Effective Ranking Fusion Methods for Personalized Metasearch Engines

Leonidas Akritidis, Dimitrios Katsaros and Panayiotis Bozanis
Department of Computer and Communication Engineering, University of Thessaly
leoakr@uth.gr, dimitris@delab.csd.auth.gr, pbozanis@inf.uth.gr

Abstract

Metasearch engines are a significant part of the information retrieval process. Most of Web users use them directly or indirectly to access information from more than one data sources. The cornerstone of their technology is their rank aggregation method, which is the algorithm they use to classify the collected results. In this paper we present three new rank aggregation methods. At first, we propose a method that takes into consideration the regional data for the user and the pages and assigns scores according to a variety of user defined parameters. In the second expansion, not all component engines are treated equally. The user is free to define the importance of each engine by setting appropriate weights. The third algorithm is designed to classify pages having URLs that contain subdomains. The three presented methods are combined into a single, personalized scoring formula, the Global KE. All algorithms have been implemented in QuadSearch, an experimental metasearch engine available at http://quadsearch.csd.auth.gr.

1. Introduction

Metasearch engines are Web services that receive user queries and dispatch them to multiple crawl-based search engines (also called component engines). Then, they collect the returned results, reorder them and present the ranked result list to the end user. The ranking fusion algorithms that meta-search engines utilize are based on a variety of parameters, such as the ranking a result receives and the number of its appearances in the component engine's result lists. These parameters are being exploited to compute a weight (also called *score*) for each collected result.

Better results classification can be achieved by employing ranking fusion methods that take into consideration additional information about a Web page. Another core step is to implicitly or explicitly collect some data concerning the user that submits the query. This will assist the engine to decide which results suit better to his informational needs.

A considerable number of papers describe how a search system should approach personalized search. Haveliwala [9] builds a topic-oriented PageRank, by computing the similarity of a user query and each of the 16 main topics of the Open Directory project [10]. In addition, many researches focus on constructing user profiles. However, none of these studies propose a ranking formula that is suitable for metasearch engines.

The existing methods assign scores according to objective criteria, such as the rank a result receives from the component engines etc. None of them can accept data varying among different users (subjective data) and produce different results respectively. In other words, the current methods lack personalization; they output the same results for the same queries, submitted by different users. All methods presented in this paper manage to confront this problem.

The first method, the *GeoKE Algorithm*, is designed to provide different results for users having different geographical origins. It is based on regional information obtained for both the Web page and the user. Moreover, we examine the possibility that the Web page is written in a language that is familiar to the user and assign proportional scores.

Although geographically oriented search has recently gained substantial attention from industry, we were unable to find any public information on the ranking algorithms used by metasearch engines. Some sporadic works we have studied concern the way that search engines should construct their index, by exploiting the geographical location information of Web pages. Nevertheless, none of these techniques can be used by metasearch engines; obtaining the content of all results at query time would result a dramatic recession to the system's performance.

The original *KE Algorithm* [1] considers all component engines as major and treats them equally.

We propose a scoring strategy in which the user can define the importance of each search engine and modify its impact in the results classification process. The *Weighted KE Algorithm* is also designed to achieve personalized results.

The third expansion analyzes the domain structure that characterizes a Web page. It is a common phenomenon to encounter results in the final ranked list, having approximate URLs. We try to confront the problem by giving the user the ability to decide whether the number of pages with similar subdomains should be limited.

Finally, we present a unified method, the *Global KE Algorithm* that combines all three proposed algorithms into a single scoring formula. The *Global KE Algorithm* retains the personalization features of the other three, by giving the user the ability to modify its parameters.

The algorithms described in this paper have been implemented in a real world metasearch engine, *QuadSearch,* and they can all be found in the *Options* menu. *QuadSearch* implements other ranking algorithms as well (the original *KE Algorithm* and the *Borda Count* method) and also has a special section for searching scientific articles and authors. Moreover, it is capable of computing bibliometric indices and generating charts at query time [12].

2. The Original KE Algorithm

KE Algorithm on its original form is a score-based method. It exploits the ranking that a result receives by the component engines and the number of its appearances in the component engines' lists. All component engines are treated equally, as all of them are considered to be reliable. Each returned ranked item is assigned a score based on the following formula

$$W_{ke} = \frac{\sum_{i=1}^{m} r(i)}{n^m \left(\frac{k}{10}+1\right)^n} \qquad (1)$$

where $\sum_{i=1}^{m} r(i)$ is the sum of all rankings that the item has taken, *n* is the number of search engine top-k lists the item is listed in, *m* is the total number of search engines exploited and *k* is the total number of ranked items that the *KE Algorithm* uses from each search engine. According to the definition above, it is clear that the less weight a result scores the better ranking it receives.

