CATEGORY-BASED RANKING OF FEDERATED PRODUCT OFFERS

Maximilian Walther, Daniel Schuster, Till Juchheim, Alexander Schill
Technische Universität Dresden, Faculty of Computer Science, Institute of Systems Architecture
Helmholtzstr. 10, 01062 Dresden, Germany
{maximilian.walther, daniel.schuster, till.juchheim, alexander.schill}@tu-dresden.de

ABSTRACT

During the last years the World Wide Web has become one of the central sites for consumer product search. Due to the wide range of online malls offering products from different categories in heterogeneous ways, finding the most adequate product is a time-consuming process. Our current research work proposes mechanisms for the federated ranking of such search results. The developed algorithms are able to automatically classify vendors regarding different product areas, categorize consumers' search queries and consequently only request information from relevant sources. Based on the previous calculations, product hits are ranked and presented to the user. Evaluations show the feasibility of the approach.

KEYWORDS

Information Retrieval, Information Federation, Semantic Web, Information Extraction.

INTRODUCTION

Since people started using the Internet as a product search platform, producers and retailers in particular started to take advantage of this development by presenting comprehensive sets of product information on the Web. The sheer number of shops and product information sources and their heterogeneous ways of product presentation as well as differing quality standards complicate the process of product information search and comparison for the average Internet user and create an annoying and longsome experience.

Federated product search offers a way out of this dilemma. Different shops provide their product information to portals such as Google Product Search (2009) for reaching additional customers. For example, if searching for a product with a query like "Canon 450D", a result list is generated containing hundreds of variations of the product mixed with accessories. The best-ranked links reference new lists of hundreds of shops ranked by price and user shop ratings. Most of the shops are not known to the user, so he is likely to end his search at this point, choose the best-ranked shop or the first shop he knows out of the list.

The sketched example shows that good ranking and filtering strategies are essential for federated product search. Unfortunately, federated ranking mechanisms are hard to realize, as there is no objective measurement like the link structure for Web pages available. Neither PageRank, developed by Page et al (1998), nor any kind of content-based ranking is applicable to solve this problem. The situation is even worse if the search string does not specify a special product, but consists of general terms such as "digital camera" which disables a federated product search system from recognizing a concrete product. Additionally, federated search providers generally do not have information about sales numbers or customer satisfaction values for products or shops.

In this paper we present an alternative view on the problem of federated product ranking. Our approach of category-based ranking tries to copy the way a user with expert knowledge in the current product domain would select shops and order product offers from these shops. The developed method executes the federation and ranking mechanisms in real-time. The query is mapped to a product category (e.g., Electronics) and the available shops are ranked according to their "competence" in the product domain. Some shops may be specialists for electronics, others for books, or may have a good reputation as general-purpose stores. Only the best-ranked shops actually receive the query and return individual result lists.

Before outlining details on the ranking algorithm, requirements for the federated product ranking are to be presented. The most obvious requirement is that an automatically generated product ranking has to correspond to an ordered list created by a human being having expert knowledge in the respective domain, when assigned to the same task. Of course the automated approach should complete in less time and with a larger amount of information taken into account. Further requirements are presented in Listing 1.

Listing 1. Requirements for federated ranking

1. Easy to understand (consumer may retrace, why a product appears higher in the list than another)
2. Well balanced (scores are calculated by overall relevance, not just single features)
3. Diverse (single sources or products are not preferred in comparison to others)
4. Repeatable (results are always scored the same way, resulting in the same outcome)
5. Scalable (scores are easy to compute to enable a real-time solution)
6. Comprehensive (all available sources are included)

We review related work in the area of federated ranking according to these requirements in the next section. Our method of category-based ranking is introduced in Section 3. We implemented category-based ranking within the Fedseeko system, a federated product information system based on Ruby on Rails (Section 4). Evaluation results in Section 5 show the feasibility of our approach.

FEDERATED RANKING

In this section various existing techniques for the federated ranking of search results will be examined. As mentioned in the introduction, ranking in such systems differs heavily from standard approaches in centralized web search engines, mainly caused by the absence of a linking structure and missing rich information on single results, as on the one hand sources do not expose their entire knowledge to the public and on the other hand requesting detailed information on single results may be too time-consuming.

