ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 2, Issue 3 (July - Sept. 2014)
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014)

# A Fingerprint Based Approach for Resource Selection in Federated Research

[I]Benjamin Ghansah, [II]Ben-Bright Benuwa
[I]School of Computer Sciences, Data Link Institute, P. O Box 2481, Tema Ghana, West Africa
[II]School of Computer Sc. and Communication Engg. Jiangsu University 301 Xuefu Road 212013 Zhenjiang, Jiangsu China

**Abstract**
*In a distributed information retrieval (DIR) environment, after the descriptions of the available resources are obtained by the broker, the next most important phase is to select resources that best answer a user query: this is often referred to as obtaining relevant resources or resource selection. For an effective result output, it would be expedient to include the entire resources available for searching, but an exhaustive resource-search would be very expensive in terms of source load, execution time, quality of results, document overlap removal, network load, among others. This paper addresses the challenge of selecting relevant resources amid duplicate resources, which competes for inclusion with novel resources in a cooperative distributed information retrieval environment. We present DelCosim, a resource selection approach which addresses this issue of duplicate resources by using a local fingerprint method to identify and remove a duplicate pair with a minimum size.*

**Keywords**
*Distributed information retrieval, Resource selection, Duplicate resources, Fingerprint*

## I. Introduction

Distributed Information retrieval (DIR) is a technique where a user query is searched from a number of multiple text resources simultaneously, with the help of a broker (central coordinator), after which results retrieved from these resources are merged into a unified whole and presented to the user. There are two environments in Distributed Information Retrieval setting: Cooperative and uncooperative. In a cooperative environment, resources inform brokers about their contents by providing information such as their term statistics and also metadata information such as the number of documents in their resources, number of terms in resource, average document size etc. According to Gravano et al. [9], this information is exchanged through a set of agreed protocol called STARTS between candidate resources. In an uncooperative environment, resources do not provide any information about their contents to the broker, the broker estimates their contents by sending probe queries to each resource: a process known as query-based sampling [5]. The phase described above, which is aimed at gathering information about candidate resources is known as resource description or resource representation. It is so called because, it gives the broker an idea of the constituents of the available resources, so as to route a user query to relevant resources during query time. The next phase after resource description is resource selection: where resources deemed relevant are selected by the broker to answer the user query. The final stage is merging the results from different resources into a unified whole and presented to the user.

### A. Motivation

Resource selection has in the past been a popular research area in distributed information retrieval due to its importance in the final result expected by a user and the perception in terms of evaluation of the overall retrieval system. Most prior research in DIR assumes that the available resources are sets of disjoint resources, with no overlap between the individually indexed resources[9, 17, 23]. This assertion is generally an invalid one and was made evident by Alan et al.[1]. They showed that duplication and near-duplication of documents between servers is a significant problem and one of the major challenges in DIR. Note that, in a typical DIR environment, resources crawl the same web to search for documents for indexing;this suggests that, various resources have the tendency of selecting the same sets of documents, primarily based on the visibility, accessibility or relevance attached to the document by the crawlers of candidate resources. In view of this, during a particular resource selection expeditionin a DIR environment, the propensity of selecting resources with overlapping or duplicate resource is very high. This phenomenon, if not checked would prevent the broker from selecting other resources which could be relevant but covers different aspects of the query. For example, assuming there are twelve (12) resources available for selection, and two (2) of the resources contains a number of similar documents. In the computation of "query- term" and "resource-term" similarity, the two resources having similar documents would be assumed to have similar weights, especially if the query terms are present in the duplicate documents in each of the two resources. Assume again that the retrieval system selects the highest ten (10) ranked resources, it would be observed that the two resources with overlapping documents would have similar weights hence would find their way among the top 10 resources, at the expense of seemingly relevant resources which have other relevant aspects that were not captured by the duplicate resources. In a more practical sense, consider various resources competing in the same sector or industry, for example the news industry. In this industry, various news websites always competes to be the first to publish a news item. In their bid to satisfy their readers, they populate their website with current articles mostly from the same external sources, sometimes without any modification. Consider again an industry such as the movie industry. Their online databases (e.g.collectorz.com/movie/,themoviedb.org/, omdbapi.com/, imfdb.org etc.) are composed of very similar categories of movies and movie titles that overlap. Work by Selberg et al.[21] and Zamir et al.[29] attempted to solve the problem of overlapping on the web by using websites that points to the same url as a bases for duplicate detection. Their method was however found to be suboptimal, especially in the Web environment. For example, a news website such as CNN.com would have a news item that is much the same as a news item on BBC.com - but BBC.com and CNN.com are different in terms of url (i.e. domain names), making

