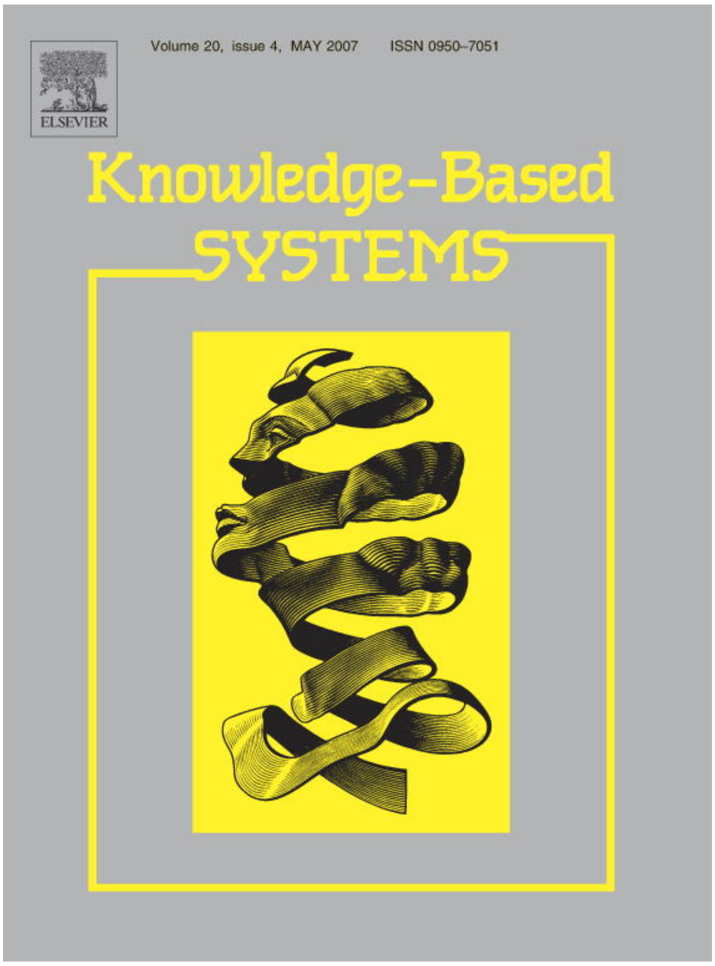
Volume 20, issue 4, MAY 2007 ISSN 0950-7051
ELSEVIER
Knowledge-Based
SYSTEMS

Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Knowledge-Based Systems 20 (2007) 321–328
Knowledge-Based SYSTEMS
www.elsevier.com/locate/knosys

Aggregation of web search engines based on users' preferences in WebFusion

Amir Hosein Keyhanipour *, Behzad Moshiri *, Majid Kazemian, Maryam Piroozmand, Caro Lucas

Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

Received 1 March 2006; received in revised form 26 July 2006; accepted 8 August 2006
Available online 7 September 2006

Abstract

The required information of users is distributed in the databases of various search engines. It is inconvenient and inefficient for an ordinary user to invoke multiple search engines and identify useful documents from the returned results. Meta-search engines could provide a unified access for their users. In this paper, a novel meta-search engine, named as WebFusion, is introduced. WebFusion learns the expertness of the underlying search engines in a certain category based on the users' preferences. It also uses the "click-through data concept" to give a content-oriented ranking score to each result page. Click-through data concept is the implicit feedback of the users' preferences, which is also used as a reinforcement signal in the learning process, to predict the users' preferences and reduces the seeking time in the returned results list. The decision lists of underling search engines have been fused using ordered weighted averaging (OWA) approach and the application of optimistic operator as weightening function has been investigated. Moreover, the results of this approach have been compared with those achieve by some popular meta-search engines such as ProFusion and MetaCrawler. Experimental results demonstrate a significant improvement on average click rate, and the variance of clicks as well as average relevancy criterion.

Keywords: Meta-search engine; User preferences modeling; Click-through data; Reinforcement learning; Optimistic OWA; Decision fusion

1. Introduction

The scale of the WWW is daily increasing, consisting more than 10 billion publicly visible web documents [1] distributed on millions of servers. Individual general-purpose search engines have been unable to track this growth. The coverage of the Web by each of the major search engines has been steadily decreasing despite their efforts to index more web pages. Several studies have shown that no single search engine has complete coverage and it is unlikely that any single web search engine ever will [2–4]. Worse yet, as these search engines get larger, higher percentages of their indexed information are becoming obsolete.

Relatively, low coverage of the web by individual search engines spurred research into meta-search engines, or tools that send user queries to various search engines and combine the results. Research has demonstrated that combining results, in the form of a meta-search engine, produces a significant improvement in coverage and search effectiveness [3,5]. A meta-search engine could be considered as an interface on the top of local search engines to provide uniform access to many local search engines.

