

Database Selection and Result Merging in P2P Web Search

Sergey Chernov¹, Pavel Serdyukov², Matthias Bender³, Sebastian Michel³,
Gerhard Weikum³, and Christian Zimmer³

¹ L3S Research Center, University of Hannover, Expo Plaza 1, 30539, Hannover, Germany, chernov@l3s.de

² Database Group, University of Twente, PO Box 217, 7500 AE Enschede, Netherlands, serdyukovpv@cs.utwente.nl

³ Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123, Saarbrücken, Germany, {mbender, smichel, weikum, czimmer}@mpi-inf.mpg.de

Abstract. Intelligent Web search engines are extremely popular now. Currently, only commercial centralized search engines like Google can process terabytes of Web data. Alternative search engines fulfilling collaborative Web search on a voluntary basis are usually based on a blooming Peer-to-Peer (P2P) technology. In this paper, we investigate the effectiveness of different database selection and result merging methods in the scope of P2P Web search engine Minerva. We adapt existing measures for database selection and results merging, all directly derived from popular document ranking measures, to address the specific issues of P2P Web search. We propose a general approach to both tasks based on the combination of pseudo-relevance feedback methods. From experiments with TREC Web data, we observe that pseudo-relevance feedback improves quality of distributed information retrieval.

1 Introduction

Current Web search technologies encounter a number of serious obstacles. The first one is the size of indexable Web, which can be hardly covered entirely enough due to limited network bandwidth and finite computational resources of search engines. Consequently, pages are only periodically updated and the outcome of most search engines is out of date. Another issue is a “Deep Web” problem, when search engines cannot get access to the information stored by commercial information providers or to the resources not linked by any page. There is also the noticeable social perspective. The most powerful player ever, Google, monopolizes Web search market. It controls a major part of Web search requests and can establish its own censorship. We think that a search engine having resort to P2P technology could be able to overcome the described limitations. Collaborative crawling can span a larger portion of the Web, if each peer would contribute its own focused crawl into the system. In addition, we disclose various opportunities of topic-oriented search by using intellectual inputs from users. These considerations induced us to launch the Minerva project [2], a P2P Web search engine.

In Minerva, each peer has its own collection of crawled or personal documents. For efficient query routing, all peers collectively maintain a global directory, which contains peer-summary information about which peer has documents for which index terms. This information is organized in peer lists constructed for each term occurring in the system. For example, each peer list contains df (document frequency) and a sum of tf (term frequency) for respective term at each peer plus total number of documents (with sum of their lengths) stored at this peer. To make these peer lists accessible to any peer, to share network load and to secure this information from loss, Minerva disseminates all peer lists using Chord distributed hash table (DHT) protocol [22]. It hashes terms and peer network addresses to know which peer is responsible for managing which peer list.

At many points, Minerva resembles a large-scale highly dynamic metasearch engine. We can leverage and adopt existing solutions from metasearch field, see Fig. 1 [8]. A query q is posed on the set of peers P . A database selection problem occurs when we select a subset of peers P' that most probably contain relevant documents. Then system sends q to every peer in P' and obtains a set of document rankings R' from the local search engines. A result merging problem occurs when all rankings in R' are merged into one ranking R_m and the top- k results from it are presented to the user.

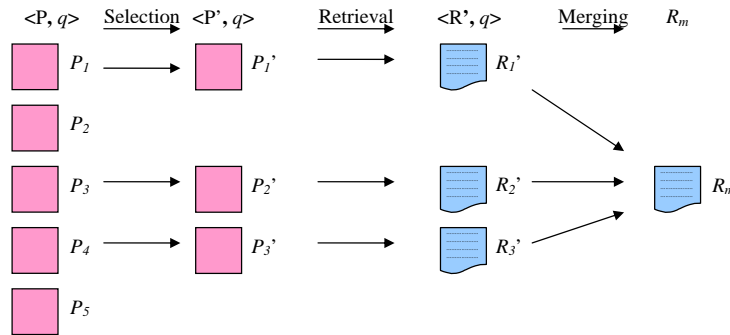


Fig. 1. A query processing scheme in the distributed search system

Looking for an appropriate scheme for the database selection and result merging for Minerva we evaluated several database selection and result merging methods in our prototype. We also proposed new methods based on pseudo-relevance feedback obtained from the best peer in a rough initial database ranking.