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

the use of url, as a yardstick for duplicate detection ineffective.

### B. Contribution

Our main contribution is to improve the performance effectiveness by explicitly removing a resource- duplicate pair from a distributed search environment. This is done with the intuition of giving way for inclusion of a rather novel resource, which hitherto would not have a place in the selected resources during a typical resource selection phase. We present a novel algorithm which removes a duplicate resource pair that has minimum size from a combination of a fingerprint method and a cosine similarity algorithm. Our overall goal is to make distributed search in large scale possible. With this paper we hope to make a crucial step towards this goal. Section 2 gives an overview of related work. Section 3 presents our novel method DelCosim. The discussion is in section 4. Finally we conclude in Section 5..

## II. Related Work

A significant amount of study in DIR has been channeled to the development of models and algorithms in addressing the issue of resource selection. A comparative study by Powell and French [18] showed that resources are ranked when representatives (resource description) of each resource is created based on their local term frequencies (tf) and document frequencies (df) information, which are compared to similar frequencies of the user query during query time. Most researchers such as [6, 9, 12-14, 24, 26, 28] employed this resource selection method in their work. Unfortunately, these researchers assumed the absence of overlap between resources.
In recent times, researches have debunked the assertion of non-existence of duplicates or overlapping resources in a DIR. Wu [25], showed using a utility objective function method comparing four parameters: relevance, time, cost and duplicates ratio. Their quest was to find out which of these parameter(s), if optimized would improve the resource selection process. Their experiments showed that, relevance and duplicate ratio has a direct relation; which implies duplicate detection is as important as relevance. COSCO, another duplication detection algorithm, developed by Hernandez et al.[11] took into consideration the coverage of individual resources and the overlap between resources before determining which resource should be called next. Shokouhi and Zobel [22], proposed a novel algorithm called Relax, an overlap-aware method that selects resources that are expected to maximize the number of unique relevant documents in the final results. Work by Bernstein et al.[3], tested Greedy Hash Vector (GHV) on three DIR testbeds with overlapping resources, and showed that GHV can be effectively used for detecting and removing duplicate documents from the merged results. However, their method was implemented at the merging phase of DIR.
Other approaches related to Overlap estimation among data sources in the area of distributed and P2P information retrieval includes the work of Bender et al.[2], they argued for the extension of existing quality measures using estimators of mutual overlap among resources. They implemented a prototype coined MINERVA 1 which was used on a P2P web search engine, this allowed handling large amounts of data in a distributed and self-organizing manner. Their experiments showed that, taking overlap into account during resource selection can drastically decrease the number of resources that have to be contacted in order to reach a satisfactory level of recall. Michel et al.[15] presented a comprehensive evaluation of overlap estimators known as Integrated Quality Novelty (IQN). They showed how their approach could be incorporated into an efficient, iterative approach to query routing. They further enhanced their approach using histograms, combining overlap estimation with the available score/ranking information. Nie et al.[16], presented a set of connected techniques that estimates the coverage and overlap statistics while keeping the needed statistics tightly under control. Their approach uses a hierarchical classification of the queries, and threshold based variants of familiar data mining techniques to dynamically decide the level of resolution at which to learn the statistics. Saleem et al. [19], presented DAW, a duplicate-aware approach to federated querying over the Web of Data. DAW is based on a combination of min-wise independent permutations and compact data summaries..

