Personalized Web Search via Query Expansion based on User's Local Hierarchically-Organized Files*

Gianluca Moro[1], Roberto Pasolini[1] and Claudio Sartori[2]
[1]*Department of Computer Science and Engineering, University of Bologna, Via Venezia 52, Cesena (FC), Italy*
[2]*Department of Computer Science and Engineering, University of Bologna, Viale Risorgimento 2, Bologna (BO), Italy*

Keywords: Information Retrieval, Personalized Search, Query Expansion, Local Files, Search Engine.

Abstract: Users of Web search engines generally express information needs with short and ambiguous queries, leading to irrelevant results. Personalized search methods improve users' experience by automatically reformulating queries before sending them to the search engine or rearranging received results, according to their specific interests. A user profile is often built from previous queries, clicked results or in general from the user's browsing history; different topics must be distinguished in order to obtain an accurate profile. It is quite common that a set of user files, locally stored in sub-directory, are organized by the user into a coherent taxonomy corresponding to own topics of interest, but only a few methods leverage on this potentially useful source of knowledge. We propose a novel method where a user profile is built from those files, specifically considering their consistent arrangement in directories. A bag of keywords is extracted for each directory from text documents within it. We can infer the topic of each query and expand it by adding the corresponding keywords, in order to obtain a more targeted formulation. Experiments are carried out using benchmark data through a repeatable systematic process, in order to evaluate objectively how much our method can improve relevance of query results when applied upon a third-party search engine.

1 INTRODUCTION

Millions of people daily use Web search engines to find Web pages they are looking for. Every search starts from an *information need* expressed by the user as a textual query. Due to haste and laziness, users express complex information needs with few keywords, often resulting in an ambiguous query yielding many irrelevant results (Jansen et al., 2000; Carpineto and Romano, 2012). In a typical example, a user interested in apple-based recipes may simply search for "apple" and to have then to set apart resulting pages which actually deal with Apple computers.

We can disambiguate the query, in order to better express the information need and get proper results, by enriching the query with context. The query could for example include the intended topic of discussion, e.g. for "apple" it could be "fruits" or "computers". As every user often has a specific set of topics of interest, these can be used to properly disambiguate simple queries without requiring him or her to explicitly provide a context: a cooking-enthusiast user could effortlessly query for "apple" to only retrieve pages about apple-based recipes.

Personalized search provides a customized experience to the specific user or for the specific task being carried on, leveraging contextual information. Personalization is usually wrapped around an existing search engine using one or both of two approaches: queries can be expanded or otherwise rewritten before sending them to the search engine, while results can be filtered or reordered to move the most pertinent on top (Pitkow et al., 2002).

Personalized search users are described by *profiles* usually indicating topics or *concepts* of their interest, such as "travels", "acoustic guitar", "videogames" and alike. Users can be profiled by monitoring their activity unobtrusively, without their explicit feedback. It is common to leverage the history of previous searches, including issued queries and visited results, while other solutions mine knowledge from the whole history of browsed pages (Ghorab et al., 2013).

A less common approach is to extract information from documents stored locally by the user on his or

*This work was partially supported by the project "Toreador", funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 688797.

Moro G., Pasolini R. and Sartori C.
Personalized Web Search via Query Expansion based on UserâĂŹs Local Hierarchically-Organized Files.
DOI: 10.5220/0006486401570164
In *Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KDIR 2017)*, pages 157-164
ISBN: 978-989-758-271-4

her own PC or device (Chirita et al., 2007). Users often store many text files, either self-produced or obtained externally, representing their recurring interests. These files are stored within an arbitrarily structured hierarchy of directories, in order to easily browse the collection and find any needed document. We claim that these directories naturally represent specific topics of interest for the user and that their hierarchical structure approximates a coherent taxonomy of these topics.

We propose a personalized search solution leveraging these hierarchically organized local documents. Text files are analyzed in order to extract recurring keywords for each directory, deemed to represent a specific topic of interest. Whenever a search engine is used, keywords of its inferred topic are added to the issued query beforehand, in order to disambiguate it according to user's interests. We use well known information retrieval (hence IR) techniques to extract keywords, weigh their relevance and determine the most likely topic for each query.

