588&amp;r=bibtex&amp;mode=intra</link>
<title>The Eurovision St Andrews collection of photographs</title></snippet>'
b'<snippet id="FW13-e002-7090-03"><link>http://liinwww.ira.uka.de/cgi-bin/bibshow?e=Njtd0ECMQ03117/fyqboefe%7d:1652797&amp;r=bibtex&amp;mode=intra</link>
<title>Comparison of Eurovision Song Contest Simulation with Actual Results Reveals Shifting Patterns of Collusive Voting Alliances</title></snippet>'
b'<snippet id="FW13-e002-7090-04"><link>http://liinwww.ira.uka.de/cgi-bin/bibshow?e=Njtd0ECMQ03117/fyqboefe%7d2329:8267&amp;r=bibtex&amp;mode=intra</link>
<title>User experiments with the Eurovision cross-language image retrieval system</title></snippet>'
b'<snippet id="FW13-e005-7090-06"><link>http://www.citeulike.org/article/11188869</link><title>Plant Contributed Papers</title></snippet>'
```

Figure 3: Sample of the snippets that contain title and URL returned by a search engine

Category 3: The snippets in the resources result list in this category combined the description and title into a single field and has a URL. Like category 2, we made this observation for some queries.

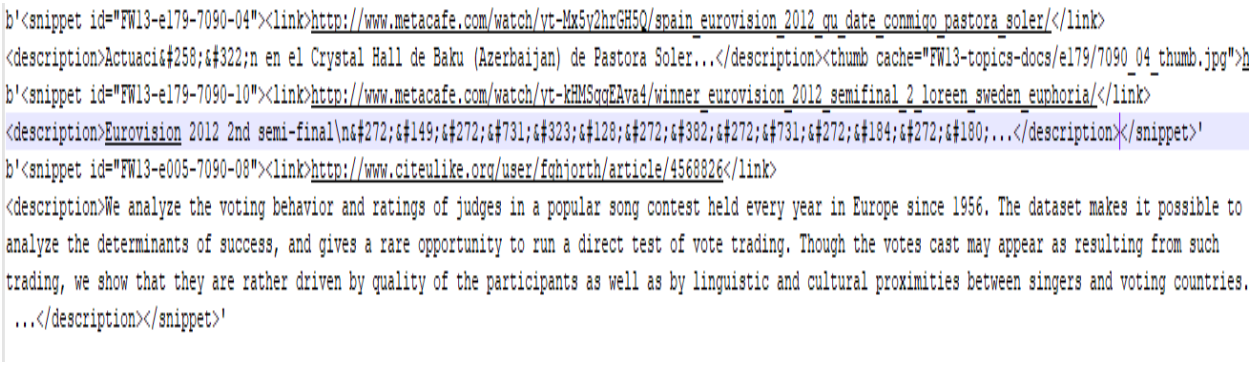


Figure 4: Sample of snippets that contain description and URL returned by a search engine

It can be seen from the above that search engines snippet do not have a uniform structure. As such, we propose a simple yet effective term weighting scheme for different types of snippets and then use it to estimate the relevance of the corresponding documents irrespective of their structure. The main idea is to treat the terms in each part in a customized way and then to combine them.

3.3. Scoring and Weighting Terms in Snippets

One of the common practices of weighting a term in IR is Term frequency and Inverse Document Frequency (TF.IDF). Inverse Document Frequency means that a rare term in the collection is more important than a common term, while term frequency suggests that the more a term t occurs in a particular document d_i, the more important the term represents that document. Due to the environment, Inverse Document Frequency cannot be considered. We only consider term frequency.

$$TF(t, R_i) = \sum_{t_q \in R_i} TF(t_q, R_i) \qquad (1)$$

where $TF(t, R_i)$ is the total number of query terms in the resource i result list.

Instead of considering all the terms, only the query terms are considered in this paper because we hypothesize they are useful, while non-query terms may not help to discriminate between relevant and non-relevant documents. Therefore, the weight of query terms is computed in each part of the snippet, as described below:

Title (T): Terms that occur in the title mostly reveal the content of the whole document. Although, to our knowledge, no study proves whether a short or long title is more informative in revealing the document content, nevertheless, we assign a higher weight to the query terms that appear in the short title than in the long one. One of our reasons is that, after a careful study of the

snippets in the dataset, we find that most of the short titles are navigational to other web pages. As such, the weight of the query terms in the snippet title is given by:

$$w(Q,\ T) = \begin{cases} \frac{\sum_{q=1}^{|Q|} TF(t_q,T)}{0.5} & if\ |T| \leq 4 \\ \frac{\sum_{q=1}^{|Q|} TF(t_q,T)}{2} & if\ |T| > 4 \end{cases} \quad (2)$$

where $TF(t_q,T)$ is the frequency of query terms in the snippet title, T_t is the number of terms in the title, 0.5 and 2 are the normalization constant we use to reward short titles and penalize long ones. The ideal values of these normalization constants are left for future work.

