Journal of Web Engineering, Vol. 10, No. 2 (2011) 153–173
© Rinton Press

AUTOMATIC MAINTENANCE OF WEB DIRECTORIES BY MINING WEB BROWSING DATA

CARLOS HURTADO
Faculty of Engineering and Science, Universidad Adolfo Ibáñez
Diagonal Las Torres 2640, Santiago, Chile
carlos.hurtado@uai.cl

MARCELO MENDOZA
Computer Science Department, Universidad Técnica Federico Santa María
Vicuña Mackenna 3939, Santiago, Chile
marcelo.mendoza@usm.cl

Received December 24, 2010
Revised May 26, 2011

Web directories allow Web users to browse a hierarchy of categories, under which different types of resources are classified. We study the problem of maintaining a Web directory, that is, the problem of continually discovering and ranking resources that are relevant to the categories of the directory. We propose an unsupervised computational method that conducts the maintenance of the directory by analyses of user browsing data. The method is based on the extraction and classification of user sessions (sequences of resources selected by users) into the categories of the directory. In addition, we show that the directory maintenance method can be slightly modified to find queries that are useful to find relevant resources allowing users to switch from directory browsing to query formulation. Experimental results allow for affirmation that the proposed methods are effective, that they attain identification of new pages in each category and also recommend related queries with high precision, without needing labeled data to conduct traditional web page and query classification tasks.

Keywords: Web directories, Web Mining, Query Logs

Communicated by: D. Schwabe & Q. Li

1 Introduction

The Web is the largest and most diverse repository of data in the world, allowing users to access information, download and upload contents, as well as performing different sorts of transactions. Due to its explosive growth, the search for information on the Web is becoming increasingly difficult for its users. In the early nineties, Web directories were by far the most popular tools to support this task. Web directories are hierarchical structures that allow grouping Web resources into different categories. Examples of directories that provide extensive categories of large number of Web resources include Yahoo! (Yahoo.com), the Open Directory Project (Dmoz.org), and Looksmart (Looksmart.com). With the arrival of Web 2.0 applications and social media, niche Web directories have gained popularity, including directories of RSS feed (Topic.com), shopping directories in commerce search, and directories of mobile applications, among others.

Web directories are usually created and maintained by human editors, which can be either experts or common users who voluntarily decide to collaborate. The latter is the case of the Open Directory Project, which has gathered more than 80,000 human editors since its launch in 1998. Together these people have classified nearly 4,5 million sites into more than 590,000 categories. However, due to the large scale and dynamism that the Web has reached, it is unfeasible to manually keep large Web directories up-to-date. This task requires to periodically delete obsolete resources, add new relevant resources, and rank the resources according to their importance to the categories.

An alternative approach is to carry out the directory maintenance task automatically, with or without a little intervention of human editors. Roughly, this can be done by retrieving resources from the Web and then classifying them, also automatically, into the categories of the directory. When Web resources are associated with text descriptions, which is the case in several applications and an assumption in this paper, the problem is similar to text categorization, a problem that has attracted a great deal of research in the last two decades [1]. A main approach to address the problem is to apply machine learning techniques to extract a model from a collection of previously classified resources. The extracted model can be used to classify new resources which do not appear in the original collection. A good survey about aforementioned approach is provided by Sebastiani [2].

Web directories are commonly integrated into Web sites that keep track of user feedback in log files. In particular, log files store user sessions, which consist of sequences of Web documents selected by users in a specific time interval. It has been widely proved that Web logs, particularly Web query logs, are a rich input to improve Web sites, and more generally, to improve different services offered in current Web sites. In this paper, we take advantage of such principle to support the maintenance of a Web directory. We propose a method in which user sessions are classified into one or more categories of the directory. The classified sessions are then used to estimate the relevance of resources to the categories.

In addition, we show that the directory maintenance method can be slightly modified to find queries that are useful to identify relevant resources. This facility allows users to switch from directory browsing to Web query formulation. We propose to identify such queries by estimating its ”utility” to the categories, a notion we introduce for measuring the level of semantic connection between queries and categories.

The remaining of the article is organized as follows. In Section 2, we present related work. In Section 3, we explain the directory maintenance method proposed. An experimental evaluation is presented in Section 4. A discussion of the experimental results is in Section 5. Finally, in Section 6 we conclude and present some prospects for future work.

2 Related Work

The state of the art shows works in two main directions related to the aim of this article. The first one is to change the directory structure, seeking to identify new sub categories defining more precisely the scope of some concepts. The methods of this line are often semi supervised (e.g. [4, 5, 6]), thus require the effort of experts to validate the results. A second line of research focuses on enriching the content of directories. Often this type of work seeks to improve the description of the concepts described in each category identifying more and better descriptive terms. This line also introduces methods able to categorize new documents in each category. Our work is part of this second research line, with a comprehensive effort to identify documents of high quality to populate the directory and improve the description of each category by the detection of queries that can clarify the concepts behind the directory. Our method requires an initial directory, but as we shall show in this article, it is possible to update the directory and improve its content in a completely automatic way, achieving a performance comparable to those obtained by human editors. Unlike methods based on web page classification (see [2] for a complete survey of these methods), our method does not require labeled data because it uses query logs as a measure of implicit user feedback, using this information source for conducting the directory maintenance process. This fact eliminates the direct dependence of web directory maintenance methods on expert editors or expert judges.

Directory structure modification

Yang et al. [3] propose clustering documents to construct sub-directories. To do this, a collection of documents related to a specific topic, which is manually created, is represented by term vectors. Throught the use of SOM networks, this allows them to generate a lattice of terms and a lattice of documents, defining a mapping function between both structures. Afterwards, a clustering strategy that obtains a hierarchy of documents is applied to the collection of documents. Later, using the mapping function between term-document lattices, each node of the hierarchy is tagged with terms extracted from the lattice of words, obtaining a sub-directory. Finally, experts execute a validation and pruning phase on the sub-directory.

A more sophisticated method was introduced by Stamou et al. [4]. The authors constructed a taxonomy from WordNet 2.0, a collection that consists of more than 118 thousand synonym groups (synsets). To do this, they used SUMO (Suggested Upper Merged Ontology), which is a data set of concepts mapped onto WordNet 2.0 and MWND (Multi Word Net Domains), which is a data set of domains structured hierarchically around 165 categories. Using the MWND's categories, which are also mapped onto WordNet 2.0, they associate a domain to each synset of WordNet. Later, a group of experts manually classify the synsets in six first level categories taken from Google's directory and add synsets like sub-domains from each of these categories obtaining a taxonomy of 143 nodes. In a second stage, a collection of documents is processed in order to identify lexical chains between categories and documents, measuring how informative a page is for a topic. The success of the proposed method depends a great deal on the quality of the process made in order to identify lexical chains and also to generate the initial taxonomy.

Chung et al. [5] introduce a framework that constructs web directories oriented towards specific domains. The framework identifies features extracted from the documents that enrich the previously established category descriptions. Combined with design decisions made by a group of experts, this knowledge allows them also to modify the structure of the directory. Using this framework the authors constructed a directory of business for the Spanish Web (SBiz).

Gerstel et al. [6] deals with the problem of modifying the structure of the directories by identifying *hotlinks* from log files, meaning links between categories that subsequently have been visited and from which directory users have selected documents on multiple opportunities. By adding these new links it was possible to reduce the length of the paths that users must traverse to find resources relevant to their information needs.