Although a meta-search engine improves coverage, it is still limited by the results returned from the underlying search engines. Another challenge for meta-search engines is the database selection problem, which is to identify local search engines that are likely to contain useful documents for a given query. The objective of performing database selection is to improve efficiency by sending each query to only potentially useful search engines. Regarding this,

* Corresponding authors. Tel.: +98 912 248 1647 (A.H. Keyhanipour).
E-mail addresses: a.keyhanipoor@ece.ut.ac.ir (A.H. Keyhanipour), moshiri@ut.ac.ir (B. Moshiri).

0950-7051/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2006.08.001

network traffic and the cost of searching useless databases could be reduced. Using the historical information of the underlying search engines and the behavior of the users, it can dynamically dispatch the queries to the suitable underlying search engines to achieve better results for the given queries.

In this paper, we proposed a new meta-search engine named as WebFusion. WebFusion provides a mapping between the users' categories and the underlying search. It uses the implicit feedback of the users as a reinforcement signal during system learning and simulates the users' preferences from their usage logs. Moreover, the returned results of the underlying search engines have been fused using Optimistic OWA operator based on the users' preferences. Considering the weights achieved by this operator, the final decision list has been re-ranked and displayed.

The rest of this paper is organized as follows: In Section 2, related works and their applications in the Web environment are reviewed and compared. In Section 3 we introduce a framework for modeling users' preferences used in collection fusion. The proposed user modeling approach is based on the historical behavior of the users and underlying search engines. Section 4 demonstrates the Optimistic OWA approach which is used for combining the returned results of underlying search engines. In Section 5, a re-ranking method which is based on this framework is described. Experimental results will be presented in Section 6. Section 7 briefly illustrates the prototype of the proposed system named as WebFusion. Finally, in Section 8, the conclusions and further research works are given.

2. Related research works

Currently, a large number of research papers on the issues related to meta-search engines or distributed collections have been published [6–9]. For database selection, most approaches rank the databases for a given query based on certain usefulness measures. For example, gGlOSS uses the sum of document similarities that are higher than a threshold [10], CORI Net uses the probability that a database contains relevant documents due to the terms in a given query [6] and Meng uses the expected number of documents whose similarities are higher than a threshold [11]. All these database ranking methods are heuristics as they are not designed to produce optimal orders based on some optimality criteria. A necessary and sufficient condition for ranking databases optimally was given in [12].

Query log analysis is also extensively investigated in recent years. Joachims [13] proposed a method of utilizing click-through data in learning of a retrieval function. Specifically, he introduced a new method for training a retrieval function on the basis of click-through data, which he called it, Ranking SVM. Xue et al. [14] extended this research by proposing a novel iterative reinforced algorithm to utilize the users' click-through data to improve search performance. The algorithm fully explores the interrelations between queries and the corresponding Web pages, and effectively finds "virtual queries" for those Web pages. More recent methods such as QueryFind [15–17] are based on users' feedbacks regarding the underlying search engines' recommendations. They can provide more relevant results with the higher rank in the results list. Another new approach is based on exploiting the filtering capabilities of search engines and the generalized use of weights and aggregation operators to rank documents [18].

For collection fusion, most earlier approaches use weighted allocation to retrieve documents, that is to retrieve proportionally more documents from databases that have higher ranking scores (e.g., CORI Net, ProFusion [19], and MRDD [20,21]), and use adjusted local similarities of documents to merge retrieved documents (e.g., D-WISE [22], and ProFusion). These approaches are all heuristics and are not aimed at guaranteeing the retrieval of all potentially useful documents for a given query. To determine what documents to be retrieved from a local database, different approaches are proposed [11,12] to find a tight local similarity threshold for the local database based on a global similarity threshold. These approaches aim at guaranteeing the retrieval of all potentially useful documents from each selected database while minimizing the retrieval of useless documents. The problem with this type of approaches is that they must know what local similarity function is used in each search engine but the similarity function is usually proprietary. The "Inquirus" meta-search engine [23] uses the real global similarities of documents to merge retrieved documents. As a result, high quality merging can be achieved. The disadvantage is that documents may need to be fetched to the meta-search engine to enable the computation of their global similarities.

These methods do not consider the expertness of search engines in a certain category simultaneously with the users' preferences, which is promising when different sources of knowledge with different coverage are used [24].

3. A framework for user modeling

The search engines' performance concerning the user preferences could be inferred from users' feedback. One of the well-known kinds of implicit feedbacks is chosen URLs by the user. Clearly, users do not randomly click on links, but make an (somewhat) informed choice. While click-through data is typically noisy and clicks are not "perfect" relevance judgments, the clicks are likely to convey some information [13]. Most users usually are likely to click on relevant results, thus we benefit from a large quantity of query logs. Experiments show that around 82% of the queries are in fact related to the topics of the clicked Web pages [14].