Description (D): The description part of the snippet contains a few lines of sentences that are considered to be enough to convey the complete information of the document. Since the length of the description varies across the snippet, a reasonable way to quantify the importance of the query terms in the description part is to count their frequency relative to the description length. Thus, the description TF is formulated as:

$$w(Q,D) = \frac{\sum_{q=1}^{|Q|} TF(t_q,D)}{len(\ D)} \quad (3)$$

where $TF(t_q,D)$ is the frequency of query terms in the snippet description and $len(D)$ is the total number of terms in a snippet description.

URL (U): Since the URL is the gateway for reaching any webpage on the web, this leads the owners of the web pages to create URLs and page titles to reflect their content. Therefore, if the query terms appear in the URL, it is an indication that the document is relevant to the search query. For this reason, higher weight is assigned to the query terms in the URL, which we calculate using the following expression:

$$w(Q,U) = \frac{\sum_{q=1}^{|Q|} TF(t_q,U)+1}{0.25} \quad (4)$$

where $TF(t_q,U)$ is the frequency of query terms in the URL, the value of 1 is acting like a smoothing parameter since some URLs do not contain query terms, and 0.25 is a normalization constant that gives higher weight to the query terms that appear in the URL.

Now that we have computed the weight of the query terms in each part of the snippet, the next step is to compute the relevance merging score of the resources returned results list.

3.4. Result Merging Score

Based on our observation about the snippets structure in the dataset, as explained in Section 3.2, the relevance merging score of each document can be calculated by summing the weight of

the query terms appearing in the part of the snippet under consideration and can be obtained using any of the equations below:

In the first case, we assumed the snippet returned by the resources contained the title, description, and URL (TDU). In this scenario, the merging score of each document can be estimated using Eq. (5)

$$TDU = \log\left(\left(\sum w(Q,T)w(Q,D)w(Q,U)\right) \times TF\left(t_q, R_i\right)\right) \quad (5)$$

In the second case, we assumed the snippet returned by the resources contain the title and URL (TU). In this scenario, the merging score of each document can be estimated using Eq. (6)

$$TU = \log\left(\left(\sum w(Q,T)w(Q,U)\right) \times TF\left(t_q, R_i\right)\right) \quad (6)$$

Finally, we assumed the snippet returned by the resources contain the title and description (TD). In that scenario, the merging score of each document can be estimated using Eq. (7)

$$TD = \log\left(\left(\sum w(Q,T)w(Q,D)\right) \times TF\left(t_q, R_i\right)\right) \quad (7)$$

where $w(Q,T)$, $w(Q,D)$, and $w(Q,U)$ are the weights of the query terms in each part of the snippet part and $TF(t_q, R_i)$ is the total number of query terms in the resource that returned that particular snippet. The reason for taking the logarithm of the scores is to damp the effect of the TF values. From Eq. (5) to (7), it can be observed that no assumption is made about the exact structure for the resources snippets. Instead, we experiment with all three equations one at a time by assuming that it is the snippets structure which all the resources returned. Based on the score obtained with each equation, we merged the resource results into a single ranking list. Finally, performance analysis was conducted to see which method had the highest merged result effectiveness.

Based on the results obtained using the proposed methods and baselines, we then investigate how utilizing resource weight in calculating the merging score affects the effectiveness of the merged result list. Specifically, we experiment with the resource score obtained in the resource selection phase, and the number of documents in the resource ranked list. Therefore, the new forms merging score is:

$$S\left(s_j\right) = S(s_{j-1}) \times R_k \quad (8)$$

where $S(s_j)$ is the new merging score, $S(s_{j-1})$ is the merging score obtained using any of the Eq. (5) to (7) or the baselines, and the R_k takes the value of resource score or number of documents in the resource ranked list.

