Using ODP Metadata to Personalize Search

Paul - Alexandru Chirita
L3S and University of Hannover
Deutscher Pavillon Expo Plaza 1
30539 Hannover, Germany
chirita@l3s.de

Wolfgang Nejdl
L3S and University of Hannover
Deutscher Pavillon Expo Plaza 1
30539 Hannover, Germany
nejdl@l3s.de

Raluca Paiu
L3S and University of Hannover
Deutscher Pavillon Expo Plaza 1
30539 Hannover, Germany
paiu@l3s.de

Christian Kohlschütter
L3S and University of Hannover
Deutscher Pavillon Expo Plaza 1
30539 Hannover, Germany
kohlschuetter@l3s.de

ABSTRACT

The Open Directory Project is clearly one of the largest collaborative efforts to manually annotate web pages. This effort involves over 65,000 editors and resulted in metadata specifying topic and importance for more than 4 million web pages. Still, given that this number is just about 0.05 percent of the Web pages indexed by Google, is this effort enough to make a difference? In this paper we discuss how these metadata can be exploited to achieve high quality personalized web search. First, we address this by introducing an additional criterion for web page ranking, namely the distance between a user profile defined using ODP topics and the sets of ODP topics covered by each URL returned in regular web search. We empirically show that this enhancement yields better results than current web search using Google. Then, in the second part of the paper, we investigate the boundaries of biasing PageRank on subtopics of the ODP in order to automatically extend these metadata to the whole web.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-based services*; F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems

General Terms

Algorithms, Experimentation, Analysis

Keywords

Personalized Search, Metadata, Open Directory, Biased PageRank

1. INTRODUCTION

Everyone working in the context of the semantic web is convinced of the utility of metadata describing the content and various other interesting properties of web pages and relationships between them. Probably almost everybody is equally convinced that we will not be able to manually annotate all web pages. But do we really need to?

This paper focuses on manually entered metadata expressing topical categorizations of web pages, as well as on the importance of these pages. This kind of metadata was one of the first metadata available on the web in significant quantities, because it is useful to provide hierarchically structured access to high-quality (recommendable) content on the web, starting with efforts like the Yahoo! Directory, collected and put together by a group of human editors. By inserting a web page into one or more categories, basically a content classification is annotated to the document.

Most notable is the annotation / categorization done in the context of the Open Directory Project (ODP). This is one of the largest efforts to manually annotate web pages, exporting all this metadata information in RDF format. Over 65,000 editors are busy keeping the directory reasonably up-to-date, and the ODP now provides access to over 4 million web pages in the ODP catalog. Still, given the fact that Google now indexes more than 8 billion pages, the ODP effort still only covers about 0.05 percent of the Web pages indexed by Google. So does search using these metadata stand any chance against Google?

One good use these metadata can be put to is to personalize search, i.e., returning search results which are both relevant to the user profile, as well as of good quality. This paper investigates the possibilities we have for building such a personalized search engine based on ODP or similar directory metadata and investigates the quality and effectiveness of such personalization. Specifically, this paper investigates two ways to personalize search and makes the following contributions:

First, using ODP entries directly, we show how to generalize personalized search in catalogs such as ODP and Google Directory beyond the currently available search restricted to specific categories. The precision of this personalized search significantly surpassed the precision offered by Google in a set of experiments on topic related searches.

Second, extending the manual ODP classifications from its cur-

rent 4 million entries to a 8 billion Web in an automated way is feasible, based on an analysis of how topic classifications for a small but important subset of a large page collection can be extended to this large collection via topic-sensitive biasing of PageRank values [21]. This generalizes earlier approaches which already investigated topic-sensitive page ranks, but relied on very simple classifications using only 16 topics.

The paper is organized as follows: In Section 2 we will give a short overview of the Open Directory Project, as well as of PageRank and Personalized PageRank as relevant algorithms for this paper. In Section 3 we discuss how we can directly use ODP and Google directory entries to implement personalized search based on user profiles corresponding to topic vectors from the ODP hierarchy, and discuss a user study comparing Google and ODP search with these personalized versions. Section 4 builds on the idea that sets of ODP or other directory entries can be used to bias PageRank appropriately, and thus to implicitly extend such annotations to the rest of the Web. We specifically investigate when biasing on such a set actually makes a difference to non-biased PageRank, presenting experiments with various kinds of biasing sets (i.e., including different kinds of entries). We then use these results to analyze biasing sets from the ODP 2001 crawl used in [11] and show that all biasing sets we investigated (up to four levels deep) can be successfully used for biasing. Finally, we sketch future work and conclude.

2. RELEVANT BACKGROUND

2.1 ODP: The Open Directory Project

Description. The "DMOZ" Open Directory Project (ODP) [20] is the largest, most comprehensive human-edited web page catalog currently available. It covers 4 million sites filed into more than 590,000 categories (16 wide-spread top-categories, such as Arts, Computers, News, Sports, etc.) Currently, there are more than 65,000 volunteering editors maintaining it.

ODP's data structure is organized as a tree, where the categories are internal nodes and pages are leaf nodes. By using symbolic links, nodes can appear to have several parent nodes. Since ODP truly is free and open, everybody can contribute or re-use the dataset, which is available in RDF (structure and content are available separately). Google for example uses ODP as basis for its Google Directory service.