A good introduction on federated search including some existing federated search systems is given in Fyer (2004). Luo and Callan (2005) present an approach for modeling the retrieval effectiveness of search engines in a federated search environment. The developed algorithm estimates the quality of such search engines by generating result lists through these engines using example queries and comparing them with lists generated by an effective centralized retrieval algorithm on the returned documents. Paltoglou et al (2007) covered the problem of hybrid results merging. The authors point out that two kinds of methods for merging results from different information sources had been analyzed so long. On the one hand approaches had been developed that are solely based on the sources' underlying ranking mechanisms which leads to an insufficient ranking quality. The second type are methods based on the exhaustive evaluation of documents which cause high costs in computation time and traffic. As a solution they propose a model approach that only relies on partial source evaluation and strong estimation of results. The Federated Search portal Science.gov (2009) can be viewed as another model approach. The engine searches millions of documents from multiple U.S agencies hosted on various different servers. Since a single query may return thousands of hits, a decent ranking method is indispensable. Over the years three different real-time ranking methods were designed for Science.gov: QuickRank, MetaRank and DeepRank. QuickRank calculates a document score based on the number of occurrences of the search terms in the document by evaluating the document's title and text snippets. MetaRank differs from QuickRank by assuming that documents within the sources' databases supply an amount of meta-data about their contents that is then evaluated by the ranking method. Currently, DeepRank is the most thorough and exhaustive method and was introduced in 2007. It evaluates the full text of documents to produce the most accurate results. The FedLemur project, presented by Avrahami et al (2006), is a federated search engine specifically designed for providing information to the Fedstats.gov website, to compare the overall performance of a single-database solution with a federated approach. The engine is based on the Lemur toolkit (2009) for language modeling and retrieval. Inter alia, the presented approach ranks integrated search engines by creating a word histogram for each engine. The histograms are built by querying the search engines with random queries and analyzing the first returned documents. Thus, specific domains for the search engines are described which can be used at query-time.

The presented approaches are only hardly applicable to federated product search systems. Information attached to products is not suitable for full-text analysis and result sets originating from different sources have shown to be very heterogeneous which makes the consolidation of results from different sources a complex task that might not fulfill the real-time requirements entailed by the federated search paradigm.

In the field of federated shopping, federated ranking mechanisms have not yet been examined in detail. Most portals rather stick to sorting results by their price or the similarity of query and product titles, since their main goal is to aid a user in finding cheap and reliable vendors instead of providing valuable product information. Lately some of them have also started to include consumer ratings to improve results. But especially when searching for rather unspecific product names (product lines, categories, etc.), the performance of existing systems still tends to be quite low.

In the academic research the shopinfo.xml standard was developed by Wolter et al (2006) which sorts search results by characteristics connected to an immediate purchase (price, shipping time etc.), while not considering the relevance of each result for the given query. The ranking is only determined by underlying sources. The IPIS system (Kim et al (2005)) uses Web Service interfaces for querying different vendors and ontologies for mapping a semantic product query to a special product category. As most vendors do not provide Web Services yet, this approach is hardly applicable in the current Internet's infrastructure. Additional research in this area includes Lim et al (2009), Park and Kim (2007) and Pu and Kumar (2004).

Our method will combine ideas of the presented works and eliminate their shortcomings.

CATEGORY-BASED RANKING

Since there are no major standards for the design of a method to rank product search results obtained from different online malls, our method shall be inspired by the way human beings perform federated shopping tasks on the Web. The model to be used throughout this paper is called "The Federated Shopper" and consists of three steps that were observed by consulting multiple users (Listing 2).

Listing 2. Product search process executed by the Federated Shopper

Precondition:	Gain experience which shops are best qualified for particular product categories.
Step 1:	Detect the category of the product to be searched.
Step 2:	Choose shops that are suitable for the corresponding category.
Step 3:	Query shops for the product and create a ranked result list taking the shops' internal rankings as well as their suitability for the current category into account.