Zaihrayeu et al. [7] planted work lines around the important problem of converting web directories into ontologies. Using this transformation would allow to map a document to a knowlegde field where recommendation and inference tasks could be treated more precisely.

Directory content improvement

Web queries have been used for directory maintenance and enrichement tasks. Chuang & Chien [8] introduce a technique to categorize query terms in nodes of a Web directory using log files. In the first stage, a group of experts determine the query terms that are the most frequently used in the formulation of queries. Then, the experts manually classify those query terms in the categories of the Web directory. For each new query term, they recover the group of queries that have been formulated using that term, and the group of documents selected in the sessions for each of such queries. Then, a function is introduced to measure the distance between each group of documents and each category of the taxonomy. The function considers terms in the documents and in the categories. Using this function the category which is closest to each query term is identified. The authors show that query terms categorized in the taxonomy are useful to improve the description of categories.

A similar approach was applied by Adami et al. [9, 10]. Their method considers that each node of the taxonomy is characterized by terms that describe it. The documents are represented using term vectors and are categorized in the taxonomy using a technique called *TaxSOM*, which is based on auto-organizational maps. Finally, a pruning process is conducted by a group of experts to remove inaccurate documents.

A more complex approach was studied by Zhang et al. [11], who deal with the problem of integrating resources extracted from different taxonomies into a master taxonomy. To do this the authors analyze the taxonomy where the resources are originally described, trying to mapping concepts from these taxonomies to the master taxonomy. Using techniques such as node shrinkage they are able to identify relations between the concepts of the taxonomies, allowing resource categorization.

Finally, Qi et al. [1] provide a good survey about Web page classification. However, unlike the methods proposed in this paper, the work revised by them are supervised, depending on the existence of labeled data to train standard text categorization methods.

3 Directory Maintenance Method

The method proposed in this paper makes use of the following four information sources:

Web content Contains a set of Web resources, each one represented by a URL and a text description; In the case of Web pages descriptive texts can correspond to snippets or to the full content of each page.

Web log Contains Web user sessions, each of which is a list of resources; Often Web user sessions correspond to query sessions (i.e. Web sessions registered in search engines), where the list of resources corresponds to the list of clicked pages / sites in each query session.

Directory content Contains the Web directory data, that is a hierarchy of categories, a list of resources associated to each category, and a text description associated to each resource; Often the list of resources for each category corresponds to a list of recommended pages / sites, and its descriptive texts correspond to snippets.

Directory log Contains Web user sessions in which users have visited categories and resources of the directory.

As Figure 1 shows, the directory log and directory content sources are used as training data to learn a model to classify Web sessions (represented by `WS` in Figure 1) into categories of the directory. The model is essentially a profile-style classifier that selects suitable features to construct category representations. The profile of each category (represented by $\mathtt{P_c}$ in Figure 1) is a term-weight vector that captures the semantics of the category.

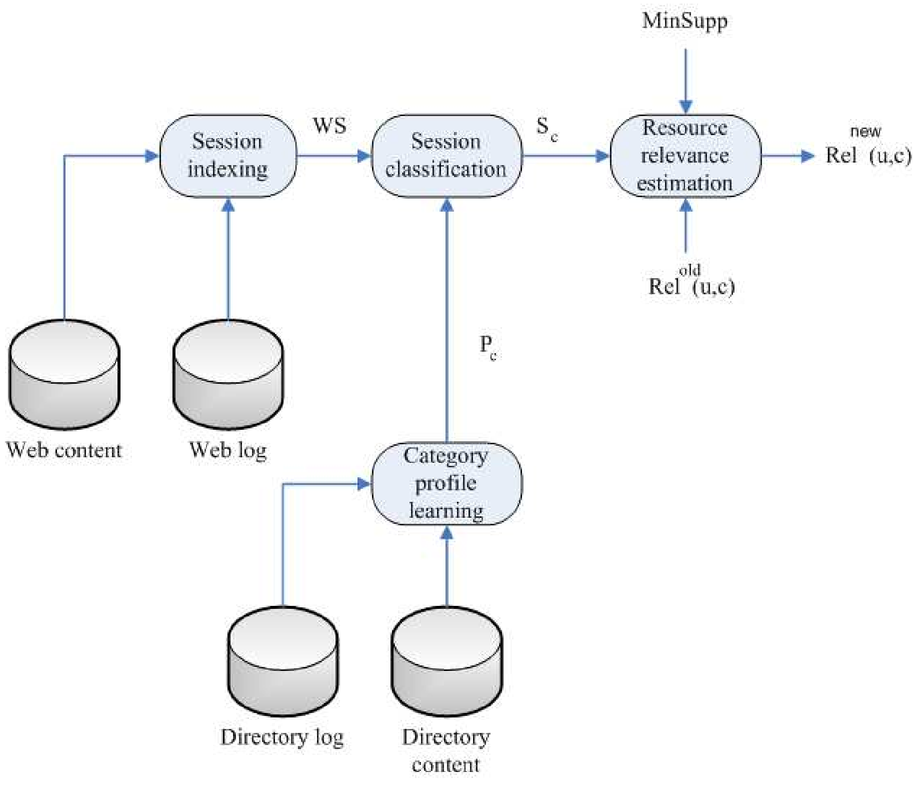


Fig. 1. Resource recommendation process

In the session indexing step, each Web session is transformed into a term-weight vector. In the session classification step, each Web session is classified into the category whose profile is closest. After this step, each category is associated to a set of Web sessions (denoted by $\mathtt{S_c}$ in Figure 1). Each session is viewed as a set of preferences made by users on the resources. This allows us to estimate a relevance measure for resources that appear in the sessions (resource relevance estimation step). However, if a resource does not appear or appears just few times in the sessions, we may not have enough evidence to support a relevance estimation. In such cases we do not update the estimated relevance.

3.1 Web Session Indexing

A Web session is a sequence of requests of Web resources by a single user made in a defined interval of time. In the Web Session Indexing step, each Web session is transformed into a term-weight vector that captures the information need associated to the Web session. Consider a fixed collection U of resources. Given a Web session s we denote by $U_s \subseteq U$ the set of resources that were selected in s.

Consider a fixed list of terms $T = t_1, \ldots, t_m$. We assume that each resource in U is described by a text. Given a Web session s, we build the term vector representation of s by considering a $\mathtt{Tf} - \mathtt{Idf}$ scheme. The i-th component $v_s[i]$ of the vector is defined as follows:

$$v_s[i] = \sum_{u \in U_s} \frac{Tf_{i,u}}{max\ Tf_u} \times \log \frac{N_U}{N_{i,U}}, \tag{1}$$

where $Tf_{i,u}$ is the number of occurrences of the term t_i in the content of u, N_U is the number of documents in U, and $N_{i,U}$ is the number of resources in U which includes the term t_i. Notice that this vector representation corresponds to an aggregation of the $\mathtt{Tf} - \mathtt{Idf}$ weighting vectors of the clicked documents in s.