If a URL on the result page is chosen by the user, generally it means that the search engine which contributed this URL performed better on the query than those search engines contributed URLs that were ignored by the user. In other words, the confidence factor for the search engine that contributed the selected URL should be higher for the query's category than search engines whose contributions are overlooked. In general, the earlier a link on the result page is selected, the more relevant the link is. Regarding this fact, we have assigned an expertness value expt_i^{c}, which illustrates the expertness of the search engine i in the category c. The expertness value associated with each category indicates that how well a particular search engine performs on queries in a particular category.

We use the rationale that if the assigned expertness values were perfect, then the higher ranked items on the results page would more likely be followed by the user. Vice versa if the user follows some lower ranked URLs, it means that the expertness values of the search engines provided that URLs were not accurate sufficiently and must be modified. This modification could be done by increasing the expertness values of successful search engines as well as decreasing these values for unsuccessful ones.

The expertness values are updated by a bootstrapping technique based on previous expertness values and the current usage behavior of the user [25]. In this way, search engines in which the higher ranked results are clicked earlier, are more rewarded than the others. The expertness updating formula and the corresponding reward function are given as follow:

$$\text{expt}_i^{\text{c}} = (1-\alpha)\text{expt}_i^{\text{c}} + \alpha \cdot \gamma, \ \text{where} \qquad \gamma = \frac{\sum_{i \in \text{hitted results}} (N-i) \times t_i}{\sum_{i=1}^{N} (N-i) \times i}, \tag{1}$$

where N is the number of returned results, i is the index of a result in the ranked list and t_i is the index of hitted result i. This reward function is reasonable, because users are more interested in results that are located in the top of returned list than the other results. The learning rate α is also modified by the following formula:

$$\alpha = \exp(\beta \times t), \tag{2}$$

where t is the iteration number and β is the regulator of the learning rate. Each iteration starts when a user submits a query and it will be finished when the user selects the results and closes the session. The learning rate permits more exploration in the beginnings of the learning process than exploitation while it is decreased during the time.

4. Ordered Weighted Averaging

The Ordered Weighted Averaging operators (OWA) were originally introduced by Yager [26] to provide a means for aggregating scores associated with the satisfaction of multiple criteria, which unifies in one operator the conjunctive and disjunctive behavior. The OWA operator of dimension n is a mapping such as: F: $R^n \to R$ and is given by

$$\text{OWA}(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} w_j x_{\sigma(j)}, \tag{3}$$

where σ is a permutation that orders the elements $x_{\sigma(1)} \leqslant x_{\sigma(2)} \leqslant \ldots \leqslant x_{\sigma(n)}$. The weights are all non-negative ($w_i \geqslant 0$) and their sum equals to one ($\sum_{i=1}^{n} w_i = 1$).

This operator has been proved to be very useful, because of its versatility, The OWA operators provide a parameterized family of aggregation operators, which include many of the well-known operators such as the maximum, the minimum, the k-order statistics, the median and the arithmetic mean. In order to obtain these particular operators we should simply choose particular weights. The Ordered Weighted Averaging operators are commutative, monotone, idempotent, they are stable for positive linear transformations, and they have a compensatory behavior. This last property translates the fact that the aggregation done by an OWA operator always is between the maximum and the minimum. It can be seen as a parameterized way to go from the *min* to the *max*. In this context, a degree of maxness (initially called orness) was introduced in [26], defined by

$$\text{maxness}(w_1, w_2, \ldots, w_n) = \frac{1}{n-1} \sum_{i=1}^{n} (n-i) w_i, \tag{4}$$

where the minimum, $\text{maxness}(1, 0, \ldots, 0) = 0$ and for the maximum $\text{maxness}(0, \ldots, 0, 1) = 1$.

A simple class of OWA operators as exponential class of OWA operators was introduced to generate the OWA weights satisfying a given degree of maxness. The optimistic and pessimistic exponential OWA operators were correspondingly introduced as follows [26]:

$$\begin{aligned}
\text{Optimistic}: & w_1 = \lambda, \\
& w_2 = \lambda(1-\lambda), \\
& w_3 = \lambda(1-\lambda)^2, \\
& \ldots \\
& w_{n-1} = \alpha(1-\lambda)^{n-2}, \\
& w_n = (1-\lambda)^{n-1}, \\
\text{Pessimistic}: & w_1 = \lambda^{n-1}, \\
& w_2 = (1-\lambda)\lambda^{n-2}, \\
& w_3 = (1-\lambda)\lambda^{n-3}, \\
& \ldots \\
& w_{n-1} = \lambda(1-\lambda), \\
& w_n = (1-\lambda),
\end{aligned} \tag{5}$$

where parameter λ belongs to the unit interval [0 1] and is related to orness value regarding the n. Here, we used an Optimistic OWA operator with $\lambda = 0.25$ which was simply obtained by a try-and-error algorithm.