3. The Geo KE Algorithm

By analyzing the result's URL someone may extract valuable information about the page hosted under it. The two or three final characters of the domain name usually declare the originating country of that page. We will refer to these two or three characters as the *domain's extension.*

Sometimes a user seeks for information that is directly linked to a specific region. For example, almost all queries related to traveling, are somehow connected to the travel's destination. A query for "*hotels in Paris"* explicitly limits the results to France and specifically, Paris. Undoubtedly, there is a significant possibility that pages hosted under *.fr* domain extension, could contain more valuable information than these hosted under other extensions.

But what happens when a hypothetical user from United Kingdom searches for *"hotels in Paris"*? The pages under *.fr* domains are usually written in French, so the hypothetical British user must be familiar to French to understand the provided information. In the opposite case, such result would be of no value to him; a page with *.uk* domain extension would probably be best result. Pages with *.us, .au* or *.ca* domain extensions would also be good choices, as there is a great possibility that they are all written in English.

Generally, there is strong evidence that a considerable number of search engine queries are geographically oriented [2]. Travel related searches are among the most popular on the Web [11]. Thus, developing a ranking algorithm that considers and combines the user's region and the locality of a Web page seams a challenging task.

The Geography Aware expansion of the *KE Algorithm* is designed for such occasions; it automatically receives the user's location and gives scores to the collected results according to the equation below:

$$W_{GAke} = G\,\frac{\sum_{i=1}^{m} r(i)}{n^m \left(\frac{k}{10}+1\right)^n} \qquad (2)$$

The new element added in (1), is the *GeoKE Coefficient G.* Once again, a result with lower score will achieve better ranking than a result with greater weight. The analysis of the result's domain name can lead to a variety of cases when compared to the user's locality. These cases are described thoroughly below.

Case 1: The domain extension of the result and the user's region are the same. A representative example of this case is when a number of results have the *.fr* domain extension and the user is from France. The value of *G* for these results is set equal to 2.

Case 2: The user can understand the language that the page is written. For example a page with *.pt* extension (Portugal) is probably a good result for a user from Brazil, because a Brazilian can understand Portuguese. A local database table is used to store the relationships between the geographical regions and their respective languages. These relationships are stored as {region - friendly region} pairs in our database. For each result, we search this table and if a relationship pair is found, the *GeoKE Coefficient* is set equal to 3.

Case 3: The domain extension of the page does not reveal any information about its locality. Examples of this case are the well known *.com*, *.net* and *.org* extensions. For these results we set *G=4*.

Case 4: When the page is written in a language that the user can't understand and does not belong to case 3, we set *G=5*.

In all four cases described above, we assigned fixed values to the *GeoKE Coefficient*, but our implementation in *QuadSearch*, gives the user the ability to define these values himself. This indicates the flexibility of our algorithm and enables its use in personalized search systems. By altering these values, the algorithm can lead to different rankings according to the user's selections.

The user's locality is automatically obtained by using a special database that matches IP addresses against geographical areas. By following this strategy, the user is not obliged to submit any information about his locality, but there is always a small possibility that the module makes an erroneous prediction.

Another different approach would be the analysis of the result's small context. This snippet could be a source of valuable information. However, this approach is unsafe; not all results have textual snippets. Additionally, even if a textual context is present, it is sometimes impossible for an automated system to understand the language it is written. For these reasons we avoided this approach.

4. The Weighted KE Algorithm

It is a common intuition, that a single search engine can't perform equally well for all types of queries. There are occasions where it can present qualitative results and others, in which the results are of medium or lower informational value.

For such occasions, an effective scoring algorithm must provide the user with the ability to modify the importance of each component engine. The proposed *Weighted KE Algorithm* takes this remark into consideration and assigns scores to the collected results according to the following equation:

$$W_{Wke} = \frac{\sum_{i=1}^{m} e(i) r(i)}{n^m \left(\frac{k}{10}+1\right)^n} \quad (3)$$

where $e(i)$ is the *Weight Factor of the ith Engine (EWF),* which can receive integer values between 1 and 10. The equation above leads to the same ranking as the original version of the algorithm, unless a different *EWF* is assigned to at least one engine.