Compared to similar solutions, our method requires neither a predefined taxonomy of concepts nor the classification of data into such taxonomy, which can be complex and error-prone. Instead, our method exploits the work which users already do to organize personal documents, in order to build taxonomies of topics upon their own needs: no computational effort is required to distinguish topics and to classify documents within them. The personalization system just extracts relevant keywords for each topic and analyzes and expands search queries issued by the user: these tasks are performed through simple and efficient techniques, allowing the system to scale up to a large volume of user documents.

To obtain an objective, quantitative and repeatable evaluation of the goodness of the method, we employ a systematic assessment process using predefined benchmark data. We employ resources generally used in IR and freely available on the Web: a collection of Web pages, a search engine indexing them and a set of test queries with known relevant documents. For each test query, the relevance of results is measured with multiple performance indicators. Tests are run on the raw search engine and on different configurations of the personalization system, in order to assess its potential benefits.

The article is organized as follows. Section 2 discusses and compares against similar methods found in literature. Section 3 describes the proposed search personalization method. Section 4 explains our objective evaluation process, whose outcomes are reported in Section 5. Finally, Section 6 sums up our work and suggests possible future directions.

2 RELATED WORK

Personalization of users' experience in Web search has been widely explored, addressing both *contextualization* of search according to the task being carried on and *individualization* of the experience according to the user executing the task (Pitkow et al., 2002). Earlier approaches used *relevance feedback* to refine results upon users' feedback (Salton and Buckley, 1997), while subsequent are often based on getting implicit feedback by monitoring users' behavior.

Personalized search (hence PS) generally works by building a profile of the user and employing it to customize the search experience. *Outride* (Pitkow et al., 2002) defines a general architecture of a personalization system mediating between the user and a general search engine. User-provided query can be refined to a more precise expression of the information need (*query rewriting*), while results provided by the engine can be filtered and reordered to promote those deemed to effectively interest the user. Many works follow this scheme, differing in how the user model is built and used (Gauch et al., 2007; Steichen et al., 2012; Ghorab et al., 2013).

Topics are often a key element in user modeling: concept-based profiles represent arbitrary mixes of different topics of interest. Possible topics are often given by a predefined taxonomy like DMOZ (discussed in Section 4.1). The *OBIWAN* project (Pretschner and Gauch, 1999; Gauch et al., 2003) is an early example: user's interests are located on a hierarchy of about 4,400 nodes used as search contexts.

Many works, including ours, combine keywords-based representation with a concept-based approach: a user profile is represented by a multiple keyword vector for each topic of interest. (Trajkova and Gauch, 2004) describe how to profile users by classifying visited Web pages in a DMOZ-based ontology. In (Liu et al., 2002) a user profile is used in conjunction with a global profile; for each query, the most likely topics are proposed to the user. In (Speretta and Gauch, 2005) a list of weighted concepts is extracted from search results clicked by the user.

Among other existing approaches to user profiling, there are *semantic networks*, represented by interlinked keywords and/or concepts, useful to explicitly address polysemy and synonymy issues (Micarelli and Sciarrone, 2004).

The user profile can be built from different information sources, like user's behavior while browsing the Web and information from local history or cache (Sugiyama et al., 2004). Some solutions leverage search history, possibly detecting the most interesting results: for example (Speretta and Gauch, 2005)

gather titles and summaries of results clicked by the user, while in (Stamou and Ntoulas, 2009) the full content of selected search results is analyzed.

Few methods explicitly use locally stored user's documents, usually merging them with other information. In (Teevan et al., 2005), both Web and local documents are similarly indexed to build models for Web search personalization: even approximate client-side-built models can improve relevance of search results. In (Chirita et al., 2007) is proposed to extract keywords from the user's "personal information repository", including text files as well as e-mails, cached Web pages and other data. Different techniques to determine the correct additional terms are evaluated and a method to adaptively choose the number of these terms is investigated.

These methods aim to build the user's model upon large amounts of unlabeled data to build the user's model, ignoring the topic of each document. In contrast, considering the generally higher value of labeled data (Domeniconi et al., 2017), our approach focuses solely on personal files organized in a hierarchy of directories: the quantity of analyzable documents may be smaller, but the use of correctly labeled data can potentially bring significant improvements in the accuracy of the resulting model and consequently in the relevance of personalized search results.