Applications. Besides its re-use in other directory services, the ODP taxonomy is used as a basis for various other research projects. In Persona [23], ODP is applied to enhance HITS [13] with dynamic user profiles using a tree "coloring" technique (by keeping track of the number of times a user has visited pages of a specific category). Users can rate a page as being "good" or "unrelated" regarding their interest. This data is then used to rank and omit interesting/unwanted results. While [23] asks users for feedback, we only rely on user profiles, i.e., a one-time user interaction. More, we do not develop our search algorithm on top of HITS, but on top of *any* search algorithm, as a refinement. In [17], a similar approach using the ODP taxonomy is applied onto a recommender system of research papers.

The Open Directory can also be used as a reference source containing "good" pages, to fight web spam containing uninteresting URLs through whitelisting [14, 24], as a web corpus for comparisons of rank algorithms [5], as well as for focused crawling towards special-interest pages [7, 3]. Unfortunately, the free availability of ODP also has its downside. A clone of the directory modified to contain some spam pages could trap people to link to this fake directory, which results in an increased ranking not only for this directory clone, but also for the injected spam pages [10].

2.2 PageRank and Personalized PageRank

PageRank [21] computes Web page scores based on the graph inferred from the link structure of the Web. It is based on the idea that "a page has high rank if the sum of the ranks of its backlinks is high". Given a page p, its input $I(p)$ and output $O(p)$ sets of links, the PageRank formula is:

$$PR(p) = (1-c) \cdot \sum_{q \in I(p)} \frac{PR(q)}{\|O(q)\|} + c \cdot E(p) \qquad (1)$$

The dampening factor $c < 1$ (usually 0.15) is necessary to guarantee convergence and to limit the effect of rank sinks [2]. Intuitively, a random surfer will follow an outgoing link from the current page with probability $(1-c)$ and will get bored and select a random page with probability c (i.e., the $\mathbf{E}$ vector has all entries equal to $1/N$, where N is the number of pages in the Web graph).

Initial steps towards personalized page ranking are already described by [21] who proposed a slight modification of the above presented algorithm to redirect the random surfer towards preferred pages using the $\mathbf{E}$ vector. Several distributions for this vector have been proposed since.

Topic-sensitive PageRank. Haveliwala [11] builds a topic-oriented PageRank, starting by computing off-line a set of 16 PageRank vectors biased on each of the 16 main topics of the Open Directory Project [20]. Then, the similarity between a user query and each of these topics is computed, and the 16 vectors are combined using appropriate weights.

Personalized PageRank. A more recent investigation, [12], uses a different approach: it focuses on user profiles. One Personalized PageRank Vector (PPV) is computed for each user. The personalization aspect of this algorithm stems from a *set of hubs* (H)[1], each user having to select her *preferred pages* from it. PPVs can be expressed as a linear combination of PPVs for preference vectors with a single non-zero entry corresponding to each of the pages from the preference set (called basis vectors). The advantage of this approach is that for a hub set of N pages, one can compute 2^N Personalized PageRank vectors without having to run the algorithm again, unlike [21], where the whole computation must be performed for each biasing set. The disadvantages are forcing the users to select their preference set only from within a given group of pages (common to all users), as well as the relatively high computation time for large scale graphs.

3. USING ODP METADATA FOR PERSONALIZED SEARCH

Motivation. We presented in Section 2.2 the most popular approaches to personalizing Web search. Even though they are the best so far, they all have some important drawbacks. In [21], we need to run the entire algorithm for each preference set (or biasing set), which is practically impossible in a large-scale system. At the other end, [11] computes biased PageRank vectors limited only to the broad 16 top-level categories of the ODP, because of the same problem. [12] improves this somewhat, allowing the algorithm to bias on any subset of a given set of pages (H). Although work has been done in the direction of improving the quality of this latter set [4], one limitation is still that the preference set is restricted to a subset of this given set H (if $H = \{CNN, FOXNews\}$ we cannot bias on MSNBC for example). More importantly, the bigger H is, the more time is needed to run the algorithm. Thus finding

[1]Note that hubs were defined here as pages with high PageRank, differently from the more popular definition from [13].

a simpler and faster algorithm with at least similar personalization granularity is still a worthy goal to pursue. In the following we make another step towards this goal.

Introduction. Our first step was to evaluate how ODP search compares with Google search, specifically exploiting the fact that all ODP entries are categorized into the ODP topic hierarchy. We started with the following two observations:

1. Given the fact that ODP "just" includes 4 million entries, and the Google database includes 8 billion, does ODP-based search stand a chance of being comparable to Google?
2. ODP advanced search offers a rudimentary "personalized search" feature by restricting the search to the entries of just one of the 16 main categories. Google directory offers a related feature, by offering to restrict search to a specific category or subcategory. Can we improve this personalized search feature, taking the user profile into account in a more sophisticated way, and how does such an enhanced personalized search on the ODP or Google entries compare to ordinary Google results?

Most people would probably answer (1) *"No, not yet"*, and (2) *"Yes"*. In the following Section we will prove the correctness of the second answer by introducing a new personalized search algorithm, and then we will concentrate on the first answer in the experiments Section.

3.1 Algorithm

Our algorithm is exploiting the annotations accumulated in generic large-scale annotations such as the *Open Directory* [20]. Even though we concentrate our forthcoming discussion on ODP, practically *any* similar taxonomy can be used. These annotations can be easily used to achieve personalization, and can also be combined with the initial PageRank algorithm [21].

We define *user profiles* using a simple approach: each user has to select several topics from the ODP, which best fit her interests. For example, a user profile could look like this:

/Arts/Architecture/Experimental
/Arts/Architecture/Famous_Names
/Arts/Photography/Techniques_and_Styles

Then, at run-time, the output given by a search service (from Google, ODP Search, etc.) is re-sorted using a calculated *distance* from the user profile to each output URL. The execution is also depicted in Algorithm 3.1.