An engine with greater importance than another, must receive a lower *EWF*. However, to make it more comprehensible for the user, our implementation in *QuadSearch* makes use of the *inversed EWF.* The use of *inversed EWF* indicates that an important engine must receive a greater *EWF,* hence (3) is being transformed to the following form:

$$W_{Wke} = \frac{\sum_{i=1}^{m} [11 - e(i)] r(i)}{n^m \left(\frac{k}{10}+1\right)^n} \quad (4)$$

5. The URL Aware KE Algorithm

In the past years, the phenomenon of Web pages appearing under subdomains of a *central domain name* has become very usual. In fact, subdomains are simple folders in a Web server's public directory and can contain pages with similar informational material. Of course, this observation is not absolute; pages under subdomains may have unique contents. Examples of such pages are the Departments of a University, or the personal pages of the academic staff of a faculty.

Until now, all search engines treat subdomains as different domains. The top-10 list that Google returns for the query *Aristotle University of Thessaloniki* contains 10 results, all having the term *auth.gr* in their domain name. It is obvious that the limitation of no more than two pages with the same domain name in the same result list does not cover subdomains. Thus, there is a danger that many pages with similar contents or many pages from one source would appear in the engine's result list. This danger is even greater in metasearch engines, where more than one component engines are being exploited.

The *URL Aware KE Algorithm* is designed to give the users the ability to define the value of Web pages having URLs that contain subdomains. The calculation of the weight factor for each Web page is performed with the use of the following formula:

$$W_{UAke} = (11 - D)\frac{\sum_{i=1}^{m} r(i)}{n^m \left(\frac{k}{10}+1\right)^n} \qquad (5)$$

where D is the *Domain Awareness Constant (DAC).* Higher values of *D* will decrease the result's weight, therefore improve its ranking. The value of *D* is determined by the following cases.

Case 1: The result has a domain name that is not repeated more than two times in the result list. The default value for *D* is 10.

Case 2: The result has a domain name that is repeated more than two times in the result list. The result appears under a subdomain of a *central domain* and *D=5*.

As with the *GeoKE Algorithm*, the implementation of the *URL Aware KE Algorithm* in *QuadSearch* allows the definition of the *DAC's* value by the user.

6. The Global KE Algorithm

In the previous subsections, we introduced three novel rank aggregation methods. One interesting feature of the proposed methods is that they can be combined to create a global scoring formula, the *Global KE Algorithm:*

$$W_{gke} = (11 - D)G\frac{\sum_{i=1}^{m} [11 - e(i)]r(i)}{n^m \left(\frac{k}{10}+1\right)^n} \qquad (6)$$

The proposed method gives the user the ability to:

1. Define the engine's importance for a single query. Not all search engines are treated equally for all types of queries.
2. Define how the geographic origin of a page and the language it is written, affect its ranking.
3. Define how the domain structure of a Web page affects its ranking.

7. Evaluation

In this subsection we evaluate the three introduced ranking fusion methods. One of the main advantages all algorithms have, is that they can be easily combined and create new scoring methods. A characteristic example of their combination is the *Global KE Algorithm* described in the previous Section.

Our implementation in *QuadSearch* gives the user the ability to define the value of the weight factors for a wide variety of cases. Thus, different users can get different results, or different rankings.

7.1. Time Complexity

None of the introduced algorithms harms the performance of the results classification process significantly. One might expect that the use of *GeoKE Algorithm* would result a slower query execution, as a connection to a database and numerous data transfers are required. In this occasion, keeping the appropriate data in the server's main memory eliminates this drawback. We tested all three algorithms' performances, and their combination (*Global KE Algorithm*), for various numbers of input results. The fact that no significant overhead comes out of their use, is made clear from the measurements recorded in the following table.

Table 1. The execution times of the proposed algorithms for variable numbers of input results.

Algorithm	40 results	80 results	120 results
Original KE	0.03	0.06	0.11
Weighted KE	0.03	0.06	0.11
GeoKE	0.03	0.06	0.13
URL Aware KE	0.03	0.06	0.11
Global KE	0.03	0.06	0.14

We have tested the efficiency of the proposed algorithms, for a variety of queries. The *Weighted KE Algorithm* modifies the final ranked list according to the selection of the weights for each component search engine. This is particularly useful for the special case that two results appear in only one search engine top-k list, but have the same ranking. The original *KE Algorithm* assigns equal weights to the results, but its weighted version also considers the engines' weights and ranks them respectively.

