Evaluation of a Recursive Weighting Scheme for Federated Web Search

Emanuele Di Buccio, Ivano Masiero, and Massimo Melucci

Department of Information Engineering, University of Padua, Italy
{dibuccio,masieroi,melo}@dei.unipd.it

Abstract. The informative resources available on the Web are not always directly accessible and cannot therefore be crawled since access is permitted only through the adoption of appropriate services, e.g. specialized search engines. On the other hand, specialized search engines can help address the problem of heterogeneity of the informative resources due to the type of content, the structure or the media. Federated Web Search systems address the problem of searching multiple, heterogeneous, and possibly uncooperative collections. One issue of Federated Web Search is resource selection, i.e. the selection of the search engines which most likely provide documents relevant to the query. This paper reports on the experimental evaluation in Federated Web Search setting of a recursive weighting scheme for ranking informative resources in architectures that involve an arbitrary number of resource levels.

1 Introduction

The informative resources available on the Web are not always directly accessible, and therefore cannot be crawled, because access is permitted only through the adoption of appropriate services, e.g. specialized search engines. Access to informative resources is only one of the issues that motivate the adoption of diverse search engines. For instance, the adoption of specialized search engines can help the problem of heterogeneity among the informative resources to be addressed. Informative resource, hereafter denoted as documents, can be heterogeneous because of the type of content (e.g. medical documents), the structure (e.g. patents), or the media (e.g. video and image).

Federated Web Search [12] concerns with the problem of searching multiple, heterogeneous, and possibly uncooperative collections. In Federated Web Search setting, the broker is a system that should select the most promising search engines to which the query should be forwarded on the basis of a description of the diverse collections handled by the engines; the selected engines will return the most promising results, that will be then merged by the broker in a final ranked list. Forms of federated search are *vertical search* or *peer-to-peer search*. In the first case, the objective is to select the most promising verticals, e.g. web pages, images or videos, and then merge the results from those verticals in a unique result page. In the second case, retrieval is performed in a Peer-To-Peer (P2P) networks, where each participating node can act both as client and server —

in an Information Retrieval (IR) perspective it can both submit a query to the other participating nodes and act as a search server, providing the most relevant documents in its local collection in response to a given query. In P2P search, besides documents in the peer collections, also peers should be ranked in order to select the most promising peers to be contacted, thus avoiding flooding that can be unfeasible for large networks.

This paper is focused on the problem of *resource selection*, i.e. on ranking resources at higher levels, e.g. search engines, verticals or peers. The problem is addressed by the adoption of Term Weighted Frequency (TWF)·Inverse Resource Frequency (IRF). This weighting framework was originally introduced in [3] to address the problem of resource selection in Hybrid Hierarchical P2P Networks. In this kind of networks there are two types of nodes, i.e. peers and super-peers. Yet, a peer has to update and transfer the data structures which summarizes its own document collection to the super-peers. A query is sent from a peer to the super-peers and then it is routed from a super-peer to the other super-peers on the basis of the summaries stored in each super-peer. While all the peers are involved when routing the query in an unstructured network, only the super-peers are involved in routing in a hybrid unstructured network. When routing the query a super-peer ranks both the other super-peers and the peers by expected recall.

TWF·IRF addresses the problem of informative resource ranking in architectures with an arbitrary number of resource levels. This paper reports on the experimental investigation of the effectiveness of this scheme in Federated Web Search settings. The evaluation was carried out through the participation to the Federated Web Search Track of the Twenty-Second Text REtrieval Conference (TREC) (FedWeb13). In the FedWeb13 setting there are three resource levels: (i) document, (ii) search engines, and (iii) set of search engines. In particular, there is a single set of 157 search engines and the objective of the resource selection task is to rank them according to (their predictive capability for) a given query. Even if search engines use features that in peer-to-peer settings could not be available,[1] the participation to the track has given insights on the TWF·IRF effectiveness in ranking peers in a group when considering a completely uncooperative environment. Indeed, a broker of FedWeb13 performs the same task as a super-peer does for forwarding queries to the most effective peers in its group. Moreover, the summaries stored in the broker are the results of query-based sampling performed on the considered search engines since the index of the distinct search engines cannot be accessed — for this reason they are considered "uncooperative".