5. Re-ranking method

The returned results of the underlying search engines are considered as their decisions for the dispatched query which are fused in the decision level of information fusion. In decision level which is the highest level of information fusion, the more weights are assigned to the higher results of the more expert search engines for the dispatched query in the corresponding category.

For each result item j from the search engine i which is classified in category c, a weight is computed as follows:

$$w(i,j,\mathrm{c}) = \mathrm{expt}_i^{\mathrm{c}} \times \left(1 - \frac{j}{N}\right). \tag{6}$$

In the first re-ranking approach, if an item were returned by more than one of the underlying search engines-obviously each contributed engine has assigned its own weight to that item, the maximum assigned weight would be assumed as the item final weight. After assigning the weights to each returned result, they are sorted by their assigned weights in a decreasing manner and the final ordered list is generated.

In the second approach, we have relaxed the constraint of the assigning each document only to one of the underlying search engines. This is done by assigning OWA weights to the search engines that return a specific item in their results list. For each result item, the weights of search engines to this item are sorted decreasingly and the Optimistic OWA weights are assigned to these search engines according to the Eq. (5). Final weight of each result item is calculated as follows:

$$w_{\mathrm{F}}(j) = \sum_{i=1}^{N} w(i,j,\mathrm{c}) \times w_{\mathrm{OWA}}(i). \tag{7}$$

After assigning the weights to each returned result, similarly to first re-ranking approach the final ordered list is generated.

6. Experimental results

In order to study the effectiveness of the proposed method, we have selected 280 sample queries in the "Computers" category: 150 queries as training queries for the learning process and the remaining as test queries. These queries were collected using a proxy application mounted on the server of the "Instrumentation and Industrial Control Lab." of the ECE department of the University of Tehran on 21st November 2005. Some of these queries are listed in Table 1.

The performance of the WebFusion is measured by factors such as average click rate, total relevancy of the returned results and the variance of the clicked results. The first measure which is the average position of the clicked results indicates that more interesting results are settled on the top of the ranked list. If average click rate is relatively little, it shows that users can find useful results at the first portion of the ranked list while saving time and energy. The second criterion, measures the relevancy of the content of the returned results according to the judgment of the users. This criterion can be considered as the indicator of nature of responses. The last factor shows the long term performance of the meta-search engine which is determined by the behavior of the users. Consideration of these criteria together, can provide a good insight about the meta-search engine. For calculating the relevancy of the results, we have extended the relevancy measure proposed in [27] as follows:

$$\text{relevancy} = \frac{(2 \times \text{number of relevant documents}) + (\text{number of undecided documents})}{(2 \times \text{number of returned documents})}.$$

In our study, a number of students of "Advanced Instruments Lab." were asked to execute the sample search queries on WebFusion. The results are supposed to be classified into three categories: relevant, undecided, and irrelevant documents. Each participant was asked to evaluate the result items in the corresponding classes.

Most of the users usually browse just a first few numbers of returned results by a search engine. The top 20 retrieved results of each underlying search engine have been used for the judgment of users. Relevance judgments were done according to users' preferences, e.g., overall value of the returned results. Relevant items were scored as 2, irrelevant ones as 0 and undecided items as 1. One sample from the judgment of a user for a sample query has shown in Table 2.

Fig. 1 shows the long term behavior of WebFusion in comparison to ProFusion [28]. As it can be seen, mean of average clicks of ProFusion is 9.701 while it is 7.786 and 6.233 for Max fusion and Optimistic OWA methods, respectively. As it mentioned before the average click rates demonstrates the amount of settlement of more preferable results in the returned list of results. The variance of ProFusion is less than these two approaches which indicates that the ProFusion is in its steady state while these techniques are not in their final state and they are still learning and adapting based on user clicks.

Fig. 2 shows the average click rate of WebFusion with two decision fusion approaches in comparison to the ProFusion along the time. Table 3 depicts sample of click-through data, order of clicks, and the corresponding average click rate in WebFusion using OWA aggregation operator. For ProFusion the mean of average clicks has

Table 1
Some sample queries

Computational intelligence techniques	Pattern recognition neural networks	Journal information and security	Artificial immune system application
Robotic fusion	Human genetics	Swarm intelligence robotic	Soft computing
Combining pattern classifiers	Information filtering techniques	Protein structure prediction	Pattern recognition neural networks
Computational biology and chemistry	Aggregation and fusion of information	Relational database management system	Space weather time-series prediction