This Web session representation has several advantages. First, it is simple and easy to compute. Notice that Tf factors are calculated when the resources are indexed, usually in an inverted index. Then, to retrieve Tf factors we only need to use the inverted index, adding the values for each term of the vocabulary and for each clicked document in s. Notice that we can avoid the storage of these vectors by using an inverted index of Web sessions, where each term entry references a posting list of Web session IDs with their aggregated Tf factors. In addition, each $N_{i,U}$ factor can be stored in the vocabulary part of the index. Finally, the inclusion of new Web sessions can be processed incrementally, adding new Tf values for old terms and new term - Web session ID entries for new terms.

Our Web session representation avoids sparse vectors because it uses terms extracted from a set of clicked Web resources. Finally, we may view Web sessions simply as documents, allowing the use of standard similarity functions for documents, as in the case of IR systems.

3.2 Category Profile Learning

In the category profile learning step, a profile-based model for classifying Web sessions is extracted from the data. We next explain how the profiles of categories are obtained. We denote by $\mathtt{CS_c}$ the set of Web sessions that consists of selections of resources that appear under a category c in the directory. Each session in $\mathtt{CS_c}$ is transformed into a term-weight vector using Equation 1. The profile of a category c, denoted p_c, is defined as the centroid of the sessions in $\mathtt{CS_c}$. The i-th component of p_c is calculated as follows:

$$p_c[i] = \sum_{s \in \mathtt{CS_c}} v_s[i] \times \frac{1}{|\mathtt{CS_c}|}, \tag{2}$$

This category profile allows us to identify relations among Web sessions and categories, measuring the similarity between each Web session representation and each centroid, as a Rocchio process [12].

3.3 Web Session Classification

In order to classify Web sessions into the directory, we measure the similarity between each Web session and each category profile, following an approach similar to a Rocchio classifier. We classify each Web session into the closest category, where the notion of closeness is defined by a distance function. We measure the distance between a Web session s and a category c by applying the standard cosine function over the vector representations of s and c, that is:

$$dist(s,c) = 1 - \frac{v_s \cdot p_c}{\mid v_s \mid \times \mid p_c \mid}. \tag{3}$$

To avoid consideration of Web sessions marginally relevant to each category, we introduce a pruning step. Let $\mathtt{WS_c}$ be the set of Web sessions categorized into c. Given a maximum distance threshold $\mathtt{MaxDist}$, the pruned set of Web sessions classified into c, denoted by $\mathtt{S_c}$, is defined as follows:

$$\mathtt{S_c} = \{\mathtt{s} \in \mathtt{WS_c} \mid dist(\mathtt{s},\mathtt{c}) < \mathtt{MaxDist}\} \cup \mathtt{CS_c} \tag{4}$$

Notice that $\mathtt{S_c}$ includes also the set of Web sessions that consists of selections of resources that appear in the directory

3.4 Resource Relevance Estimation

During a Web session, several resources are presented to a user. However, the user selects only a subset of such resource. Therefore, each Web session can be viewed as a sample of user preferences. The sessions classified into a category provide a collection of such preferences that can be used to estimate the relevance of resources to a category.

Given a resource u and a category c, the support of u in c, denoted $S(u,c)$, is the number of sessions in S_c in which the resource was presented to the user. Thus, we estimate the relevance of u to c as follows:

$$\mathtt{rel}^{new}(u,c) = \begin{cases} \frac{|\{s \in S_c | u \in U_s\}|}{S(u,c)} & \text{if } S(u,c) \geq \mathtt{MinSupp} \\ \mathtt{rel}^{old}(u,c) & \text{otherwise} \end{cases} \tag{5}$$

As we can see in Equation (5), we estimate the relevance by using the fraction of Web sessions of S_c where u was selected. To avoid non-representative estimations, we introduce a condition of minimum support, which requires a minimum number of sessions in which the resource was presented to users. In other case, the estimation is discarded preserving the previous value, if exists (considering 0 as default value).

Example. We illustrate how the relevance of resources to a category can be estimated through the following case. Suppose that for a given category c the Web session classification process was conducted determining that five sessions $\{s_1, s_2, s_3, s_4, s_5\}$ are relevant to c. For a set of resources $\{u_1, u_2, u_3, u_4, u_5\}$ we estimate the relevance to c as follows:

Session/Resource	u_1	u_2	u_3	u_4	u_5
s_1	$\boxminus$ / –	$\boxminus$ / $\boxtimes$	$\boxminus$ / $\boxtimes$	$\boxminus$ / $\boxtimes$	– / –
s_2	$\boxminus$ / $\boxtimes$	$\boxminus$ / –	$\boxminus$ / –	– / –	– / –
s_3	$\boxminus$ / $\boxtimes$	– / –	$\boxminus$ / $\boxtimes$	– / –	$\boxminus$ / $\boxtimes$
s_4	$\boxminus$ / –	– / –	$\boxminus$ / $\boxtimes$	– / –	$\boxminus$ / –
s_5	– / –	– / –	$\boxminus$ / –	$\boxminus$ / –	– / –
$S(u,c)$	4	2	5	2	2
$\texttt{rel}^{new}(u,c)$	0.5	0	0.6	0	0

Symbols $\boxminus$ and $\boxtimes$ represent that the resource was presented to the user and was clicked by the user, respectively. For example, u_2 was seen and clicked in s_1, was seen but not clicked in s_2 and was not seen (and not clicked) in s_3, s_4, s_5. Suppose that we consider a minimum support equals to 4 sessions (a relative support equals to 0.8) and old relevance values are equal to zero. In this case u_3 and u_1 are expected to be relevant to c, in this order.

$\boxdot$

3.5 Connecting Directory Categories and Web Queries

In the previous section, we introduce a method for updating resources listed in a directory. In this section, we show that the method can be slightly adapted to associate Web queries and categories of the directory, allowing users to switch from directory browsing to query formulation. Intuitively, the utility of a query q to a category c is large if many resources that are relevant to c appear in the answer returned by q. Now we introduce a notion to measure the utility of a query to a category. Given a category c and a query q, we define the utility of q to c as follows:

$$\texttt{utility}(q,c) = \sum_i \texttt{rel}(u_{i,q}, c), \tag{6}$$

where $u_{i,q}$ is the document that the query q returns at the i-th position of its document list. Therefore, the utility of a query q to a category c considers for each document u that appears in the answer list of q the relevance of u to c.

Example. To illustrate how our query utility function works, we continue with the previous case. Suppose that three queries registers in the query log clicks to documents in $\{u_1, u_2, u_3, u_4, u_5\}$. We calculate the utility of each query to c as follows:

Query/Resource	u_1	u_2	u_3	u_4	u_5	$\texttt{utility}(q,c)$
q_1	$\boxminus$ / 0.5	– / –	– / –	$\boxminus$ / 0	– / –	0.5
q_2	$\boxminus$ / 0.5	$\boxminus$ / 0	$\boxminus$ / 0.6	– / –	– / –	1.1
q_3	– / –	– / –	$\boxminus$ / 0.6	– / –	$\boxminus$ / 0	0.6

The symbol $\boxminus$ represents that the resource is recommended in the answer list of q. For example, q_2 returns documents u_1, u_2, and u_3. Adding the relevance of these resources, we obtain a utility value for q_2 equals to 1.1, which indicates to us that q_2 is expected to have the best resources for c.

$\boxdot$

The utility measure allows us to associate a ranked list of queries to each category in the directory. This list of queries lends usefulness to the users by allowing them to formulate different queries to retrieve relevant resources.

