SocialSearch
A Social Platform for Web 2.0 Search

Claudio Biancalana, Fabio Gasparetti, Alessandro Micarelli and Giuseppe Sansonetti
Department of Engineering, Artificial Intelligence Laboratory, Roma Tre University,
Via della Vasca Navale, 79, 00146 Rome, Italy

Keywords: Query Expansion, Social Bookmarking Services, Personalization.

Abstract: In the last decade, social bookmarking services have gained popularity as a way of annotating and categorizing a variety of different web resources. The idea behind this work is to exploit such services for enhancing traditional query expansion techniques. Specifically, the system we propose relies on three-dimensional co-occurrence matrices, where the further dimension is introduced to represent categories of terms sharing the same semantic property. Such categories, named *semantic classes*, are related to the folksonomy mined from social bookmarking services such as *Delicious*, *Digg*, and *StumbleUpon*. The paper illustrates a comparative experimental evaluation on real datasets, such as the one collected by the Open Directory Project and the TREC 2004. We also include the results of a specific disambiguation analysis aimed to evaluate the effectiveness of our approach in comparison with state-of-the-art techniques when satisfying queries characterized by polysemic and ambiguous terms.

1 INTRODUCTION

The Social Semantic Web combines together the core principles of the Semantic and Social Web: it includes, on the one hand, the idea of associating a semantic description with web resources for enabling machines to access and process them, on the other hand, the idea of exploiting social content information for that purpose. This development, however, leads to the need to revise the classical techniques for the traditional Web (Micarelli et al., 2006; Lops et al., 2007; Gentili et al., 2001; Gasparetti and Micarelli, 2003), as they could not be more efficient in the new Web design.

Automatic query expansion (QE) is a well-known technique for enabling users to better characterize their search domain by supplementing the original query with additional terms that are somehow linked to the frequency of the term the user specified in his query (Bai et al., 2005). This method can significantly improve the performance of information retrieval systems. However, traditional QE techniques, even those providing users with personalized results, may suffer from some drawbacks if extended to the Social Semantic Web. In particular, additional terms can (i) be simple synonyms, (ii) not consider the existence of different lexicons, given that each user has his own custom dictionary. As a result, QE process may fail to contextualize the research domain of interest if multiple users annotate the web content.

Our research objectives include (i) to find a solution to the lack of expression of the candidate terms for query expansion, (ii) to customize the search results taking into account the semantic domain of the user interests, (iii) more generally, to explore novel approaches combining semantic, social, and adaptive aspects.

The proposed system - named *SocialSearch* - is an extension of the traditional QE techniques, which are based on the computation of two-dimensional co-occurrence matrices (Biancalana and Micarelli, 2009; Biancalana et al., 2012). Our approach makes use of three-dimensional co-occurrence matrices, where the added dimension is represented by *semantic classes* (i.e., categories comprising all the terms that share a semantic property) related to the folksonomy extracted from social bookmarking services (Musto et al., 2009) such as *Delicious* [1], *Digg* [2] and *StumbleUpon* [3]. These web sites allow users to store, organize, share, and search bookmarks associated with web resources, through the input of additional data

[1] delicious.com
[2] digg.com
[3] www.stumbleupon.com

Biancalana C., Gasparetti F., Micarelli A. and Sansonetti G..
SocialSearch - A Social Platform for Web 2.0 Search.
DOI: 10.5220/0004943000700081
In *Proceedings of the 10th International Conference on Web Information Systems and Technologies* (WEBIST-2014), pages 70-81
ISBN: 978-989-758-023-9

(e.g., tags or short summaries), freely available to the entire community of users. In our approach, the expansion process takes place by analyzing multiple occurrences divided into categories related to semantic classes, which are analyzed in the folksonomy. The full process is entirely transparent to the user, implicitly occurring based on his past choices related to the terms of the submitted queries and the corresponding visited pages. The input queries are analyzed according to collected data, and if they reflect the interests already shown by the user in previous searches, the system returns different QEs, before performing the search phase. All of these QEs are related to the terms of the user query, but each of them involves a different semantic field. The final results are displayed in different blocks - each one classified through keywords - thereby supporting the user in determining what is most relevant to him (Acampora et al., 2010). This way, our system provides the contextualization and categorization of the information by analyzing and extracting the semantic domain of the user interests.

A comparative analysis of our findings with those obtained through some state-of-the-art techniques, such as relevance feedback, shows that our approach is able to achieve better results. This reveals that our system can offer a stronger correlation with the actual user interests, which confirms the validity and usefulness of their categorization in semantic classes.

The rest of the paper is structured as follows. Section 2 reviews some related works, Section 3 illustrates the system architecture. The main algorithms are detailed in Section 4, while Section 5 is devoted to the presentation and discussion of the experiments we performed. Finally, Section 6 concludes the paper and highlights some future directions.

2 RELATED WORK

Automatic query expansion (QE) has been widely used in Information Retrieval (Carpineto and Romano, 2012; Biancalana et al., 2013b). Among the various QE approaches proposed in literature, some of them take advantage of the implicit relevance feedback through pseudo-relevance feedback (PRF) (Manning et al., 2008). All these methods follow the basic assumption: documents classified higher by an initial search contain many useful terms that can help discriminate relevant documents from irrelevant ones. Despite the large number of studies, a crucial issue is that the expansion terms identified through traditional methodologies from the pseudo-relevant documents may not be all useful (Cao et al., 2008).

Bilotti et al. (Bilotti et al., 2004) analyze the effect of some QE approaches on document retrieval in the context of question answering, mainly targeted to the so-called "factoid" questions, namely, fact-based, natural language questions that usually can be answered by a short noun phrase. More specifically, the authors describe a quantitative comparative analysis between two different strategies for tackling term variation: i) employing a stemming algorithm at indexing time, or ii) carrying out a morphological query expansion at retrieval time. The findings show that, when compared to the baseline (no stemming nor expansion), stemming yields a lower recall, while morphological expansion results in higher recall. However, higher recall is paid at the cost of retrieving more irrelevant documents and ranking relevant documents at lower positions.