The algorithm is based on estimated scores that are assigned to each product hit from different vendors depending on the relevance of the corresponding vendor for the product's category as well as the vendor's internal ranking for the considered product. Each vendor's relevance scores for the different categories need to be calculated initially. Before executing the calculation, several sample lists of products from different categories have to be created. Then, for every tied vendor it needs to be checked which fraction of these products is contained in the underlying database. The estimated vendor score for the product's category is merged with the vendor's internal ranking score of each product to calculate a total ranking for all products. Thus, a result page still retains the ranking order of the vendors, but tends to rank products of vendors that are more relevant for the current category higher. The respective steps are explained in-depth below.

3.1 Product Sampling

To evaluate each vendor's relevance for a certain product category the system samples the source with a representative set of product titles. This short section shall be concerned with the creation of such sets.

To receive precise estimations of a vendor's relevance for a certain product category, it is obviously necessary that the product list that is used for sampling each source contains products that clearly shape the market segments. Inaccurate or unbalanced sets may easily lead to overspecialization in certain areas (e.g. a sample set for “Books” only containing titles on sports) which may cause the vendor's relevance estimations in those areas to be incorrect.

Different methods to create the sample sets have been evaluated. At first the sets were created using expert knowledge in certain market domains. This manual method has proven to be a very time consuming task, but showed good results in evaluation. Subsequently, a random sampling algorithm was developed to automatically query several shopping portals using keyword lists. This method showed some difficulties, as the automatic creation of word lists being representative for every domain involves additional research work.

The finally chosen method creates product title sets by crawling the different categories of big shopping portals like Amazon.com. For every category the crawler selects a random page, followed by the selection of a random product and the extraction of its title. This process continues until a suitable set has been created. As the employment of only one big vendor for the category crawling might produce biased product sets, a clean product sampling procedure should include the usage of several online malls.

3.2 Source Ranking

As mentioned before, each vendor is assigned a relevance score for each category known to the system. This is done by sampling the vendors with representative sets of product names for every category (Figure 1). The sampling and scoring process has to be done initially once for every vendor to be included in the federated search system. The total product ranking uses these scores during runtime.

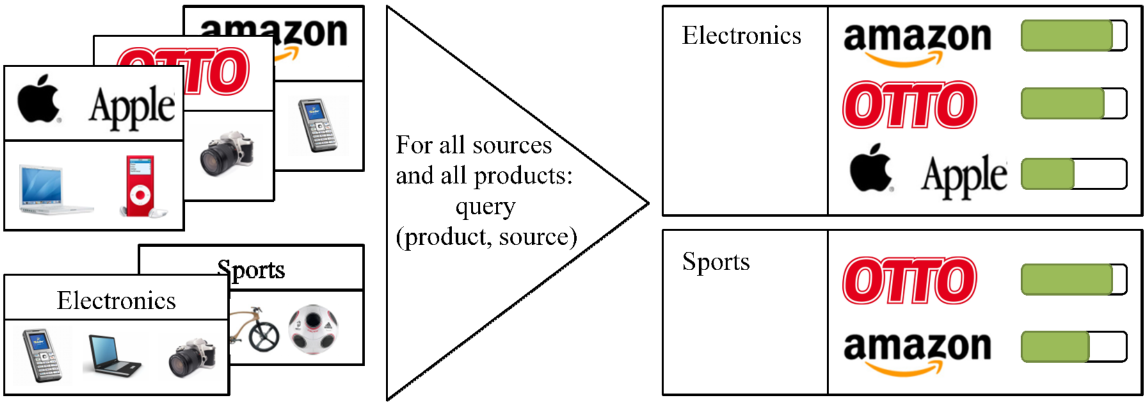


Figure 1. Source ranking algorithm

Every source returns the total number of hits for each category. While this could already be viewed as an estimation of its relevance, it does though not fully reflect the model of the Federated Shopper. The formulas for calculating a source’s relevance are given below.

$$\mathrm{products(source)} = \sum\nolimits_{t=1}^{n} \mathrm{hits}(\mathrm{category}_t, \mathrm{source}) \tag{1}$$

$$\mathrm{relevance(source, category)} = \frac{\mathrm{hits(category,source)}^2}{\mathrm{products(source)}^2} \tag{2}$$

As shopping portals generally do not provide information about the number of products contained in their catalogue, we define the number of products available from one source as the sum of hits for all queries of all categories sent to one source (1). The squared quotient of the products found for one category and the number of products available from this source results in the relevance of the source for the considered category (2). Dividing the hits count by the total number of products estimates the degree of specialization of a source’s catalog for the current category. This causes highly specialized stores to be scored better than stores covering many different categories. Analyses showed that squaring the quotient results in better overall ranking results.