4 Experimental Results

4.1 TodoCL Dataset

To test our method we use data from the TodoCL[a] Chilean search engine. TodoCL mainly covers the .CL domain and some pages included in the .COM and .NET domains that are hosted by Chilean ISP providers. The directory provided by TodoCL has 468 categories distributed over 16 main categories (at the first level) and belongs to the Open Directory Project (ODP). Each category is associated to a list of Web pages suggested by users and validated by human editors.

The log file we considered for the experiments registers a period of 6 months of activity. It contains 245,170 non-empty sessions, in which 127,642 different queries were submitted. The sessions involve 617,796 selections, which are over 238,457 different URLs. Therefore, on average users clicked 2.51 documents per query session and 4.84 documents per query. The vocabulary of query terms and documents was processed to eliminate accents, digits, and punctuation. Moreover, the top-50 stop words of the collection were deleted. In what follows, we refer to these data as the *TodoCL dataset*.

Figure 2 shows some interesting characteristics of our dataset. In Figures 2a and 2b we plot query and term frequency, respectively, observing that both distributions follow power laws with decay factors equal to 1.42 and 1.27, respectively. This indicate to us that only a few queries (and terms) are used many times and an important fraction of the queries and terms register only few occurrences in the query log.

In Figure 2c we show the number of clicks per query session. Finally, Figure 2d shows how many documents and clicks are registered in the query log for each query. Notice that the click-through data distributions showed in Figures 2c and 2d follow power laws, indicating to us that only a few queries and query sessions registers a significant number of clicks in the query log.

4.2 Evaluating the Web Session Classifier

After running the category profile learning step (Section 3.2), using the directory log and directory content of TodoCL dataset, we obtained a profile-based model (i.e., a profile for each category in the directory). In order to obtain a test set to evaluate the model, we used the dataset of the 2005 KDD Cup [13] and combined it with the TodoCL dataset. The KDD dataset consists of 800 queries manually classified by experts into first and second level categories of the ODP directory. Many of the queries were classified into more than one category. The TodoCL directory and the KDD Cup dataset have 67 categories in common. For each of such categories and for each query classified in it (according to the KDD Cup dataset), we identified the query sessions related to the query and classified them into the category. At the end of this task, we obtained a collection of 5,012 sessions classified into the categories of the TodoCL directory. The directory with the classified query sessions compounds the test data to evaluate the model. Therefore, we applied the model to classify the 5,012 query sessions and compared the results with the test data.

Following the evaluation methodology used in related work (see [2] for a complete survey of this methodology) we considered each category as a binary classifier and obtained the number of true positives (`TP`), false positives (`FP`), true negatives (`TN`), and false negatives (`FN`) for

[a]http://www.todocl.cl

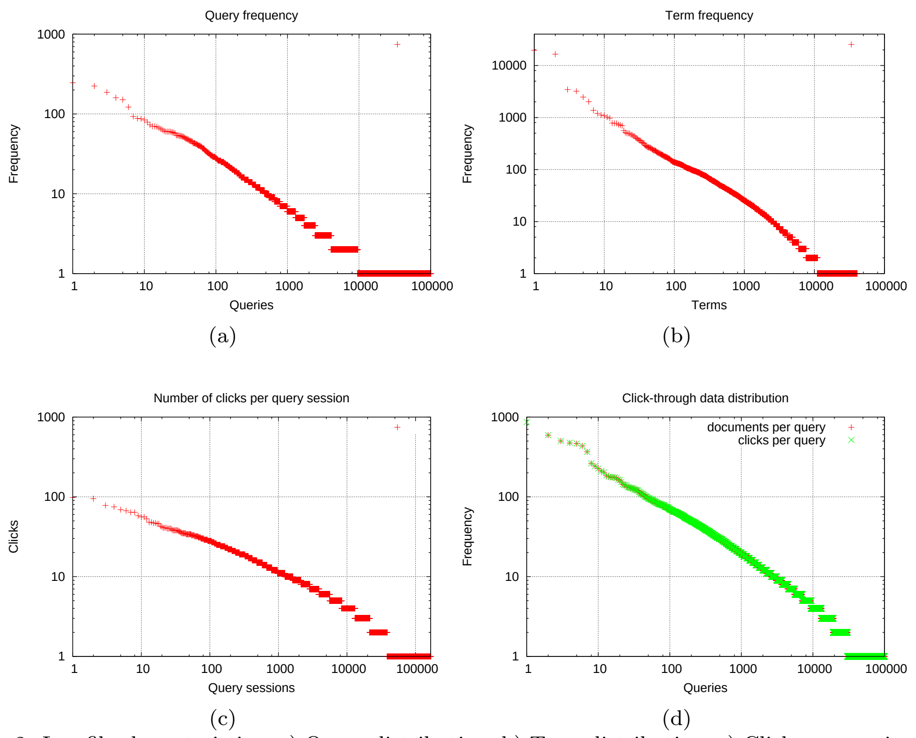


Fig. 2. Log file characteristics. a) Query distribution, b) Term distribution, c) Clicks per sessions, and d) Clicks and documents per query.

each category. Then we calculated the following performance measures: $\texttt{Precision} = \frac{\texttt{TP}}{\texttt{TP}+\texttt{FP}}$, $\texttt{Recall} = \frac{\texttt{TP}}{\texttt{TP}+\texttt{FN}}$ and $F_1 - \texttt{measure} = \frac{2}{\frac{1}{\texttt{Precision}}+\frac{1}{\texttt{Recall}}}$. In the case of the F-measure we considered the F_1 (the harmonic mean between the $\texttt{Precision}$ and $\texttt{Recall}$ measures), which captures a good balance between both criteria. Finally, to evaluate the global performance of the model, we calculated the average of the previous measures over all the categories considered in the experiment.

As the classifier is parameterized by a maximum distance threshold ($\texttt{MaxDist}$), Figure 3 shows the results obtained for the global measurements of $\texttt{Precision}$, $\texttt{Recall}$ and F_1 for values of $\texttt{MaxDist} \in [0.1, 1]$ (in increments of 0.1). We do not considered $\texttt{MaxDist} = 0$ because it rules out the entire dataset ($\texttt{Recall} = 0$).

Figure 3 shows that as $\texttt{MaxDist}$ increases, therefore allowing for a more permissive classifier, we achieve higher recall but lower precision. The $\mathtt{F_1}$ measure indicates to us that the precision-recall tradeoff achieves a good balance in the range $[0.5, 0.8]$, with an optimum value of $\widehat{\mathtt{F}}_1 = 0.63$ ($\texttt{Precision} = 0.74$ and $\texttt{Recall} = 0.55$), which is obtained for $\texttt{MaxDist} = 0.6$.