[1] We consider the case where each participating peer provides search functionalities to access their local collection, e.g. part of the personal documents of a user.

2 A Recursive Weighting Scheme

According to the literature in Distributed IR reported for example in [2, 4], the specific approach adopted in this work is to describe the informative resources at the diverse levels (document, search engines) in terms of document descriptors, e.g. terms. Therefore, a search engine is described as a set of document descriptors, specifically the distinct descriptors appearing in the documents stored in it.

The innovative contribution of our approach has been the computation of the weights. In order to support the description of the TWF·IRF, let us consider an architecture with three resource levels, e.g. that depicted in Figure 1. Examples of resource levels are (1) documents, (2) peers, and (3) super-peers in Hybrid P2P networks or (1) documents, (2) search engines and (3) sets of search engines in Federated Web Search setting. For instance, the search engines adopted in the FedWeb13 test collection have been categorized by the track organizers according to a set of categories that include news, books, academic, travel.

In our approach the weight of a descriptor t in a resource i at level z is

$$w_{i,t}^{(z)} = twf_{i,t}^{(z)} \cdot irf_t^{(z)}, \tag{1}$$

where

$$twf_{i,t}^{(z)} = \sum_{r \in R_i^z} twf_{i,t}^{(z-1)} \cdot irf_t^{(z-1)} \tag{2}$$

and R_i^z denotes the sets of resources in the ith resource at level z. For instance, in Figure 1, when $z = 2$ the r's are search engines and the R_i^3's are sets of search engines; in Equation 2 the sum for the resource $R_i^3 = se_3$ is computed over the engines e_8 and e_9. For a given query q, resources at level z can be ranked according to $\sum_{t \in q} w_{i,t}^{(z)}$. Equation 1 shows how the weight of a descriptor t in a resource is the product of two components: TWF and IRF. The TWF is peculiar of this scheme and its definition is recursive since it relies on the TWF of the resources at lower levels — e.g. the TWF of a set of search engines is computed as the weighted sum of the TWF of the search engines in the considered set.

The Inverse Resource Frequency (IRF) is a generalization of the Inverse Document Frequency (IDF), that is,

$$irf_t^{(z)} = \log N^{(z)} / n_t^{(z)} \tag{3}$$

where t denotes the term, $N^{(z)}$ is the number of resources at level z contained by the resource at level $z + 1$ and $n_t^{(z)}$ is the number of those resources that are indexed by t. For instance, the search engine e_2 in Figure 1 is contained in the set se_1 and, for that set, $N^{(2)} = 4$. A generalization of the IDF was proposed in [2] to rank collections (Inverse Collection Frequency, ICF) and another was proposed in [4] to rank peers (Inverse Peer Frequency, IPF). ICF and IPF are instances of the IRF weight at level 2. In the FedWeb13 informative resources at level 2 are search engines.

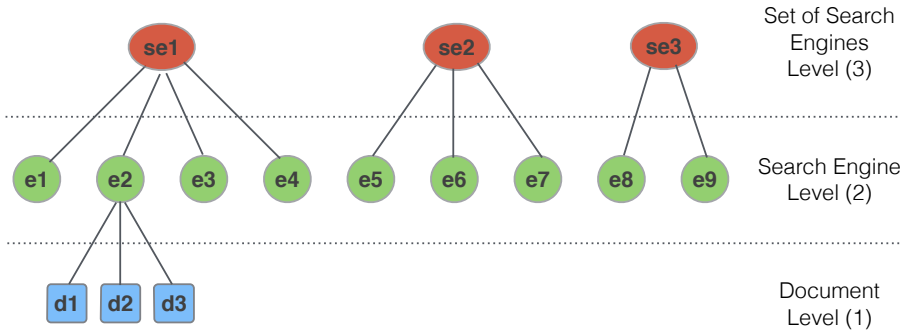


Fig. 1. Architecture with three resource levels.