### 1. Other Document Similarity methods

Other methods used in determining similarity between documents could be grouped as Keyword similarity matching method (KSM), Substring similarity matching method (SSMM) or Fingerprint Matching method (FMM). In the KSM Fullam et al [8], Keywords are extracted from topic of a document and converted to numeric weights. This weight is compared to other documents topic-weight and when the similarity measure (S) exceeds a particular threshold, the documents involved are classified as similar. The disadvantage with this method is that, it assumes similarity occurs in topically similar documents.
SSMM[29] use text compression algorithms to identify matches between documents. The method compress a document to canonical sequence by removing morphological connotations and stopwords, and finally the resultant substrings are compared with other candidate documents that have gone through similar transformation for overlap detection. Two other methods mostly used in traditional IR techniques are Jaccard similarity method: which was used by Haveliwala and Taher [10] and also the basic Cosine Similarity method: used by Xiao et al. [27]. Document fingerprinting is an efficient and effective technique for detecting full and partial similarity between documents. Most techniques for detecting full or partial copies in the past make use of an n-gram of contiguous substring of length n; usually a document is divided into a number of n-grams where n is a parameter chosen by the user. For example, consider a hypothetical text "love hurts, love wisely ". Removing morphological, whitespaces and stopwords would transform the text into a canonical text "lovehurtslovewisely". The next stage is to break the resultant text to a sequence of a specified gram. In this example we choose 4-grams; creating various individual words like - love oveh vehu ehur hurt urts rtsl tslo slov love ovew vewi ewis wise isel sely . After deriving the 4-gram sequence, the next stage is to hash the derived sequence into a set of numeric values. Assume the following hash values represent sequence of the 4-grams sequence above: 65 32 55 64 38 71 54 34 65 24 53 21 14 45 63 23. Once hash values are obtained, the next stage is to select subset of the available hash values that could be used to represent the document. Note that the selected hash values derived are referred to as document fingerprints or signature. In this example we use 0 mod 5 as the criterion for 'shortlisting' the various 4-grams values derived, applying the criterion limits the hash values to 65 55 65 45. Thefour derived values is referred to as the documentfingerprint or signature. For a very effective similarity comparison, the entire hash values would have been ideal to use, but unfortunately, using the entire hash values would be very expensive in terms of bandwidth size, processing time among others. For efficiency purposes, only a subset of the hashes should be retained as the document's

fingerprint. A very common practice as shown above, is to select subset of available hashes that are 0 mod m, for a fixed number of m. The resultant fingerprint derived from this method is a good measure in representing the originated document and could be used to compare fingerprints from other documents as shown in the above example. One apparent disadvantage of this method is the fact that it is possible that for an entire sequence of hash values, very limited or sometimes none available hash values would satisfy the 0 mod m condition, and thus no fingerprint would be obtained for comparison with other documents. Consider the following sequence; 62 32 51 64 38 71 54 34 61 24 53 21 14 43 63 23, supposing the same 0 mod 5 used in our previous example is applied; it would mean none of the hash sequence qualifies as a fingerprint and hence there would not be any bases for comparison with the procedure aforementioned. Secondly Schleimer et al. [20] observed that, there are situations where real data does not generate adequately random sequences of hashes. In particular, there are clusters of low-entropy strings on the Web, such as 1111111111111111, or more complex patterns such as cddccddccddccddccddc… which could generate a very large fingerprints sequence or no fingerprints at all.

In this paper, we present an efficient algorithm for selecting the fingerprints from a sequence of hashes that guarantees that at least enough hashes would be selected which has the tendency to represent the document as its fingerprint and can be used for subsequent resource-similarity comparison.