Algorithm 3.1. Personalized Search.

Input: $Prof_u$: Profile for user *u*, given as a vector of topics
Q : Query to be answered by the algorithm.
Output: Res_u : Vector of URLs, sorted after user *u*'s preferences

---

1: Send Q to a search engine S (e.g., Google)
2: Res_u = Vector of URLs, as returned by S
3: **For** $i = 1$ **to Size**(Res_u)
$Dist[i] = Distance(Res_u[i], Prof_u)$
4: **Sort** Res_u using $Dist$ as comparator

We additionally need a function to estimate the distance between a URL and a user profile. Let us inspect this issue in the following discussion.

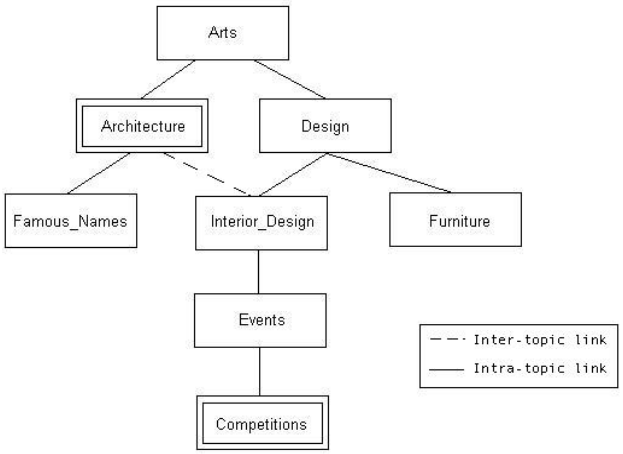

Figure 1: Example tree structure of topics from ODP

3.1.1 Distance Metrics

When performing search on Open Directory, each resulting URL comes with an associated ODP topic. Similarly, a good amount of the URLs output by Google [9] is connected to one or more topics within the Google Directory (almost 50%, as discussed in Section 3.2). Therefore, in both cases, for each output URL we are dealing with two sets of nodes from the topic tree: (1) Those representing the user profile (set A), and (2) those associated with the URL (set B). The distance between these sets can then be defined as the minimum distance between all pairs of nodes given by the Cartesian product $A \times B$. Finally, there are quite a few possibilities to define the distance between two nodes. Even though, as we will see from the experiments, the simplest approaches already provide very good results, we are now performing an optimality study[2] to determine which metric best fits this kind of search. In the following, we will present our best solutions so far.

Naïve Distances. The simplest solution is the minimum tree-distance, which, given two nodes a and b, returns the sum of the minimum number of tree edges between a and the subsumer (the deepest node common to both a and b) plus the minimum number of tree edges between b and the subsumer (i.e., the shortest path between a and b). On the example from Figure 1, the distance between */Arts/Architecture* and */Arts/Design/Interior_Design/Events/Competitions* is 5, and the subsumer is */Arts*.

If we also consider the inter-topic links from the Open Directory, the simplest distance becomes the graph shortest path between a and b. For example, if there is a link between *Interior_Design* and *Architecture* in Figure 1, then the distance between *Competitions* and *Architecture* is 3. This solution implies to load either the entire topic graph or all the inter-topic links into memory. Furthermore, its utility is subjective from user to user: the existence of a link between *Architecture* and *Interior_Design* does not always imply that a famous architect (one level below in the tree) is very close to the area of interior design. We can consider these links in our metric in three ways:

1. Consider the graph containing all intra-topic links and output the shortest path between a and b.
2. Consider graph containing only the intra-topic links directly connected to a and b and output the shortest path.

[2]We refer the reader to [16] for an in-depth view of the approach we took in this study.

3. If there is an intra-topic link between a and b, output 1. Otherwise, ignore all intra-topic links and output the tree-distance between a and b.

Complex Distances. The main drawback of the above metrics comes from the fact that they ignore the depth of the subsumer. The bigger this depth is, the more related are the nodes (i.e., the concepts represented by them). This problem is solved by [16], which investigates ten intuitive strategies for measuring semantic similarity between words using hierarchical semantic knowledge bases such as WordNet [18]. Each of them was evaluated experimentally on a group of testers, the best one having a 0.9015 correlation between the human judgment and the following formula:

$$S(a,b) = e^{-\alpha \cdot l} \cdot \frac{e^{\beta \cdot h} - e^{-\beta \cdot h}}{e^{\beta \cdot h} + e^{-\beta \cdot h}} \quad (2)$$

The parameters are as follows: α and β were defined as 0.2 and 0.6 respectively, h is the tree-depth of the subsumer, and l is the semantic path length between the two words. Considering we have several words attached to each concept and sub-concept, then l is 0 if the two words are in the same concept, 1 if they are in different concepts, but the two concepts have at least one common word, or the tree shortest path if the words are in different concepts which do not contain common words.

Although this measure is very good for words, it is not perfect when we apply it to the Open Directory topical tree because it does not make a difference between the distance from a (the profile node) to the subsumer, and the distance from b (the output URL) to the subsumer. Consider node a to be */Top/Games* and b to be */Top/Computers/Hardware/Components/Processors/x86*. A teenager interested in computer games (level 2 in the ODP tree) could be very satisfied receiving a page about new processors (level 6 in the tree) which might increase his gaming quality. On the other hand, the opposite scenario (profile on level 6 and output URL on level 2) does not hold any more, at least not to the same extent: a processor manufacturer will generally be less interested in the games existing on the market. This leads to our following extension of the above formula:

$$S'(a,b) = ((1-\gamma) \cdot e^{-\alpha \cdot l_1} + \gamma \cdot e^{-\alpha \cdot l_2}) \cdot \frac{e^{\beta \cdot h} - e^{-\beta \cdot h}}{e^{\beta \cdot h} + e^{-\beta \cdot h}} \quad (3)$$

with l_1 being the shortest path from the profile to the subsumer, l_2 the shortest path from the URL to the subsumer, and γ a parameter in $[0,1]$.