One of the failure reasons of the query expansion has been identified in the lack of relevant documents in the local collection. Consequently, some works advance the use of an external resource for query expansion in order to improve the effectiveness of query expansion, such as thesaurus (Nanba, 2007), Wikipedia (Xu et al., 2009), key-phrases from corpus of documents (Biancalana et al., 2013a; Biancalana et al., 2011), browsed web pages (Gasparetti et al., 2014) and search engine query logs (Cui et al., 2003). Abouenour et al. (Abouenour et al., 2010) point out that the adoption of a thesaurus, typically constructed through statistical techniques, poses several drawbacks. First of all, the construction of a thesaurus is time-consuming because of the great deal of data to process. Effective semantic QE techniques can also rely on ontologies instead of thesauri. Indeed, ontologies describe both semantic and concept relations, and enable semantic reasoning as well as cross-language information retrieval. The authors specifically deal with the enhancement of question answering in Arabic, a complex language for its peculiarities. They propose an approach that implements a semantic QE based on the WordNet [4]ontology in Arabic. As a result, the described QE method bears the following semantic relations: synonymy, hypernymy (supertypes), hyponymy (subtypes), and the Super Upper Merged Ontology (SUMO) [5] concept definition. SUMO is a top-level ontology that defines general terms and can be used as a foundation for middle-level and more specific domain ontologies. The documents retrieved through the previous process are then re-ranked using a structure-based approach based on the Distance Density n-gram model.

Recently, several authors have focused on social

[4]wordnet.princeton.edu

[5]www.ontologyportal.org

annotations as external resource, largely motivated by their increasing availability through many Web-based applications. Among these, Carman et al. (Carman et al., 2009) explore how useful tag data may be to improve search results, but they focus primarily on data analysis rather than retrieval experiments. Zhou et al. (Zhou et al., 2012) propose a query expansion framework relied on user profiles extracted from the annotations and resources bookmarked by users. The main difference with our approach is that the selection of expansion terms for a given query is not based on semantic classes, but on the assumption that they are likely to have similar weightings influenced by the documents best ranked for the original query.

3 SYSTEM ARCHITECTURE

In this section we present the architecture of the system we propose (see Fig. 1), describing the functionalities of each module and the modalities which they actively collaborate through.

- **Interface:** the mail role of the this module is readdressing external requests to the specialized modules and processing the achieved results so as to show them in a more understandable form;
- **Expansion:** after the user has submitted his search query, this module is responsible of the query expansion process. To perform multiple expansions, this module has to access the user interests stored in the user model;
- **Search:** this module is in charge of the real search process, receiving (possibly expanded) queries in input and returning the corresponding results;
- **Persistence:** all the necessary information is retained in this module: login data, encountered terms (both before and after stemming), tags, co-occurrence values between terms, tag relevance, and URLs of documents visited by the user; it interacts mainly with the interface (for user login and saving URLs) and the user model (for data needed for the construction and analysis of the user model);
- **UserModel:** this is the largest module in that it has to constantly update the user profile realized as a three-dimensional co-occurrence matrix. The interaction with the persistence module is the first step for achieving data (visited URLs and corresponding queries) from which to infer information for the model update. Before the necessary processing, this module makes use of two other sub-modules: Parser and TagFinder;
  - **Parser:** its main role is to filter out the unnecessary information related to the user interests collected by the system, and to provide the user model with a sorted set of terms for the three-dimensional matrix computation. It includes parsing functionalities (i.e., filtering the HTML pages visited by the user), stemming, and stop-word removal;
  - **TagFinder:** it is devoted to the search of tags to be associated with the pages visited by the user. It interacts with external resources (social bookmarking services) to extract complete tags of a relevance index, in order to supply them to the user model.

Results obtained in each search session are then presented to the user so as to underline the different semantic categories of each group of them. The search of the tags associated with the pages visited by the user is performed by analyzing the information provided by main sites that offer social bookmarking services. In this case, data collection occurs directly by parsing the HTML pages containing the necessary information. In order to model the user visits, the system employes matrices based on co-occurrence at the page level: terms highly co-occurring with the issued keywords have been proven to increase precision when appended to the query (Biancalana et al., 2009). The generic term t_x is in relation with all other n terms t_i (with $i = 1,\ldots,n$) according to a coefficient c_{xi} representing the co-occurrence measure between the two terms. In a classical way, we can construct the co-occurrence matrix through the Hyperspace Analogue to Language approach (Burgess and Lund, 1995): once a term is given, its co-occurrence is computed with n terms to its right (or its left); in particular, given a term t and considered the window f_t of n terms w_i to its right $f_t = \{w_1,\ldots,w_n\}$, we have $cooc(t,w_i) = \frac{w_i}{i}$, $i = 1\ldots,n$. A pair (a,b) is equal to pair (b,a), that is, the co-occurrence matrix is symmetrical. For each training document a co-occurrence matrix is generated, whose lines are then normalized to the maximum value. The matrices of the single document are then summed up, thus generating one single co-occurrence matrix representing the entire corpus.