4. Experiment

4.1.Dataset

All the experiments in this paper were conducted using the TREC 2013 FedWeb Greatest Hits dataset[1]. This dataset has been used for various TREC federated search web tracks. The dataset was created to provide resources similar to the real-world federated setting in order to stimulate research on federated search and discourage artificial creation of the resources by dividing the TREC Web track datasets and arbitrarily assigning retrieval models to them. The dataset was sampled from 157 real-world search engines in 24 vertical categories (i.e., news, entertainment, games, academics, videos, images, etc.). In generating, a total of 2000 queries were issued to each search engine and a total of 1,973,591 snippets were extracted in all. Each search engine outputs an average of 12,570.6 results and saved in XML format.

Its advantage over the artificially created ones is that the search engines retrieve the sampled documents with their customized retrieval models. We considered the search engines as the resources, and the 50 topics provided with the dataset were used as search queries. The TREC assessors graded each query result in the dataset using five level relevance judgments. That is, the results were judged as navigational (Nav), top relevant (Key), highly relevant (HRel), relevant (Rel), and non-relevant (NRel).

4.2. Resource Selection

In federated search, the user's query is not routed to all the resources, as some may not contain relevant documents for the given query. The resource selection phase aims to select a few of the most relevant resources to search for each given query. In this paper, we modified the central-rank-base resource selection algorithm [34] to select the relevant resources. Specifically, we indexed all the snippets provided in the dataset and ran the provided queries on the index. Resource weight (score) is determined by the number of relevant documents it has in the top *N*-ranked list as well as its rank position. The contribution of each document to the score of the resource that returned it is expressed using the equation below:

$$R(s_l) = \gamma\, e^{(-\delta \times r)} \tag{9}$$

where $R(s_l)$ is the contribution score of the document s_l, δ and γ are constant values, and r is the rank of the document. Then, the weight of the resources is calculated using the expression below:

$$R_k = n \sum_{j=1}^{n} R(s_l) \tag{10}$$

where n is the total number of documents resource i has in the top N ranked list. The values of N, γ, and δ are set to 40, 0.28, and 1.2, respectively, in the experiment. These are the same values used in [34].

4.3. Experimental Setup and Evaluation Metric

[1] https://fedwebgh.intec.ugent.be/fedwebgh

To perform the experiment, the snippets provided in the dataset were indexed in the Lucene Apache Solr[2] retrieval system. For resource selection, we queried both the title and description of each indexed snippet. Then the two scores were summed as the snippet relevance score. For those snippets with only title or description, a value of zero was assigned to the missing field. The ranking was generated using the VSM retrieval model. For each given query, the top three and five most relevant resources were selected to merge their results. Also, the same approach of querying title and description was followed to calculate the baselines' merging scores.

For merging the results using the proposed methods, three separate experiments were conducted, one for each of the Eq. (5) to (7). In each experiment, only the parts specified in Eq. (5) to (7) were queried. When a missing part was encountered in a particular snippet, that snippet was not included in the ranking of the merged result. For instance, assuming we were calculating the merging scores of the documents using Eq. (6); any snippet found without a URL would be dropped before we merged the result.

The results were evaluated using nDCG@k [35], which is the official evaluation metric for the 2013 FedWeb result merging task. The effectiveness of the methods is measured at top *k* results; the value of *k* is set to 5, 10, 15, and 20, respectively.

We compared the effectiveness of the proposed result merging methods with the results obtained by ranking the snippets using the three well-known retrieval models, i.e., Vector Space Model (VSM) using standard TF.IDF weighting scheme, Divergence from Randomness (PL2), and BM25 as standard baseline methods. All the parameters for the classical retrieval models were in their default Solr setting. Furthermore, CORI [20], SSL [21], and TF-RF proposed by Vo [9] are used as the state-of-the-art baseline methods.

5. Experimental Results and Discussion

5.1. Result

Table 1 shows the effectiveness of the proposed methods compared to the baselines when the top three and five resources are selected. Our methods are denoted as TDU-TF, TD-TF, and TU-TF. The performance metrics were averaged across all the given queries. Then, we used a paired t-test to assess significant differences between the proposed methods and the baselines. The results show that the improvement of the proposed methods over the baselines at some of the cut-off rank is statistically significant at p-value <0.05.