Other PS solutions based on user-labeled data exist, like *social tagging* systems where users can assign free-form tags to Web resources, thus obtaining a very large corpus of tagged pages (Zhou et al., 2012). Our solution works with fewer labeled data, but it is specifically representative of the target user and its hierarchical nature is useful to distinguish different levels of detail; also, our solution reduces users' privacy concerns and removes additional server-side effort.

3 METHOD DESCRIPTION

After formalizing the considered context, we describe how local knowledge is extracted and used for query expansion. Figure 1 shows the interconnections between the components of the method.

3.1 Application Context

Our personalization system works on top of any search engine Σ responding to any query q with a list $\Sigma(q)$ of relevant documents. We formally consider each query as either a single keyword or a set of subqueries combined by AND ($\wedge$) or OR ($\vee$) operators.

A collection $\mathbf{UD} = \langle \mathcal{F}, \mathcal{D}, \prec, C \rangle$ of local user documents is defined by a set $\mathcal{F}$ of *files*, a tree $\mathcal{D}$ of *directories* partially ordered by the antisymmetric, transitive relation $\prec \subset \mathcal{D} \times \mathcal{D}$ and a function $C : \mathcal{F} \to \mathcal{D}$ mapping each file to the directory containing it. We denote with $L(d)$ the set of files directly contained in the directory d and with $L_{\mathrm{R}}(d)$ the set of files contained in d itself or a subdirectory within.

$$L(d) = \{f \in \mathcal{F} : C(f) = d\} \quad (1)$$

$$L_{\mathrm{R}}(d) = \{f \in \mathcal{F} : C(f) = d \vee C(f) \prec d\} \quad (2)$$

This model can represent a user "home" directory, containing his or her personal documents conveniently arranged in sub-directories. For the purposes of our method we only consider text files.

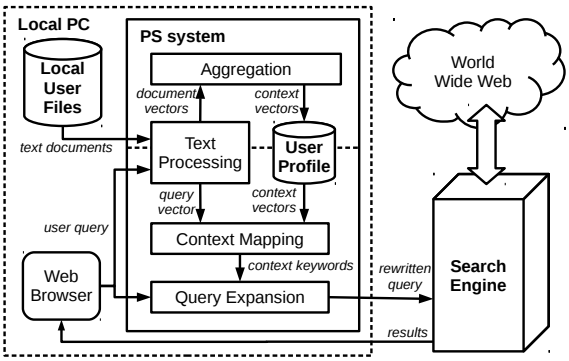

Figure 1: General diagram of the proposed method. The upper part of the PS system concerns the construction of the user profile, while the lower part deals with its application.

3.2 Extraction of Local Knowledge

We describe how to build a local knowledge base U constituted by a set of *contexts*, each representing a topic. Each context corresponds to a directory $d \in \mathcal{D}$. A context is constituted by the words which represent the modeled topic, extracted from the text documents in the corresponding directory.

Each user file in $\mathcal{F}$ is first pre-processed to extract single words contained within it, filtered using standard IR techniques: casefolding, stopwords removal and stemming. From each document we obtain a multiset of filtered words or *terms* extracted from it. We denote with $\#(t, f)$ the number of occurrences of term t in file f; for convenience, $t \in f$ indicates that t appears at least once in f, i.e. $\#(t, f) > 0$.

Now, knowing the set $\mathcal{T}$ of distinct terms found within the files, we represent each file $f \in \mathcal{F}$ with a vector $\mathbf{w}_f = (w_{f,1}, \ldots, w_{f,|\mathcal{T}|}) \in \mathbb{R}^{|\mathcal{T}|}$, indicating the relevance of each term within f. Such relevance is computed according to one of the possible weighting schemes explained shortly.

A vector $\mathbf{c}_d$ for each directory $d \in \mathcal{D}$ is trivially computed as the sum of vectors of every single file

within the directory itself.

$$\mathbf{c}_d = \sum_{f \in L(d)} \mathbf{w}_f \tag{3}$$

The set of these vectors constitutes the user profile $U : \mathcal{D} \to \mathbb{R}^{|\mathcal{T}|}$: in fact, each vector corresponds to a topic in the hierarchy of user interests and indicates which are the most relevant words for that topic. These vectors will be used for query expansion.