We then used the model to classify our 245,170 non-empty query sessions extracted from the TodoCL dataset. These sessions were placed into 468 categories in the directory. Figure 4 shows the distribution of distances between sessions and category profiles, obtained for the collection of classified sessions. The frequency axis indicates the fraction of session-category pairs at a given distance over the total number of session-category pairs. Using the distance

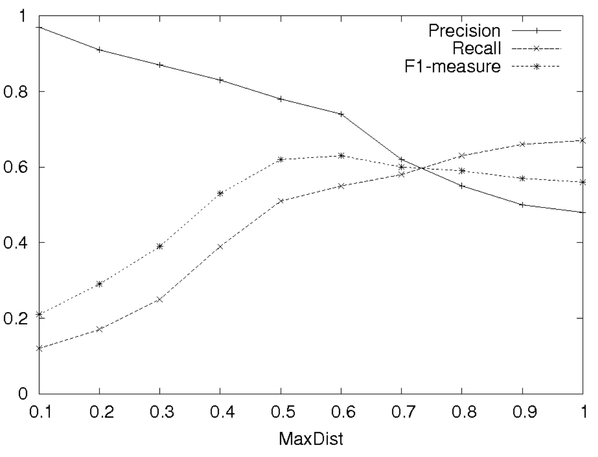


Fig. 3. Precision, Recall and F_1 measures for values of `MaxDist` $\in$ [`0.1`, `1`]

threshold found in the previous experiment (`MaxDist` = `0.6`), we found 31,912 sessions below the distance threshold. This set represents approximately 13% of the complete collection of sessions in the TodoCL dataset. This fraction of the dataset represents information needs that potentially can be related to the directory.

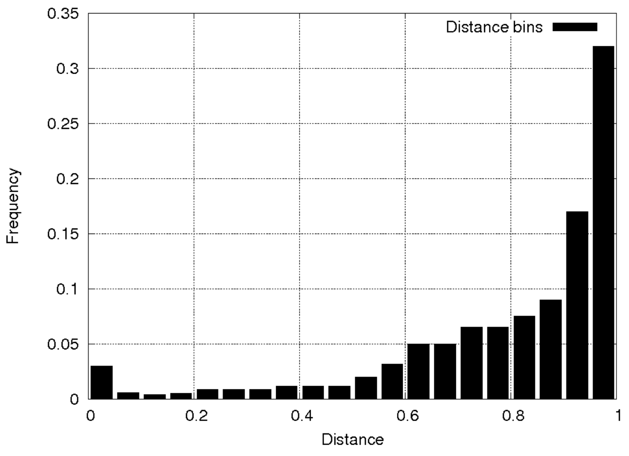


Fig. 4. Histogram of distances for the session - category pairs

The 31,912 sessions classified into the 468 categories of the TodoCL directory conform a list of query sessions for each node that definitely matches the number of observations that each category uses to elaborate its list of recommended documents. Each of these sessions was classified in one of the categories in the TodoCL directory. Figure 5 shows the distribution of the number of sessions for the first level of the directory.

As Figure 5 shows, the five categories with the largest number of sessions are, in decreasing order, Business and Economy, Computers and Internet, News and Media, Sports and Recreation, and finally, Society and Culture. The sessions classified into these categories account

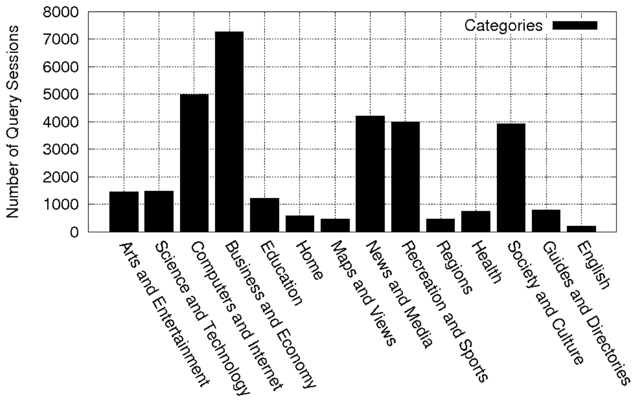


Fig. 5. Histogram of Number of Query Sessions per Category

for 78% of the total number of sessions classified in the directory. This indicates to us that these categories are expected to concentrate more clicked resources, and as a consequence, our method will have a greater amount of information in order to make recommendations.

4.3 Evaluating the Directory Maintenance Method

We evaluated the quality of the directory computed by our method using the methodology described in Baeza-Yates & Ribeiro-Neto [14], which consists of calculating performance measures for the top-N results. We considered Precision, Recall, and F-measure, and compared the measurements obtained by the directory returned by our method with the ones obtained by the TodoCL directory. As explained before, the TodoCL directory was elaborated by human editors so it allows us to compare our method that generates the document lists automatically with document lists generated manually. From this point of view, we consider the editor's lists as a ground truth for our evaluation strategy.

We extracted a sample of 50 categories from the set of 279 categories of the directory with at least 10 documents. This sample of 50 categories corresponds to the 17.9%, which is equivalent to the use of a sampling fraction of 0.18. For each category in the sample, experts from our labs evaluated the relevance of the top-10 document lists computed by our method. This evaluation allow us to reevaluate the editor's lists, avoiding the fact that these lists can introduce noise in our evaluation. Thus, our strategy allow to perform a comparison between our method and the editor's lists.

We consider a minimum support equals to zero, retrieving the top results ordered by our relevance measure. This involved the evaluation of 500 documents. We retrieve also the top-10 documents recommended by the directory for each of the 50 categories evaluated. Both document collections have an overlap of 212 documents, so at the end, the collection evaluated comprised 788 different documents.

For each category, the document list was randomized in a way such that the expert's evaluation was not biased by the position of the documents in the answer lists. The experts carried out their evaluation by expressing judgments of relevance on a scale from 0-4, from least to greatest relevance. Later, for each document-category profile pair, we have calculated the average relevance for the evaluators of the pair. Each pair was described using the URL and title of the document and the title of the category, allowing each reviewer could clicking the URL and read the content of the document. For each category we provide also a hyperlink to the directory node, allowing that the content of the category could be revised. We were careful not to show the same documents that they were evaluated, preventing by this that the judgments of experts may be biased.

Of the 788 pairs evaluated, 767 obtained consensual results (this is 0 or 1 in case of no relevance, 3 or 4 in case of relevance) equivalent to 97.3% of the total. The remaining 21 pairs (2.7%) were reevaluated achieving a consensus in a second round. The experts determined that 561 pairs were relevant, which is equivalent to 71.1% of the evaluation set.

We compare the performance of our method to a standard unsupervised text classifier, discarding the comparison with a supervised learning method to avoid the use of labeled data and the need to construct a labeled dataset. Then, the evaluation compares our method with a standard Rocchio classifier. The Rocchio classifier associates each document with its closest category, according to a distance function. The baseline uses a Tf-Idf vector representation for the documents using document terms, which is considered state of the art for text representation. Notice that our method also use a Tf-Idf weighting schema for text representation, but performing aggregation in each Web session (see Equation 1). This comparison allows us to assess, for the same text representation schema, the impact of the use of clicks in the performance of our method. To represent categories we consider the terms retrieved from each category description and for the documents recommended on each of them, also considering a Tf-Idf vector weighting schema. We test two distance functions, Cosine and Euclidean.

Performance evaluation was conducted for each test document at two levels of granularity. At macro-level, we capture overall scores for each test document, evaluating the performance measures considering all the documents in the dataset. At micro-level, we evaluate each performance measure per category. In the end the results are averaged over the total number of categories. Results are shown in Table 1.

Table 1. Micro and macro evaluation results for web page classification.