From the result, it can be seen that the proposed methods have superior performance compared to the baselines. Boldface is used to indicate the method with the highest score in each set of the experiment. As the results show, TD-TF maintains a consistent performance when the top three resources are selected as it outperforms the two other proposed methods by over 2.8% on NDCG@5 and over 4.4% on NDCG@10. It has the highest improvement when compared with

[2] https://lucene.apache.org/solr/

classical retrieval models and state-of-the-art methods for NDCG@5 and NDCG@10, which is over 16% and is statistically significant. However, when the top five resources are selected, TU-TF and TDU-TF show the highest performance. At the same time, the performance of TD-TF was drastically reduced by over 27% on NDCG@5 and 26.4% on NDCG@10 compared to its effectiveness when the top three resources were selected. Additionally, we found that there was no statistically significant difference among the proposed methods across all the cut-off ranks when the top 3 and 5 resources were selected.

5.2. Baseline Results

For merging the results using the traditional retrieval models (i.e., BM25, PL2, and VSM), we indexed the results returned by the top three and five resources one at a time. For each document snippet in the resources result list, we queried both the title and description parts separately and then combined the two scores as the document relevance score. The scores are used to merge the results into a single list. For the SSL [21] and TF-RF [9] models, we obtained the merged results following the procedure described in their papers.

Among the baselines result, SSL showed strong performance on NDCG @15 and NDCG@20 when the top 5 resources are considered. It achieved its highest performance on NDCG@20, where it outperformed the proposed methods. This might be the result of having more overlap documents between the central database generated and the resources generated ranking lists. Moreover, it was reported in the SSL paper [21] that its highest performance was achieved by selecting more resources compared to selecting less resources. Similarly, CORI performs slightly higher compared to the remaining baselines on the NDCG@5 and NDCG@10. A possible explanation for this may lie in the way that we derived the local score of the documents in the resources returned list. That is, we obtained the local score of each document by taking the reciprocal of its rank and then multiplying it by the BM25 score of its description. Based on the results in Table 1, we can conclude that using snippets is helpful for results merging to improve performance because all related methods perform well and better than some other methods involved.

Table 1

Performance comparison of the different methods used to merge the results when the top three and five most relevant resources are selected.

	Top Three Resources Selected				Top Five Resources Selected			
	nDCG				**nDCG**			
	@5	@10	@15	@20	@5	@10	@15	@20
BM25	0.3459	0.3374	0.3293	0.3220	0.2975	0.2958	0.2935	0.2886

VSM	0.3466	0.3401	0.3351	0.3253	0.2999	0.3025	0.2978	0.2948
PL2	0.3300	0.3293	0.3266	0.3219	0.2595	0.2741	0.2640	0.2718
TF-RF	0.3499	0.3383	0.3340	0.3249	0.2653	0.2932	0.2940	0.2945
CORI	0.3713	0.3559	0.3412	0.3329	0.3247	0.3011	0.2887	0.3095
SSL	0.3080	0.3154	0.3441	0.3592	0.2845	0.2979	0.3055	**0.3242**
TDU-TF	0.4093	0.3949	0.3985	0.3931	0.3724	0.3316	**0.3152**	0.3171
TD-TF	**0.4298**	**0.4122**	**0.4113**	**0.3952**	0.3118	0.3035	0.3030	0.3085
TU-TF	0.4181	0.3862	0.3886	0.3890	**0.3758**	**0.3322**	0.3150	0.3155

In Table 1, we can also observe an overall decrease in the effectiveness of the merged result list when the number of selected resources increasing from three to five. The results suggest that selecting fewer relevant resources for results merging may lead to higher effectiveness.

Table 2

Retrieval performance of the different merging methods when the resources score obtained in the resource selection phase is utilized in calculating the documents merged scores. In all the cases, the top three and five most relevant resources are selected.

	Top Three Resources Selected				Top Five Resources Selected			
	nDCG				**nDCG**			
	@5	@10	@15	@20	@5	@10	@15	@20
BM25	0.3790	0.3526	0.3396	0.3322	0.3243	0.3098	0.3047	0.3026
VSM	0.3777	0.3566	0.3406	0.3364	0.3324	0.3185	0.3138	0.3141
PL2	0.3846	0.3661	0.3567	0.3407	0.3382	0.3411	0.3297	0.3248
TF-RF	0.3852	0.3559	0.3444	0.3200	0.3411	0.3323	0.3329	0.3164
CORI	0.3872	0.3593	0.3422	0.3325	0.3449	0.3202	0.3098	0.3017
SSL	0.3312	0.3223	0.3389	0.3474	0.3168	0.3218	0.3330	0.3352
TDU-TF	0.4263	0.4183	0.4061	0.3952	**0.3832**	0.3579	0.3423	0.3432

TD-TF	**0.4521**	**0.4405**	**0.4267**	**0.4011**	0.3698	0.3556	**0.3499**	**0.3452**
TU-TF	0.4341	0.4203	0.4038	0.3943	0.3790	0.3579	0.3401	0.3397

Table 3

Retrieval performance of the different merging methods when the number of documents in the resources rank is utilized in calculating the documents merged scores. In all the cases, the top three and five most relevant resources are selected.