Combining the Distance Function with Google PageRank. And yet something is still missing. If we use Google to do the search and then sort the URLs according to the Google Directory taxonomy, some high quality pages might be missed (i.e., those which are top ranked, but which are not in the directory). In order to integrate that, the above formula could be combined with the Google PageRank. We propose the following approach:

$$S''(a,b) = \delta \cdot \frac{1}{1 + S'(a,b)} + (1-\delta) \cdot PageRank(b) \quad (4)$$

δ is another parameter in $[0,1]$ which allows us to keep the final score $S''(a,b)$ also inside $[0,1]$ (for normalized PageRank scores). Finally, if a page is not in the directory, we take $S'(a,b)$ to be ∞.

Conclusion. Human judgment is a non-linear process over information sources [16], and therefore it is very difficult (if not impossible) to propose a metric which is in perfect correlation to it. A thorough experimental analysis of all these metrics (which we are currently performing, but which is outside the scope of this paper) could give us a good enough approximation. In the next Section we will present some experiments using the simple metric presented first, and show that it already yields quite reasonable improvements.

3.2 Experimental Results

To evaluate the benefits of our personalization algorithm, we interviewed 17 of our colleagues (researchers in different computer science areas, psychologists, pedagogues and designers), asking each of them to define a user profile according to the Open Directory topics (see Section 3.1 for an example profile), as well as to choose three queries of the following types:

- One *clear* query, which they *knew to have one or maximum two meanings*[3]
- One *relatively ambiguous* query, which they knew to have two or three meanings
- One *ambiguous* query, which they knew to have at least three meanings, preferably more

We then compared test results using the following four types of Web search:

1. "Plain" Open Directory Search
2. *Personalized Open Directory Search*, using our algorithm from Section 3.1 to reorder the top 1000 results returned by the ODP Search
3. Google Search, as returned by the Google API [8]
4. *Personalized Google Search*, using our algorithm from Section 3.1 to reorder the top 100 URLs returned by the Google API[4], and having as input the Google Directory topics returned by the API for each resulting URL.

For each algorithm, each tester received the top 5 URLs with respect to each type of query, 15 URLs in total. All test data was shuffled, such that testers were neither aware of the algorithm, nor of the ranking of each assesed URL. We then asked the subjects to rate each URL from 1 to 5, 1 defining a very poor result with respect to their profile and expectations (e.g., topic of the result, content, etc.) and 5 a very good one[5]. Finally, for each sub-set of 5 URLs we took the average grade as a measure of importance attributed to that $< algorithm, querytype >$ pair. The average values for all users and for each of these pairs can be found in table 1, together with the averages over all types of queries for each algorithm.

We of course expected the "plain" ODP search to be significantly worse than the Google search, and that was the case: an average of 2.41 points for ODP versus the 2.76 average received by Google. Also predictable was the dependence of the grading on the query type. If we average the values on the three columns representing each query type, we get 2.54 points for ambiguous queries, 2.91 for semi-ambiguous ones and 3.25 for clear ones - thus, the clearer was the query, the better rated were the URLs returned.

Personalized Search using ODP. But the same table 1 also provides us with a more surprising result: The personalized search algorithm is *clearly better than Google search*, regardless whether we use Open Directory or Google Directory as taxonomy. Therefore, a personalized search on a well-selected set of 4 million pages often provides better results than a non-personalized one over a 8 billion set. This a clear indicator that taxonomy-based result sorting is indeed very useful. For the ODP experiments, only our clear queries did not receive a big improvement, mainly because for some of

[3]Of course, that did not necessarily mean that the query had no other meaning.

[4]We were forced to use only the top 100 URLs, because of the limitations imposed by the Google API, as well as the limited number of Google API licenses we had available.

[5]This is practically a weighted P@5.

Algorithm	Ambiguous Queries	Semi-ambiguous Queries	Clear Queries	Average / Algorithm
ODP Search	2.09	2.29	2.87	2.41
Personalized ODP Search	3.11	3.41	3.13	3.22
Google Search	2.24	2.79	3.27	2.76
Personalized Google Directory Search	2.73	3.15	3.74	3.20

Table 1: Survey results for the analyzed web search approaches

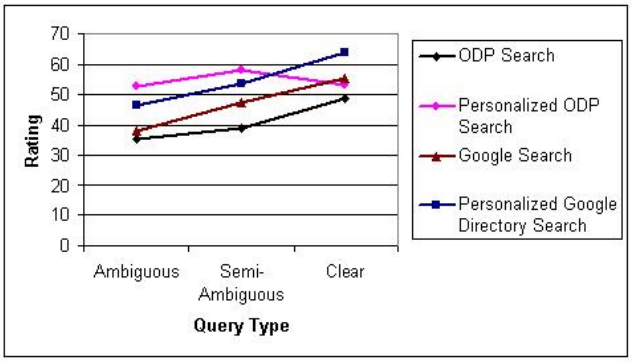

Figure 2: Algorithm grading for each query type

these queries ODP contains less than 5 URLs matching both the query and the topics expressed in the user profile.