An overview of metasearch and recent work on P2P information retrieval (IR) is introduced in Sect. 2. We present details of our approach for database selection and result merging in Sect. 3. The experimental setup, evaluation methodology, and our results are presented in Sect. 4 and . In Sect. 5 we make some topical conclusions.

2 Related Work

2.1 P2P Search Platforms

ODISSEA [23] uses two-layered search engine architecture and global index structure distributed over the nodes of the system. The distributed version of Fagin’s threshold algorithm [11] is used for result aggregation over the inverted lists. In PlanetP [5] each node maintains an index of its content and summarizes the set of terms in its index using a Bloom filter. This approach is effective for several thousands peers, but it is hardly scalable to long queries.

The issue of efficient score aggregation in P2P IR environment with a structured topology was addressed in [3]. An algorithmic framework KLEE for distributed top-k queries was presented in [16].

2.2 Database Selection

All database selection methods fall in two classes: ad-hoc and language model based.

The ad-hoc database selection algorithms suggested in [27, 4] use the document frequency of a term in a database as the most important evidence for the database usefulness. The most popular representative of ad-hoc family is CORI [4]. It adapts classic $tf \times idf$ formula, which proved its effectiveness for document retrieval, to the problem of database selection. In that case tf (term frequency) turns into a document frequency, while idf (inverted document frequency) becomes analogous icf (inverted collection frequency).

Another group of ad-hoc approaches considers word counts and merely utilizes the summarized similarity of documents as a usefulness measure [10]. In that case, similarity is just a sum of $tf \times idf$ weights for each query term in each document. The most sophisticated among such methods, GLOSS, was presented in [12].

Several works [26, 21] contain quite successful attempts to apply language modeling framework for database selection task. They, again, simply work with a database as a “virtual document” by concatenating all its documents and simulate document retrieval. However, experimental results show that the performance of the language model based selection is not inferior to any of ad-hoc selection methods.

2.3 Result Merging

For consistent merging all search engines must produce a document’s relevance score using same retrieval algorithm and global statistics. However, this requirement is not realistic. For example, under $tf \times idf$ scoring scheme the tf component is document-dependent and fair across all databases. In contrast the idf component is collection-dependent and we should globally normalize it.

When environment is cooperative, i.e. scores and documents statistics are provided by peers, usually Kirsch’s algorithm [13] is used. At query time it

collects local statistics from the selected databases and normalizes document scores. The semi-supervised method for merging results in a hierarchical P2P network [15] uses a centralized database of collection samples. In our setup we cannot afford learning-based approaches, since system is highly dynamic.

A series of publications [7, 4, 14] describes a CORI merging strategy, which heuristically combines both resource and document scores. In [7] it was also suggested that the result merging based on raw *tf* values is a worthwhile approach when involved databases are homogeneous. Result merging techniques for topically organized collections were studied in [14]. Their experiments showed that global *idf* normalization is the best method, but they did not consider real Web pages and overlap between collections.

The approach in [Bau99] is designed for the cooperative environment. A probabilistic ranking algorithm is based on the exported statistics from the search engines. The language modeling based merging from [21] is developed for uncooperative setup, the probability of generating a query from the language model of a document is approximated.

3 Pseudo-Relevance Feedback for Distributed IR

Our analysis of the distributed IR state of art showed that there is a lack of original task-oriented approaches. In particular, the authoritative database selection methods are mostly based on the well-known query-document similarity measures applied with minor changes to the database selection task. The main assumption is that the estimated collections can be represented by a concatenation of their documents. At the same time, being merged together text sources lose their individuality and become very heterogeneous and multi-topically oriented. Consequently, the ambiguity of short Web queries becomes extremely high with respect to these enormously long virtual documents.

There are several ad-hoc query expansion and language model based query modeling methods operating on the top-*k* ranked documents. At the same time, nobody applied these methods in the scope of distributed IR. Moreover, they have never been used simultaneously for IR, what appears to be an omission in our opinion.

