Social Tagging in Query Expansion: a New Way for Personalized Web Search

Claudio Biancalana
Department of Computer Science and Automation
Artificial Intelligence Laboratory
Roma Tre University
Via della Vasca Navale, 79, 00146 Rome, Italy

Technical Scientific Committee
LAit S.p.A.
Via Adelaide Bono Cairoli 68, 000145 Rome, Italy

email: claudio.biancalana@dia.uniroma3.it

Alessandro Micarelli
Department of Computer Science and Automation
Artificial Intelligence Laboratory
Roma Tre University
Via della Vasca Navale, 79, 00146 Rome, Italy

email: micarel@dia.uniroma3.it

***Abstract*—Social networks and collaborative tagging systems are rapidly gaining popularity as primary means for sorting and sharing data: users tag their bookmarks in order to simplify information dissemination and later lookup. Social Bookmarking services are useful in two important respects: first, they can allow an individual to remember the visited URLs, and second, tags can be made by the community to guide users towards valuable content. In this paper we focus on the latter use: we present a novel approach for personalized web search using query expansion. We further extend the family of well-known co-occurence matrix technique models by using a new way of exploring social tagging services. Our approach shows its strength particularly in the case of disambiguation of word contexts. We show how to design and implement such a system in practice and conduct several experiments on a real web-dataset collected from Regione Lazio Portal[1]. To the best of our knowledge this is the first study centered on using social bookmarking and tagging techniques for personalization of web search and its evaluation in a real-world scenario.**

I. Introduction

The considerable quantitative increase in the amount of documents on the World Wide Web has led to a scenario in which disorganization gained the upper hand, due to the many different languages composing the documents, typically drafted by a huge number of authors on the Web. This fact leads to the need of supporting the user more efficiently in retrieving information on the web. Users easily find problems in retrieving information by means of a simple Boolean search to check the presence of the searched-for term in the web texts [8]. Indeed, some web texts, consisting of terms that are often synonyms, or related to similar topics only, do not allow to conduct a proper search, and only take into consideration a few terms, which could be input by a user who is likely to have no or little experience in web searches. The Query Expansion (QE) technique fits in this disordered scenario to support the user in his/her search and allow them to widen the search domain, to include sets of words that are somehow linked to the frequency of the term the user specified in his/her query [1]. These may be simple synonyms or terms that are apparently not connected to syntactic meaning, but nevertheless linked to a context that is similar or identical to the one expressed by the original search provided by the user [3]. Such information may be obtained in several ways, the main difference being the source used to obtain further information, which can be retrieved through the preferences explicitly indicated by the user, through the user's interaction with the system [6], through the incremental collection of information that links the query terms to document terms [5](for instance the search session logs) or by means of a simple syntactic analysis of the phrase forms that compose the documents [7],[11]. The approach presented in this paper has its origin from the calculation of co-occurrence matrices. We put forward an extension of the user model which makes use of information collected through Social Bookmarking services such as *del.icio.us*[2] and *StumbleUpon*[3]. The above websites allow us to store, organize, share and search bookmarks associated to Web pages, by the input of additional data (such as tags or short summaries), freely available to the whole community of users. The use of information with a social content, i.e., data based on the active participation of all involved users, is the subject of recent studies which underline both its positive and negative sides. Data reliability is sometimes spoiled by the introduction of erroneous or personal information, or by spamming phenomena [13]. Nevertheless, most of annotations turn out to be often consistent with the categories of the associated bookmarks [9]. The presence of a vast number of users, who agree in assigning a tag to a resource, has been shown to be a very reliable criterion [10]. Recently, the literature has offered several examples of interaction with the

[1]http://www.regione.lazio.it

[2]http://del.icio.us/

[3]http://www.stumbleupon.com/

data retrieved by Social Bookmarking services. In particular we report an example [2] that has represented one of the most relevant input for the conception of our system: the Social Similarity Ranking (SSR), used to estimate the similarities between a search query and a Web page candidate for result; and the Social Page Ranking (SPR), which determines the rank of each page, based exclusively on the amount of input data associated to the page itself.
In our case, the two-dimensional matrix of co-occurrence is extended by inserting a third dimension containing metadata corresponding to tags assigned to the available resources on the Web, as inserted by the users of of the most noticeable Social Bookmarking service, *del.icio.us*. The present paper is organized as follows: in Section II we illustrate the developed system. In Section III we describe the algorithms that distinguish the main innovation that we introduce. In Section IV we present the experimentations carried out on the developed system, and the corresponding results; finally, we discuss our conclusions in section V.