Table 2
Judgment of a user for a sample query

	LookSmart	Lycos	Altavista	MSN	Yahoo	Teoma	WiseNut	MetaCrawler	ProFusion	WebFusion-Max	WebFusion-OWA
Result 1	R	R	R	R	U	R	R	U	R	R	R
Result 2	U	U	R	U	R	R	U	R	R	R	R
Result 3	I	I	U	R	R	I	R	I	R	U	R
Result 4	R	R	R	I	R	U	R	R	I	R	R
Result 5	I	U	U	I	U	U	U	R	U	I	R
Result 6	R	U	I	R	U	R	I	R	U	R	U
Result 7	I	R	R	U	I	I	U	I	R	R	U
Result 8	R	R	U	R	I	R	R	R	R	U	U
Result 9	I	I	R	R	R	R	R	I	R	U	U
Result 10	I	U	R	U	R	R	I	U	I	U	R
Result 11	R	U	U	U	U	U	I	R	U	R	R
Result 12	U	R	I	R	U	I	I	R	R	R	R

Query, "Fuzzy Decision making algorithms"; R, related; I, irrelative; U, undecided.

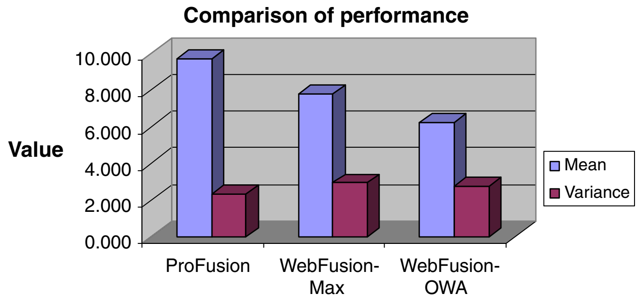

Fig. 1. Performance of WebFusion in comparison to ProFusion.

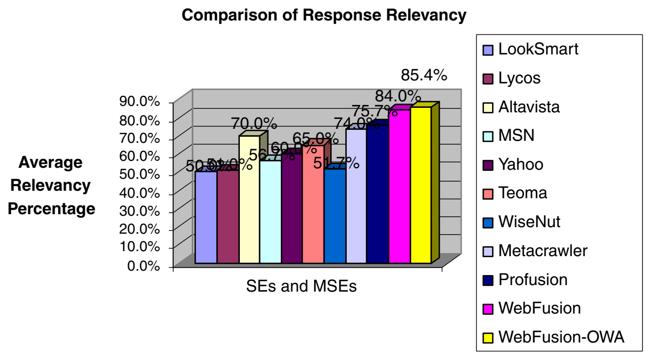

Fig. 3. Response relevancy.

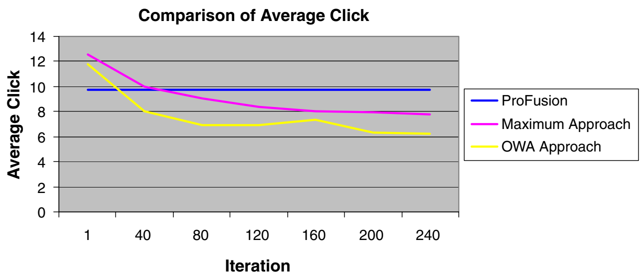

Fig. 2. Average click rate along the time.

been shown because the ProFusion has reached to the steady state and the iterations along the time could not affect its results while the proposed approaches are adapted during the time.

Fig. 3 shows the relevancy of the results of the WebFusion comparing to the MetaCrawler [29], ProFusion and underlying search engines. As it shown, the average relevancy of WebFusion with these decision fusion approaches are 84.0% and 85.4%, respectively, while it is about 75.7% for ProFusion and 74.0% for MetaCrawler. It is also noticeable that the average relevancy of WebFusion is 15.4% better than those achieved by the best underlying search engine in this sense. This significant enhancement depicts that the knowledge of each underlying search engine can just cover a limited region of WWW in a certain category. Therefore, knowledge fusion of these search engines in their expertness regions will increase the total coverage as well as quality of returned results.

Table 4
Underlying search engines of WebFusion

1	LookSmart, http://www.looksmart.com
2	Netscape, http://home.netscape.com
3	Lycos, http://www.Lycos.com
4	AltaVista, http://www.altavista.com
5	MSN, http://www.msn.com
6	Yahoo!, http://www.yahoo.com
7	AlltheWeb, http://www.alltheWeb.com/
8	Teoma, http://www.teoma.com
9	WiseNut, http://www.wisenut.com

Table 3
Sample of click-through data in WebFusion-OWA

Query	Click orders in returned results of WebFusion-OWA							Average click rate
Neuro-fuzzy systems ANFIS	1	4	5	6	7	12	19	7.71
Soap web-services	3	4	6	7	8	11	12	7.29
Supply chain management multi-agent	1	2	4	5	8	10	13	6.14
Digital library federated search	1	2	3	4	11	9	15	6.43

7. A prototype system

To elicit data and provide a framework for testing the proposed method, we have implemented a WWW meta-search engine called "WebFusion". WebFusion combines the results of nine well-known search engines on WWW. These search engines are listed in Table 4. Selected search engines are considered to be most familiar ones that accept

