The Journal of Systems and Software 84 (2011) 130–143

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Effective rank aggregation for metasearching

Leonidas Akritidis, Dimitrios Katsaros*, Panayiotis Bozanis

Department of Computer and Communication Engineering, Univarsity of Thessaly, Glavani 37, Volos 38221, Greece

ARTICLE INFO

Article history:
Received 6 May 2010
Received in revised form 30 July 2010
Accepted 1 September 2010
Available online 22 September 2010

Keywords:
Ranking
Rank aggregation
Rank fusion
Metasearch
Borda Count
Search engines
Information search
Information retrieval
Web

ABSTRACT

Nowadays, mashup services and especially metasearch engines play an increasingly important role on the Web. Most of users use them directly or indirectly to access and aggregate information from more than one data sources. Similarly to the rest of the search systems, the effectiveness of a metasearch engine is mainly determined by the quality of the results it returns in response to user queries. Since these services do not maintain their own document index, they exploit multiple search engines using a rank aggregation method in order to classify the collected results. However, the rank aggregation methods which have been proposed until now, utilize a very limited set of parameters regarding these results, such as the total number of the exploited resources and the rankings they receive from each individual resource. In this paper we present *QuadRank*, a new rank aggregation method, which takes into consideration additional information regarding the query terms, the collected results and the data correlated to each of these results (title, textual snippet, URL, individual ranking and others). We have implemented and tested *QuadRank* in a real-world metasearch engine, *QuadSearch*, a system developed as a testbed for algorithms related to the wide problem of metasearching. The name *QuadSearch* is related to the current number of the exploited engines (four). We have exhaustively tested *QuadRank* for both effectiveness and efficiency in the real-world search environment of *QuadSearch* and also, using a task from the recent TREC-2009 conference. The results we present in our experiments reveal that in most cases *QuadRank* outperformed all component engines, another metasearch engine (Dogpile) and two successful rank aggregation methods, Borda Count and the Outranking Approach.

1. Introduction

The lack of any specific structure and the vast amount of information published on the Web, makes it extremely difficult for the user to find the information s/he desires without any external help. As of February 2010, there are at least 19 general-purpose search engines[1], as well as numerous special-purpose search engines. Their population is mainly justified by two reasons: (a) no ranking algorithm is broadly acceptable, although many users tend to consider *Google*'s ranking method as the most successful; (b) no engine can achieve large coverage and high scalability. It is a common belief (Sugiura and Etzioni, 2000; Manning et al., 2008) that a single general purpose search engine for all Web data is unrealistic, since its processing power cannot scale up to the rapidly increasing and unlimited amount of Web data.

The tool which rapidly gains acceptance among the users is *metasearch engines* (Meng et al., 2002). These systems operate like a filter of the various crawler-based or directory-based search engines which they combine. Metasearch engines run simultaneously a user query across multiple *component search engines*, retrieve the generated results and then aggregate them. Finally, they present the best among them to the user.

The advantages of metasearch engines against search engines are significant (Meng et al., 2002):

- They increase the search coverage of the Web, providing a higher *recall*. The overlap among the major search engines is usually very small (Spink et al., 2006) and it can be as small as 3% of the total results retrieved. On the other hand, the unique results can be as high as 85% of the total results retrieved by all component engines.
- They solve the scalability problem of searching the Web and they facilitate the exploitation of multiple search engines enabling consistency checking (Aslam and Montague, 2001a).
- They improve the retrieval effectiveness providing higher precision, due to the "chorus effect" (Vogt, 1999).

Consequently, metasearch engines and their Web 2.0 successors, mash-up services are important tools and they are becoming increasingly popular. The core of any such system is the ranking function it employs, because this function defines the final ranked

* Corresponding author.
E-mail address: dkatsar@inf.uth.gr (D. Katsaros).
[1] See http://www.searchenginewatch.com.

0164-1212/$ – see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2010.09.001

result list from the results provided by the component search engines. Hence, finding effective ranking algorithms is a problem of critical significance for metasearch engines and mash-up services.

The problem of rank aggregation is quite old and has been studied for a century, starting from a need to design fair elections. It can be thought of as the unsupervised analog to regression, with the goal of discovering a combined ranking which minimizes the distance to each individual ranking. Despite its seeming simplicity it is surprisingly complicated; finding the optimal combined ranking is NP-hard (Dwork et al., 2001) under certain conditions. Thus, several recent efforts describe approximation algorithms for the rank aggregation problem (Ailon et al., 2005; Ailon, 2007; Coppersmith et al., 2006), after showing its relation to the *feedback arc set problem on tournaments* (Ailon et al., 2005). Some of these are extensively applied to many different research domains, such as bioinformatics (DeConde et al., 2006), Web spam detection (Dwork et al., 2001), pattern ordering (Tan and Jin, 2004), metasearching (Liu et al., 2007; Renda and Straccia, 2003; Sculley, 2007; Shokouhi, 2007; Oztekin et al., 2002) and many more.