II. Social Query Expansion

To explore the potential of personalized query expansion by a social approach, we have developed an innovative search engine that can record and interpret users' behavior, in order to provide personalized search results, according to their interests. The whole procedure of personalization is completely transparent to the user, because it occurs in an implicit way based on of his/her choices, related to the terms in the submitted queries, and to the corresponding visited pages. The generation of a user profile occurs through the creation of a model, updated dynamically with the information derived from the searches (visited pages and corresponding search queries). The input queries are analyzed according to collected data, and if the comparison has a positive outcome (i.e., if the queries reflect the interests that the user has already shown in previous searches), then the system makes different Query Expansions, each one to a different semantic field, related to the terms inputed by the user, before carrying out the search. The final result is a page in which results are grouped in different blocks, each of them categorized through keywords, in such a way to facilitate the user in the choice of the result that is most coherent with his/her interests.

The system can be thought of a group of modules in close collaboration amongst each other; the roles of modules, together with the modalities with which they actively collaborate, can be described in the following way: interface the system interface is the contact point with the user. It has the main role of readdressing requests coming externally, to the specialized modules, processing the results obtained in order to show them in a more understandable form.

queryexpansion this is the module responsible of the Query Expansion, after the input of a search query from the user, through the interface. It talks to the usermodel module because, in order to produce the multiple expansion, it needs to access to the users interests which are collected in a matrix form.

search it deals with the actual search, and it receives the queries (possibly expanded) as an input, and it provides the corresponding results. Each search is actually proposed to the popular search engine yahoo.com through specific libraries, after which the results are uniformed to a standard scheme and fed to the interface.

persistence it is responsible for recording all information necessary to the system: login data, the set of encountered terms (both those in original forms and those successive to the stemming), tags, co-occurrence values between terms and tags relevance, and the users visited urls. It interacts mainly with the two modules: interface (for the user login and for saving the urls visited during navigation), and usermodel (for the necessary data used for the construction and consultation of the user model).

usermodel it is the largest module because it deals with constantly updating the user module, realized by means of a tridimensional co-occurrence matrix. The interaction with persistence is the first step in order to obtain data (visited urls and queries) from which to extrapolate information for the model update. Before carrying out the necessary calculations, usermodel makes use of two other sub-modules (parser and tagfinder), respectively for filtering the large amount of data and for obtaining the tags associated to the resources chosen by the user.

parser the main role of this module is to eliminate the superfluous information on the users interests collected by the system, and it provides the usermodel module a sorted set of terms for the calculation of the tridimensional matrix. It includes parsing functionalities (i.e. the format filtering in the html pages visited by the user), elimination of the stopwords, and stemming.

tagfinder it is the module dedicated to the search of tags to be associated to the pages visited by the user. It interacts with external resources (stumbleupon.com and del.icio.us) to find complete tags of a relevance index, in order to provide them to the usermodel module.

The system takes advantages of some resources freely available on the Web: *yahoo.com*, one of the most popular search engines. Results obtained in each search session are then shown to the user in such a way to underline the different categories of each group of results. The search of the tags associated to the pages visited by the user, is carried out by analyzing the information provided by two of the main sites of Social Bookmarking, In this case, the 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 shown to increase precision when appended to the query. Many statistical measures have been developed to the best *term relationship* levels, either analyzing entire documents, lexical affinity relationship (i.e., pairs of closely related words contain exactly one of the initial query terms), etc.