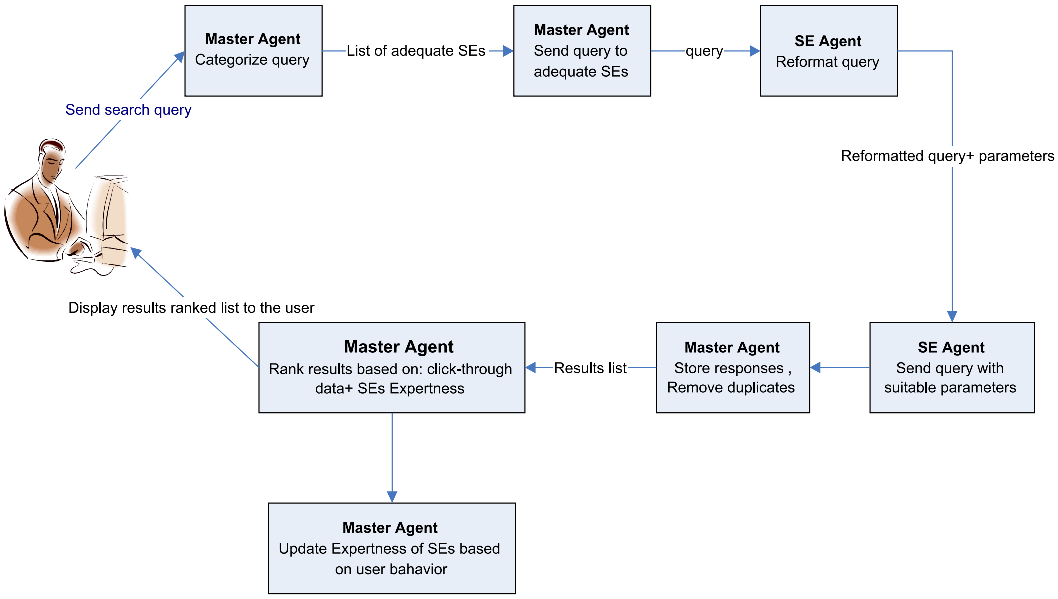

Fig. 4. Working diagram of WebFusion.

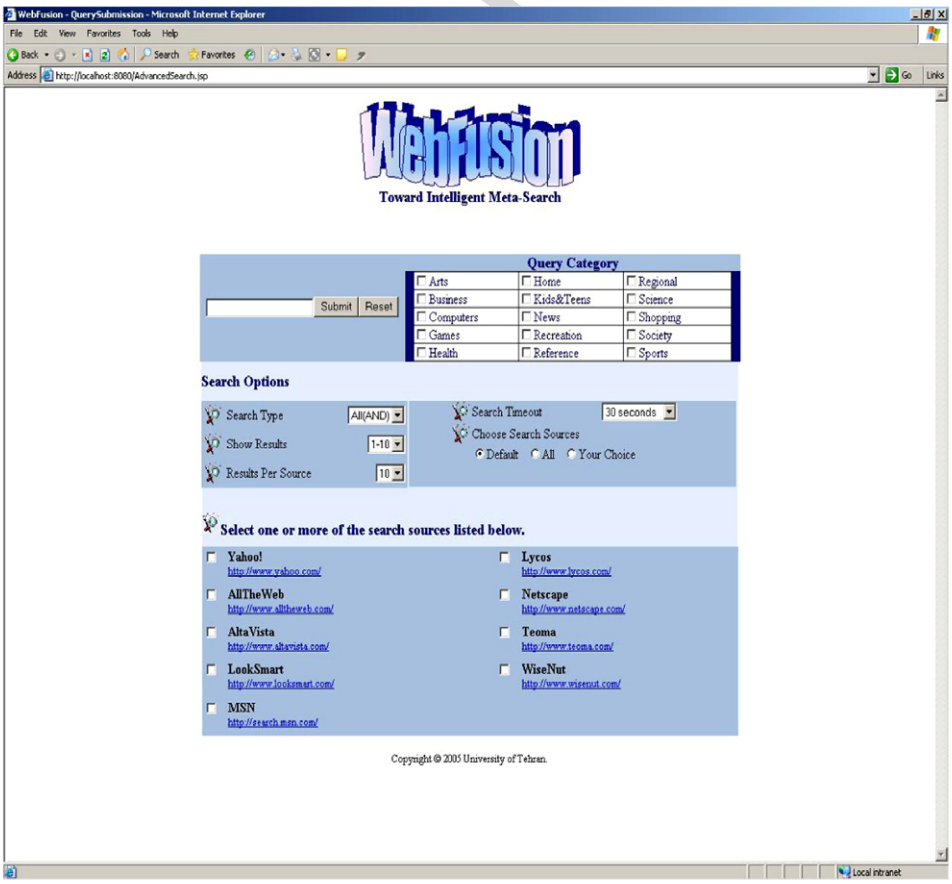

Fig. 5. Advanced search page of WebFusion.

automated querying mechanism which is developed for WebFusion. Some other search engines like Google do not allow sending automated queries [30].