## 2. What constitutes similarity?

According to Bernstein et al. [3] circular and substandard definitions of what constitutes a duplicate has weakened several previous works in the area of identifying duplicate documents in an information retrieval system. Detecting duplicate or similarity between text resources is not explicit, as in the case of detection of duplicate attributes of records between different tables in a DBMS. In a more general perspective, duplicate in text resources can be defined as the exact syntactic terms, baring formatting differences between candidates. According to Broder [4], duplicate could be seen as a measure of resemblance between documents: this measure of resemblance is determined by a predetermined threshold. For the purposes of this paper, we define duplicate between two text resources as a numeric quantity that are similar above a certain specified threshold. We are aware of a potential difficulty of implementing this algorithm due to the fact that a resource could be intertwine with other resources making the computation of duplicate detection a daunting task. But we assume a number of isolated resources for this experiment. In a distributed information retrieval environment, due to the fact that enormous resources are available, various complicated scenarios are possible. It is possible that Resource I is similar to resource II, which is similar to resource III, but Resource I and resource III have no similarity. There could be another situation where resource I and resource II are similar, resource II similar to resource III and resource I similar to resource III, but the composite resources I, II, and III share no common similarity. Or perhaps Resources I, II and III can share a similarity ratio of a little below 50% which could be classified as significant. Again Resource I and II could share a similarity ratio of 45%, resources II and III could share 61% and so on.

IIn this paper, we consider similarity between resource-pairs only. This is due to the varying complexity described above. We also assume that, classification of resource-pairs is based on a similarity threshold. According toFinkelet al.[7], a similarity analysis between ColI and ColII that is based on hash values of RECSI designated as h(HECSI) and that of RECSII designated by h(RECSII) measuring the portion of the fingerprint intersection, with respective cardinalities |h(RECSI) | and |h(RECSII)|, will yield

$$\text{local}(\text{RECSI}, \text{RECSII}) = \frac{|hh(\text{RECSI}) \cap h(\text{RECSII})|}{|h((\text{RECSI}) \cup h(\text{RECSII})|} \quad (1)$$

Where local(RECSI,RECSII) is the measure of similarity. The above similarity measure is termed local similarity or overlap similarity. This is because it directly reflects the quantum of intersection between the two resources.

## III. THE DelCosim Method

We first convert a resource to its index and subsequently stream the document again to do away with morphological connotations that found its' way back to the index resource. During indexing whitespaces, punctuation, and extraneous features are removed from the resource. After the process of indexing, specific words that are repetitive are deemed relevant to be used to represent the resources. With our new index, we hash each index term to obtain a sequence of numeric values. The intuition is, a resource on a particular subject would have a number of repetitive set of words or terms that would run through the entire resource. For example, a resource on information retrieval could be assumed to have words such as information, retrieval, information retrieval, index etc. appearing almost frequently throughout the resource. Based on this assumption, it would be worthwhile to use such repetitive words to represent the resource. After hashing, we select hash values whose counts n (h), exceeds a particular threshold j, ie. (n(h)>j). These sets of hash values were used as fingerprint to represent the resource. With this procedure, each resource would have a unique signature or fingerprint that could be used as bases in detecting rate of similarity or duplicates between other resources with similar fingerprints.