3.3 Term Weighting

Term weighting schemes are employed to properly estimate the relevance of terms in single documents. The use of suitable weighting schemes can be considerably beneficial in many practical contexts (Domeniconi et al., 2016). The weight of a term within a document f is generally composed of a *local* factor based solely on the contents of f and a *global* factor computed on the whole set of documents.

As the local weight factor for a term t in f, known as *term frequency* (*tf*), we use the number of occurrences of t normalized according to the maximum number of occurrences of a term in f itself.

$$\mathrm{tf}(t,f) = \frac{\#(t,f)}{\max_{\tau \in f} \#(\tau,f)} \tag{4}$$

The most common choice for global factor would be the *inverse document frequency* (idf), assigning higher weights to terms appearing in fewer documents. However, taking inspiration from *supervised* term weighting schemes used in automated text categorization to compute different weights for each topic category (Domeniconi et al., 2016), we also consider the directory where a file is located when computing weights for its terms.

A first supervised global measure we propose, called *idfd* (***idf** in **d**irectory*), is in practice the standard idf measured only on documents of the directory being considered, recursively including its subdirectories. This allows to evaluate the discriminative power of a term within the specific topic being considered, rather than in the whole taxonomy.

$$\mathrm{idfd}(t,d) = \log\left(\frac{|L_\mathrm{R}(d)|}{|f \in L_\mathrm{R}(d) : t \in f|}\right) \tag{5}$$

The second option we propose, called *idfod* (***idf** **o**utside of **d**irectory*), is to instead compute idf excluding documents in the considered directory and its nested subdirectories. The rationale is to avoid underestimating terms which appear frequently in the considered directory, but rarely elsewhere.

$$\mathrm{idfod}(t,d) = \log\left(\frac{|\mathcal{F} \setminus L_\mathrm{R}(d)|}{|f \in \mathcal{F} \setminus L_\mathrm{R}(d) : t \in f|}\right) \tag{6}$$

Either one G of these global factors is multiplied by the local factor (tf) to obtain the final weight of a term t in a file f in a directory d.

$$\mathrm{tf} \cdot G(t,f,d) = \mathrm{tf}(t,f) \cdot G(t,d) \tag{7}$$

3.4 Query Expansion

To expand a query, we have first to determine its effective context among the ones inferred from the user's directories. One option would be to let the user manually select a context, as in e.g. (Liu et al., 2002): this requires some more effort from the user, although he or she may prefer to pick a category from a limited set rather than typing additional search keywords.

However, to reduce the user's effort, we use a simple method to automatically map each query to its most likely context. Using the same pre-processing steps and term weighting schemes employed for user documents, a query q is converted to a vector $\mathbf{q} \in \mathbb{R}^{|\mathcal{T}|}$ in the same form of the contexts. In this way, employing the common *cosine similarity* measure between vectors, we pick the context most similar to the query.

$$C(q) = \operatorname*{argmax}_{d \in \mathcal{D}} \frac{\mathbf{q} \cdot \mathbf{c}_d}{||\mathbf{q}|| \cdot ||\mathbf{c}_d||} \tag{8}$$

Once the context of the query is determined, the actual expansion can be performed. At an high level, this involves determining a set $K = \{k_1, k_2, \ldots\}$ of keywords and adding them to the user's query.

As the set of keywords, the n_E terms with the highest weight in the vector of the picked context is used. $n_\mathrm{E} > 0$ is a parameter of the method.

Finally, for any query q given by the user, the expanded query q' is obtained from the conjunction (AND) between q and the disjunction (OR) of all the keywords picked from the context. This new query q' is sent to the search engine in place of q.

$$q' = q \wedge \bigvee_{k \in K} k = q \wedge (k_1 \vee k_2 \vee \ldots) \tag{9}$$

4 EVALUATION PROCESS

To objectively assess our solution for personalized search, we set up a repeatable evaluation process using available benchmark data, providing quantitative measures of the relevance of results given by a search engine. We can thus measure how much our method can improve the relevance of topmost search results when applied to a "plain" search engine.