Personalized Search using Google. Similarly, personalized search using Google Directory was far better than the usual Google search. We would have expected it to be even better than the ODP-based personalized search, but results were probably negatively influenced by the fact that the ODP experiments were run on 1000 results, whereas the Google Directory ones only on 100, due to the limited number of Google API licenses we had.

The grading results are summarized in Figure 2. Generally, we can conclude that personalization significantly increases output quality for ambiguous and semi-ambiguous queries. For clear queries, one should prefer Google to Open Directory search, but also Google Directory search to the plain Google search. Also, the answers we sketched in the beginning of this Section proved to be true: Google search *is* still better than Open Directory search, but we provided a personalized search algorithm which *outperforms* the existing Google and Open Directory search capabilities.

Another interesting result is that 40.98% of the top 100 Google pages were also contained in the Google Directory. More specifically, for the ambiguous queries 48.35% of the top pages were in the directory, for the semi-ambiguous ones 41.35%, and for the clear ones 33.23%[6].

Finally, let us add that we performed statistical significance tests[7] on our experiments [1], obtaining the following results:

- Statistical significance with an error rate below 1% for the "algorithm" criterion, i.e., there is significant difference between each algorithm grading.
- An error rate below 25% for the "query type" criterion, i.e., the difference between the average grades with respect to query types is less statistically significant.
- Statistical significance with an error rate below 5% for the inter-relation between query type and algorithm, i.e., the results are overall statistically significant.

[6]There were more pages for the ambiguous queries, because they were covering multiple topics.

[7]More specifically, we used an Analysis of Variance (ANOVA).

Src. of variance	QS	Deg. of Free.	F-value [25]
Query Type	17.092	2	F(2,32,75%) = 2.114
Algorithm	22.813	3	F(3,48,99%) = 6.812
Inter-Relation	7.125	6	F(6,96,95%) = 2.512

Table 2: Survey results for the analyzed web search approaches

For a more in-depth view, the statistical analysis data is collected in table 2.

4. EXTENDING ODP ANNOTATIONS TO THE WEB

In the last Section we have shown that using ODP entries and their categorization directly for personalized search turns out to be amazingly good. Can this huge annotation effort invested in the ODP project (with 65,000 volunteers participating in building and maintaining the ODP database) be extended to the rest of the Web? This would be useful if we want to find less highly rated pages not contained in the directory. Just extending the ODP effort does not scale, because first, significantly increasing the number of volunteers seems improbable, and second, extending the selection of ODP entries to a larger percentage obviously becomes harder and less rewarding once we try to include more than just the "most important" pages for a specific topic.

We start with the following questions:

- Given that PageRank for a large collection of Web pages can be "biased" towards a smaller subset, can this be done with sets of ODP entries corresponding to given categories / subcategories as well?
- Specifically, ODP entries consist of many of the "most important" entries in a given category. Do we have enough entries for each topic such that biasing on these entries makes a difference?

4.1 When does biasing make a difference?

One of the most important work investigating PageRank biasing is [11]. It first uses the 16 top levels of the ODP to bias PageRank on and then provides a method to combine these 16 resulting vectors into a more query-dependant ranking. But what if we would like to use one or several ODP (sub-)topics to compute a Personalized PageRank vector? More general, what if we would like to achieve such a personalization by biasing PageRank towards some generic subset of pages from the current Web crawl we have? Many authors have used such biasings in their algorithms. Yet none have studied the boundaries of this personalization, the characteristics the biasing set has to exhibit in order to obtain relevant results (i.e., rankings which are different enough from the non-biased PageRank). We will investigate this in the current Section. Once these boundaries are defined, we will use them to evaluate (some of) the biasing sets available from ODP in Section 4.2.

First, let us establish a characteristic function for biasing sets, which we will use as parameter determining the effectiveness of biasing. Pages in the World Wide Web can be characterized in quite a few ways. The simplest of them is the out-degree (i.e., total number of out-going links), based on the observation that if biasing is targeted to such a page, the newly achieved increase in PageRank score will be passed forward to all its out-neighbors (pages to which it points). A more sophisticated version of this measure is the hub value of pages. *Hubs* were initially defined in [13] and are pages pointing to many other *high quality* pages. Reciprocally, high quality pages pointed to by many hubs are called *authorities*. There are several algorithms for calculating this measure, the most common ones being HITS [13] and its more stable improvements SALSA [15] and Randomized HITS [19]. Yet biasing on better hub pages will have less influence on the rankings because the "vote" a page gives is propagated to its out-neighbors divided by its out-degree. Moreover, there is also an intuitive reason against this measure: PageRank biasing is usually performed to achieve some degree of personalization and people tend to prefer highly valued authorities to highly valued hubs. Therefore, a more natural measure is an authority-based one, such as the non-biased PageRank score of a page.

Even though most of the biasing sets consist of high PageRank pages, in order to make this analysis complete we have run our experiments on different choices for these sets, each of which must be tested with different sizes. For comparison to PageRank, we used two degrees of similarity between the non-biased PageRank and each resulting biased vector of ranks. They are defined in [11] as follows:

1. **OSim** indicates the degree of overlap between the top n elements of two ranked lists τ_1 and τ_2. It is defined as
$$\frac{|Top_n(\tau_1) \cap Top_n(\tau_2)|}{n} \tag{5}$$
2. **KSim** is a variant of Kendall's τ distance measure. Unlike OSim, it measures the *degree of agreement* between the two ranked lists. If U is the union of items in τ_1 and τ_2 and δ_1 is $U \setminus \tau_1$, then let τ_1' be the extension of τ_1 containing δ_1 apearing after all items in τ_1. Similarly, τ_2' is defined as an extension of τ_2. Using these notations, KSim is defined as follows:
$$KSim(\tau_1, \tau_2) = \frac{|(u,v) : \begin{array}{l}\tau_1' \text{ and } \tau_2' \text{ agree on order (u,v),} \\ \text{and } u \neq v\end{array}|}{|U| \cdot |U-1|} \tag{6}$$

Even though [11] used $n = 20$, we chose n to be 100, after experimenting with both values and obtaining more stable results with the latter value. A general study of different similarity measures for ranked lists can be found in [6].