The limit of this structure lies in the latent ambiguity of collected information: in presence of polysemy of the terms adopted by the user, the result of the query expansion risks to misunderstand the interests, so leading to erroneous results. In order to overcome this problem, in our system the classical model of co-occurrence matrix has been extended. The user model consists of a three-dimensional co-occurrence matrix. Each term of the matrix is linked to an intermediate level containing the relative belonging classes, each

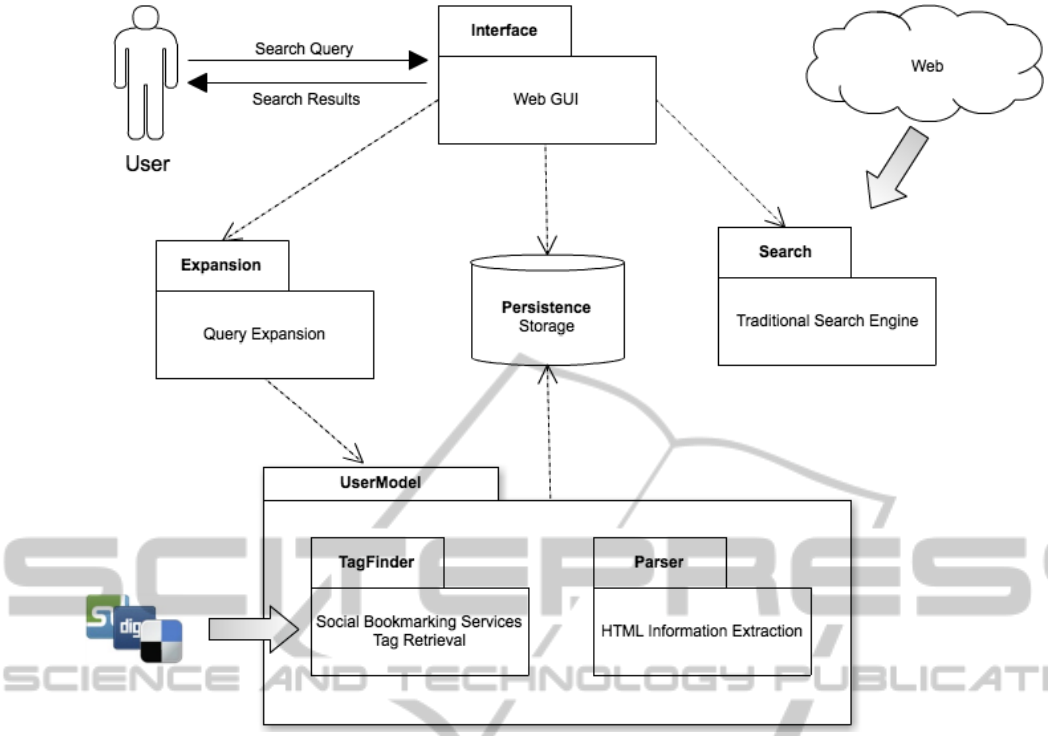

Figure 1: The system architecture.

accompanied by a relevance index. This way, each term is *contextualized* before being linked to all the other terms present in the matrix, and led to well determined semantic categories that are identified by tags.

4 SOCIAL SEARCH

In this section we describe in detail the two main algorithms of our approach. The former is designed for the user model creation and update (discussed in Section 4.1), the latter for the query expansion process (discussed in Section 4.2). With reference to the pseudocode shown below, we notice that the co-occurrence matrix is represented through a map of maps for encoding knowledge and connecting such knowledge to relevant information resources. Maps of maps are organized around *topics*, which represent subjects of interest; *associations*, which express relationships between the subjects; and *occurrences*, which connect the subjects to pertinent information resources.

4.1 User Model Creation and Update

The creation and update of the user model are based on the pages chosen by the user while searching. Starting with an empty model, every time the user clicks on a result after typing a search query, the system records the visited URL, together with the query originally submitted for the search. Our system performs the analysis of the visited URLs in incremental way, according to the following algorithm (see Algorithm 1, where capital deltas (Δ) denote comments):

- a temporary map M is initialized, where it is possible to store the extracted data, before updating the pre-existent model (empty at first execution). The map keys are the encountered tags, the values are the relative two-dimensional co-occurrence matrices;
- for each visited URL, the corresponding HTML page is obtained, from which the textual information is extracted through a parser, as a list of terms;
- the list of terms is filtered in order to eliminate stopwords (i.e., all those terms that are very frequent in all documents, so irrelevant to the creation of the user model);
- the list of terms undergoes a stemming process by means of the Porter's algorithm (Porter, 1980). At the same time the system retains the relations between stemmed terms and original terms;
- the co-occurrence matrix corresponding to the most relevant k_{term} keywords is evaluated. The relevance is measured by counting the occur-

Algorithm 1: User Model Creation and Update.

```
begin
    Δ Initialize the global co-occurrence matrix M (map of maps);
    M ← Map([]);
    Δ Analyze training documents;
    for (doc, query) in D do
        Δ Parse the document (stemming and stopword removal);
        doc = parse(doc);
        Δ Initialize the co-occurrence matrix of different terms;
        terms ← Map([]);
        Δ Compute the co-occurrence value of every term;
        terms = frequency_occurrences(doc);
        Δ Initialize the co-occurrence matrix of document;
        co_occ ← Map([]);
        Δ Compute the co-occurrence matrix of document;
        co_occ = co_occurrences(terms);
        Δ Get the site list of social bookmarking for tag search;
        sites = get_social_bookmarking_sites();
        Δ Initialize URL list tags;
        tags ← Set([]);
        Δ Retrieve tags by URL;
        for i = 0; i < sites.size() & tags.size() = 0; i++ do
            tags = retrieve_tags(url, sites[i]);
        Δ Update the matrix M;
        update(M, tags, terms);
    Δ Initialize all terms in documents;
    all_terms ← Set([]);
    Δ Get unique terms set;
    all_terms = get_term_set(M);
    Δ Get subset of user model;
    user_matrix ← get_user_matrix(all_terms);
    Δ Update user model by the intermediate matrix;
    update(user_matrix, M, all_terms);
    Δ Store updated user model;
    save(user_matrix);
```

rences within the document itself, with the exception of terms used in the query (retained by the system together with the corresponding URL), to which is assigned the maximum weight;

- tags concerning the visited URLs are obtained by accessing different sites of social bookmarking. Each extracted tag has a weight which depends on its relevance (i.e., the number of users which agree to associate that tag to the visited URL);
- the update of the temporary map M is performed by exploiting all the information derived from the co-occurrence matrix and the extracted tags in a combined fashion. For each tag_i the system updates the co-occurrence values just calculated, according to the tag relevance weight. After that, the vectors M_{tag_i,t_i} related to each term t_i are updated by inserting the new (or summing to the previous) values;
- the set $terms$ is calculated, which contains all the terms encountered during the update of the temporary map M;
- from the persistence module a subset UM_{terms} of the user model is obtained as a three-dimensional matrix of co-occurrences, corresponding only to the terms contained in $terms$;
- the matrix UM_{terms} is updated with the values of M. For each t_i belonging to $terms$, the set of keys ($tags$) is extracted from M, which points to values corresponding to t_i. For each tag_i belonging to $tags$, the vector M_{tag_i,t_i} is added to the pre-existent vector UM_{t_i,tag_i}, updating the values for the terms already present and inserting new values for the terms never encountered.