While methods proposed for personalized search are usually accompanied by the results of an experimental evaluation of their goodness, comparisons

across different methods are generally unfeasible, due to the use of different evaluation processes, particularly when they are not objective, or because of data sets not made available by the authors. Especially, many works (e.g. (Pitkow et al., 2002; Micarelli and Sciarrone, 2004)) employ a user-oriented assessment, where the system is tested by an heterogeneous group of people: a feedback about usability is obtained through questionnaires or by tracking time and actions required by users for each query. This approach has the advantage of estimating how much the tested system actually improves the search experience, especially in terms of personalization to the different users. However, for the sake of objectiveness and ease of repeatability, we favor the quantitative measurement of retrieval effectiveness as described in the following.

For the systematic evaluation of an IR system, we first need a corpus $\mathcal{P}$ of documents to be indexed. To evaluate the system behavior, we then use a set $Q = \{q_1, q_2, \ldots\}$ of queries: in each test, the system responds to a query with a relevance-sorted list of documents picked from $\mathcal{P}$. To evaluate their correctness, we need a gold standard to compare against: this is constituted, for each query q, by an expert-made relevance judgment on each document in $\mathcal{P}$. For each query, a perfect IR system should return all the relevant documents and nothing else. As users often look only at the topmost results of a search engine, we only consider the first 20 results for each query response: we use these results and the gold standard given for each query to compute four different performance measures, as explained later.

Since personalized search works around an existing IR system, another necessary element is a reference search engine upon which to apply our method. Similarly to other works, we perform an evaluation on the plain search engine to obtain baseline results, which are then compared to those obtained by applying personalization on top of the same engine.

To test a personalized search system, other than the "global" test data described above, is also needed either an example user profile or data useful to build it. In our case, to build a profile, we employ a corpus of text documents organized in a hierarchy of directories, reflecting the personal files of a generic user.

4.1 Benchmark Data

The benchmark corpus of Web pages we use is extracted from the ClueWeb09 Dataset by Lemur Project[1]: our collection is constituted by the 503,903,810 pages in English language.

[1]http://www.lemurproject.org/

Table 1: Sample queries used in experimental evaluation.

horse hooves	uss yorktown charleston sc
avp	ct jobs
discovery channel store	penguins
president of the united states	how to build a fence
iron	bellevue

To simulate a Web search engine indexing these pages we use the Indri search engine, also by Lemur Project, providing a flexible query language including the AND and OR operators used in query expansion.

The IR system, either personalized or not, has to be tested with a set of sample search queries representing different information needs and having a known set of actually relevant documents within the indexed ones. We employ data used in the 2010 Web Track[2] of Text REtrieval Conference (TREC) for evaluation of participant IR systems, based on the ClueWeb09 collection. For the competition, NIST created 50 search *topics*, each including a query string likely to be searched by a user interested in that topic: 10 of these queries are listed in Table 1. To each topic is associated a set of documents within ClueWeb09 judged as relevant to it. Each relevant document is also graded in an integer scale ranging from 1 (little information about the topic) to 4 (very specific information).

Finally we need a corpus of text documents resembling files stored on a user's PC: documents should discuss different topics and be accordingly organized in a hierarchy of directories. We take advantage of DMOZ (http://dmoz.org), an open-content directory of about four million Web sites organized in a tree-like hierarchy of about one million categories, ranging from generic, e.g. "Science", to specific ones such as "Bayesian Analysis". We extract a data set constituted by the first two levels of the hierarchy, excluding the "World" and "Regional" top-level categories containing non-English pages. The resulting set of 5,184 Web pages is considered as a collection of personal documents, with the 13 top-level and 308 nested categories treated as directories.

We choose to only consider the two topmost levels of DMOZ basing on the hypothesis that the average user employs a hierarchy of the same depth, with good balance between specificity of topics and ease of organization. A deeper taxonomy would require more user effort and would be generally worthless for the typical amount of documents stored locally.