The WebFusion is a meta-search engine based on J2EE technology [31] with a multi-agent architecture. The working diagram of WebFusion is briefly described in Fig. 4. Users send their queries through the interface of WebFusion. These queries are forwarded to the underlying search engines. The result pages that are returned by these search engines can be controlled by parameters such as time-out, results per search engine and etc. These returned rank lists are processed and redundant results are eliminated. Afterward, by using the proposed re-ranking method, the final ranked list is generated and delivered back to the user interface. Details of implementation and some previous results of WebFusion are reported in [32,33]. The Figs. 5, 6 show some snapshots of the implemented system.

WebFusion has a multi-layer/multi-agent architecture which consists of two layers of different types of agents. At the first layer, there are many agents; each is responsible for communication with a specified search engine. These agents are self-contained black boxes which handle all the representation, computation, reasoning, and execution that is necessary for its particular search engine. By learning, these agents gradually become specialist for communication with the dedicated search engine. Learning is done by learning signal, which are send from the master agent. Master agent is responsible of gathering search results from agents and training them. Fig. 7 shows the architecture of WebFusion.

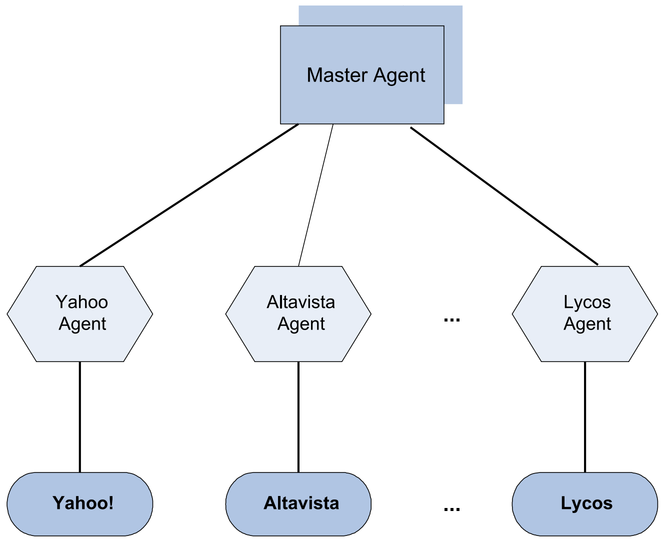

Fig. 7. Architecture of WebFusion.

Fig. 6. Results page of WebFusion.

Agents operate concurrently. Because of this distributed operation, this system is able to react flexibly to changes in the environment and make the corresponding adjustments.

8. Conclusion and further research

The optimum integration of decision lists of search engines and awareness of the users' preferences are the most challenging problems in the area of meta-search engines. In this paper, Optimistic OWA operator has been used to combine the decisions of underlying search engines which have different degrees of importance in a certain category. In this regard, a multi-agent architecture has been proposed for WebFusion which is easy to extend, maintain, and distribute. The experimental results show that concerning the user preferences as well as integration of decision lists has significant effect on reducing users' time and effort for finding the required information.

There are still some open issues ahead: usually users are not convenient with the explicit feedbacks which ask them to vote for the quality of the returned results. Therefore, using other implicit information, such as the spending time or the depth of browsing in each returned page, is more interesting. As it shown in this research, the fusion of decision lists has an important effect in users' preferences satisfaction. In this regard, developing customized information fusion techniques in decision level for better re-ranking are still promising.