Figure 2 reports an example of the computation of TWF in an architecture that involves four search engines. The first matrix on the left reports the index terms, the IDFs and the Term Frequencys (TFs) in the documents indexed by the search engine e_2. The TWF of a term t in e_2 can be computed by accessing the associated posting list and computing the sum of the

$$twf^{(1)}_{e_2,t} \cdot irf^{(1)}_{t}$$

over the documents in that posting list; in the considered example,

$$twf^{(1)}_{e_2,t} = tf(t, e_2)$$

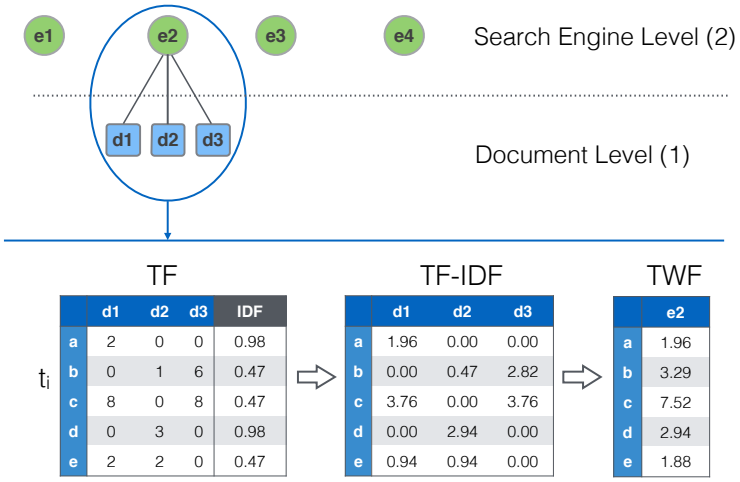
but normalized values of TF can be adopted.

Figure 3 illustrates how search engine ranking can be performed on the basis of the TWFs of the diverse engines. The matrix on the right basically reports information stored in the broker index; this is a search engine level index that contains the TWFs and IRFs for all the terms in the search engine indexes (or a subset of them in uncooperative environments). The score assigned to a search engine e_i is the sum of the TWF·IRF scores computed over the query terms. In the reported example, the final ranking will be: e_2, e_4, e_1, e_3. Some remarks on how this information can be stored in an inverted index and the actual implementation adopted in the experiments are reported in Section 3.3.

3 Experiments

3.1 Research Task and Questions

The experiments reported in this paper consider the following task. Given a set S of search engines, a set $Q_{\mathcal{T}}$ of queries and a set of sample documents obtained by query-based sampling performed on each of the search engines, a Federated Web Search system should return a ranked list of search engines for each query

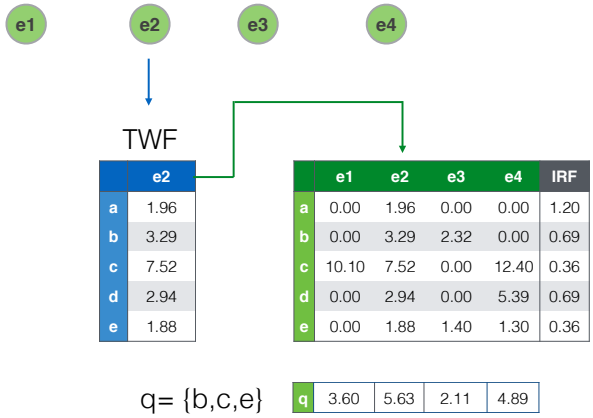


	d1	d2	d3	IDF
a	2	0	0	0.98
b	0	1	6	0.47
c	8	0	8	0.47
d	0	3	0	0.98
e	2	2	0	0.47

	d1	d2	d3
a	1.96	0.00	0.00
b	0.00	0.47	2.82
c	3.76	0.00	3.76
d	0.00	2.94	0.00
e	0.94	0.94	0.00

	e2
a	1.96
b	3.29
c	7.52
d	2.94
e	1.88

Fig. 2. Computation of the TWF for a search engine.



	e2
a	1.96
b	3.29
c	7.52
d	2.94
e	1.88

	e1	e2	e3	e4	IRF
a	0.00	1.96	0.00	0.00	1.20
b	0.00	3.29	2.32	0.00	0.69
c	10.10	7.52	0.00	12.40	0.36
d	0.00	2.94	0.00	5.39	0.69
e	0.00	1.88	1.40	1.30	0.36

q	3.60	5.63	2.11	4.89

Fig. 3. Search engine ranking by TWF·IRF.

in $Q_{\mathcal{T}}$ ranked by a measure of the capability of satisfying the user's information need expressed by the query.