4.2 Query Expansion

The query expansion process is performed beginning from the original terms entered into the search engine by accessing the information collected in the user model. The result is a set of expanded queries, each of them associated with one or more tags. This way, it is possible to present the user with different subgroups of results grouped in categories. Using low level boolean logic, every expansion assumes the following form:

$$(t_{11}\ OR\ \ldots\ OR\ t_{1x})\ AND\ (t_{21}\ OR\ \ldots\ OR\ t_{2x})\ \ldots\ AND\ (t_{y1}\ OR\ \ldots\ OR\ t_{yx})$$

where t_{yx} represents the generic term x corresponding to the stemmed root y. The different terms coming from the same root undergo *OR* operation amongst them, since the result has to contain at least one of them. (see examples in Table 1).

Table 1: Example of multiple expansions.

Original query	Categorization tags	Expansions
amazon	**e-commerce, shopping:**	buy **AND** (books **OR** book) **AND** amazon
amazon	**nature:**	(rivers **OR** river) **AND** amazon

The algorithm of multiple expansion is the following (see Algorithm 2):

- let us suppose that the query Q is given, which consists of n terms q_i (with $i = 1, \ldots, n$). For each of them the system evaluates the corresponding stemmed term $q_i^{'}$, so obtaining the new query $Q^{'}$ as a new result;

- for each term belonging to $Q^{'}$, the corresponding two-dimensional vector q_i is extracted from the three-dimensional co-occurrence matrix. Each of those vectors may be viewed as a map, whose keys are the tags associated with the terms $q_i^{'}$ (which have a relevance factor), and the values are themselves *co-occurrence vectors* between $q_i^{'}$ and all the other encountered terms;
- for each encountered tag the relevance factor is recalculated, adding up the single values of each occurrence of the same tag in all two-dimensional vectors. This way, the result is a vector T in which tags are sorted according to the new relevance factor;
- amongst all tags contained in T, only the higher k_{tag} are selected and considered for the multiple expansions;
- for each selected tag t_i the vector sum_{t_i} is computed, which represents the sum of the co-occurrence values of the three-dimensional matrix, corresponding to all terms $q_i^{'}$ of the query $Q^{'}$;
- for each vector sum_{t_i}, the most relevant terms k_{qe} (corresponding to higher values) are selected. Combining the extracted terms with those of the query Q, a new query $EQ^{'}$ (made up of stemmed terms) is initialized;
- for each expanded query $EQ^{'}$, the corresponding query EQ is calculated through the substitution of stemmed terms with all the possible original terms stored into the system, exploiting the boolean logic according to the scheme previously shown;
- the query EQ and the original tag t_i are entered into the map M_{EQ}, whose keys are expanded queries and values are sets of tags. If M_{EQ} already contains an expanded query identical to the input one, the tag t_i is added to the corresponding set of tags.

Algorithm 2: Multiple Query Expansion.

```
begin
    Δ Initialize the query to be expanded (a list of n terms);
    query ← [q_1, q_2, ..., q_n];
    Δ Stemming of query terms;
    query ← stemming(query);
    Δ Get the subset of the user model related to the query;
    user_matrix = get_user_matrix(query);
    Δ Initialize the tag map for multiple query expansion;
    expansion_tags ← Map([]);
    Δ Compute tags for multiple expansion;
    expansion_tags = find_expansion_tags(query, user_matrix);
    Δ Initialize the expanded query map related to tags;
    exp_queries ← Map([]);
    Δ Compute expanded queries for every tag;
    for (tag, ranking) in expansion_tags do
        Δ Compute the expanded query by choosing most
        relevant terms;
        exp_query = select_relevant_terms(query, user_matrix);
        Δ Enter the result in the expanded query map;
        insert_expanded_query(exp_query, tag, ranking, exp_queries);
    return exp_queries;
```

5 EVALUATION

We now present the experimental results of the proposed approach. Specifically, we describe a comparative evaluation analysis between *SocialSearch*, our social-based search engine, and some state-of-the-art techniques.

A number of different aspects must be evaluated in order to assess the real effectiveness of search engines, such as index coverage, search capabilities, presentation, and user effort in seeking tasks. In this evaluation, we are particularly interested in the standard relevance measures to evaluate the efficacy of the retrieval of web documents and the quality of the results. Several relevant factors make this comparative analysis somewhat difficult. Personalized search aims at enhancing user interaction by understanding the user needs, the context, and the applications and information being used, typically across a wide set of user goals. Usage data that might be of potential interest for recognizing and assessing information consumption patterns of each user and the various information foraging strategies must be accurately collected. Moreover, personalization is influenced by the selection of particular topics on which the evaluation is to be performed. It can create an authoring bias where the topics selected by a group of peers influence the relative results of one approach when compared with others. For example, one approach might exploit a topic characterized by a wealth of documents and references, while a different one is critically affected by the presence of several polysemous words in the query set. In spite of these issues, implementing an experimental evaluation of personalized approaches in a real setting is still the most significant method to measure the scalability and the overall quality of search effectiveness, in terms of both coverage and accuracy of the produced search results. While coverage measures the ability of engines to produce all the references that are likely to be visited by the user, accuracy is essential in evaluating the quality of such references.