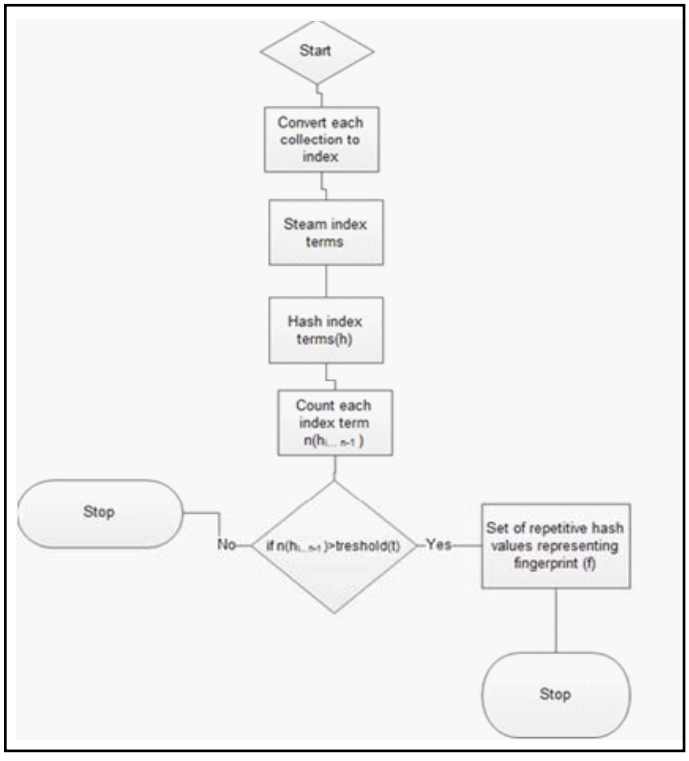


Fig. 1: Flowchat diagram for computing fingerprint value

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Onceastoppingcriterionforthe numberofhashvaluesrepresenting .Theideaisto computethesimilaritybetweentheresources. Givennvectorsof attributes,Ai..n,,Bi..n,…. Nn..ithecosinesimilarity,cos(θ),is representedusingadotproductandmagnitude.AtypicalCosine Similarity of two vectors (v1 and v2) is computed as follows:

$$\cos(v1, v2) = \text{dot}(v1,v2) / \|v1\| \|v2\| \quad (2)$$

Where $\text{dot}(v1,v2) = v1[0]*v2[0] + v1[1]*v2[1] \ldots$ (3)

AndWhere $\|v1\| = \text{sqrt}(d1[0]\text{^}2 + d1[1]\text{^}2 \ldots)$ (4)

The resulting similarity ranges from −1 meaning exactly opposite; 1 meaning exactly the same, and 0 indicating independence, and in-between values indicating intermediate similarity or dissimilarity.

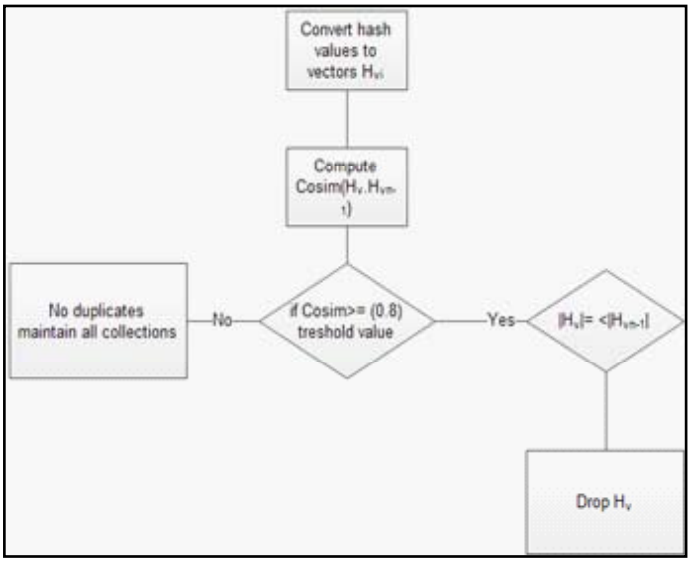


Fig. 2: Flowchat diagram for DelCosim

## IV. Experimental Setup

### Dataset

Our experiment was conducted on two familiar testbeds; first, Trec123-100col-bysource: 100 databases which was created from TREC CDs 1, 2 and 3. They are organized by source and publication date. This testbed has been extensively used in prior research [1-3] . Second, Trec4-kmeans: 100 databases were created from TREC 4 data. A k-means clustering algorithm was used to organize the databases by topic[3], so the databases are similar and the word distributions are very tilted. The characteristics of these two testbeds are shown in Table 1. For the purposes of our experiment, we intentionally deleted 10 random databases from the 100 databases created in the first testbed (Trec123-100col-bysource), and replaced them with 10 random copied databases from the remaining 90 databases left (this is done to check for full overlap). We also deleted some contents (documents) of 10 of the remaining 90 databases and replaced them with documents from other databases (this is done to check for partial overlap). This process eventually enable us to arrive at the 100 database stated above. The same process is applied to the second testbed (Trec4-kmeans). In summary we have 10 full-duplicate databases in each testbed as well as 10 partial-duplicate databases in each collection.