The objective of our work is to investigate the effectiveness of TWF·IRF for search engine selection in Federated Web Search setting. In particular, we have investigated the following research questions:

– Is TWF·IRF effective when adopted to rank the most promising search engines at high rank positions?
– Which is the effect of IRF when TWF·IRF is adopted for search engine ranking? Is TWF, which is peculiar to this scheme, "sufficient"?

The results reported in this paper are representative of the results obtained in the TREC 2013 Federated Web Search Track. We compared the effectiveness of the search engine ranking based only on TWF with the effectiveness of the original TWF·IRF. We report the comparison with one well known retrieval model for distributed collection selection, i.e. bGlOSS [6], the boolean version of the Glossary of Servers Server (GlOSS). bGlOSS ranks the collection according to the estimated number of documents that satisfy the query q:

$$\text{ESize}_{Ind}(q, e_i) = |e_i| \prod_{t \in q} \frac{df(t, e_i)}{|e_i|} \tag{4}$$

where $df(t, e_i)$ denotes the number of documents in the collection e_i – in our setting e_i is a search engine – that is indexed with the term t; $|e_i|$ denotes the number of documents in the ith collection.

3.2 Test Collection and Effectiveness Measures

The research questions described in Section 3.1 were addressed using the FedWeb13 test collection. This collection is constituted of a list of 157 search engines[2] and a set of sample search results obtained by performing query-based sampling on those search engines. A set $Q_{\mathcal{S}}$ of 2000 queries was adopted to perform the sampling. For each search engine and for each query in the given query set, the top 10 results were retrieved – both snippets and landing documents. Half of the queries in $Q_{\mathcal{S}}$ was obtained using the *Zips method*, which exploits "single term queries taken evenly from the binned term distribution in ClueWeb09, where terms were binned on a log-scale of their document frequency (df) to ensure that there are queries from the complete frequency distribution." [10]. The other half of the queries was built by randomly selecting terms from the sample documents collected from the search engine.

A set of 200 queries, $Q_{\mathcal{T}}$, were provided by the track organizers to address the two research tasks described in Section 3.1.

The evaluation for the two tasks was performed on a subset of 50 queries among those in $Q_{\mathcal{T}}$. The primary effectiveness measure adopted for the *resource selection* task was NDCG@20. The Normalized Discounted Cumulative Gain (NDCG) [7] version adopted in the experiments is that proposed in [1]. The relevance of a search engine was computed by using the graded precision [8] on the top 10.[3]

[2] The list of search engines is available at the following url:
`http://snipdex.org/datasets/fedweb2013/FW13-engines.txt`
[3] Details are provided in the FedWeb13 Track web page:
`https://sites.google.com/site/trecfedweb/`

3.3 Parsing and Indexing

The indexing module of our system relies on an XML parser written in Java for extracting the document fields from the sample searches and the sample documents in the test collection, and on the Apache Lucene library. The sample documents in the FedWeb13 Test Collection were indexed by creating a distinct index for each of the 157 search engines. These indexes are *document-level* indexes. Each (Lucene) document in a document-level index is constituted of four fields: link, title, description, and the content of the document associated to the sample search result. For each field, the document-level index stores information on the frequency of the descriptors in each document and in the collection, as well as their positions in each document.

Starting from these indexes, a search engine-level index was built. The set of descriptors in this index is the union of all the distinct descriptors in the distinct document-level indexes associated to the search engines. As for the document-level index, in the search engine-level index a list of posting is associated to each descriptor. Each posting stores information on the identifier of the search engine, the number of documents in the search engine were the descriptor appears, and the TWF of the descriptor. In the specific Lucene-based implementation adopted, TWF was stored in the payload that can be associated to each term; the weight value was approximated and stored as a float.[4]