Five different search engines have been included in the comparative analysis: Google (denoted simply as *Google* in figures), the personalized version of Google (*PersGoogle*), a query expansion search engine based on co-occurrence data (*CoOcc*), a traditional search engine with Relevance Feedback (*RF*), and our system (*SocialSearch*). In the first personalized version of Google back in 2004, the search engine showed a directory like category drop-down menu, where users could select the categories that matched their interests. During the search process, the search engine adapts the results according to each user needs, assigning a higher score to the resources related to what the user has seen in the past. A slider in the graphic user interface allows the user to control the level of personalization in the results. For example, if the user earlier chose the category of *Computers* as one of his interests, results such as *Apple*, *Acer* or *HP* would rank among the first positions. Unfortunately, no details or evaluations are presently available for the algorithms exploited for the re-ranking process, except the ones contained in the patent application filed in 2004 (Zamir et al., 2004). Our comparative evaluation takes into account the current version of personalized Google. It basically reorders the search results based on gathered usage data, such as previous queries, web navigation behavior and, possibly, visited sites that serve Google ads, computers with Google Applications installed, such as Desktop Search and personal information, which may be implicitly or explicitly provided by the user.

Relevance feedback aims at modifying the initial query using words extracted from top-ranked or identified relevant documents. If both documents and queries are represented in a vector space model (Salton and Buckley, 1997), the Rocchio feedback approach alters the initial query by combining the vectors of the relevant documents increasing the recall of the search engine, and possibly its precision as well (Manning et al., 2008).

Query expansion based on co-occurrences is a well-known approach that collects the correlations between pairs of terms in a given corpus. It is a straightforward approach that limits the computational complexity through the idea of associating contexts to the current user needs. The two fundamental problems of information retrieval, namely, synonymy and polysemy, are addressed during the construction of the query vector. Ambiguous words have only one lemma for all their meanings. If one meaning is mentioned in a query, the documents in which the term appears with the other meanings are also retrieved and estimated as closer to the query. In case of polysemy there will be terms associated to more than one meaning, but if the query is composed by a number of keywords, the intended meaning is more likely to be referenced. These terms and their associated terms will form a cluster, which is associated to the intended meaning and outweighs the unintended meanings. Several studies in the literature have proven the effectiveness of this approach, but have also raised some doubts on its real improvements in the performance of document retrieval systems, because of the following potential issues:

- Weighting terms that occur more frequently in the whole dataset, so favoring the more popular (see, for example, (Peat and Willett, 1991));
- Expanding each single term in the query in isolation, ignoring the potential meaning of the all terms as a whole;
- Co-occurrences data extracted from small collections of documents;
- Collection of documents not including relevant concepts and information during the query expansion.

In order to play down those issues mainly related to the documents selected for the initial dataset, the co-occurrence matrix used for expansion is built on the corpus of documents retrieved during the learning process. In this way, it is certain that enough relevant documents for the expansion are included and there are less chances to see several common terms that cover several different topics of interests. The comparative analysis consists in the following two evaluations:

- TREC corpus-based evaluation;
- ODP corpus-based evaluation;

Corpus-based evaluations have the advantage of showing a zero test-retest variability if the same closed corpus is employed in future experiments that include different approaches. We also include a specific disambiguation analysis in order to measure the efficacy of the search engines to tackle queries characterized by polysemic and ambiguous terms.

5.1 TREC Corpus-based Evaluation

In the first evaluation, we consider the TREC [6] 2004 Robust Track on TREC disks 4 and 5. It contains over 500K documents, a subset of them marked relevant or irrelevant according to a given topic. On average, each document consists of 467 terms. All the 249 queries are included in the evaluations. The approaches considered in this evaluation are *RF*, *CoOcc*,

[6] trec.nist.gov/data.html

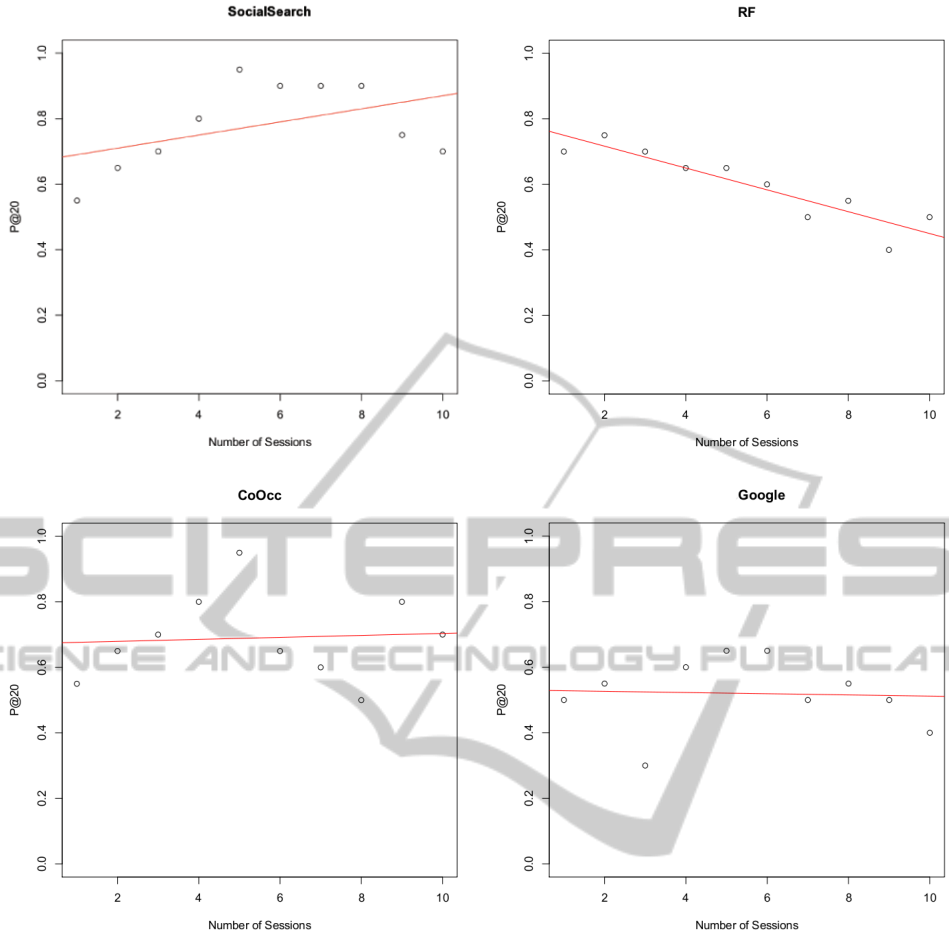

Figure 2: P@20 average values after a certain number of feedbacks.