The merging task is even more sensitive to the ambiguity of short queries and lack of context. Obviously, we cannot rely on learning methods, since information is outdated quickly in P2P system. Thereupon, pseudo-relevance feedback can be useful. The language model based retrieval algorithms seem the most promising nowadays, so we want to investigate their pseudo-relevance feedback opportunities for merging. We intend not to merely normalize the similarity scores, but also to improve the retrieval algorithm itself.

In the Minerva system, Web pages are crawled with respect to the user's bookmarks and assumed to reflect specific interests. We can exploit this fact using a pseudo-relevance feedback for finding the "preference" language model from the most relevant database. We assume that it was crawled by the user with query-related interests and contains some pages on the relevant topic. For

the pseudo-relevance based model estimation we suggest to use the top- k ranked documents, obtained after query execution on the most highly ranked database for the current query. We increase the number of attributes that can be used to compare databases (*query expansion*) and weight different attributes w.r.t. their level of importance for the relevance evaluation (*query modeling*).

Unfortunately, there is no comparative analysis of pseudo-relevant feedback techniques. We consider two methods to be good representatives of ad-hoc expansion methods, based on the popular IR heuristics Robertson’s method [19] and on the Language Model based approach to IR, Ponte’s method [17]. Both of them assume that a term is more significant if it appears more often in the top- k documents than in any document of the collection in average. Two query modeling methods have been proposed in [28] by Zhai and Lafferty and in [24] by Tao and Zhai. These methods suppose that any term in the pseudo-relevant documents is generated from two sources: the pseudo-relevance model of a user PR , since they were chosen by the users query, and the background language model GE , which can be approximated by the collection language model well enough. The latter method takes into account that the probability of a term being generated by the relevance model is decreasing in parallel with the document score. In both approaches the estimation of $p(t_k|PR)$ for each term t_k is done using Expectation-Maximization (EM) algorithm [9].

3.1 Database Selection

We propose a two-step database selection method. At first, from the set of all available peers P and a query q we build a peer ranking P' by the database selection method described in [21]. For the Global English language model GE we use an approximation from peer lists corresponding to a specific query. Thus, peers are scored by:

$$Score(q, P_i) = - \sum_{k=1}^{|q|} \log p(t_k|P_i) \quad (1)$$

$$p(t_k|P_i) = \lambda \cdot \frac{ctf_{t_k}}{|P_i|} + (1 - \lambda) \cdot p(t_k|GE) \quad (2)$$

$$p(t_k|GE) = \frac{\sum_{i=1}^{|P|} ctf_{t_k}^{P_i}}{\sum_{k=1}^{|T|} \sum_{i=1}^{|P|} ctf_{t_k}^{P_i}}. \quad (3)$$

Here, $p(t_k|P_i)$ is the generation probability of term t_k of language model for collection P_i ; λ is a empirically set smoothing parameter between 0 and 1; ctf_{t_k} is the collection term frequency, the number of term occurrences in the database; T is a system vocabulary, the full set of distinct terms on all peers; $p(t_k|GE)$ is the generation probability of term t_k of Global English language model.

At the next step, the query q is executed on the best database in the ranking P'_1 . At first, the top- k ranked documents are used by ad-hoc query expansion techniques [19, 17], to add new terms to the query. Then, this top- k is utilized by

query modeling techniques [28, 24] to estimate generation probability $p(t_k|PR)$ for each query term from the pseudo-relevance language model PR . As a result, we build new database ranking P'' using expanded query and term generation probabilities. We apply cross-entropy, an information-theoretic measure of distance between two distributions:

$$H(q, P_i) = - \sum_{k=1}^{|q|} p(t_k|PR) \cdot \log p(t_k|P_i). \quad (4)$$

Apparently, the lower cross-entropy of a database language model w.r.t. the pseudo-relevance based language model, the higher similarity of these models. It also turns out that in our formula we combine two values expressing term importance: query-specific $p(t_k|PR)$ and global $p(t_k|GE)$.