Macro-evaluation					
Method	TP-Rate	FP-Rate	Precision	Recall	F_1
Query log	0.718	0.146	0.713	0.718	0.715
Cosine	0.657	0.186	0.651	0.657	0.654
Euclidean	0.563	0.196	0.573	0.563	0.568
Micro-evaluation					
Method	TP-Rate	FP-Rate	Precision	Recall	F_1
Query log	0.735	0.178	0.673	0.735	0.703
Cosine	0.559	0.215	0.581	0.559	0.57
Euclidean	0.529	0.195	0.559	0.529	0.544

As Table 1 shows, the proposed method outcomes standard text classifiers by several points in all the comparisons. Notice that Cosine distance outcomes Euclidean distance by several points at macro level, and this difference decreases at micro level. In particular, our method achieves better results at micro level, illustrating that the use of query logs allow us to get better results for the full directory. From our point of view a good method should perform well both at macro level and micro level. The results obtained show that our method achieves the best results in both levels of granularity.

We test the statistical significance of each F_1 measure obtained at macro level in this evaluation by performing a Fisher's randomization significance test. We compare our method to each baseline, given a null hypothesis that assumes that there is no difference in both methods. Since there are 788 document - category pairs, there are 2^{788} ways to label the results under the null hypothesis. Then we measure the difference between the methods for each permutation. To avoid the evaluation of the 2^{788} permutations, we create 10,000 random permutations, measuring the difference in F-measure for each arrangement. In our evaluation the difference in F-measure is 0.061 and 0.147, for comparisons against Cosine and Euclidean baselines, respectively. Of the 10,000 random permutations 72 and 87 are $\leq$ -0.061 and -0.147, and 76 and 80 are $\geq$ 0.061 and 0.147 for Cosine and Euclidean-based comparisons, respectively. These differences give us two-sided p-values of $\frac{72+76}{10000} = 0.0148$ and $\frac{87+80}{10000} = 0.0167$ for Cosine and Euclidean-base comparisons, thus, accordingly, we reject the null hypothesis.

We analyze how the evaluated methods performs in each category. To do this we plot the F-measure for each of the 14 first level categories of the directory. Results are shown in Figure 6.

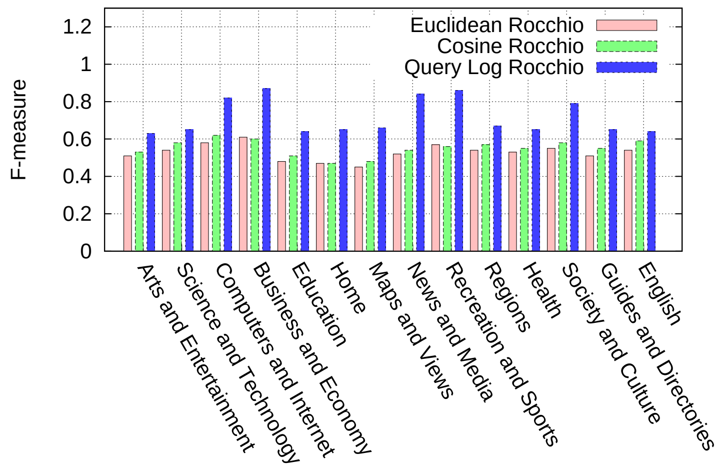


Fig. 6. F-measure performance per category

As Figure 6 shows, the results obtained by the baselines are independent of the number of clicked resources that each category registers. On the other hand, our method achieves the best results in the categories with a most significant fraction of query sessions in the dataset (see Figure 5). Notice that in categories such as "Computers and Internet" or "Business and Economy" the performance of our method is greater than 80%, outcoming baselines by more than 20 percentual points. This fact indicates to us that the use of clicks allows significant improvements in performance. Improvements in those categories with fewer clicked resources are less significant.

In a second stage, we study also the impact of the minimum support threshold in the performance of the proposed method. In Table 2 we show the number of documents evaluated as relevant by the experts for five different values of the `MinSupp` parameter. The recall is calculated for a total of 561 relevant pairs at macro level of granularity.

Table 2. Global performance measures calculated for five values of `MinSupp`. The second and third columns represent the number of relevant and recommended resources, respectively. Precision, Recall and F-measure are depicted in the last columns, in this order.

`MinSupp`	Rel.	Rec.	Prec.	Recall	F_1
0.1	356	499	0.713	0.634	0.671
0.2	386	437	0.883	0.688	0.773
0.3	378	401	0.942	0.674	**0.786**
0.4	331	352	0.940	0.590	0.725
0.5	290	302	0.960	0.517	0.672

In Table 2 we observe that for our dataset the precision-recall trade off achieves the best performance for `MinSupp` = `0.3`, where the F_1 measure obtains its maximum value (78.59%). This fact illustrates that the `MinSupp` parameter allows us to discard page recommendations based on few observations (with low support values) without discarding relevant recommendations.

The setting of a minimum support value is an important process for our method. In this work we have been able to make a proper tuning of this value because we have a set of labeled data that allows us to validate the quality of our recommendations. However the value of this parameter is not generalizable to other data sets requiring new tuning processes. A way to drive this tuning process without the need for labeled data would be to conduct a post-testing step, requiring the collaboration of experts or collecting implicit user feedback measures by processing click-through data.

Table 2 also shows that the method proposed in this paper obtains a high precision. Several non relevant documents were filtered out thanks to the support threshold, which can be seen in Table 2, where the precision rises from 71.34% for `MinSupp` = `0.1` up to 94.26% for `MinSupp` = `0.3`.

In Table 3 we show the relationship between the directory computed by our method (using `MinSupp` = `0.3`) and the TodoCL directory. In Table 3, A (automatic) and H (human) represent the directory calculated by our method (A) and TodoCL directory (H), respectively. The numbers in the table count pairs category-document in each of the subcollections. These results provide evidence that the proposed method compares favorably with the TodoCL directory.

Table 3. Global performance measures calculated for the two directories.

Set	Rel.	Rec.	Prec.	Recall	F_1
A	378	401	94.26	67.38	78.59
H	470	500	94.01	83.78	88.61
H ∩ A	289	297	97.31	51.52	67.37
H - A	181	203	89.16	32.26	47.38
A - H	89	104	85.58	15.86	26.77

In Table 3 we can also observe the following important facts. First, the intersection between the two directories is significant and its precision is the highest among all the sub-collections shown in the table (97.31%). Second, the precisions of A and H are similar, above 94% in both cases. Furthermore, the precision of our method is slightly better than the one of TodoCL. In addition, the method we propose is able to recommend documents that the editors have not identified as relevant (89 total), which represents a recall of 15.86%. Finally, if we combine both collections (relevant documents of A and H) we recover 559 of the 561 pairs ($A+H-A\cap H$) evaluated as relevant by the experts. This suggests that both directories could be combined to obtain a better directory. The condition `MinSupp` = `0.3` is effective to filter out non relevant documents.

Figure 7 shows that both directories present high precision, though the proposed method is slightly more precise than TodoCL directory. Regarding recall, as we increase the list of documents, the recall increases in both cases, with better values for the TodoCL directory. This is reflected in the F-measure, which is slightly better for the TodoCL directory. Figure 7 shows that both directories are comparable in recall and precision, which is a good result for our work. It support our claim that without involvement of human editors, and by analyzing browsing behavior, we can update a directory without affecting its precision, and also guaranteeing a good recall.