Google, and *SocialSearch*. The closed nature of this corpus has not allowed us to include *PersGoogle* in this comparative analysis as well. The precision at 20 (P@20) measures the performance of the retrieval. It evaluates the fraction of the retrieved documents that are relevant to the user information needs. The average number of result pages viewed by a typical user for a query is 2.35 (Jansen et al., 2000), and a more recent study (Jansen et al., 2005) reports that about 85.92% of users view no more than two result pages. For these reasons, the precision is evaluated at a given cut-off rank, considering only the top 20 results returned by the system. Figure 2 shows the P@20 after collecting a certain number of feedbacks (Biancalana et al., 2013b).

Google approach shows the worst outcomes with a low average precision. This is an expected result because *Google* does not exploit the suggestions that feedbacks might provide. Better average outcomes are obtained by employing the relevance feedback, even though the slope of the linear model of data is negative. That is to say that the amount of information collected by means of the relevance feedback negatively affects the precision by including irrelevant keywords during the expansion of the queries. Better outcomes are obtained through both *CoOcc* and *SocialSearch* approaches. It must be noted how several web references included in the corpus do not find a correspondence in the sites of social bookmarking services such as *Delicious*. For this reason, *SocialSearch* is put in a unfavorable position in comparison with *CoOcc* trained on the collection of documents related to the relevant topics. The same issue also affects the ODP corpus-based evaluation (see Sect. 5.2).

5.2 ODP Corpus-based Evaluation

Our goal is to build profiles of users that show interests in some specific topics. Each topic must be associated with more than one document, whose content is extracted by personalized search engines and used

to build a user profile representation.

Open Directory Project [7] (ODP) is a multi-language directory of links belonging to the Web. ODP has a hierarchic structure: the links are grouped into categories and subcategories, also known as *topics*. It is therefore possible to identify a level-based organization within the hierarchy. An example of topic is *Top/Business/Forestry and Agriculture/Fencing*; excluding the *Top* level common to all the topics, we have:

- Level I: Business;
- Level II: Forestry and Agriculture;
- Level III: Fencing.

Given the large quantity of links contained in ODP, we have decided to limit to the third level the links taken into consideration for the evaluation. The pages corresponding to such links are retrieved from the Web and indexed. The obtained index consists of 131,394 links belonging to 5,888 topics. Thereafter, ten topics are randomly chosen, five of which corresponding to potential user information needs, and five whose function is exclusively that of representing the pages visited by the user whose content is not relevant, that is, transient needs. The links of each topic were then subdivided into a training set, corresponding to 25% of the links, and the remaining links for test sets. The ten topics are summarized in Table 2.

It is clear now that this methodology allows us to build several different profiles of potential users. Once these profiles are built, it is possible to compare the precision of the search engines. In this evaluation, *Google*, *RF* and *SocialSearch* approaches are compared in terms of F1 score (or F-measure), a standard statistical measure that combines both the precision and the recall of the test to compute the resulting score. A query is built for each topic belonging to the user needs. The query is composed by the terms that form the topic name in ODP (e.g., query: "shopping craft papers"). The evaluation aims at measuring the fraction of document retrieved by the search engine from the whole collection of indexed documents that are also included in the test set for each need. Table 3 shows the variation of F1 score for the three engines. In this evaluation, *RF* engine does not take any sensible advantage of the content extracted from the training documents. *SocialSearch* outperforms the other approaches, even though several links in the training set do not have any reference in the sites of social bookmarking services. Part of the training documents are indeed very old or not very popular, therefore users are not likely to attach metadata to these resources.

[7] www.dmoz.org

5.3 WordNet-based Disambiguation Analysis

Personalization has an important role when users submit ambiguous queries, that is, consisting of terms with multiple different meanings. Past and current contexts might help disambiguate polysemous words and improve result accuracy. For this reason, this evaluation aims at gathering ambiguous queries and performs a comparison of how the different approaches behave in correctly disambiguating their meanings.

A straightforward methodology that involves the analysis of the WordNet [8] lexical database has been defined. Briefly, WordNet is a collection of *synsets*, namely, groups of nouns, adjectives and adverbs all expressing a common concept (e.g., house, home, dwelling, habitation, etc.) (Fellbaum, 1998). Synsets are interlinked by means of semantic and lexical relations. In this way, it is very easy to find terms that have potentially several different meanings (Hirst and Budanitsky, 2005).
A random choice of these ambiguous terms enables us to focus on the 12 keywords shown in Table 4. For each term, two synsets (or semantic contexts) are identified by the ones included in the database. Formally, it is possible to define a triple as follows:

$$< T, X_T, Y_T >$$

where T is a polysemic term with different meanings depending on its context, for example, $T = draft$; X_T and Y_T are two sets of tags, each of which consists of five tags that briefly describe a semantic context, for example

$$X_T = \{beer, kegerator, homebrew, keg, brewing\}$$

and

$$Y_T = \{nba\_draft, basketball, nbadraft, nba, basket\};$$

of course, T gets a different meaning in each of the two semantic contexts X and Y.