Table1. :Summary statistics for the Trec123-100 col by source and Trec4-k means test beds.

| Name | Query Count | Size (GB) | NumberofDocs | | | Megabytes(MB) | | |
|---|---|---|---|---|---|---|---|---|
| | | | Min | Avg | Max | Min | Avg | Max |
| Trec123 | 50 | 3.2 | 75 | 10782 | 39713 | 28.1 | 32 | 41.8 |
| Trec4 kmean | 50 | 2.0 | 30 | 5675 | 82727 | 3.9 | 20 | 248.6 |

### Query

We used 50 short queries created from the title fields of TREC topics 51-100 for the Trec123-100bycol, relevant and nonrelevant testbeds. We also created 50 longer queries created from the description fields of TREC topics 201-250 for the Trec4-kmeans testbed.

## V. Discussions

As expected, DelCosim performs very well on random trials. For example, the testbeds Trec123-100col-bysource and Trec4-kmeans had substitutions (with duplication), thus changing about 20% of the file (but leaving significant portions unchanged). We then ran DelCosim (in the one-against- all mode) setting the threshold at a very low 6%. We ran this experiment 60 times. Each time DelCosim found the right file (100 database) . The similarity that DelCosim reported ranged from 41% to 62%, averaging 52%. The average running time for one test (user + system time) was 3.1 seconds (not counting, of course, the time it took originally to build the index. The running time for computing all fingerprints was 3-6 seconds per database. It takes longer if there are many documents in the collection. The sorting of the fingerprints takes from a third to a half of this time (sorting is not a linear-time algorithm, so it takes longer for large number of fingerprints.

We again used a state-of-the art resource selection algorithm, CORI, the results showed that, CORI selected both full and partial databases which have already been selected. This shows that, in environments such as the web, where duplication is inevitable, conscious effort is needed to remove duplicate and near duplicate collections in order to reduce the tendency of 'selecting' two servers that would invariable produce the same documents. Again when duplication is prevented at the resource selection stage, it reduces the cost of preventing seemly relevant databases who were unable to be selected due to the selection of duplicate pairs.

.

## VI. Conclusion

This paper addressed the issue of resource selection for information retrieval in an environment composed of overlapping resources. We presented a method called DelCosim which adapts a state-of-the-art relevance based information retrieval method, cosine, to consider resource overlap. We presented a systematic evaluation of the effectiveness of DelCosim over existing methods. Our experiments showed that DelCosim outperforms current best methods when there are overlappingresources with a very small increase in processing cost.

## References

*[1]. Allan, J., V. Lavrenko, and H. Jin. First story detection in TDT is hard. in Proceedings of the ninth international conference on Information and knowledge management. 2000. ACM.*

*[2]. Bender, M., S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer. Improving resource selection with overlap awareness in p2p search engines. in Proceedings of the 28th annual international ACM SIGIR conference on Research*

*and development in information retrieval. 2005. ACM.*

*[3]. Bernstein, Y., M. Shokouhi, and J. Zobel. Compact features for detection of near-duplicates in distributed retrieval. in String Processing and Information Retrieval. 2006. Springer.*

*[4]. Broder, A.Z. Filtering near-duplicate documents. in Proc. FUN 98. 1998. Citeseer.*

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

*[6]. Callan, J.P., Z. Lu, and W.B. Croft. Searching distributed resources with inference networks. in Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval. 1995. ACM.*

*[7]. Finkel, R.A., A. Zaslavsky, K. Monostori, and H. Schmidt. Signature extraction for overlap detection in documents. in Australian Computer Science Communications. 2002. Australian Computer Society, Inc.*

*[8]. Fullam, K. and J. Park. Improvements for scalable and accurate plagiarism detection in digital documents. in Proceedings of the 8th International Conference on Parallel and Distributed Systems. 2002.*