Let us start by analyzing the biasing on high quality pages (i.e., with a high PageRank). We consider the most common set to contain pages in the range $[0-10]\%$ of the sorted list of PageRank scores. We varied the sum of scores within this set between 0.00005% and 10% of the total sum over all pages (for simplicity, we will call this value TOT hereafter). For very small sets, the biasing produced an output only somewhat different: about 38% Kendall similarity (see Figure 3). The same happened for large sets, especially those above 1% of TOT. Finally, the graph makes also clear where we would get the most different rankings from the non-biased ones (in a set size from 0.003% to 0.1%)[8].

[8]Generally, if the similarity (y-axis value) is below the threshold line, then we consider the biased ranks to be relevant, i.e., different enough from the non-biased ones.

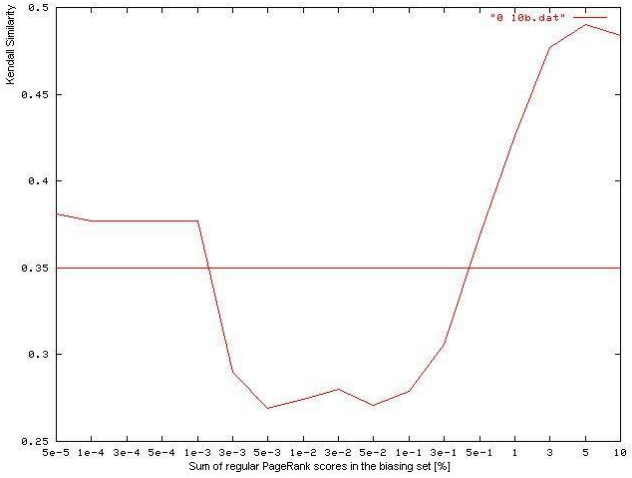

Figure 3: Biasing behavior for top 0 - 10% PageRank pages

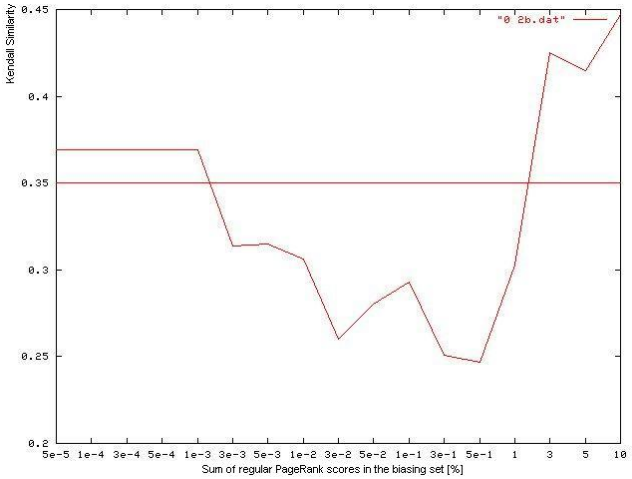

Figure 4: Biasing behavior for top 0 - 2% PageRank pages

Someone could wish to bias only on the best pages (the top $[0-2]\%$, as in Figure 4). In this case, the above results would only be shifted a little bit to the right on the x-axis of the graph, i.e., the highest differences would be achieved for a set size from 0.02% to 0.75%. This was expectable, as all the pages in the biasing set were already top ranked, and it would therefore take a little bit more effort to produce a different output with such a set.

Another possible input set consists of randomly selected pages (Figure 5). Such a set most probably contains many low PageRank pages. This is why, although the biased ranks are very different for low TOT values, they start to become extremely similar (up to almost the same) after TOT exceeds 0.01% (because it would take *a lot* of low PageRank pages to accumulate a TOT value of 1% of the overall sum of scores, for example).

The extreme case is to bias *only* on low PageRank pages (Figure 6). In this case, the biasing set will contain too many pages even sooner, around $TOT = 0.001\%$.

The last experiment is mostly theoretical. One would expect to obtain the smallest similarities to the non-biased rankings when using a biasing set from $[2-5]\%$ (because these pages are already close to the top, and biasing on them would have best chances to overturn the list). Experimental results support this intuition (Fig-