While the spectrum of topics considered in this data set is considerably wide compared to the set of interests of a typical user, we claim that it is suitable in our evaluation process. In fact, as the test queries

[2]http://trec.nist.gov/data/web10.html

cover a wide range of different topics, the corpus of personal documents should include them all in order to have the chance of picking a suitable context. In a real use case, the local user's files would likely include a narrower set of topics, but also the issued queries would still mostly fit into the same set: our selection of benchmark data reflects this latter point.

4.2 Performance Measures

We consider four different performance measures proposed in literature, used to evaluate IR systems in the TREC 2010 Web Track cited above. Each measure is computed for each query q by comparing the list $\Sigma(q) = (r_1, r_2, \ldots)$ of search results to known relevance judgments; results for each measure are averaged across test queries. We denote with $\Sigma_k(q)$ the list $\Sigma(q)$ truncated at the topmost k entries, where $k = 20$ in our evaluation.

The first two measures are simply based on a binary labeling of test documents as either "relevant" or "not relevant" to each query. $R_q \subset \mathcal{P}$ denotes the set of documents relevant to the query q.

When search results are in the form on unranked sets, the standard precision and recall measures are usually employed: especially, the *precision* is the ratio of actually relevant documents among the retrieved ones. In the case of a ranked list of results, it is common to measure the precision only on the topmost k results (with k usually within 10 and 30), termed *precision@k* or *P@k* for short.

$$P@k(q) = \frac{|\Sigma_k(q) \cap R_q|}{|\Sigma_k(q)|} \tag{10}$$

The *Mean Average Precision* (MAP) is another commonly used measure for evaluation in IR. The average precision (AP) on a query is obtained by averaging for each relevant document the precision@k with k equal to its position in the results, considering 0 for documents excluded from the results. MAP is simply the average of AP on all test queries.

$$AP(q) = \frac{1}{|R_q|} \sum_{r_i \in R_q} P@i(q) \tag{11}$$

The following two measures support a non-binary notion of relevance: rather than marking a document r as either "relevant" or "not relevant" to a query q, a real relevance score $R(q,r) \in [0,1]$ evaluates how much r satisfies the information need expressed by q. In our case, where relevance of r to q is evaluated by a grade $g \in \{0,1,2,3,4\}$ with 0 indicating a non-relevant document, the score is computed as follows.

$$R(q,r) = \frac{2^g - 1}{16} \tag{12}$$

One measure using such relevance scores is the *normalized discounted cumulative gain* (NDCG), computed in the formula below, where Z is a normalization factor ensuring that the maximum achievable NDCG@k is 1.

$$NDCG@k(q) = Z \sum_{m=1}^{k} \frac{2^{R(q,r)} - 1}{\log_2(1+m)} \tag{13}$$

Finally, the *expected reciprocal rank* (ERR) (Chapelle et al., 2009), contrarily to other measures, is based on the so-called *cascade* model where the user just searches for a single relevant document, starting from the top of the results. The ERR measures the expected number of documents the user has to check before finding the first relevant one.

$$ERR@k(q) = Z \sum_{m=1}^{k} \frac{R(q,r_m)}{m} \prod_{n=1}^{m-1} (1 - R(q,r_n)) \tag{14}$$

5 EXPERIMENTAL RESULTS

We compare different test configurations, testing in each different combinations of values for the considered parameters, which are the used term weighting scheme and the number n_E of maximum keywords used to expand each query. Table 2 summarizes the results, grouped by configuration. For every configuration and for both tf-idfd and tf-idfod weighting schemes, we report the best results obtained by varying the n_E parameter between 1 and 50, along with the lowest value of the parameter which yielded the reported measure.

In the first test no personalization is applied: test queries are issued as they are to the Indri search engine and the considered performance metrics are measured by comparing the topmost 20 results for each query with the known relevant documents. These results serve as a baseline for tests on the personalization system: for each measure, both its absolute value and the relative gain on the baseline are considered.

To test our approach, we first devise an ideal case where the local user documents perfectly match the information needs expressed by the test queries. In practice, we suppose that the user's home directory has a subdirectory for each test query $q \in Q$ exactly containing the relevant documents R_q for it. We also assume that the system maps each query to the corresponding directory. While this configuration is obviously unrealistic, these tests show the plausible maximum potential of our personalization approach.