3.2 Result Merging

Our merging approach also exploits pseudo-relevance feedback adapted for the distributed setup. Executing the query on the peer, which won the highest rank from the database selection algorithm, we obtain the top- k best results for our pseudo-relevance based model estimation. This language model is then used for adjusting merging results, one user with a highly specified collection of documents implicitly helps another user to refine the final document ranking.

As in the database selection approach above, we estimate the probability $p(q|PR)$ from the pseudo-relevance based cluster of documents PR . The probability $p(q|GE)$ is again approximated using the peer lists information. After the probabilities $p(q|GE)$, $p(q|PR)$, and the query q were sent to every peer in ranking P' , we compute cross-entropy between the pseudo-relevance based language model PR and the document language models for every document D_{ij} :

$$H(t_k, D_{ij}) = -p(t_k|PR) \cdot \log p(t_k|D_{ij}). \quad (5)$$

At second step we compute the ordinary language modeling similarity score, smoothed by General English language model:

$$p(t_k|D_{ij}, GE) = \log(\lambda \cdot p(t_k|D_{ij}) + (1 - \lambda) \cdot p(t_k|GE)). \quad (6)$$

Finally, we combine both scores in a heuristic manner, the empirically set parameter β lies in interval from zero to 1:

$$s = \sum_{k=1}^{|q|} \beta \cdot p(t_k|D_{ij}, GE) + (1 - \beta) \cdot H(t_k, D_{ij}). \quad (7)$$

The search results are sorted in descending order of similarity scores s and the best top- k URLs are presented to the user.

4 Experiments

4.1 Experimental Setup

The Minerva system is implemented in Java and document databases associated with peers are managed by Oracle DBMS. We conducted experiments with 50 databases created from the TREC-2002, 2003 and 2004 Web Track datasets from the “.GOV” domain. For these three volumes, four topics were selected. The relevant documents from each topic were taken as a training set for the support vector machine classification algorithm and 50 collections were created. The non-classified documents were randomly distributed among all databases. Each classified document was assigned to two collections from the same topic. For example, for the topic “American music” we had the subset of 15 small collections and all classified documents were replicated twice in it. The topics with the numbers of corresponding collections are summarized in the Tab. 1, each collection was managed by one dedicated peer. Assuming that the search in our

Table 1. Topic-oriented experimental collections

N	Topic	Number of collections
1	Health and medicine	15
2	Nature and ecology	10
3	Historic preservation	10
4	American music	15

system should be topic-oriented as well as crawling, we selected the set of the 25 out of 100 title queries from the topic distillation task for the TREC 2002 and 2003 Web Track. Queries are selected with respect to two requirements: at least 10 relevant documents exist and query is related to the “Health and Medicine” or “Nature and Ecology” topics. The set of selected queries is presented in [6], relevance judgments are available on the NIST site (<http://trec.nist.gov>).

4.2 Database Selection Experiments

The methodology for evaluation of database selection performance is not standardized. The most popular approach is to measure a quality of selection by *cumulative recall*. It shows which method accumulates relevant documents faster by selecting databases from the top of its ranking. Let $|D_i^r|$ be a number of relevant documents on peer P_i ; N is the number of collections selected from the top of the database ranking P' . Cumulative recall is a fraction, where term of fraction is a sum of all relevant documents on first N peers from ranking P' and denominator is a total number of relevant documents on all peers in P' :

$$Recall = \frac{\sum_{i=1}^N |D_i^r|}{\sum_{i=1}^{|P'|} |D_i^r|}. \quad (8)$$

Resulting evaluation is based on macro averaging of cumulative recall over all test queries. Usually, it is calculated for every N up to sensible system-dependent number, so we consider selection of at most 20 databases.

We evaluate Robertson’s [19] and Ponte’s expansion [17] methods and find the best combination of the following parameters for each method: 5, 10 and 20 expansion terms, and 5, 10, 15, 20, 25 pseudo-relevant documents. Boundaries for the latter parameter are derived from the experiments with document retrieval in original papers, where the size of analyzed top- k did not exceed 20. To estimate the best k for the usage of top- k ranked documents in query modeling methods, we take 7 values from top-10 to top-70 pseudo-relevant documents, it conforms with original papers [28, 24]. In addition, we study the possibility to expand a query by terms having the greatest $p(t_k|PR)$ assigned by the respective modeling methods. Earlier, they have been applied for massive expansion (with full vocabulary) coupled with modeling itself what is completely unfeasible in distributed IR.