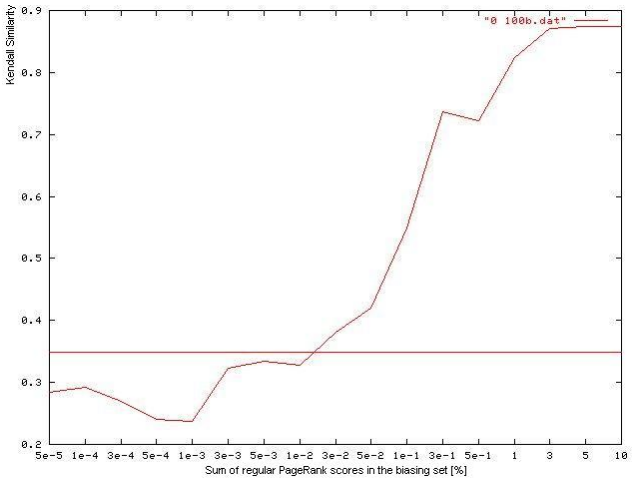

Figure 5: Biasing behavior for random pages

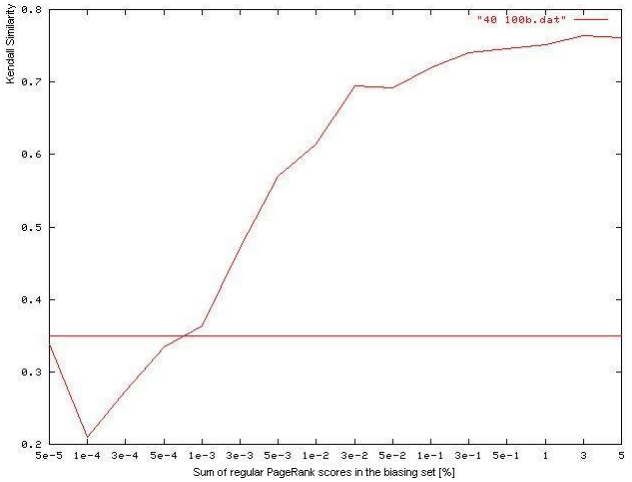

Figure 6: Biasing behavior for random low PageRank pages

ure 7), generating very different rankings for very small biasing sets and up to sets of $TOT = 0.1\%$, that is for a large scale of sizes for the biasing set.

The graphs above were initially generated based on a crawl of 3 million pages. Once all of them had been finalized, we selectively ran similar experiments on the Stanford WebBase crawl [22], obtaining similar results. For example, a biasing set of size $TOT = 1\%$ containing randomly selected pages produced rankings with a 0.622% Kendall similarity to the non-biased ones, whereas a set of $TOT = 0.0005\%$ produced a similarity of only 0.137%. This was necessary in order to prove that the above discussed graphs are not influenced by the crawl size. Even so, the limits they establish are not totally accurate, because of the random or targeted random selection (e.g., towards top $[0-2]\%$ pages) of our experimental biasing sets.

4.2 Is biasing possible in the ODP context?

The URLs collected in the Open Directory are manually added Web pages supposed to (1) cover the specific topic of the ODP tree leaf they belong to and (2) be of high quality. Both requirements are not fully satisfied. Sometimes (rarely though) the pages are not really representing the topic in which they were added. More important for PageRank biasing, they usually cover a large interval of page ranks, which made us decide for the random biasing model. However, we are aware that in this case, the human editors chose much more high quality pages than low quality ones, and thus the decisions of the analysis are susceptible to errors.

Generally, according to the random model of biasing, every set with TOT below 0.015% is good for biasing. According to this, all possible biasing sets analyzed in tables 4, 5 and 3 would generate a different enough PageRank vector[9].

We can therefore conclude that biasing is (most probably) possible on *all* subsets of the Stanford Open Directory crawl.

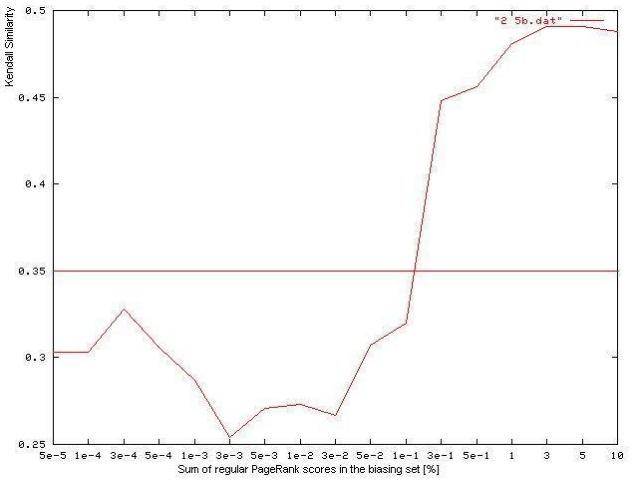

Figure 7: Biasing behavior for top 2 - 5% PageRank pages

Topic	TOT Value	Topic	TOT Value
/Arts	0.01062%	/Business	0.01046%
/Computers	0.02343%	/Games	0.00297%
/Health	0.00596%	/Home	0.00528%
/Kids & Teens	0.00532%	/News	0.00707%
/Recreation	0.00541%	/Reference	0.01139%
/Regional	0.00839%	/Science	0.01314%
/Shopping	0.00296%	/Society	0.01201%
/Sports	0.00235%	/World	0.01091%

Table 4: Low-level ODP biasing analysis for the Stanford ODP crawl

5. CONCLUSIONS

Given that directories like ODP represent some of the largest manual metadata collections today (ODP contains topic classifications for about 0.05% of all Google indexed pages), this paper investigated the impact these efforts have and specifically their feasibility to implement personalized search based on these metadata. We investigated two possibilities to do that, and made the following contributions:

First, using ODP entries directly, we showed how to generalize personalized search in catalogs such as ODP and Google Directory beyond the currently available search restricted to specific categories. The precision of this personalized search significantly surpassed the precision offered by unpersonalized search in a set of

[9]Only biasing on the entire topic set of "Computers" seems to exceed this limit a little bit, but running the biased PageRank with it produced a good enough similarity - most probably because of the special structure of the ODP topic sets, as we discussed above in this Section.