From the results obtained with this configuration (section A of Table 2), we see that our personalization system, in ideal conditions, could significantly boost the relevance of search results.

Table 2: Relevance measures for different search configurations with the best values of the n_E parameter.

configuration	weighting scheme	MAP			P@20			ERR@20			nDCG@20		
		n_E	acc.	impr.	n_E	acc.	impr.	n_E	acc.	impr.	n_E	acc.	impr.
baseline	-	-	*0.073*	-	-	*0.180*	-	-	*0.062*	-	-	*0.073*	-
A	tf-idfd	18	0.137	87.7%	8	0.360	100.0%	11	0.172	177.4%	8	0.195	167.1%
perfect mapping	tf-idfod	20	0.127	74.0%	20	0.270	50.0%	14	0.116	87.1%	14	0.130	78.1%
B	tf-idfd	20	0.094	28.8%	16	0.275	52.8%	15	0.104	67.7%	16	0.120	50.7%
optimal mapping	tf-idfod	24	0.064	-12.3%	24	0.185	2.8%	24	0.081	30.6%	24	0.077	5.5%
C	tf-idfd	23	0.078	6.8%	9	0.220	22.2%	18	0.095	53.2%	18	0.084	15.1%
automatic mapping	tf-idfod	15	0.090	23.3%	17	0.230	27.8%	15	0.090	45.2%	17	0.084	15.1%

In the next two more realistic configurations, documents extracted from DMOZ as described in Section 4.1 are used as the corpus of local user documents, so that there is no match between the documents indexed by the search engine and those upon which the user profile is built. Also, test queries do not directly correspond to user directories anymore, thus requiring a non-trivial mapping between them.

We first devise a case where this mapping is supervised: the correct context of each test query is the one under which most actual relevant documents are classified by a k-NN classifier trained on the ODP documents used in the local files collection. Despite this process would not be feasible in a real case where the relevant documents are not known in advance, it provides an indication of how much a proper reformulation of a query would improve search results and simulates cases where the query-directory mapping is based on knowledge beyond the query itself. This would happen for example if the user explicitly indicates the intended context of the query, picking among his or her directories.

Results for this configuration (section B of Table 2), while notably inferior to the previous ideal ones, still show significant improvements over the baseline, especially where the tf-idfd term weighting scheme is employed. On the other side, use of the tf-idfod scheme does not notably improve results in this case, which sometimes are even worse than the baseline.

Lastly, we consider the actual procedure described in Section 3.4: for each query, the context most similar to the query itself is selected, without employing additional knowledge. This is intended to be the most common use case, where the user's input is limited to a short and possibly ambiguous query and the personalization system ought to automatically infer the correct context to be used to expand it.

Results on this configuration (section C of Table 2) show an overturning regarding the term weighting schemes. While tf-idfd brought better results than tf-idfod in previous configurations, here generally brings scarce improvements over the baseline. On the contrary, tf-idfod performs best according to most performance measures, even improving the results obtained with the previous configuration, where optimal query-directory mapping was used. In any case, more or less significant improvements over the no-personalization baseline are obtained, especially using ERR@20 as the reference measure.

Regarding the number n_E of keywords added to each query to personalize it, in most cases its optimal value ranges between 15 and 20. Numbers in this interval seem thus to constitute the correct balance between providing enough terms to effectively make the query unambiguous and avoiding to add terms which are not relevant enough to the context.

6 CONCLUSIONS

We proposed a method to personalize Web search by expanding outgoing queries according to the specific user's interests. With respect to known methods, the major novelty of our solution consists into leveraging text files locally stored by the user and specifically considering their natural hierarchical organization into directories. From this collection of documents already subdivided by topics, we can extract representative keywords for each topic, infer the topic of each query issued by the user and expand it by adding keywords indicating the topic.

To obtain an objective assessment of the effectiveness of our solution, we set up a systematic evaluation process based on predefined benchmark data. A reference search engine has been tested on a fixed corpus of Web pages using a set of test queries, in order to obtain baseline indicators of the retrieval performance. The same evaluation has then been applied on different configurations of our personalization method, running upon the same reference search engine.