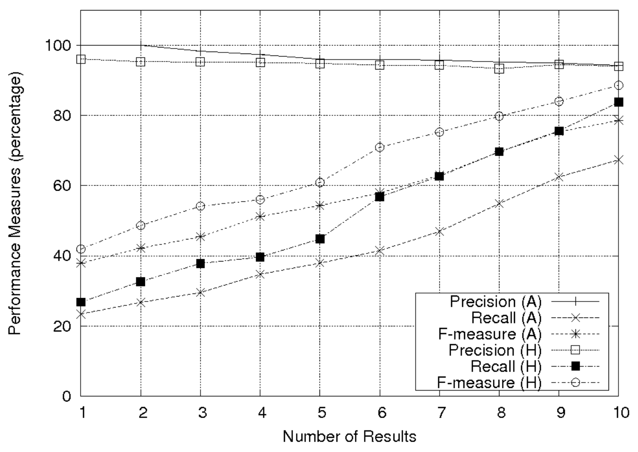


Fig. 7. Average performance results over the top ten results

4.4 Evaluating Queries Suggested to Categories

We also evaluate the precision of the query lists suggested by our method to the directory categories. For each category in the sample of 50 categories used in the previous experiment, we evaluated the precision of the list of top-5 suggested queries ordered by utility. Experts from our laboratory evaluated the relevance of the queries to the category using an evaluation scale 0-4, from least to greatest relevance. After averaging the evaluations done for each category-query pair, the results obtained were interpreted in the following format: 0 or 1 represents a false positive, 3 or 4 represents a true positive, 2 must be reevaluated. Unlike the previous experiment, the experts coincidenced in all of their judgments, so there was no need to reevaluate any of the pairs. Of the 250 total pairs evaluated, only 27 have been classified as false positives, which is equivalent to a global precision of 89.5%.

In Table 4 we show the average precision obtained for the 50 categories and for different sizes of the list of queries suggested. We consider as a baseline a text-based query classifier. The baseline use a Tf-Idf vector representation for the queries using query terms. To represent categories we consider the terms retrieved from each category description and for the documents recommended in each of them, also considering a Tf-Idf vector weighting schema. We consider a Cosine and Euclidean distance functions for the evaluation. Performance evaluation was conducted for each query at macro-level and micro-level of granularity. Results are shown in Table 4.

Table 4. Average precision for the query recommendation method over the top five results.

Macro-evaluation					
Method	P@1	P@2	P@3	P@4	P@5
Query logs	98.65	96.75	96.52	94.12	89.52
Cosine	86.23	82.24	80.87	79.42	78.32
Euclidean	84.85	81.64	80.01	78.54	77.23
Micro-evaluation					
Method	P@1	P@2	P@3	P@4	P@5
Query logs	96.23	95.86	95.23	94.64	92.54
Cosine	87.64	86.23	84.42	84.12	83.52
Euclidean	86.53	85.32	84.02	83.22	82.11

As we can observe, Table 4 shows the trade off between precision and number of results: the precision declines slightly as more queries are considered in the result. We also note that the results obtained are highly precise, observing an average precision higher than 94% for the top-4 queries suggested. Our method increases the precision over the baselines by more than 10 percentual points in several comparisons, being this difference more significant at macro-evaluation granularity.

We calculate the average precision of each query list and then we calculate the mean value across the fifty tested categories obtaining a MAP measure. MAP values are 0.92, 0.81 and 0.8 for Query logs, Cosine and Euclidean methods, respectively. We test the statistical significance of MAP values by performing a Fisher's randomization significance test. Again we establish a null hypothesis that assumes that there is no difference between both methods. To test the hypothesis we create 5,000 random permutations (we have 50 different category pairs), measuring the difference in MAP for each arrangement. In our evaluation the difference is 0.11 and 0.12, for comparisons against Cosine and Euclidean baselines, respectively. We obtained two-sided p-values of 0.0112 and 0.0108 for Cosine and Euclidean-based comparisons, thus, accordingly, we reject the null hypothesis.

We evaluate also how the recommended queries are useful to find new relevant resources. To do this, we calculate the gain which each query contributes to the recommended answer list. We define the gain of each query as the fraction of new relevant resources that each query provides to the total set of relevant resources. To do this we conduct the following experiment. For each of the 50 evaluated categories, we retrieved the top-3 recommended queries by our method. Then, for each of these queries, we retrieved the top-10 results from the TodoCl search engine. The relevance of each of these documents to each category was evaluated by our experts, using an evaluation scale 0-4, from least to greatest relevance. This process involved the evaluation of 1500 documents. This document collection overlaps the directory in 354 documents (previously evaluated by our experts in the directory maintenance method testing step), so at the end, the collection evaluated comprised 1146 different documents. As in previous evaluations, for each document-category profile pair we have calculated the average relevance for the evaluators of the pair. We used the same testing interface as in the previous document evaluation process, so each pair was described using the URL and title of the document and the title of the category, allowing each reviewer could clicking the URL and read the content of the document. For each category we provide also a hyperlink to the directory node, allowing that the content of the category could be revised.

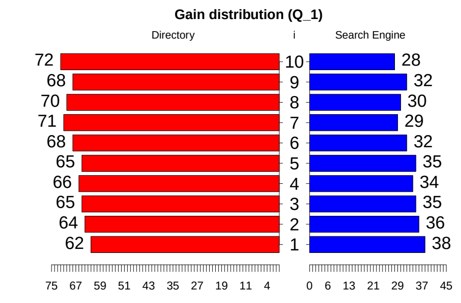
Of the 1146 pairs evaluated, 1098 obtained consensual results equivalent to 95.8% of the total. The remaining 48 pairs (4.2%) were reevaluated achieving a consensus in a second round. The experts determined that 752 pairs were relevant, which is equivalent to 65.6% of the evaluation set. Note that the fraction of relevant documents is lower by 6 percentual points to the results obtained by the documents recommended in the directory (see Section 4.3), this because some recommended queries are not relevant to the categories, as was shown in the previous evaluation.

To evaluate the ability of each query to identify new relevant resources, we count the number of new relevant documents for top i positions, for $i \in \{1, \dots, 10\}$. We do not consider overlapping documents, so if a document was recommended in the directory, we discard the document from the query recommendation list. Then, for a given position i and a given query - category pair, we have the number of relevant documents for the pair, determining the fraction of these documents recommended in the directory and in the search engine. Note that these measures correspond to the fraction of recall that each list contributes over the total number of relevant documents considered until this position. We average these results over the 50 category - query pairs evaluated. These results are shown in Figure 8.