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 correlation matrix using the HAL approach [4]: the co-occurence matrix is generated as follows: once a term is given, its co-occurrence is calculated with N terms to its right (ot its left). In particular, given a term t and considered the window of N terms to its right $f_t = \{w_1, \ldots, w_n\}$ we get $co - oc(t, w_i) = \frac{w_i}{i}$, $i = 1 \ldots, N$. A pair (a, b) is equal to pair (b, a): the co-occurence matrix is symmetrical. For each one of the training documents a co-occurence matrix is generated, whose lines are then normalized (on the maximum value). The matrices fo the single document are then summed up, generating one single co-occurence matrix representing the entire corpus. The limit of this structure consists 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, leading to erroneous results. In order to overcome the above problem, in our system the classical model of co-occurrence matrix has been extended. The user model consists of a three-dimensional correlation matrix (see an example in figure 1). Each term of the matrix is linked to an intermediate level, containing the relative belonging classes, each accompanied by a relevance index. In 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, identified by tags.

In the example, the term *amazon*, if referred to the semantic class *nature*, shows high values of co-occurrence with the term *river*. Viceversa, if it is referred to the category *shopping*, is in strong relation with terms such a *books* and *buy*.

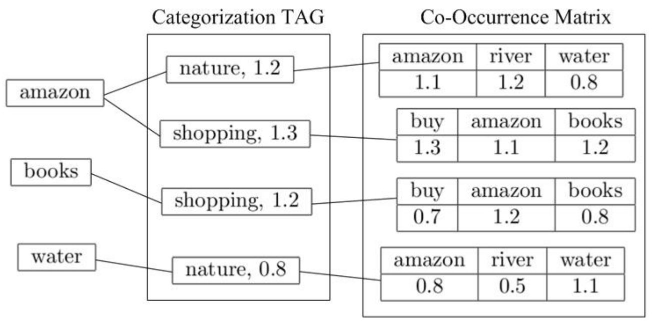

Fig. 1. An Example of Three-Dimensional correlation Matrix used for User Model

III. Algorithms

The system is based on two main algorithms: the first refers to the building and updating of the user model (discussed in subsection III-A), the second to the query expansion (discussed in subsection III-B). With reference to the pseudocode we notice that the co-occurrence matrix is represented by a map of maps for encoding knowledge and connecting this encoded knowledge to relevant information resources. Map of maps are organized around topics, which represent subjects of discourse; associations, representing relationships between the subjects; and occurrences, which connect the subjects to pertinent information resources.

A. Building and Updating the user model

The building and updating of the user model are based on the pages chosen by the user during the searches. 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 used for the search.

Our system performs the analysis of the visited URLs in incremental way, according to the following algorithm (see algorithm 1, capital deltas (Δ) signify comments):

Algorithm 1: User Model Creation & Update

```
1  begin
2      Δ co-occurrence global matrix initialization, represented by a map of maps
3      $M \leftarrow Map([])$
4      Δ training documents analysis
5      for $(doc, query)$ in $D$ do
6          Δ term occurrence map initialization (stemming and stopword removing)
7          $doc = parse(doc)$
8          Δ term frequency calculation
9          $terms \leftarrow Map([])$
10         Δ term frequency calculation for every terms in the document
11         $terms$ = frequency_occurrences(doc)
12         Δ co-occurrence matrix initialization
13         $co\_occ \leftarrow Map([])$
14         Δ co-occurrence document matrix initialization
15         $co\_occ$ = co_occurrences(terms)
16         Δ get sites list of Social Bookmarking for tag search
17         $sites = get\_social\_bookmarking\_sites()$
18         Δ initialization of URL list tags
19         $tags \leftarrow Set([])$
20         Δ retrieve tags by URL
21         for $i = 0; i < sites.size()$ & $tags.size() = 0; i{+}{+}$ do
22             $tags = retrieve\_tags(url, sites[i])$
23         Δ update intermediate matrix $M$
24         update(M,tags,terms)
25     Δ initialization of all terms in documents
26     $all\_terms \leftarrow Set([])$
27     Δ get unique terms set
28     $all\_terms$ = get_term_set(M)
29     Δ get the subset of user model
30     $user\_matrix \leftarrow get\_user\_matrix(all\_terms)$
31     Δ update user model by the intermediate matrix
32     updadte(user_matrix,M,all_terms)
33     Δ saving the updated user model
34     $save(user\_matrix)$
35 end
```