3.3 Product Categorization

The developed approach needs to know the category that a queried product belongs to, so that it can pick the appropriate sources to forward the query and choose the correct relevance estimations from the database. Since there are already large sets of classified product data available on the Web represented by online shops and portals that offer products and sort them into categories, we created an algorithm to categorize a product by passing the query to these sources and using a majority vote on the actual product category. In the current implementation the system queries two vendors, Amazon and EBay, matches their category names onto the system’s internal categories and then picks the category with the most hits for the current query. Those two vendors proved good performance in categorizing product queries. However, additional vendors may be integrated easily for meliorating the categorization process. The matching of vendor site category names to the system's internal categories is done by using the Levenshtein distance as well as pattern matching.

As the classification of a single query may require multiple requests to vendor services, the system also has a concept for caching classifications and offline inferences of a query’s category. For each query passed to the system it classifies the query's substrings and stores the results in a local database. When the system is addressed with a new query, it first checks the database for already classified substrings of the query. In case the amount of previously classified substrings exceeds a certain threshold, the category is inferred from the database. This method takes advantage of the similar naming within product lines and drastically reduces requests to vendor services after an initial training period.

3.4 Product Ranking

Finally, the actual scoring and ranking of product entries in a result set is executed (Figure 2). First, the provided query is classified. The system uses the method explained in the previous section to assign a category name to a query. This information is then used to select vendors that are expected to return the best results by calculating the ranking values of the product search hits they would return.

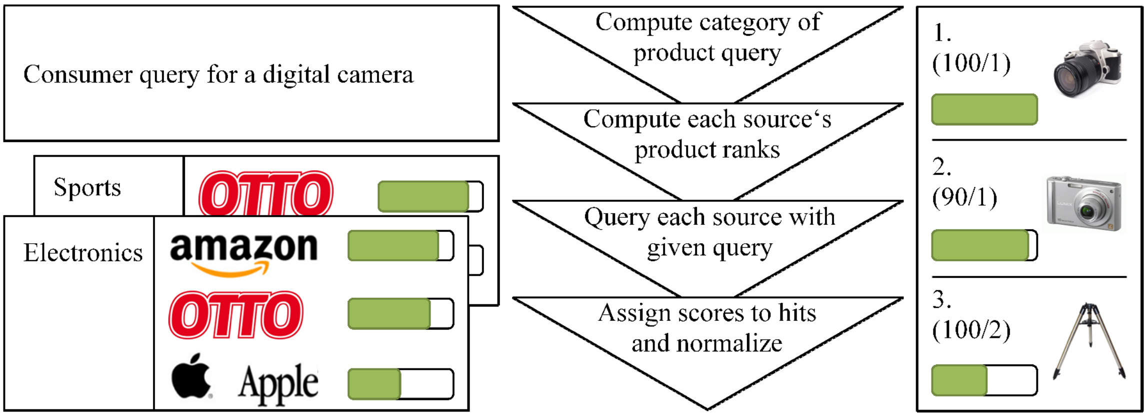

Consumer query for a digital camera
Sports
OTTO
Electronics
amazon
OTTO
Apple
Compute category of product query
Compute each source‘s product ranks
Query each source with given query
Assign scores to hits and normalize
1.
(100/1)
2.
(90/1)
3.
(100/2)

Figure 2. Product Ranking Algorithm

Finally, the chosen vendors are queried for their result lists which are merged using the previously calculated ranking values. The scoring value is calculated as shown below.