	Top Three Resources Selected				Top Five Resources Selected			
	nDCG				**nDCG**			
	@5	@10	@15	@20	@5	@10	@15	@20
BM25	0.3308	0.3221	0.3176	0.3119	0.2607	0.2894	0.2834	0.2784
VSM	0.3337	0.3256	0.3207	0.3144	0.2610	0.2831	0.2777	0.2723
PL2	0.3478	0.3273	0.3218	0.3213	0.2799	0.3236	0.3006	0.3004
TF-RF	0.3573	0.3371	0.3278	0.3200	0.2537	0.2799	0.2911	0.3045
CORI	0.2914	0.3244	0.2987	0.2889	0.2673	0.2979	0.3089	0.3011
SSL	0.3237	0.3290	0.3314	0.3474	0.2743	0.2900	0.3103	0.3187
TDU-TF	0.4073	0.3961	0.3964	0.3868	**0.3732**	**0.3501**	**0.3304**	0.3181
TD-TF	**0.4376**	**0.4200**	**0.4137**	**0.3938**	0.3535	0.3335	0.3294	**0.3318**
TU-TF	0.4094	0.3919	0.3875	0.3824	0.3675	0.3486	0.3282	0.3190

5.3. Discussion

The experimental result presented in Table 1 shows the performance of the proposed methods as each one outperformed the baselines method except SSL at NDCG@20 when the top 5 resources were selected. Using the classical retrieval models as the baselines methods, VSM performs slightly better compared to the BM25 and PL2 on NDCG@5 and NDCG@10. This may not be

unconnected to the way each method computes its query-document similarity and the way we query the title and description separately before summing the two scores as relevance score of the documents.

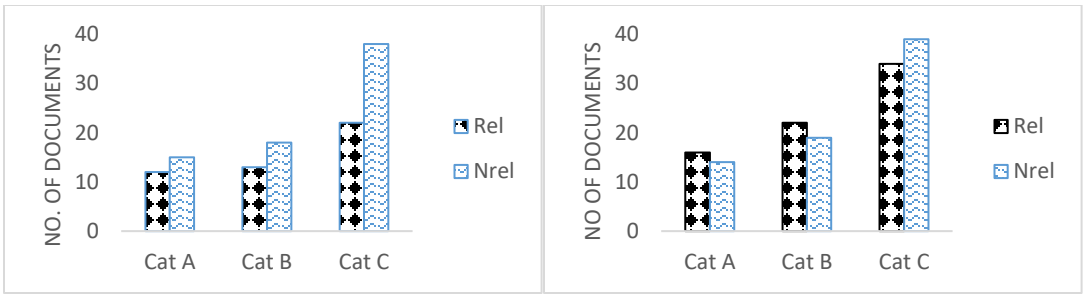
Now that we have observed the overall performance of the merged result list, let us first look at the reason that affords the proposed methods superior performance compared to the baselines. Then, we will finally discuss the results in the form of comparative analysis on how including resource weight in calculating the merging score affects the effectiveness of the merged result list.

5.3.1. Effect of the snippets' title length on document relevancy

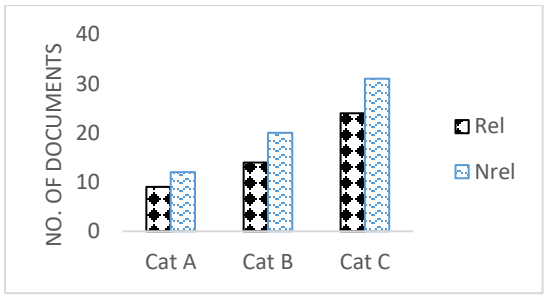
We attribute the robustness of the proposed methods to our hypothesis that a short title in the snippets signals a high probability of relevant documents. As can be observed in computing the weight of query terms in the title, we penalize long titles and reward short ones. However, prior to making this hypothesis, we did some analysis about the length of the titles under two conditions: (i) with stop words and (ii) without the stop words.

i. With stop words, the longer title is 17 terms, the shorter title is 1 term, the mode is 6 terms, and the mean is 6.87 terms.
ii. Without the stop words, the longer title is 14 terms, the shorter title is 1 term, the mode is 4 terms, and the mean 4.29 terms.