1) a temporary map M is initialized, where it is possible to record 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; **(lines 2-3)**
2) for each visited URL one obtains the corresponding html page, from which the textual information is extracted through a parser, in form of a list of terms; **(lines 6-7)**
3) the list of terms is filtered in order to eliminate the stopwords, i.e., all those terms that are very frequent

but irrelevant to the creation of the user model; **(lines 6-7)**

4) the list of terms undergoes a stemming by means of the Porter's algorithm. At the same time the system records the relations between stemmed terms and original terms; **(lines 6-7)**
5) the co-occurrence matrix corresponding to the most relevant k_{term} keywords is evaluated. The relevance is measured by counting the occurrences within the document itself, with the exception of terms used in the query (recorded by the system together with the corresponding URL), to which is assigned the maximum weight; **(lines 8-15)**
6) 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., on the number of users which agree to associate that tag to the visited URL); **(lines 16-22)**
7) the update of the temporary map M is performed, by exploiting all information derived from the co-occurrence matrix and the extracted tags in a combined fashion. For each tag_i the system updates the values of the co-occurrence just calculated, according to the tag relevance weight. After that, the vectors M_{tag_i,t_i}, relative to each term t_i are updated by inserting the new (or summing to the previous) values; **(lines 23-24)**
8) the set $terms$ is calculated, containing all terms encountered during the update of the temporary map M; **(lines 25-26)**
9) from the persistence layer one obtains a subset UM_{terms} of the user model in form of a three-dimensional matrix of co-occurrence, corresponding only to the terms contained in $terms$; **(lines 29-30)**
10) 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. **(lines 31-34)**

B. Query Expansion

Query expansion is performed starting from the original terms as inputted into the search engine, accessing the information collected in the user model. The result is a set of expanded queries, each of them associated to one or more tags. In such a way it is possible to present to the user different subgroups of results divided in categories. Exploiting the possibilities of submitting queries containing boolean logic of low level offered by *yahoo.com*, every expansion assumes the following form:

$$\begin{gathered}(t_{11}\ OR\ t_{12}\ OR \ldots OR\ t_{1x})\\ AND\ (t_{21}\ OR \ldots OR\ t_{2x})\\ AND \ldots AND\ (t_{y1}\ OR \ldots OR\ t_{yx})\end{gathered}$$

original query	*categorization tags*	*expansions*
amazon	**internet:**	buy **AND** (books **OR** book) **AND** amazon
amazon	**e-commerce, shopping:**	(rivers **OR** river) **AND** amazon

TABLE I
EXAMPLE OF MULTIPLE EXPANSIONS

where t_{yx} represents the generic term x corresponding to the stemmed root y, and the different terms coming from the same root undergo OR operation amongst them, because it is necessary that the result contains at least one of them. See examples in table I.

The algorithm of multiple expansion is the following:

1) let us suppose that the query Q is given, which is made of n terms q_i (with $i = 1, \ldots, n$). For each of them the system evaluates the corresponding stemmed term $q_i^{'}$, obtaining as a new result the new query $Q^{'}$;
2) for each term belonging to $Q^{'}$, from the three-dimensional co-occurrence matrix one extracts the corresponding two-dimensional vector q_i. Each of the extracted vectors may be thought as a map, whose keys are the tags associated to the terms $q_i^{'}$ (which possess a relevance factor) and the values of which are themselves *co-occurrence vectors* between $q_i^{'}$ and all the other encountered terms;
3) 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. In this way, one produces a vector T in which tags are sorted according to the new relevance factor;
4) amongst all tags contained in T, only the higher k_{tag} are selected, on which the multiple expansion is performed;
5) for each selected tag t_i, one evaluates the vector sum_{t_i}, containing the sum of the co-occurrence values of the three-dimensional matrix, corresponding to all terms $q_i^{'}$ of the query $Q^{'}$;
6) for each vector sum_{t_i}, the most relevant terms k_{qe} (corrisponding to higer values) are selected. Combining the extracted terms with which if the query Q, a new query $EQ^{'}$ (made up of stemmed terms) is initialized;
7) for each expanded query $EQ^{'}$, the corresponding query EQ is calculated, through the substitution of stemmed terms with all the possible original terms recorded by the system, exploiting the boolean logic according to the scheme previously shown;
8) the query EQ obtained above, and the original tag t_i is inputed into the map M_{EQ}, in which the keys are expanded query and the 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.