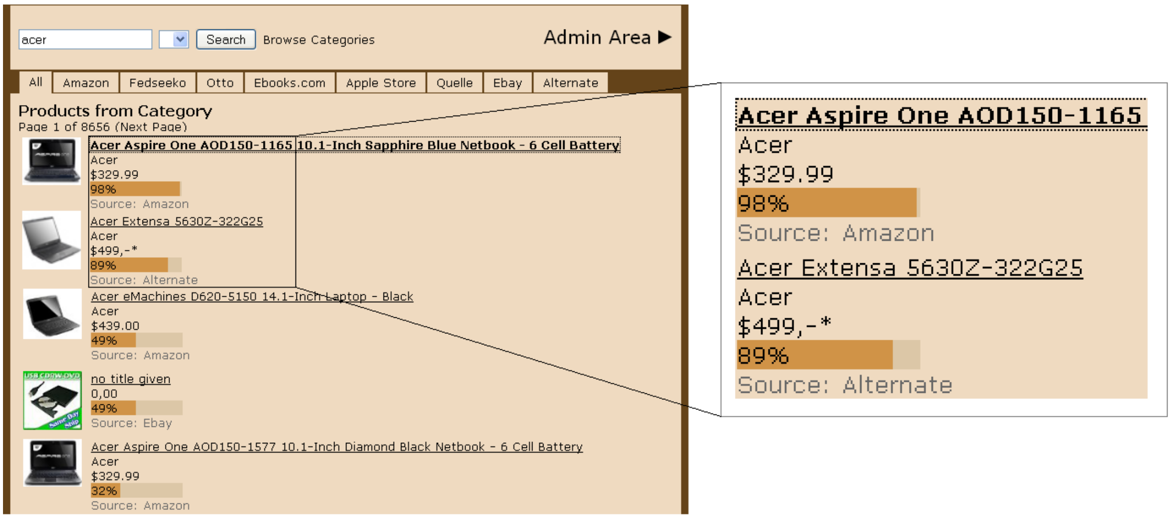
$$\text{score}(\text{hit}, \text{source}, \text{category}) = \frac{\text{relevance(source,category)}}{\text{position(hit,source)}} \tag{3}$$

The score is made up of two components: the initially calculated relevance of a source for the entry's category and the score that the source originally assigned to that entry which is assumed to be equal to the entry's position in the source's result list. Therefore, product search hits originating from different vendors are arranged corresponding to the sources’ relevance estimations, while preserving the order of entries from each vendor respectively. In our implementation the formula also includes additional factors for the normalization of scores. In each result set the highest ranking entry is scored with 100%, putting all other entry scores in relation to this one. Thus, 100% relevance means that for the queried category the corresponding shop is the best online mall currently known to the system. The normalization has been left out in the formula.

IMPLEMENTATION IN FEDSEEKO

The ranking is implemented as part of the Fedseeko system, a federated product information portal. The system is described explicitly in Schuster et al (2008) and Walther et al (2009).

Fedseeko is a Ruby on Rails application being executed with JRuby which allows the comfortable integration of Java code in the system. The ranking algorithm has been implemented as a Java Library.



Products from Category
Page 1 of 8656 (Next Page)

Acer Aspire One AOD150-1165 10.1-Inch Sapphire Blue Netbook - 6 Cell Battery
Acer
$329.99
98%
Source: Amazon

Acer Extensa 5630Z-322G25
Acer
$499,-*
89%
Source: Alternate

Acer eMachines D620-5150 14.1-Inch Laptop - Black
Acer
$439.00
49%
Source: Amazon

no title given
0,00
49%
Source: Ebay

Acer Aspire One AOD150-1577 10.1-Inch Diamond Black Netbook - 6 Cell Battery
Acer
$329.99
32%
Source: Amazon

Acer Aspire One AOD150-1165
Acer
$329.99
98%
Source: Amazon
Acer Extensa 5630Z-322G25
Acer
$499,-*
89%
Source: Alternate

Figure 3. Screenshot of the Fedseeko implementation

The calculated ranking is visualized in Fedseeko using ordered result lists, as well as little bars displaying each entry's score in relation to the best possible result (Figure 3). Vendor information providers can be accessed in Fedseeko using Web scraping technologies, thus enabling the inclusion of every existing vendor.