Table 4 summarizes the set of triples used in this evaluation.
For every triple, we collected 400 documents from the Web, subdivided into:

- 100 documents for each of the two contexts X and Y. These two collections are divided into two parts of 50 documents each one, which we used for training and test;
- 200 "noisy" documents, namely, that belong to both of the two semantic classes;

[8] wordnet.princeton.edu

Table 2: Benchmark statistics: ODP topic, number of links for test and training, and if topic is part of user needs.

Topic	Test links	Training links	Need
Sports/Cycling/Human Powered Vehicles	15	5	+
Computers/Home Automation/Products and Manufacturers	27	7	+
Business/Mining and Drilling/Consulting	74	18	+
Games/Roleplaying/Developers and Publishers	52	14	+
Business/Agriculture and Forestry/Fencing	100	27	+
Shopping/Crafts/Paper	35	7	
Arts/Performing Arts/Magic	25	6	
Science/Publications/Magazines and E-zines	26	7	
Science/Social Sciences/Linguistics	13	5	
Recreation/Guns/Reloading	15	5	
	382	101	

Table 3: Comparison in terms of F1 score.

Topic	PersGoogle	RF	SocialSearch
Computers/Home Automation/Products and Manufacturers	0.05	0.08	0.16
Sports/Cycling/Human Powered Vehicles	0.09	0.13	0.09
Games/Roleplaying/Developers and Publishers	0.10	0.18	0.18
Business/Mining and Drilling/Consulting	0.19	0.14	0.19
Business/Agriculture and Forestry/Fencing	0.05	0.14	0.57
Average F1	0.10	0.13	0.24

Table 4: Terms and semantic contexts.

Term	**Tags A context**	**Tags B context**
amazon	[geography, south_america, rivers, cruise, river]	[shop, books, bargains, shopping, deals]
cancer	[horoscopes, tarot, zodiac, horoscope, astrology]	[medical, medicine, health, disease, research]
capital	[dc, washington, washingtondc, washington_dc, capitolhill]	[marxism, communism, economics, socialism marx]
depression	[neuroscience, mentalhealth, psychology, health, science]	[recession, financialcrisis, economy, imf, thegreatdepression]
draft	[beer, kegerator, homebrew, keg, brewing]	[nba_draft, basketball, nbadraft, nba, basket]
hamilton	[lewishamilton, formula1, racing, mclaren, f1]	[urban canada, canadian, ontario, city]
harrison	[film, harrisonford, indianajones, movies, ford]	[george, beatles, guitar, rock, the_beatles]
lee	[kungfu, brucelee, martial, martialarts, karate]	[wii, wiimote, interaction, interface, multitouch]
mercury	[msn, java, chat, im, linux]	[planets, nasa, solarsystem, space, astronomy]
oxford	[elearning, university, courses, education, academic]	[words, language, dictionary, english reference]
porter	[5forces, marketing, strategy, management, business]	[stemmer, programming, stemming, algorithms, language]
victoria	[guide, melbourne, australia, tourism, travel]	[waterfall, safari, zambia, falls, africa]

so obtaining a collection of 4800 documents. The documents of the three collections related to the two contexts X and Y and the noisy collection are retrieved by submitting to *Delicious* the following queries, respectively:

- `q: T (tag:x1 OR tag:x2 ... tag:x5) -tag:y1 -tag:y2 ... -tag:y5`
- `q: T -tag:x1 -tag:x2 ... -tag:x5 (tag:y1 OR tag:y2 ... tag:y5)`
- `q: T -tag:x1 ... -tag:x5 -tag:y1 ... -tag:y5`

Each document retrieved by *Delicious* is annotated with a set of tags. As might be expected, two profiles U_X and U_Y are built by analyzing the documents and tags of the context X and Y, respectively. Also the noisy collection is included in both the profiles.

At the end of the training phase, the initial terms T are submitted to the search engines. For each term, the following measures are evaluated for both the profiles U_X and U_Y: P precision , R recall and $F1$ F-measure. Table 5 summarizes the results. While the order of the topics that obtain better results are similar among the considered approaches, the average precision and recall measures differ significantly. Topics such as *draft*, *mercury*, *lee* and *amazon* are clearly easier to disambiguate while *cancer*, *capital* and *depression* need more sophisticated approaches. The average precision favors *SocialSearch* and the approach based on co-occurrences. In terms of average recall and F1 score *SocialSearch* outperforms both *RF* and *CoOcc*. In particular, the average of the two standard deviation measures of F1 score over the contexts *a* and *b* shows that *SocialSearch* is able to disambiguate the same term over both the considered contexts, while *CoOcc* obtains more dispersion from the average precision.

Table 5: Average measures over all topics: Precision, Recall, F1, and its Standard Deviation.

	RF	CoOcc	SocialSearch
Avg P	0.5	0.58	0.60
Avg R	0.41	0.45	0.50
Avg F1	0.39	0.44	0.51
σ_{F1}	0.26	0.29	0.23

6 CONCLUSIONS

User generated content represents a unique source of information that can be exploited for different purposes. In this paper we have described a novel approach that takes advantage of social bookmarking services for enhancing classic query expansion techniques. Specifically, we rely on web sites such as *Delicious*, *Digg* and *StumbleUpon* in order to define semantic classes, namely, categories of terms sharing the same semantic property, based on which to categorize multiple occurrences of the query expansion process. The results of a comparative experimental evaluation confirm that the proposed approach makes for a stronger correlation among expansion terms and real user interests, thereby providing an effective solution to deal with term ambiguity.

This study shows the benefits of categorizing user interests in semantic classes related to the folksonomy extracted from social bookmarking services. As future work we plan to devise ways of integrating more social knowledge, such as social structures, in our approach. A further research effort will also concern the use of natural language processing techniques to better classify the user interests in semantic classes.

REFERENCES

Abouenour, L., Bouzouba, K., and Rosso, P. (2010). An evaluated semantic query expansion and structure-based approach for enhancing arabic question/answering. *International Journal on Information and Communication Technologies*, 3(3):37–51.