7.2. Precision Evaluation

The *GeoKE Algorithm* proved very useful for travel related queries and generally for cases where the quality of the presented results is affected by geographical information. For example, we compared the top-10 list returned by the *GeoKE Algorithm,* to the list returned by the original *KE Algorithm*, for the query *Athenian Acropolis.* The query is submitted by a hypothetical user located in Greece. The exploited result resources were the four major commercial search engines, Google, Yahoo, Live and Ask; all engines were considered to be of equal weight. The two top-10 lists returned by the two algorithms are shown in the following table.

Table 2. The top10 lists generated by *KE* and *GeoKE,* for the query *Athenian Acropolis.*

	Original KE Algorithm	GeoKE Algorithm
1	witcombe.sbc.edu	witcombe.sbc.edu
2	www.bluffton.edu	www.bluffton.edu
3	www.metrum.org	www.acropolisofathens.gr
4	people.hsc.edu	www.metrum.org
5	plato-dialogues.org	people.hsc.edu
6	www.wikitravel.org	plato-dialogues.org
7	en.wikipedia.org	www.wikitravel.org
8	www.acropolisofathens.gr	en.wikipedia.org
9	www.amazon.com	www.amazon.com
10	www.reconstructions.org	www.culture.gr

From the table above it becomes obvious that the *GeoKE Algorithm* assigns scores to the collected results with respect to the user's and the result's locality. The algorithm's scoring formula has given a higher rank to the Greek oriented result www.acropolisofathens.gr, than the original *KE Algorithm* did. Moreover, the top-10 list formed by the *GeoKE Algorithm* contains one more result with Greek origin, placed in the tenth position (www.culutre.gr).

8. Conclusions

In this paper we presented three innovative ranking fusion methods and their generalization, the *Global KE Algorithm* that derives from their combination. All three methods have been implemented and can be tested in QuadSearch, our experimental metasearch engine located in http://quadsearch.csd.auth.gr.

The introduced algorithms can be used to achieve better rankings according to the profile of an individual user. Different users usually have different informational needs even if they submit the same query terms. Moreover, the computational cost a ranking system suffers from their use is negligible. As a conclusion, any metasearch engine having a flexible personalized system, could capitalize on them

9. References

[1] L. Akritidis, G. Voutsakelis, D. Katsaros and P. Bozanis, "QuadSearch: A novel metasearch engine", *Current Trends in Informatics,* pp. 453-466, May 2007.

[2] C.D. Manning, P Raghavan and H. Schütze, "*Introduction to Information Retrieval*", Cambridge University Press, 2008.

[3] C. Dwork, R. Kumar, M. Naor and D. Sivakumar "Rank aggregation methods for the Web", *Proceedings of ACM Conference on World Wide Web (WWW),* pp. 613-622, 2001.

[4] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar and E Vee "Comparing partial rankings", *Proceedings of ACM Symposium on Principles of Database Systems (PODS)*, pp. 47-58, 2004.

[5] R. Fagin, R. Kumar and D. Sivakumar "Comparing top k lists" *SIAM Journal on Discrete Mathematics,* vol. 17, no. 1, pp. 134-160, 2003.

[6] H. C. Lee, H. Liu and R. J. Miller "Geographically sensitive link analysis", *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence (WI),* pp. 628-634, 2007.

[7] O. Buyukkokten, J. Cho, H. Garcia-Molina, L. Gravano and N. Shivakumar, "Exploiting geographical location information of Web pages", *Proceedings of WebDB Workshop,* pp. 91-96, 1999.

[8] L. Wang, C. Wang, X. Xie, J. Forman, Y. Lu, W.-Y. Ma, and Y. Li. "Detecting dominant locations from search queries", *Proceedings of ACM International Conference on Research and Development in Information Retrieval (SIGIR),* pp. 424-431, 2005.

[9] T.H. Haveliwala, "Topic-sensitive PageRank", *Proceedings of ACM Conference on World Wide Web (WWW),* pp.517-526, 2002.

[10] Open Directory Project. http://dmoz.org.

[11] E. Amitay, N. Har'El, R.Sivan, and A. Soffer, "Web-a-where: Geotagging Web content". *Proceedings of ACM International Conference on Research and Development in Information Retrieval (SIGIR),* pp. 273-280, 2004.

[12] D. Katsaros, L. Akritidis and P. Bozanis, "Spam: It's not just for inboxes and search engines! Making Hirsch h-index robust to scientospam", *arxiv.org*, 2008.

[13] YY Chen, T Suel and A Markowetz, "Efficient query processing in geographic Web search engines", *Proceedings of ACM international conference on Research and Development in Information Retrieval (SIGIR),* pp. 277-288, 2006.