To test the hypothesis, we first performed a post-retrieval analysis of the merged results under each of the conditions. Specifically, we extracted the documents with a snippet title length of less than or equal to four terms (without stop words) and less than or equal to six terms (with stop words) from the merged results list. Next, their relevancy was checked using TREC relevance judgment. Furthermore, for each condition, we categorized the titles based on the number of terms. That is, those with less than or equal to two terms of Cat-A, three terms of Cat-B, and more than three terms of Cat-C. Fig. 5 shows the results of both relevant and non-relevant documents with stop words in the top 20 documents of the merged result for the given queries, whereas those without stop words are shown in Fig. 6. From the results in Figs. 5 and 6, it can be observed that there is no difference among the two conditions in Cat-A and Cat-B under each of the proposed methods. For example, if we look at Figs. 5(b) and 6(b), we can see that the Cat-A and Cat-B results are the same in both figures. This is because we have not observed any document among the top 20 documents with a title length less than or equal to three terms that has stop words among the terms. However, when it comes to Cat-C, the conditions without the stop words have more relevant documents than the condition with the stop words, as shown in Figs. 5 and 6. A conscious observation of Figs. 5 and 6 reveals another interesting fact: the majority of the resources return more documents with long titles than those with short ones. The findings also show that more non-relevant documents are typically found under longer titles. From these results, we can safely conclude that a short title in the snippets is more informative in revealing relevant documents than the long ones when the snippets are returned by multiple resources.

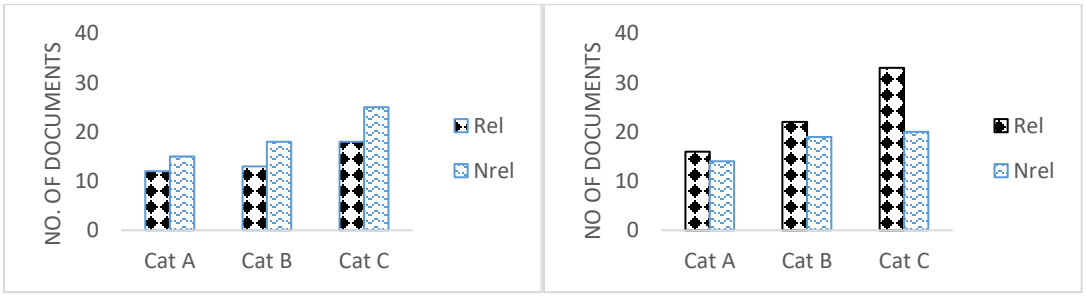


(a) TDU-TF Merging Method **(b) TD-TF Merging Method**

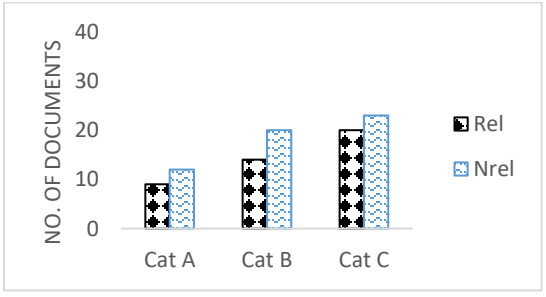


(c) TU-TF Merging Method

Figure 5: Graph showing the number of relevant and non-relevant documents based on their snippet title length with the stop words.



(a) TDU-TF Merging Method **(b) TD-TF Merging Method**



(c)TU-TF Merging Method

Figure 6: Graph showing the number of relevant and non-relevant documents based on their snippet title length without stop words.

5.3.2 Effect of utilizing resource weight in calculating the documents merging score

We investigated to what extent does using the resources score or number of documents in the resource ranked list improves the effectiveness of the merged result list. Table 2 depicts the effectiveness of the merged result list when the resource score obtained in the resource selection phase is utilized in computing the documents merging scores. By comparing Tables 1 and 2, we can observe an increase of the merged result effectiveness across all the nDCG ranks in Table 2. This increase in the merged result effectiveness manifests more when the top five resources are selected. From this result, it can be concluded that utilizing the resources score in calculating the documents merging score improves the effectiveness of the merged results list.