References

[1] A. Gulli, A. Signorini, The indexable Web is more than 11.5 billion pages, in: Poster Proceedings of the 14th International Conference on World Wide Web, ACM Press, Chiba, Japan, 2005, pp. 902–903.
[2] S. Lawrence, C.L. Giles, Searching the World Wide Web, Science 280 (1998) 98–100.
[3] S. Lawrence, C.L. Giles, Accessibility of information on the web, Nature 400 (1999) 107–109.
[4] M. Gordon, P. Pathak, Finding information on the World Wide Web: the retrieval effectiveness of search engines, Information Processing and Management 35 (2) (1999) 141–180.
[5] E.W. Selberg, Towards Comprehensive Web Search, Ph.D. thesis, University of Washington, 1999.
[6] J. Callan, Z. Lu, W. Croft, Searching distributed collections with inference networks, in: Proceedings of the ACM SIGIR Conference, Seattle, 1995, pp. 21–28.
[7] D. Dreilinger, A. Howe, Experiments with selecting search engines using metaserach, ACM Transactions on Information Systems 15 (3) (1997) 195–222.
[8] M. Meng, K. Liu, C. Yu, W. Wu, N. Rishe, Estimating the usefulness of search engines, in: Proceedings of the IEEE International Conference on Data Engineering, Sydney, Australia, 1999, pp. 146–153.
[9] K. Liu, C. Yu, W. Meng, W. Wu, N. Rishe, A statistical method for estimating the usefulness of text databases, IEEE Transaction on Knowledge and Data Engineering 14 (6) (2002) 1422–1437.
[10] L. Gravano, H. Garcia-Molina, Generalizing GlOSS to vector-space databases and broker hierarchies, VLDB (1995).
[11] W. Meng, K. Liu, C. Yu, X. Wang, Y. Chang, N. Rishe, Determine text databases to search in the internet, VLDB (1998).
[12] T. Kirk, A. Levy, Y. Sagiv, D. Srivastava, The Information Manifold, AAAI Spring Symposium on Information Gathering in Distributed Heterogeneous Environments, 1995.
[13] T. Joachims, Optimizing search engine using click-through data, in: Proceedings of the ACM Conference on Knowledge Discovery and Data Mining, 2002.
[14] Gui-Rong Xue, Huajun Zeng, Zheng Chen, Yong Yu, Wei-Ying Ma, Wensi Xi, Weiguo Fan, Optimizing Web Search Using Web Click-through Data, Thirteenth ACM Conference on Information and Knowledge Management (CIKM 2004), USA, 2004.
[15] T.X. Lin, HitRank: Search Ranking Based on Mining User-oriented Feedback, Master Thesis, National Taiwan University of Science and Technology, 2002.
[16] N. Sato, M. Udagawa, M. Uehara, Y. Sakai, H. Mori, Query based site selection for distributed search engines, in: Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW'03), IEEE, 2003.
[17] P.H. Wang, J.Y. Wang, H.M. Lee, QueryFind: search ranking based on users' feedback and expert's agreement, in: Proceedings of IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04), 2004.
[18] M. Gomez, J.C. Abasolo, Improving meta-search by using query-weighting and numerical aggregation operators, in: Proceedings of the Fifth Congress of the Catalan Association for Artificial Intelligence, 2002.
[19] Y. Fan, S. Gauch, Adaptive Agents for Information Gathering from Multiple, Distributed Information Sources, Intelligent Agents in Cyberspace, Technical Report SS-99-03, 40-46, The AAAI Press, Menlo Park, CA, 1999.
[20] E. Voorhees, N. Gupta, B. Johnson-Laird, The Collection Fusion Problem. TREC-3, 1995.
[21] L. Gravano, H. Garcia-Molina, Merging ranks from heterogeneous internet sources, in: Proceedings of 23rd International Conference on Very Large Data Bases, Greece, 1997, pp. 196–205.
[22] B. Yuwono, D. Lee, Server Ranking for Distributed Text Resource Systems on the Internet, DASFAA'97, 1997
[23] S. Lawrence, C.L. Giles. Inquirus, the NECi Meta Search Engine, Seventh International World Wide Web Conference, 1998.
[24] R.W. White, I. Ruthven, J.M. Jose, The use of implicit evidence for relevance feedback in Web retrieval, in: Proceedings of 24th BCS-IRSG European Colloquium on IR Research. Lecture notes in Computer Science 2291, Glasgow, 2002, pp. 93–109.
[25] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 1998.
[26] R.R. Yager, On ordered weighted averaging aggregation operators in multi-criteria decision making, IEEE transactions on Systems, Man and Cybernetics 18 (1988) 183–190.
[27] S. Gauch, G. Wang, Information fusion with profusion, in: Proceedings of WebNet'96: The First World Conference of the Web Society, San Francisco, CA, October 1996.
[28] ProFusion Web site, http://www.profusion.com, December 2005.
[29] MetaCrawler Web site, http://www.metacrawler.com, December 2005.
[30] Google Web site, http://www.google.com/terms_of_service.html, December 2005.
[31] E. Armstrong, J. Ball, S. Bodoff, D.B. Carson, I. Evans, D. Green, K. Haase, E. Jendrock, The J2EE 1.4 Tutorial, Sun Microsystems (2005).
[32] A.H. Keyhanipour, M. Piroozmand, B. Moshiri, C. Lucas, A multi-layer/multi-agent architecture for meta-search engines, in: Proceedings of ICGST International Conference on Artificial Intelligence and Machine Learning (AIML-05), Cairo, Egypt, 2005.
[33] A.H. Keyhanipour, B. Moshiri, M. Piroozmand, C. Lucas, Application of Behavioral User Modeling Used in Meta Search Engines by Information Fusion, in: Proceedings of International Conference on Machine Intelligence (ACIDCA-ICMI) Sponsored by IEEE, Tozeur, Tunisia, 2005.