IV. Tuning and Experimental Results

In this section we present the experimental results. The tuning phase consists of three main stages: *training*, *expansion test*, *tuning*.3 During the first stage (*training*) the users (a pool of 12 independent referees) simulated an actual search session, centred on two topics which are chosen beforehand. The system records the various correspondences between the search query and the selected results. The users are asked to submit to the system one last query relative to at least one the chosen topics. In the second stage (*expansion test*) the system presents two lists of adjacent results, one obtained without modifying the query, and the other through Query Expansion, users evaluate which one corresponds better to their interests. Finally, in the third stage (*categorization test*) results obtained with the Query Expansion - which include categories - are shown, and the users are asked a correctness of the results by a questionnaire.

A. Settings

After data collection (*training* stage), we started by formulating a vast set of trial results, each time varying some parameters of the system. Once the tuning phase was over, we have independently chosen the set of most satisfying parameters, by submitting the referees' results. The main values relative to the terms candidate to the expansion are the following three:

- MAX EXPANSION TERMS
  is the maximum number of terms to be added to the original query, chosen to be equal to 3.
- MIN EXPANSION TERMS RELATIVE VALUE
  indicates the percentage threshold for selection of relevant terms after the first one, and it has a value of 0.9. In other words, if for instance the first candidate term has a relevance $r_1 = 2$, the system will judge as acceptable only the those terms with relevance of at least $r_2 = 0.9 \cdot 2 = 1.8$.
- LIMIT EXPANSION TERMS
  this parameter represents a constraint, depending on the number of term of the original query. Based on it, a query can be expanded with a number of terms equivalent, at most, to the number of original terms.

On the other hand, values relative to the tags are:

- MAX EXPANSION TAGS
  is the maximum number of tags selected for the multiple expansion, and has a value of 5.
- MIN EXPANSION TAGS RELATIVE VALUE
  as in the case of the corresponding parameter relative to the terms candidate to the expansion, it is a percentage threshold for the selection of tags relevant to the expansion, fixed to a value of 0.7.

The comparison between different algorithms is made by using comparative performance values obtained from the algorithm under examination, on one single topic or the entire benchmark. Such performances are expressed in F_1-measures, so as to summarize, in one single measure, precision (π) and recall (ρ) values. As for the performance measures, we have precision, recall and F_1-measure (F_1):

$$\pi(t) = \frac{n_t}{50} \qquad \rho(t) = \frac{n_t}{N_t} \qquad F_1 = \frac{2 \times \pi \times \rho}{\pi + \rho}$$

where n_t stands for the number of returned links belonging to topic t, only the first 50 pages are taken in consideration for our tests, and N_t the overall number of test links belonging to topic t present in the index.