After we observed the effect of utilizing the resource score on the performance of the merged result list, the next step was to look at how the use of the number of documents in the resources ranked list also affects the performance of the merged result. Table 3 depicts the effectiveness of the merged result when the number of documents in the resources ranked list is utilized in computing the documents merging score. Comparing Tables 1 and 3, we can observe a mixed performance of the merged result list in Table 3. A closer look at the result in Table 3 would reveal two things; first, except for the TD-TF method and SSL at NDCG@5 and NDCG@10, we can observe a decrease in the effectiveness of the merged result list when the top three resources are selected. Second, except for the baselines, an increase of effectiveness can be observed when the top five resources are selected. The second point proves the robustness of the proposed methods in estimating the documents' merging score as documents from the relevant resources were assigned higher merging scores. Our conclusion from this result is that using the number of documents in the resources rank list in computing the documents merging score may or may not improve the retrieval effectiveness of the merged results list.

In general, our findings can be summarized as follows:

- Assigning different weights to the query terms based on their occurrences in each part of the snippets is effective, considering the performance of our methods compared with the baselines.
- Utilizing only the snippets' title and description in computing the merging score of the multiple resources results list is more effective than using all the three snippet parts or using title and URL.
- Irrespective of the approach used to merge the multiple results list, including the resources score obtained in the resource selection phase in computing the documents merging score improves the effectiveness of the merged result list. In contrast, the use of the number of documents in the resources ranked list in some cases decreases the effectiveness of the merged result list.
- Given the nature of the federated settings, that is, multiple results are returned for the issued

query to the resources; in the results list, the snippets with short title length signal some level of relevance.

6. Conclusion

This paper presents a new result merging method for the uncooperative environment in federated search. Our method uses only the snippets returned in the resources' ranked list to calculate the merging scores. To accomplish this, we proposed a term weighting scheme for snippets, which quantifies the importance of query terms based on their occurrence in each part of the snippet. We used this weighting scheme to calculate the merging score by summing the weight of query terms in the part of snippet it occurs. Then, we multiplied it by the frequency of query terms of the resource that returned it. The experimental results conducted with the TREC 2013 FedWeb dataset reveal that our proposed model has superior performance compared to the baselines.

We also investigated how resource weight in computing the merging score affects the effectiveness of the merged result list. In particular, we first experimented with the resource score obtained in the resource selection phase and second with the number of documents in the resource ranked list. Our experimental results show that a further performance improvement is achievable by including resource scores obtained in the resource selection phase for results merging. On the other hand, the use of the number of documents has a mixed impact on results merging performance.

Our experiment has been affected by the inconsistent snippet structure returned by the resources (search engines), which has led us drop many documents. For this reason, in our future work, we want to investigate how the snippet structure affects the effectiveness of our methods. Additionally, since our work assumed that the resources with more frequency of query terms would be more relevant, we will also investigate whether the number of documents in the resources result list impacts this assumption. Further, the effectiveness of our methods on short and long resource results lists can also be investigated.

References

[1] P. Liakos, A. Ntoulas, A. Labrinidis, and A. Delis, "Focused crawling for the hidden web," *World Wide Web,* vol. 19, pp. 605-631, 2016.

[2] D. Hong and L. Si, "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*, 2012, pp. 821-830.

[3] T. T. Avrahami, L. Yau, L. Si, and J. Callan, "The FedLemur project: Federated search in the real world," *Journal of the American Society for Information Science and Technology,* vol. 57, pp. 347-358, 2006.

[4] D. Nguyen, T. Demeester, D. Trieschnigg, and D. Hiemstra, "Resource Selection for Federated Search on the Web," *arXiv preprint arXiv:1609.04556,* 2016.

[5] J. Callan and M. Connell, "Query-based sampling of text databases," *ACM Transactions on Information Systems (TOIS)*, vol. 19, pp. 97-130, 2001.
[6] A. Garba, S. Khalid, I. Ullah, S. Khusro, and D. Mumin, "Embedding based learning for collection selection in federated search," *Data Technologies and Applications*, 2020.
[7] L. Li, Z. Zhang, and S. Wu, "LDA-Based Resource Selection for Results Diversification in Federated Search," in *International Conference on Web Information Systems and Applications*, 2018, pp. 147-156.
[8] M. Shokouhi and J. Zobel, "Robust result merging using sample-based score estimates," *ACM Transactions on Information Systems (TOIS)*, vol. 27, pp. 1-29, 2009.
[9] H. T. Vo, "New re-ranking approach in merging search results," *Informatica*, vol. 43, 2019.
[10] N. Craswell, D. Hawking, and P. B. Thistlewaite, "Merging Results From Isolated Search Engines," in *Australasian Database Conference*, 1999, pp. 189-200.