/Computers	TOT Value	/Computers	TOT Value
/CAD/Mapping_and_GIS	0.000072%	/Companies/Product_Support	0.003163%
/Education/Internet	0.000001%	/Education/Hardware/HowTos_and_Tutorials	0.000198%
/Internet/Consulting	0.000041%	/Internet/Statistics_and_Demographics	0.000101%
/Internet/Bulletin_Board_Services	0.000018%	/Internet/Cyberspace	0.000167%
/Internet/E-mail	0.000001%	/Internet/Organizations	0.000377%
/Internet/Resources	0.000207%	/Internet/Telephony	0.000008%
/Internet/Broadcasting/Video_Shows	0.000065%	/Internet/E-mail/Electronic_Postcards/Humor	0.000007%
/Internet/Commercial_Services/Web_Hosting/Free/Games_Related	0.000001%	/Programming/Games	0.000124%
/Programming/Internet	0.000052%	/Publications/Mailing_Lists	0.000603%
/Security/Anti_Virus	0.000110%	/Security/Internet	0.001193%

Table 3: Low-level ODP biasing analysis for the Stanford ODP crawl

/Computers	TOT Value	/Computers	TOT Value
/Algorithms	0.000072%	/Artificial_Intelligence	0.000146%
/Artificial_Life	0.000127%	/Bulletin_Board_Syst.	0.000063%
/CAD	0.000078%	/Companies	0.004042%
/Data_Comm.	0.000001%	/Data_Formats	0.000059%
/Desktop_Publishing	0.000038%	/E-Books	0.003534%
/Ethics	0.000253%	/Graphics	0.000033%
/Hacking	0.000002%	/Hardware	0.001286%
/Home_Automation	0.000001%	/HCI	0.000223%
/Internet	0.002062%	/Multimedia	0.000713%
/Organizations	0.000008%	/Parallel_Computing	0.000055%
/Programming	0.000188%	/Publications	0.000626%
/Robotics	0.000226%	/Security	0.001308%
/Software	0.007318%	/Speech_Technology	0.000008%
/Supercomputing	0.000835%	/Usenet	0.000089%
/Virtual_Reality	0.000066%	/History	0.000511%
/Education	0.000460%		

Table 5: Low-level ODP biasing analysis for the Stanford ODP crawl

experiments.

Second, extending the manual ODP classifications from its current 4 million entries to a 8 billion Web is feasible, based on an analysis of how topic classifications for subsets of large page collection can be extended to this large collection via topic-sensitive biasing of PageRank values.

While the theoretical framework we presented in this Section is generally applicable, so far we were only able to apply it on an existing ODP crawl from 2001 (the one used in [11]). Our crawler is currently collecting a new crawl based on the current ODP directory, which we will then use to evaluate the current status and quality of the ODP entries and their suitability for biasing.

6. REFERENCES

[1] J. Bortz. *Statistics for Social Scientists*. Springer Verlag, 1993.

[2] S. Brin, R. Motwani, L. Page, and T. Winograd. What can you do with a web in your pocket? *Data Engineering Bulletin*, 21(2):37–47, 1998.

[3] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific web resource discovery. In *Proceedings of the 8th Intl. WWW Conference*, 1999.

[4] P.-A. Chirita, D. Olmedilla, and W. Nejdl. Pros: A personalized ranking platform for web search. In *Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Aug 2004.

[5] C. Ding, X. He, P. Husbands, H. Zha, and H. D. Simon. Pagerank, hits and a unified framework for link analysis. In *Proceedings of the 25th annual International ACM SIGIR Conference*, pages 353–354. ACM Press, 2002.

[6] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International WWW Conference*. ACM Press, 2001.

[7] M. Ester, H.-P. Kriegel, and M. Schubert. Accurate and efficient crawling for relevant websites. In *Proceedings of the 30th International VLDB Conference*, 2004.

[8] Google search api. http://api.google.com.

[9] Google search engine. http://www.google.com.

[10] Z. Gy'ongyi, H. Garcia-Molina, and J. Pendersen. Combating web spam with trustrank. In *Proceedings of the 30th International VLDB Conference*, 2004.

[11] T. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th International WWW Conference*, 2002.

[12] G. Jeh and J. Widom. Scaling personalized web search. In *Proc. of the 12th Intl. WWW Conference*, 2003.

[13] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[14] O. Kolesnikov, W. Lee, and R. Lipton. Filtering spam using search engines, 2003.

[15] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks (Amsterdam, Netherlands: 1999)*, 33(1–6):387–401, 2000.

[16] Y. Li, Z. A. Bandar, and D. McLean. An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):871–882, 2003.

[17] S. E. Middleton, D. C. D. Roure, and N. R. Shadbolt. Capturing knowledge of user preferences: ontologies in recommender systems. In *Proceedings of the First International Conference on Knowledge Capture*, 2001.

[18] G. Miller. Wordnet: An electronic lexical database. *Communications of the ACM*, 38(11):39–41, 1995.

[19] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Stable algorithms for link analysis. In *Proc. 24th Annual Intl. ACM SIGIR Conference*. ACM, 2001.

[20] Open directory project. http://dmoz.org/.

[21] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.

[22] Stanford webbase project. http://webbase.stanford.edu.

[23] F. Tanudjaja and L. Mui. Persona: A contextualized and personalized web search. In *Proceedings of the 35 Annual Hawaii International Conference on System Sciences*, 2002.

[24] M. Williamson. Using dmoz open directory project lists with novell bordermanager, 2003.

[25] J. B. Winer. *Statistical principles in experimental design*. McGraw Hill, 1962.