It is interesting to observe that both ad-hoc expansion methods spoil the selection being applied without consequent query modeling. Their best parameter setup allows only to approach the performance of non-expanded queries. This observation proves that expansion methods are sensitive to the number of pseudo-relevant documents used, retrieval quality depends on the fraction of indeed relevant documents in pseudo-relevant subset. Moreover, if we use query model as a source for expansion terms, the performance decreases dramatically.

Both query modeling methods improve the retrieval quality, when used in addition to query expansion methods. The modeling method of Tao and Zhai [24] shows marginally better result. We infer that it is important to apply ad-hoc expansion and query modeling simultaneously.

To get the baseline for our experiments, we measured 4 most popular existing selection methods: two language model based methods from [26, 21], CORI [4] and GLOSS [12]. Our results showed that Language model based methods are constantly more effective and method by Si, Callan and others [21] is the best. GLOSS appeared to be the worst. We can observe the performance of these methods and our approach, which uses the combination of Robertsons expansion with Tao and Zhais query modeling, in Table 2. The improvement of our approach upon the performance of baseline method is comparable to the improvement of baseline method over the worst method. Our approach reaches its maximum selection performance with use of only 5 expansion terms what allows its integration into P2P Web search without a significant loss of scalability.

4.3 Result Merging Experiments

For the merging methods evaluation in the Minerva system we used the following score normalization techniques: TF is a merging by raw tf values; TFIDF is merging by local $tf \times idf$ score; TFGIDF uses $tf \times idf$ score with globally normalized idf ; CORI is a merging method from [4]; LM is a language modeling retrieval algorithm. More details about methods can be found in [6].

Table 2. Cumulative recall at different levels of database rankings

	1	2	3	4	5	10	15	20
Our approach	0,128	0,229	0,304	0,375	0,406	0,619	0,759	0,829
LM of Si & Callan	0,092	0,187	0,27	0,352	0,399	0,606	0,733	0,81
LM of Xu & Croft	0,091	0,187	0,27	0,354	0,41	0,603	0,735	0,811
CORI	0,089	0,195	0,244	0,324	0,37	0,588	0,73	0,809
GLOSS	0,102	0,179	0,249	0,322	0,376	0,596	0,715	0,781

For the evaluation, we utilized the framework from [21]. For all tested algorithms, the average precision measure is computed over the 25 queries at the level of the top-5, 10, 15, 20, 25, and 30 documents. The parameter λ in LM method is empirically adjusted, different approaches vary it from 0.4 to 0.7. After preliminary experiments we set λ to 0.4, as it produced the most robust and effective results.

We also measured the retrieval accuracy for non-distributed case on the *single database*, which contains all the documents from the 50 peers. Two non-distributed retrieval algorithms were used: $tf \times idf$ is coded as SingleTFIDF, and language modeling retrieval is coded as SingleLM.

Only 10 selected databases participate in a query execution and, therefore, the effectiveness of the query routing algorithm influences the quality of result. We assessed the result merging methods with several database rankings. Due to space limit, we present results only for manually created IDEAL ranking, where the collections are sorted in a descending order of the number of relevant documents.

In Tab. 3 we summarize the results from the result merging experiments with six merging methods and two non-distributed retrieval algorithms. The best results in every category are shown in bold. Here are the main observations:

- Retrieval effectiveness of all result merging methods is similar;
- The LM method shows the best performance, it is robust under every ranking;
- Surprisingly, the TFIDF method is more effective than the TFGIDF technique, It might be the case that GIDF values, which are averaged over all databases, are more influenced by noise, while local IDF values are more topic-specific;
- An effective database ranking allows to outperform single database baseline;

In the second series of experiments, we evaluated our LMPR technique. Ranking PR is purely based on the cross-entropy between pseudo-relevance based and document language models, see Eq. 5; LM and SingleLM methods remain the same; LMPR is a heuristic combination of LM and PR rankings, as described in Eq. 7.