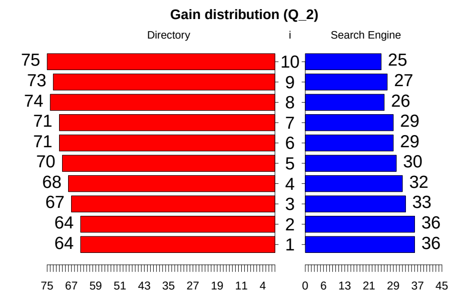
Figures 8a, 8b, and 8c show results for the first, second, and third query recommended, respectively. The average of these results is shown in Figure 8d. The vertical axis dissagregates the results by the top-i positions considered in answer result lists. The horizontal axis shows the fraction of new resources recommended by position. Thus, a centered bar indicates that both repositories allow to discover the same number of new resources. Another situation illustrates the contribution of each repository to the total collection of new relevant resources. As Figure 8d shows, when users look for new resources in each query answer list by switching from the directory to the search engine, they discover about $\frac{1}{3}$ of the total relevant results, illustrating that query suggestions are able to identify new resources which originally were not included in the directory. On the other hand, about $\frac{2}{3}$ of the relevant resources are recommended in the directory, illustrating how useful and accurate are the recommendations of our method. Notice that Figures 8a, 8b, and 8c illustrate that the contribution for the discovery of new resources decreases when the utility of the query decreases. This fact indicates to us that our query utility function correlates with the expected contribution of each query to the discovery of new resources in the search engine.

5 Discussion

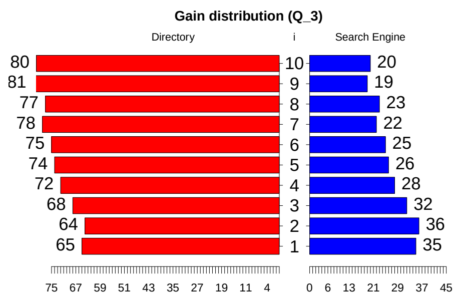
In this section we explain the scope and main limitations of the proposed method. We introduced a method for the automatic maintenance of Web directories assuming that we



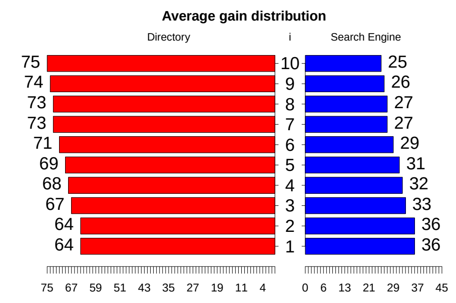
(a)



(b)



(c)



(d)

Fig. 8. Gain distribution comparison between directory and search engine recommended resources: a) First query recommended, b) Second query recommended, c) Third query recommended, and d) Average gain distribution for the top 3 recommended queries.

have full access to logs. Two requirements are needed in this point, first, a portion of the logs is extracted by processing directory user traces. Second, we need logs extracted from another repository, such as a search engine. These requirements are flexible in the following sense: If we don't have access to directory logs we can use only search engine logs to estimate our resource relevance measures. This can help us in situations where the directory has few users, which is usual at the start-up of every Web information system.

Our method is based on user traces so there are some limitations which are inherent to the use of clicks. First, our method tends to recommend resources which register a significant amount of clicks. In fact, our recommendations are based on support notions, then in practice we can identify only resources which appear to be relevant for most people. Thus, relevant resources for marginal information needs are very dificult to detect. This case is usual in ambiguous/polisemic search engine queries, where most people tend to select resources that are related to the most common information need behind the query. Another limitation of our method relies on the recommendation of new resources which does not register clicks. This fact happens in the following situations: When new resources are added to a search engine, or when these new resources are added to the directory. In both cases the lack of click-through data in one repository can be addressed by using the other repository. This limitation cannot be addressed when the resource is completely new, i.e. we don't have enough clicks for support estimation in both log repositories. Currently we are working on extensions of our method to deal with new resources. Since in these cases we do not have log history, we are trying to solve this problem by determining closest resources to them.

6 Conclusion

In this work we have presented a method able to maintain up-to-date Web directories without the intervention of human editors. In addition, the method proposed makes it possible to estimate how useful a query is to a category of a directory, so that the users can switch from directory browsing to query formulation in order to obtain new useful documents, or to disambiguate or specialize the informational need captured by the category. Experimental results provide evidence that the proposed method is effective and attains identification of new relevant documents and queries with high precision.

There are several open challenges for future work. A first challenge is to find further representations of sessions and categories to obtain a better precision in the session classification step. Also, it is interesting to research on the automatic discovery of new information needs that are not captured by the directory being maintained, such as emerging topics or specializations of current categories. This raises the problem of updating the structure of the directory by adding and deleting categories and changing the level of categories according to the browsing behavior observed in Web logs.

Acknowledgements

We thank the anonymous reviewer for useful suggestions and Ricardo Baeza-Yates, Ariel Cid and Angelo Figueroa for his comments on an early draft of this paper. Mr. Mendoza was partially supported by project UTFSM-DGIP 24.11.19.

References

1. X. Qi and B. Davison (2009), *Web page classification: Features and algorithms*, ACM Computing Surveys,41(2):1-31.
2. Sebastiani, F., (2002). *Machine learning in automated text categorization.* ACM Computing Surveys,34(1):1-47.
3. Yang, H.-C., Lee, C.-H., (2004). *A text mining approach on automatic generation of web directories and hierarchies.* Expert Syst. Appl. 27 (4), 645663.
4. Stamou, S., Ntoulas, A., Krikos, V., Kokosis, P., Christodoulakis, D., (2006). *Classifying web data in directory structures.* In: Zhou, X., Li, J., Shen, H. T., Kitsuregawa, M., Zhang, Y. (Eds.), APWeb. Vol. 3841 of Lecture Notes in Computer Science. Springer, pp. 238-249.
5. Chung, W., Lai, G., Bonillas, A., Xi, W., Chen, H., (2008). *Organizing domain-specific information on the web: An experiment on the spanish business web directory.* Int. J. Hum.-Comput. Stud. 66 (2), 5166.
6. Gerstel, O., Kutten, S., Laber, E., Matichin, R., Peleg, D., Pessoa, A., de Souza, C. (2007), *Reducing human interactions in Web directory searches.* ACM Trans. Inf. Syst. 25 (4), 1-28.
7. Zaihrayeu, I., Sun, L., Giunchiglia, F., Pan, W., Ju, Q., Chi, M., Huang, X., (2007). *From web directories to ontologies: Natural language processing challenges.* In: et al., K. A. (Ed.), ISWC/ASWC. Vol. 4825 of Lecture Notes in Computer Science. Springer, pp. 623636.
8. Chuang, S.-L., Chien, L.-F., (2003). *Enriching web taxonomies through subject categorization of query terms from search engine logs.* Decision Support Systems 35 (1), 113127.

9. Adami, G., Avesani, P., Sona, D., (2003). *Clustering documents in a web directory.* In: Chiang, R. H. L., Laender, A. H. F., Lim, E.-P. (Eds.), WIDM. ACM, pp. 6673.
10. Adami, G., Avesani, P., Sona, D., (2005). *Clustering documents into a web directory for bootstrapping a supervised classification.* Data Knowl. Eng. 54 (3), 301325.
11. Zhang, D., Lee, W. S., (2004). *Learning to integrate web taxonomies.* J. Web Sem. 2 (2), 131151.
12. Rocchio, J., (1971). *Relevance feedback in information retrieval.* In: G. Salton (Ed.), The SMART Retrieval System - Experiments in Automatic Document Processing. Prentice Hall Inc., Englewood Clifs, NJ, USA.
13. SIGKDD, (2005). KDD CUP 2005 dataset. http://www.sigkdd.org/kdd2005/kddcup.html.
14. Baeza-Yates, R., Ribeiro-Neto, B., (1999). *Modern Information Retrieval.* Addison-Wesley, ACM Press, New York.