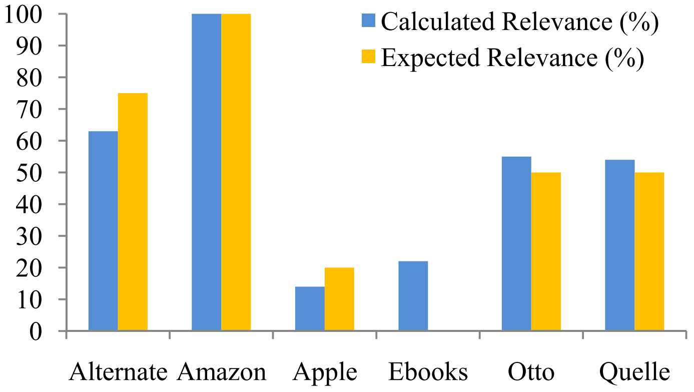
EVALUATION

This section gives a brief summary on the evaluation of the ranking's quality. The complete evaluation was split in two parts. The first one measures the accuracy of the category relevance estimation of the sources, while the second one evaluates the quality of the of search results being ranked with the complete algorithm.

Since there are no reference systems or standards to compare to we created our own gold standard. For both parts reference data was created by examining the market segments and shopping portals manually using a fixed and objective procedure. Thus, integrated lists consisting of products from different online shops were created for queries from varying categories which were ranked by human beings. The most important criteria we measured was the value for precision which is presented for the examined categories in the first part of the evaluation.

5.1 Evaluation of Source Ranking

To evaluate the quality of the source relevance estimations we chose eight product categories and six different sources that serve different market segments and originate from different countries. For each category we measured the relevance value calculated using our method and the expected relevance given by the gold standard. Figure 4 shows the evaluation for the category "Electronics".



Store	Calculated Relevance	Expected Relevance
Alternate	63%	75%
Amazon	100%	100%
Apple	14%	20%
Ebooks	22%	0%
Otto	55%	50%
Quelle	54%	50%
Ave. Diff.	8.17	
Precision	83%	

Figure 4. Evaluation of the algorithm for the "Electronics" category

Apparently, Ebooks received a higher relevance for electronic products than Apple. As Apple's product portfolio is extremely specialized on a small set of items and Ebooks offers a lot of print products dealing with electronic manufactures (e.g. books on how to use an iPhone), the calculated relevance scores fall into place. The table on the right side additionally presents the value for the average difference in percentage points (8.17), as well as the average precision. Although the precision is a very coarse measure, it is still as high as 83% in this example. The recall is not printed in this figure as it is always 100%.

The overall evaluation of the precision for all eight categories using six vendors is presented in Figure 5.

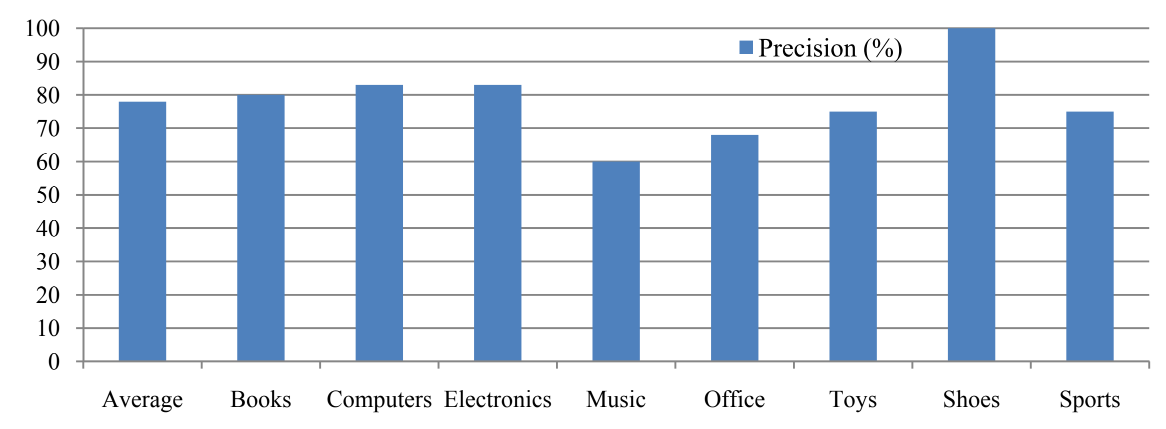


Figure 5. Precision in different categories

It shows that the overall precision is slightly below 80%. Interestingly, the system performed much better in some categories than in others. We have been able to trace this to differences in the product sets contained by vendors from different countries. But even with these slight variances the overall performance remains very good and serves as a sound foundation for the actual product ranking.