3.4 Resource Selection

The experiments exploit two specific instantiations of the weighting scheme described in Section 2. The first instantiation exploits only TWF for search engine ranking:

$$\sum_{t \in q} twf_{i,t}^{(2)} \tag{5}$$

where $twf_{i,t}^{(2)} = \sum_{d_j \in D_i} \; twf_{i,t}^{(1)} \cdot irf_t^{(1)}$ and D_i denotes the sets of documents in the ith search engine, $twf_{j,t}^{(1)} = tf(t,j)$ is the term frequency of term t in the document d_j. The IRF at the document level was implemented as:

$$irf_t^{(1)} = \log\left(1 + \frac{N^{(1)} - n_t^{(1)} + 0.5}{n_t^{(1)} + 0.5}\right) \tag{6}$$

The second instantiation exploits both TWF and IRF; search engines are ranked according to:

$$\sum_{t \in q} twf_{i,t}^{(2)} \; irf_t^{(2)} \tag{7}$$

where $twf_{i,t}^{(2)}$ is computed as above and $irf_t^{(2)}$ is computed as

$$irf_t^{(2)} = \log\left(1 + \frac{N^{(2)} - n_t^{(2)} + 0.5}{n_t^{(2)} + 0.5}\right) \tag{8}$$

[4] Single-precision 32-bit IEEE 754 floating point

where $N^{(2)}$ is the number of search engines – in this test collection $N^{(2)} = 157$ – and $n_t^{(2)}$ is the number of those search engines that are indexed by t.

For each instantiation, we considered two runs, the label of which ends with `sh` and `mu`. In the `sh` runs, the query is built by performing an OR among the terms appearing in the query.[5] The ranked list of search engines that constitute the `mu` runs are obtained by appending three ranked lists:

L_1: the list of search engines ranked by their TWFs with regard to the query, and using the AND boolean constraint among the occurrence of the distinct terms in the query[6];

L_2: the list of search engines that did not belong to L_1 and ranked by their TWFs with regard to the query by using the OR boolean constraint among the occurrence of the distinct terms in the query;

L_3: the list of search engines that did not belong to L_1 and L_2, ranked by their identifier — the identifier associated to the search engine in the test collection.

The final ranked list of search engines was obtained by appending L_2 to L_1, and then L_3 to the fusion of the first two lists.

3.5 Results

Results are reported in Figure 4. With regard to the first research question, TWF·IRF in the two instantiations outperforms bGlOSS for all runs. Moreover, the `mu` runs perform better than the `sh` runs for both `sh` and `mu`. The only drawback of the `mu` runs is that two queries should be performed – one performing the AND and one performing the OR among the query terms. However, the number of resources to rank is much lower than the number of documents in a collection – e.g. hundreds of search engines versus billion of documents in web search setting; therefore the additional computational load of the `mu` runs can be acceptable.

With regard to the second research question, IRF provides an improvement in terms of NDCG@20. An 7.37 % increment is observed for the `sh` runs, while the increment, 1.73 %, is negligible for the `mu` runs – the difference is not significant. Therefore, for this test collection, when it is wanted that all the query terms must occur in the documents, TWF seems to be "sufficient" for search engine ranking. In contrast, when it is accepted that only some of the query terms occur in the documents, IRF, i.e. both TWF and IRF appear "necessary" for search engine ranking.

[5] The Lucene query was a BooleanQuery constituted of PayloadTermQuery connected by SHOULD clause.

[6] The Lucene query was a BooleanQuery constituted of PayloadTermQuery connected by MUST clause.