At first, we conducted experiments for the separate PR ranking in order to find the optimum n for estimating our pseudo-relevance based model, for IDEAL ranking the best choice was $n = 10$. After we fixed the n parameter, we conducted

Table 3. The macro-average precision for evaluated merging methods with the database ranking IDEAL

	SingleTFIDF	SingleLM	TFIDF	TFGIDF	CORI	TF	LM
top-5	0,208	0,224	0,248	0,240	0,240	0,264	0,264
top-10	0,176	0,200	0,204	0,204	0,204	0,184	0,220
top-15	0,160	0,181	0,192	0,163	0,189	0,155	0,184
top-20	0,158	0,158	0,180	0,164	0,178	0,142	0,168
top-25	0,142	0,149	0,165	0,150	0,163	0,125	0,157
top-30	0,135	0,141	0,141	0,141	0,144	0,120	0,145

experiments with different values of the β parameter. We carried out experiments for $\beta = \{0.1, \dots, 0.9\}$ and obtained the best combination with $\beta = 0.6$. In Tab. 4 we present our combined LMPR method and show the separate performance of each methods for comparison.

The single PR ranking, which is purely based on the pseudo-relevance feedback, is poor with the IDEAL ranking. The average precision of the LMPR is the same or slightly better in comparison with LM. We conclude that the LMPB combination of the cross-entropy ranking PB with the LM language model with $\beta = 0.6$ is more effective than the single LM method.

Table 4. The macro-average precision with the database ranking IDEAL, top-10 documents for the pseudo-relevance based language model estimation, $\beta = 0.6$

	PR	LMPR	SingleLM	LM
Top-5	0,248	0,272	0,224	0,264
Top-10	0,192	0,220	0,200	0,220
Top-15	0,165	0,187	0,181	0,184
Top-20	0,146	0,170	0,158	0,168
Top-25	0,130	0,157	0,149	0,157
Top-30	0,119	0,144	0,141	0,145

5 Conclusion and Future work

In this paper, we evaluated existing database selection and result merging methods. We also proposed and evaluated our approach. Its novelty is in that peers ranking and final documents ranking are refined with use of the pseudo-relevance feedback from the best peer in the preliminary peers ranking. In most cases our methods are more effective than existing ones. We come to the conclusion that pseudo-relevance feedback information from topically organized collections allows to improve a quality of distributed IR. The presented result indicates that in future we can think about methods using pseudo-relevance models from several databases considering different levels of their query expertise.

6 Acknowledgment

We would like to thank Wolfgang Nejdl and Paul-Alexandru Chirita for helpful comments on a paper draft.