Web metasearching in contrast to rank aggregation, is a problem posing its own unique challenges. The results that a metasearch system collects from its component engines, are not similar to votes or any other single dimensional entities: Apart from the individual ranking it is assigned by a component engine, a Web result also includes a title, a small fragment of text which indicates its relevance to the submitted query (textual snippet) and a uniform resource locator (URL). Apparently, the traditional rank aggregation methods are insufficient for providing a robust ranking mechanism suitable for metasearch engines, because they ignore the semantics accompanying each Web result.

Based on these remarks, we conclude that ranking in Web metasearching is a more complex problem than rank aggregation. Individual rankings might be noisy, incomplete or even disjoint, hence they should not be the only parameter affecting ranking. Further processing is required in order to filter the results and allow the final result list of the metasearch engine to be free of unwanted, devious and unfairly highly ranked Web pages. Since commercial interests might frequently and unpredictably affect the results of searching, the user is not clearly protected against the interests of individual search engines. Therefore, the ranking algorithm employed by a real metasearch engine, should be able to provide results that are as free as they can be from paid listings and links.

In this paper we propose *QuadRank*, a new rank aggregation method suitable for metasearch engines. QuadRank is a positional ranking method designed to deal with top-k lists returned by web search engines. Its main features are:

- It assigns scores to the candidate results by considering multiple parameters such as the number of the search engines where a particular item appeared, the total number of exploited search engines, the size of the top-k list returned by each search engine, the number of the occurrences of the query terms in each document, term proximity, zone scoring and others.
- It refrains from using any training data in order to perform the rank aggregation, because, there is usually no evidence about the underlying data properties and their distributions.
- It does not count upon the *scores* of the individual search engine rankings in order to perform the rank aggregation, because, most of the search engines do not provide such scores.

The new algorithm is evaluated on real world data drawn from four major search engines against individual search engines listings as well as results returned by metasearch engine Dogpile[2], using QuadSearch[3], a metasearch engine developed, among others, as a testbed for rank fusion. There is also an independent performance study of metasearch engines (Allen, 2009), comparing QuadSearch, Dogpile and Mamma, which showed that QuadSearch was the best of the three for that (limited) query load.

We also compare our proposed method to two other existing rank aggregation methods. The first is the well-established *Borda Count* method which assigns scores to the collected documents, by accumulating the individual rankings they received by the component engines. The second method is the *Outranking Approach*, an order-based method presented in Farah and Vanderpooten (2007), which orders the items by specifying a set of thresholds and by comparing each document with all the other collected documents. You can see Section 2 for a brief description of these two methods and a discussion on their differences from our proposed algorithm.

Initially, we test these methods by utilizing the results from the Web Adhoc task of the Web Track of the TREC-2009 Conference (Soboroff et al., 2009). In the sequel, we report the performance of the examined methods in the real-world environment of QuadSearch.

The rest of this article is organized as follows: in Section 2 we provide some necessary background material and survey the relevant rank aggregation methods. In Section 3, which presents the main article ideas, we describe the new rank aggregation method and the implementation issues behind the developed metasearch engine. In Section 4 we present an evaluation of the proposed method, and finally, Section 5 concludes the work.

2. Preliminaries and relevant rank aggregation methods

We start with a universe U of items (documents in the context of metasearching); each item has a unique identifier c. A ranked list r of items $c_1, c_2, \ldots, c_n$ drawn from the universe U, is an ordered subset $S \subseteq U$, such that $r = [c_1 \geq c_2 \geq \cdots \geq c_n]$, where $\geq$ is an ordering relation on S. Each item $c \in S$, has the attribute $r(c)$ which represents the ranking of c in list r. Rankings are always positive, the best ranking an item could get is 1, and higher ranks show lower preference (reduced relevance to a query, in the context of metasearching).

If r contains all the items of U, then it is said to be a *full or complete list*; if $|r| < |U|$, then it is said to be a *partial list*, and if $|r| = k$, where k is a fixed constant, it is said to be a *top-k list*. Apparently, a top-k list is a special case of a partial list. The ideal scenario for rank aggregation is when each search engine gives a complete list of all the items of the universe related to the keyword terms of a given query. Unfortunately this is not possible since either each component engine has a partial coverage of the Web, or for reasons of speed or protection of the proprietary ranking algorithms, the engine returns only a top-k list. The worst but unusual scenario is when the result lists of component search engines have no overlapping elements. In this case there is nothing that a standard rank aggregation algorithm can do. However, as we will see later, QuadRank takes into account the metadata accompanying each item, in addition to the individual rankings of the search engines and this is an advantage of our method over the other methods.

Two families of rank aggregation techniques exist (Renda and Straccia, 2003): (a) the *score-based* policies (Vogt and Cottrell, 1999), which assign a score to each entity of the individual ranking lists and then use these scores to perform the ranking, and (b) the *order-based* (or *rank-based*) policies (Dwork et al., 2001; Sculley, 2007; Beg and Ahmad, 2003), which work upon the order

[2] http://www.dogpile.com.

[3] A publicly accessible prototype of QuadSearch is available under http://quadsearch.csd.auth.gr.