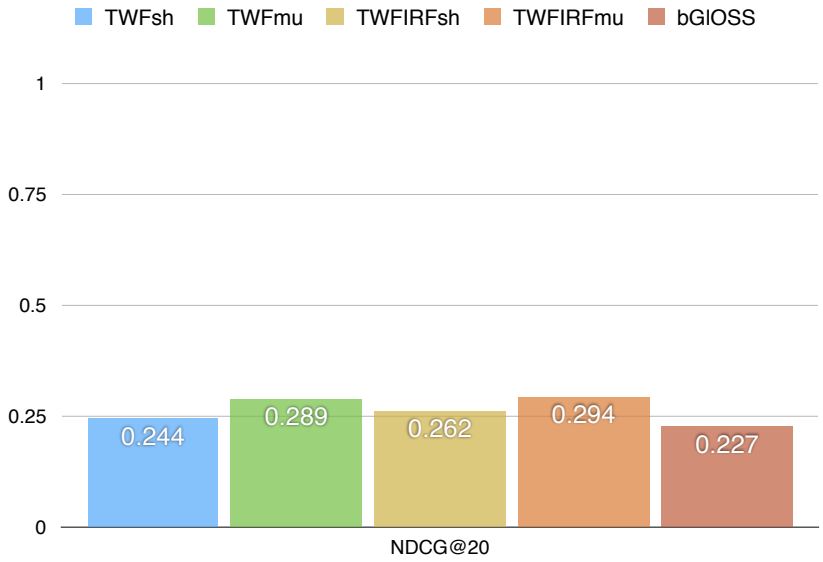


Fig. 4. Results.

4 Final Remarks

This paper reports on the investigation of TWF·IRF in Federated Web Search setting. We participated in the TREC 2013 Federated Web Search Track. This weighting scheme was shown to be effective to address the problem of loss in recall in Hierarchical Hybrid P2P Networks [9]. The results reported in this paper show that this weighting scheme can also support Federated Web Search.

Future works will be focused on further experimental investigations, particularly:

- the comparison with the most effective retrieval models, e.g. Collection Retrieval Inference Network (CORI) [2], Decision Theoretic Framework (DTF) [5], Relevant Document Distribution Estimation (ReDDE) [13], and Central-Rank-Based Collection Selection (CRCS) [11];
- the effect of different resource descriptions, e.g. based on result snippet or combination of snippet and document content – snippets are available in the FedWeb13 test collection, or the adoption of external resources, e.g. to perform "resource description expansion";
- the effect of the sampling strategy on resource selection effectiveness;
- the effect of IRF in diverse test collections.

References

1. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 89–96, New York, NY, USA, 2005. ACM.
2. J. P. Callan, Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '95, pages 21–28, New York, NY, USA, 1995. ACM.
3. R. Castiglion and M. Melucci. An evaluation of a recursive weighing scheme for information retrieval in peer-to-peer networks. In *Proceedings of the 2005 ACM workshop on Information retrieval in peer-to-peer networks*, P2PIR '05, pages 9–16, New York, NY, USA, 2005. ACM.
4. F. M. Cuenca-Acuna and T. Nguyen. Text-Based Content Search and Retrieval in Ad-hoc P2P Communities. In E. Gregori, L. Cherkasova, G. Cugola, F. Panzieri, and G. Picco, editors, *Web Engineering and Peer-to-Peer Computing*, volume 2376 of *Lecture Notes in Computer Science*, pages 220–234. Springer Berlin Heidelberg, 2002.
5. N. Fuhr. A decision-theoretic approach to database selection in networked ir. *ACM Transactions on Information Systems*, 17(3):229–249, July 1999.
6. L. Gravano, H. García-Molina, and A. Tomasic. The effectiveness of gioss for the text database discovery problem. *ACM SIGMOD Record*, 23(2):126–137, May 1994.
7. K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, October 2002.
8. J. Kekäläinen and K. Järvelin. Using graded relevance assessments in ir evaluation. *Journal of the American Society for Information Science and Technology*, 53(13):1120–1129, 2002.
9. M. Melucci and A. Poggiani. A Study of a Weighting Scheme for Information Retrieval in Hierarchical Peer-to-peer Networks. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, pages 136–147, Berlin, Heidelberg, 2007. Springer-Verlag.
10. D. Nguyen, T. Demeester, D. Trieschnigg, and D. Hiemstra. 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*, CIKM '12, pages 1874–1878, New York, NY, USA, 2012. ACM.
11. M. Shokouhi. Central-rank-based collection selection in uncooperative distributed information retrieval. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, pages 160–172, Berlin, Heidelberg, 2007. Springer-Verlag.
12. M. Shokouhi and L. Si. Federated search. *Foundations and Trends in Information Retrieval*, 5(1):1–102, Jan. 2011.
13. L. Si 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*, SIGIR '03, pages 298–305, New York, NY, USA, 2003. ACM.