Our experiments show that query expansion could boost significantly the relevance of topmost results returned by a search engine and that our method is able to improve relevance with information extracted solely from the user's local documents. While the user could manually select the correct context of each

query in order to maximize accuracy, the trivial automatic mapping still guarantees interesting improvements over the use of the plain search engine.

The method could be refined in many ways. In this work we used simple and well-established procedures to extract knowledge from hierarchically-organized local documents and to expand search queries: the experimentation of more advanced techniques may lead to improvements of the proposed general method. While we focused on enhancing user's queries, the personalization system could also work on the output of the underlying search engine, filtering or reordering results better matching known user's interests: this approach could be used in substitution of or even in combination with query expansion, exploiting the same user model. Finally, methods could be devised to make use of local user's documents alongside other information sources, such as search or browsing history.

REFERENCES

Carpineto, C. and Romano, G. (2012). A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)*, 44(1):1.

Chapelle, O., Metlzer, D., Zhang, Y., and Grinspan, P. (2009). Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 621–630. ACM.

Chirita, P.-A., Firan, C. S., and Nejdl, W. (2007). Personalized query expansion for the web. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 7–14. ACM.

Domeniconi, G., Moro, G., Pagliarani, A., and Pasolini, R. (2017). On deep learning in cross-domain sentiment classification. In *Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*.

Domeniconi, G., Moro, G., Pasolini, R., and Sartori, C. (2016). A comparison of term weighting schemes for text classification and sentiment analysis with a supervised variant of tf.idf. In *Data Management Technologies and Applications (DATA 2015), Revised Selected Papers*, volume 553, pages 39–58. Springer.

Gauch, S., Chaffee, J., and Pretschner, A. (2003). Ontology-based personalized search and browsing. *Web Intelligence and Agent Systems: An international Journal*, 1(3, 4):219–234.

Gauch, S., Speretta, M., Chandramouli, A., and Micarelli, A. (2007). User profiles for personalized information access. In *The adaptive web*, pages 54–89. Springer.

Ghorab, M. R., Zhou, D., OConnor, A., and Wade, V. (2013). Personalised information retrieval: survey and classification. *User Modeling and User-Adapted Interaction*, 23(4):381–443.

Jansen, B. J., Spink, A., and Saracevic, T. (2000). Real life, real users, and real needs: a study and analysis of user queries on the web. *Information processing & management*, 36(2):207–227.

Liu, F., Yu, C., and Meng, W. (2002). Personalized web search by mapping user queries to categories. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 558–565. ACM.

Micarelli, A. and Sciarrone, F. (2004). Anatomy and empirical evaluation of an adaptive web-based information filtering system. *User Modeling and User-Adapted Interaction*, 14(2-3):159–200.

Pitkow, J., Schütze, H., Cass, T., Cooley, R., Turnbull, D., Edmonds, A., Adar, E., and Breuel, T. (2002). Personalized search. *Communications of the ACM*, 45(9):50–55.

Pretschner, A. and Gauch, S. (1999). Ontology based personalized search. In *Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on*, pages 391–398. IEEE.

Salton, G. and Buckley, C. (1997). Improving retrieval performance by relevance feedback. *Readings in information retrieval*, 24(5):355–363.

Speretta, M. and Gauch, S. (2005). Personalized search based on user search histories. In *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*, pages 622–628. IEEE.

Stamou, S. and Ntoulas, A. (2009). Search personalization through query and page topical analysis. *User Modeling and User-Adapted Interaction*, 19(1-2):5–33.

Steichen, B., Ashman, H., and Wade, V. (2012). A comparative survey of personalised information retrieval and adaptive hypermedia techniques. *Information Processing & Management*, 48(4):698–724.

Sugiyama, K., Hatano, K., and Yoshikawa, M. (2004). Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the 13th international conference on World Wide Web*, pages 675–684. ACM.

Teevan, J., Dumais, S. T., and Horvitz, E. (2005). Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 449–456. ACM.

Trajkova, J. and Gauch, S. (2004). Improving ontology-based user profiles. In *Proceedings of RIAO 2004*, pages 380–390.

Zhou, D., Lawless, S., and Wade, V. (2012). Improving search via personalized query expansion using social media. *Information retrieval*, 15(3-4):218–242.