Acampora, G., Loia, V., and Gaeta, M. (2010). Exploring e-learning knowledge through ontological memetic agents. *IEEE Comp. Int. Mag.*, 5(2):66–77.

Bai, J., Song, D., Bruza, P., Nie, J.-Y., and Cao, G. (2005). Query expansion using term relationships in language models for information retrieval. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 688–695.

Biancalana, C., Flamini, A., Gasparetti, F., Micarelli, A., Millevolte, S., and Sansonetti, G. (2011). Enhancing traditional local search recommendations with context-awareness. In *Proceedings of the 19th International Conference on User Modeling, Adaption, and Personalization*, UMAP'11, pages 335–340, Berlin, Heidelberg. Springer-Verlag.

Biancalana, C., Gasparetti, F., Micarelli, A., and Sansonetti, G. (2012). Enhancing query expansion through folksonomies and semantic classes. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom)*, pages 611–616.

Biancalana, C., Gasparetti, F., Micarelli, A., and Sansonetti, G. (2013a). An approach to social recommendation for context-aware mobile services. *ACM Trans. Intell. Syst. Technol.*, 4(1):10:1–10:31.

Biancalana, C., Gasparetti, F., Micarelli, A., and Sansonetti, G. (2013b). Social semantic query expansion. *ACM Trans. Intell. Syst. Technol.*, 4(4):60:1–60:43.

Biancalana, C., Lapolla, A., and Micarelli, A. (2009). Personalized web search using correlation matrix for query expansion. In Cordeiro, J., Hammoudi, S., and Filipe, J., editors, *WEBIST (Selected Papers)*, volume 18 of *Lecture Notes in Business Information Processing*, pages 186–198. Springer.

Biancalana, C. and Micarelli, A. (2009). Social tagging in query expansion: A new way for personalized web search. In *Computational Science and Engineering, 2009. CSE '09. International Conference on*, volume 4, pages 1060–1065.

Bilotti, M. W., Katz, B., and Lin, J. (2004). What works better for question answering: Stemming or morphological query expansion? In *Proceedings of the Information Retrieval for Question Answering (IR4QA) Workshop at SIGIR 2004*, Sheffield, England. ACM SIGIR.

Burgess, C. and Lund, K. (1995). Hyperspace analog to language (hal): A general model of semantic representation. In *Proceedings of the annual meeting of the Psychonomic Society*.

Cao, G., Nie, J.-Y., Gao, J., and Robertson, S. (2008). Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st Annual Interna-*

tional ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08, pages 243–250, New York, NY, USA. ACM.

Carman, M. J., Baillie, M., Gwadera, R., and Crestani, F. (2009). A statistical comparison of tag and query logs. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 123–130, New York, NY, USA. ACM.

Carpineto, C. and Romano, G. (2012). A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50.

Cui, H., Wen, J.-R., Nie, J.-Y., and Ma, W.-Y. (2003). Query expansion by mining user logs. *IEEE Trans. on Knowl. and Data Eng.*, 15:829–839.

Fellbaum, C. (1998). *WordNet. An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Gasparetti, F. and Micarelli, A. (2003). Adaptive web search based on a colony of cooperative distributed agents. In Klusch, M., Ossowski, S., Omicini, A., and Laamanen, H., editors, *Cooperative Information Agents*, volume 2782, pages 168–183. Springer-Verlag.

Gasparetti, F., Micarelli, A., and Sansonetti, G. (2014). Exploiting web browsing activities for user needs identification. In *Proceedings of the 2014 International Conference on Computational Science and Computational Intelligence (CSCI'14)*. IEEE Computer Society, Conference Publishing Services.

Gentili, G., Marinilli, M., Micarelli, A., and Sciarrone, F. (2001). Text categorization in an intelligent agent for filtering information on the web. *IJPRAI*, 15(3):527–549.

Hirst, G. and Budanitsky, A. (2005). Correcting real-word spelling errors by restoring lexical cohesion. *Nat. Lang. Eng.*, 11(1):87–111.

Jansen, B. J., Spink, A., and Pedersen, J. (2005). A temporal comparison of altavista web searching: Research articles. *J. Am. Soc. Inf. Sci. Technol.*, 56(6):559–570.

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. *Inf. Process. Manage.*, 36(2):207–227.

Lops, P., De Gemmis, M., and Semeraro, G. (2007). Improving social filtering techniques through wordnet-based user profiles. In *Proceedings of the 11th International Conference on User Modeling*, UM '07, pages 268–277, Berlin, Heidelberg. Springer-Verlag.

Manning, C. D., Raghavan, P., and Schtze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Micarelli, A., Gasparetti, F., and Biancalana, C. (2006). Intelligent search on the internet. In Stock, O. and Schaerf, M., editors, *Reasoning, Action and Interaction in AI Theories and Systems*, volume 4155 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Heidelberg, and New York.

Musto, C., Narducci, F., Lops, P., De Gemmis, M., and Semeraro, G. (2009). Content-based personalization services integrating folksonomies. In *Proceedings of the 10th International Conference on E-Commerce and Web Technologies*, EC-Web 2009, pages 217–228, Berlin, Heidelberg. Springer-Verlag.

Nanba, H. (2007). Query expansion using an automatically constructed thesaurus. In Kando, N. and Evans, D. K., editors, *Proceedings of the Sixth NTCIR Workshop*, pages 414–419, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan. National Institute of Informatics.

Peat, H. J. and Willett, P. (1991). The limitations of term co-occurrence data for query expansion in document retrieval systems. *Journal of the American Society for Information Science*, 42:378–383.

Porter, M. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Salton, G. and Buckley, C. (1997). *Readings in information retrieval*, chapter Improving retrieval performance by relevance feedback, pages 355–364. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Xu, Y., Jones, G. J., and Wang, B. (2009). Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 59–66, New York, NY, USA. ACM.

Zamir, O. E., Korn, J. L., Fikes, A. B., and Lawrence, S. R. (2004). Us patent application #0050240580: Personalization of placed content ordering in search results.

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