[11] Y. Rasolofo, D. Hawking, and J. Savoy, "Result merging strategies for a current news metasearcher," *Information Processing & Management*, vol. 39, pp. 581-609, 2003.
[12] D. Pal and M. Mitra, "ISI at the TREC 2013 Federated task," in *TREC*, 2013.
[13] E. Di Buccio and M. Melucci, "University of Padua at TREC 2014: Federated Web Search Track," PADUA UNIV (ITALY)2014.
[14] A. Giachanou, I. Markov, and F. Crestani, "Opinions in federated search: University of lugano at TREC 2014 federated web search track," LUGANO UNIV (SWITZERLAND)2014.
[15] T. Demeester, D. Trieschnigg, D. Nguyen, K. Zhou, and D. Hiemstra, "Overview of the TREC 2014 federated Web search track," GHENT UNIV (BELGIUM)2014.
[16] S. Palakodety and J. Callan, "Query transformations for result merging," CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE2014.
[17] T. Demeester, D. Trieschnigg, D. Nguyen, and D. Hiemstra, "Overview of the TREC 2013 Federated Web Search Track," Ghent University (BELGIUM)2013.
[18] T. Wu, X. Liu, and S. Dong, "LTRRS: A Learning to Rank Based Algorithm for Resource Selection in Distributed Information Retrieval," in *China Conference on Information Retrieval*, 2019, pp. 52-63.
[19] B. Han, L. Chen, and X. Tian, "Knowledge based collection selection for distributed information retrieval," *Information Processing & Management*, vol. 54, pp. 116-128, 2018.
[20] J. Callan, "Distributed information retrieval," in *Advances in information retrieval*, ed: Springer, 2002, pp. 127-150.
[21] L. Si and J. Callan, "A semisupervised learning method to merge search engine results," *ACM Transactions on Information Systems (TOIS)*, vol. 21, pp. 457-491, 2003.
[22] A. Bellogín, G. G. Gebremeskel, J. He, J. Lin, A. Said, T. Samar, *et al.*, "CWI and TU Delft at TREC 2013: Contextual suggestion, federated web search, KBA, and web tracks," in *Proceedings of the Text REtrieval Conference (TREC)*, 2013.
[23] F. Guan, Y. Xue, X. Yu, Y. Liu, and X. Cheng, "ICTNET at Federated Web Search Track 2013," in *TREC*, 2014.
[24] A. Mourao, F. Martins, and J. Magalhaes, "NovaSearch at TREC 2013 Federated Web Search Track: Experiments with rank fusion," in *TREC*, 2013.
[25] G. V. Cormack, C. L. Clarke, and S. Buettcher, "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, 2009, pp. 758-759.

[26] P. J. Angeline, "Genetic programming: On the programming of computers by means of natural selection: John R. Koza, A Bradford Book, MIT Press, Cambridge MA, 1992, ISBN 0-262-11170-5, xiv+ 819pp., US $55.00," ed: Elsevier, 1994.

[27] V. Stamatis and M. Salampasis, "Results Merging in the Patent Domain," in *24th Pan-Hellenic Conference on Informatics*, 2020, pp. 229-232.

[28] R. Takanobu, T. Zhuang, M. Huang, J. Feng, H. Tang, and B. Zheng, "Aggregating e-commerce search results from heterogeneous sources via hierarchical reinforcement learning," in *The World Wide Web Conference*, 2019, pp. 1771-1781.

[29] P. Jiang, Y. Yang, G. Bierner, F. A. Li, R. Wang, and A. Moghtaderi, "Ranking in Genealogy: Search Results Fusion at Ancestry," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2754-2764.

[30] D. Maxwell, L. Azzopardi, and Y. Moshfeghi, "A study of snippet length and informativeness: Behaviour, performance and user experience," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 135-144.

[31] T. Paek, S. Dumais, and R. Logan, "WaveLens: A new view onto internet search results," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2004, pp. 727-734.

[32] M. Kaisser, M. A. Hearst, and J. B. Lowe, "Improving search results quality by customizing summary lengths," in *Proceedings of ACL-08: HLT*, 2008, pp. 701-709.

[33] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power, "End-to-end neural ad-hoc ranking with kernel pooling," in *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*, 2017, pp. 55-64.

[34] M. Shokouhi, "Central-rank-based collection selection in uncooperative distributed information retrieval," in *European Conference on Information Retrieval*, 2007, pp. 160-172.

[35] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, *et al.*, "Novelty and diversity in information retrieval evaluation," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008, pp. 659-666.