References

1. Christoph Baumgarten. A probabilistic solution to the selection and fusion problem in distributed information retrieval. In *Proceedings of the 22th Annual International Conference on Research and Development in Information Retrieval ACM SIGIR '99*, pages 246–253. ACM Press, 1999.
2. M. Bender, S. Michel, G. Weikum, and C. Zimmer. Bookmark-driven query routing in peer-to-peer web search. In: Callan, J., Fuhr, N., and Nejdl, W., *Workshop Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval ACM SIGIR '04*, pages 46–57, 2004.
3. W.-T. Balke, W. Nejdl, W. Siberski, and U. Thaden. Progressive Distributed Top-k Retrieval in Peer-to-Peer Networks. *Proceedings of the 21th International Conference on Data Engineering ICDE '05*, Tokyo, Japan, pages 174–185, 2005.
4. J. Callan, *W.B. Croft, editor, Advances in information retrieval*, Chapter Distributed Information Retrieval, Kluwer Academic Publishers, pages 127–150, 2000.
5. F. M. Cuenca-Acuna, C. Peery, R. P. Martin, and T. D. Nguyen. PlanetP: Using gossiping to build content addressable peer-to-peer information sharing communities. *Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing*, pages 236–249, 2003.
6. S. Chernov: Result Merging in a Peer-to-Peer Web Search Engine, Saarland University, Master thesis, 2005.
7. J. P. Callan, Z. Lu, and W. B. Croft. Searching Distributed Collections with Inference Networks. In E. A. Fox, P. Ingwersen, and R. Fidel, editors, *Proceedings of the 18th Annual International Conference on Research and Development in Information Retrieval ACM SIGIR '95*, pages 21–28, Seattle, Washington, ACM Press, 1995.
8. N. E. Craswell: Methods for Distributed Information Retrieval, PhD thesis, 2001.
9. A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, Vol. 39(1), pages 1-38, 1977.
10. N. Fuhr. A decision-theoretic approach to database selection in networked IR. *ACM Transactions on Information Systems*, Vol. 17(3), pages 229-249, 1999.
11. R. Fagin, A. Lotem, and M. Naor. Optimal Aggregation Algorithms for Middleware. *ACM Symp. on Principles of Database Systems*, Santa Barbara, USA, pages 102–113, 2001.
12. L. Gravano, H. Garcia-Molina, and A. Tomasic. GLOSS: text-source discovery over the Internet. *ACM Transactions on Database Systems*, Vol. 24(2), pages 229–264, 1999.
13. S. T. Kirsch. Distributed search patent. u.s. patent 5,659,732, 1997.
14. L. S. Larkey, M. E. Connell, and J. P. Callan. Collection selection and results merging with topically organized u.s. patents and trec data. In *Proceedings of the Tenth International Conference on Information and Knowledge Management CIKM '00*, ACM Press, pages 282–289, 2000.
15. J. Lu and J. Callan. Merging retrieval results in hierarchical peer-to-peer networks. In *Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval ACM SIGIR '04*, pages 472–473, 2004.

16. S. Michel, P. Triantafillou, and G. Weikum. KLEE: A Framework for Distributed Top-K Query Algorithms. In *Proceedings of the 31st International Conference on Very Large Data Bases VLDB '05*, pages 637–648, 2005.
17. J. M. Ponte. A Language Modeling Approach to Information Retrieval. PhD thesis, University of Massachusetts Amherst, 1998.
18. J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21th Annual International Conference on Research and Development in Information Retrieval ACM SIGIR '98*, pages 275–281, 1998.
19. S. E. Robertson and S. Walker. Okapi/keenbow at trec-8. In *Proceedings of the 8th Text REtrieval Conference (TREC '99)*, pages 151–162, 1999.
20. P. Serdyukov. Query routing in a peer-to-peer web search engine, Saarland University, Master thesis, 2005.
21. L. Si, R. Jin, J. P. Callan, and P. Ogilvie. A language modeling framework for resource selection and results merging. In *Proceedings of the 10th International Conference on Information and Knowledge Management CIKM '02*, pages 391–397, ACM Press, 2002.
22. I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications. In *Proceedings of the 2001 ACM SIGCOMM Conference*, pages 149–160, 2001.
23. T. Suel, C. Mathur, J. Wu, J. Zhang, A. Delis, M. Kharrazi, X. Long, and K. Shanmugasundaram. Odissea: A peer-to-peer architecture for scalable web search and information retrieval. In *International Workshop on the Web and Databases (WebDB '03)*, pages 67–72, 2003.
24. T. Tao and C. Zhai. A mixture clustering model for pseudo feedback in information retrieval. In *Proceedings of the Meeting of the International Federation of Classification Societies*, 2004.
25. G. G. Towell, E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird. Learning collection fusion strategies for information retrieval. In *International Conference on Machine Learning*, pages 540–548, 1995.
26. J. Xu and B. W. Croft. Cluster-based language models for distributed retrieval. In *Proceedings of the 22th Annual International Conference on Research and Development in Information Retrieval ACM SIGIR '99*, Berkeley, CA, USA, pages 254–261, 1999.
27. B. Yuwono, D. L. Lee, R. W. Topor, and K. Tanaka. Server ranking for distributed text retrieval systems on the internet. In *Proceedings of the 5th International Conference on Database Systems for Advanced Applications DASFAA '97*, Melbourne, Australia, pages 41–50, 1997.
28. C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the 10th International Conference on Information and Knowledge Management CIKM '01*, ACM Press, pages 403–41, 2001.