5.2 Evaluation of Product Ranking

For being able to evaluate the system's ranking mechanism, the gold standard was extended with ordered lists of products for each category. The lists corresponded to representative sample queries. Subsequently, the federated search system was executed with these queries to create ranked product lists. Then we used the top-k precisions as quality characteristics, i.e. we measured how many of the top-k results from the reference lists also showed up in Fedseeko's result lists.

Within the top-15 results of this set, meaning the 15 most important results for the consumer, the precision is around 93% while the precision within the top-3 entries reaches 98%. Evaluations with other categories showed that our method never falls below 66% for the top-15 precision.

CONCLUSIONS

In this paper we presented a new algorithm for ranking results in a federated search environment that strongly relies on an initial source evaluation. Originally, it was developed for the federated product information system Fedseeko, but may very likely be used in other federated search environments as well. Main requirements are that the queries can be classified in categories and the quality of hits in a certain search category is strongly connected to the source that they originate from.

The main advantages of this method are the low computation requirements during runtime and the possibility to rank all types of entries from heterogeneous sources since individual entries are not examined theirselves. However, this last advantage is probably also its biggest drawback. The ranking tries to estimate the quality of information a source can provide for a certain product category but cannot detect or correct estimation errors in a ranked list of results during runtime. Additionally, the integration of a new vendor requires the system to execute the source ranking algorithm before being able to include the vendor in a reasonable way. Depending on the vendor's response time, this process might be very time-consuming.

Our evaluation proved the feasibility of the approach. The source estimations in our sample runs showed that the average difference to the gold standard was around 10%. For a completely ranked result list the top-15 precisions ranged from 66% - 93%. Multiple topics that could meliorate the ranking are currently being discussed, including ideas to improve the ranking by examining individual results during runtime and possibilities to cluster similar results into a single list entry.

REFERENCES

Avrahami, T.T. et al, 2006. The FedLemur Project: Federated Search in the Real World. *Journal of the American Society for Information Science and Technology*, Vol. 57, No. 3, pp. 347-358.

Fyer, D., 2004. Federated Search Engines. *Online*, Vol. 28, No. 2, pp. 16-20.

Google Inc., 2009. Google Product Search. *http://www.google.com/products*, Mountain View, USA.

Kim, W. et al, 2005. Product Information Meta-search Framework for Electronic Commerce Through Ontology Mapping. *2nd European Semantic Web Conference*, Heraklion, Greece.

The Lemur Project, 2009. The Lemur Toolkit for Language Modeling and Information Retrieval. *http://www.lemurproject.org*, Massachusetts, USA.

Lim, S.C.J. et al, 2009. Faceted Search and Retrieval Based on Semantically Annotated Product Family Ontology. Exploiting Semantic Annotations in Information Retrieval, Barcelona, Spain.

Luo, S. and Callan, J., 2005. Modeling Search Engine Effectiveness for Federated Search. *28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Salvador, Brazil, pp. 83 – 90.

Page, L. et al, 1998. The PageRank Citation Ranking: Bringing Order to the Web, *Stanford Digital Library Technologies*, Palo Alto, USA.

Paltaglou, G. et al, 2007. Hybrid Results Merging. *16th ACM Conference on Information and Knowledge Management*. Lisbon, Portugal, pp. 321-330.

Park, S., Kim, W., 2007. Ontology Mapping Between Heterogeneous Product Taxonomies in an Electronic Commerce Environment. *International Journal of Electronic Commerce*, Vol. 12, No. 12, pp. 69-87.

Pu, P.H.Z., Kumar, P., 2004. Evaluating Example-based Search Tools. *5th ACM Conference on Electronic Commerce*. New York, USA, pp. 208-217.

Schuster, D. et al, 2008. Towards Federated Consumer Product Search from Heterogeneous Sources. *IADIS International Conference WWW/Internet*. Freiburg, Germany, pp. 453-456.

Science.gov Alliance, 2009. Science.gov. *http://www.science.gov*, Oak Ridge, USA.

Walther, M. et al, 2009. Federated Product Search with Information Enrichment using Heterogeneous Sources. *12th International Conference on Business Information Systems*. Poznan, Poland, pp. 73-84.

Wolter, T. et al, 2006. Der shopinfo.xml-Standard. *XML-Tage*. Berlin, Germany.