*[9]. Gravano, L., C.-C.K. Chang, H. García-Molina, and A. Paepcke, STARTS: Stanford proposal for Internet meta-searching. Vol. 26. 1997: ACM.*

*[10]. Haveliwala, T.H. Topic-sensitive pagerank. in Proceedings of the 11th international conference on World Wide Web. 2002. ACM.*

*[11]. Hernandez, T. and S. Kambhampati. Improving text resource selection with coverage and overlap statistics. in Special interest tracks and posters of the 14th international conference on World Wide Web. 2005. ACM.*

*[12]. Huberman, B. and R. Lukose, A metasearch engine that learns which search engine to query. Science, 1997. 277: p. 535-537.*

*[13]. Ipeirotis, P.G. and L. Gravano. Distributed search over the hidden web: Hierarchical database sampling and selection. in Proceedings of the 28th international conference on Very Large Data Bases. 2002. VLDB Endowment.*

*[14]. Meng, W., C. Yu, and K.-L. Liu, Building efficient and effective metasearch engines. ACM Computing Surveys (CSUR), 2002. 34(1): p. 48-89.*

*[15]. Michel, S., M. Bender, P. Triantafillou, and G. Weikum, Iqn routing: Integrating quality and novelty in p2p querying and ranking, in Advances in Database Technology-EDBT 2006. 2006, Springer. p. 149-166.*

*[16]. Nie, Z., S. Kambhampati, and U. Nambiar, Effectively mining and using coverage and overlap statistics for data integration. Knowledge and Data Engineering, IEEE Transactions on, 2005. 17(5): p. 638-651.*

*[17]. Nottelmann, H. and N. Fuhr, Decision-theoretic resource selection for different data types in MIND, in Distributed Multimedia Information Retrieval. 2004, Springer. p. 43-57.*

*[18]. Powell, A.L. and J.C. French, Comparing the performance of resource selection algorithms. ACM Transactions on Information Systems (TOIS), 2003. 21(4): p. 412-456.*

*[19]. Saleem, M., A.-C.N. Ngomo, J.X. Parreira, H.F. Deus, and M. Hauswirth, Daw: Duplicate-aware federated query processing over the web of data, in The Semantic Web–ISWC 2013. 2013, Springer. p. 574-590.*

*[20]. Schleimer, S., D.S. Wilkerson, and A. Aiken. Winnowing: local algorithms for document fingerprinting. in Proceedings of the 2003 ACM SIGMOD international conference on Management of data. 2003. ACM.*

*[21]. Selberg, E. and O. Etzioni, The MetaCrawler architecture for resource aggregation on the Web. IEEE expert, 1997. 12(1): p. 11-14.*

*[22]. Shokouhi, M. and J. Zobel, Robust result merging using sample-based score estimates. ACM Transactions on Information Systems (TOIS), 2009. 27(3): p. 14.*

*[23]. Si, L. and J. Callan. 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. 2003. ACM.*

*[24]. Voorhees, E., N.K. Gupta, and B. Johnson-Laird, The resource fusion problem. NIST SPECIAL PUBLICATION SP, 1995: p. 95-95.*

*[25]. Wu, S., Fusing Results from Overlapping Databases, in Data Fusion in Information Retrieval. 2012, Springer. p. 149-180.*

*[26]. Wu, Z., W. Meng, C. Yu, and Z. Li. Towards a highly-scalable and effective metasearch engine. in Proceedings of the 10th international conference on World Wide Web. 2001. ACM.*

*[27]. Xiao, C., W. Wang, X. Lin, J.X. Yu, and G. Wang, Efficient similarity joins for near-duplicate detection. ACM Transactions on Database Systems (TODS), 2011. 36(3): p. 15.*

*[28]. Yuwono, B. and D.L. Lee. Server Ranking for Distributed Text Retrieval Systems on the Internet. in DASFAA. 1997.*

*[29]. Zamir, O. and O. Etzioni, Grouper: a dynamic clustering interface to Web search results. Computer Networks, 1999. 31(11): p. 1361-1374.*