Topic	Test links	Training links	Information Needs
Agricoltura	15	5	yes
Ambiente_e_Cooperazione	27	7	yes
Attivita_Produttive	74	18	yes
Protezione_Civile	52	14	yes
Bilancio_Economia_Partecipata	100	27	yes
Casa	35	7	no
Cultura_Spettacolo_Sport	25	6	no
Demanio_Personale_Patrimonio	26	7	no
Sanita	13	5	no
Protezione_Civile	15	5	no
Tot.	382	101	

TABLE II
THE EMPLOYED BENCHMARK: STATISTICS

Topic	no QE	RF	***Our System***
Ambiente_e_Cooperazione	0.05	0.08	0.16
Agricoltura	0.09	0.13	0.09
Protezione_Civile	0.10	0.18	0.18
Attivita_Produttive	0.19	0.14	0.19
Bilancio_Economia_Partecipata	0.05	0.14	0.57
Average	F_1 0.10	F_1 0.13	F_1 **0.24**

TABLE III
COMPARATIVE F_1-MEASUREMENT

B. The employed benchmark: Lazio Region Portal Data

We extract the benchmark from Lazio Region Portal[4]: we crawled and collected a big set of web-pages in a dataset named LRDP (Lazio Region Portal Data). Lazio Region Portal Data (LRPD), is an italian direcotry of links belonging to the Lazio Region Portal; LPRD has a hierarchic structure: the links are grouped into categories, also known as topics, and subcategories. It is therefore possible to identify a level-based organization within the hierarchy. An example of topics is *Top/Sala Stampa/Presidente/Biografia*; excluding the level corresponding to Top, common to all topics, we have:

- *Level I: Sala Stampa;*
- *Level II: Presidente;*
- *Level III: Biografia.*

Given the large quantity of links contained in LRPD, we decided to consider only Level III links.

The pages corresponding to such links were downloaded from the World Wide Web, by using a parser; the textual information was taken from it, and then it was indexed by means of the Lucene indexing system [5]. Ten topics were then

[4]http://www.regione.lazio.it
[5]http://lucene.apache.org

chosen from the Level III topics, five of which corresponding to the user's information needs, and five whose function was exclusively to generate noise in the creation of the user model. The links of each topics were then subdivided in a training set, corresponding to 25% of the links, and set of tests, corresponding to 75% of the links (see table II).

C. Experimentation methods

Once the user model is generated, it is possible to carry out real tests as follows. A query is built for each topic belonging to the user's information needs. The terms of the query are simply the terms that form the topic name. This query is then expanded according to the user model, and used to search for web pages within the created index, starting from all third-level links. The pages belonging to the training set of the considered topic are removed from the returned pages; only the first fifty are taken into consideration, which include the number of pages belonging to the topic under consideration. The index obtained with Lucene, starting from the third-level link of LRDP, consists of 1,313 links belonging to 42 topics.

Personalization in this paper relies on the strength of the user community as it requires that search result documents have been tagged by users. For documents without bookmarks or tags, our personalization approach is not possible in practice because metadata about them is missing. An important task is therefore to analyze the expected availability of metadata for search result documents in real-world. For this reason it is not possible a direct comparison to other similar approach in the state of the art on the same benchmark.

Table III shows the results obtained by a system based on a traditional content-based user-modeling approach, where documents are represented in the Vector Space Model and without Query Expansion (QE), in comparative terms. This system particularly focuses on the update of the user model by means of Relevance Feedback (RF) techniques [12], applied to the training pages content: for each category, the first ten keywords are taken from the corresponding training pages.

The keywords are obtained in terms of $tf \times idf$, and then used to expand the query. In our experimentation, our system obtained the best results, in terms of performance, on the reference benchmark. To the best of our knowledge this is the first study on using social bookmarking and tagging techniques for personalization of web search and its evaluation in a real-world scenario.

V. Conclusions

In this research we developed an Information Retrieval system, based on Query Expansion and Personalization, that may help the user search for information on the Web, according to his/her information needs. The state of the art analysis of the main mechanisms of *personalized search* (in particular the query expansion) and the development of social bookmarking, were the starting points for the next realization of the system. As for personalization, the comparison amongst different methodologies in the literature allowed us a critical review, and at the same time it contributed to build a strong theoretical foundation. The study of the thematics linked to *Web 2.0*, and in particular the collaborative categorization, represented for us the initial inspiration for introducing the more original and innovative aspects which are distinctive of our system. Experimental results were encouraging and confirmed the correlation with users' interests. and the effective coherence and utility of their categorization. In the future we intend to test this approach in comparative terms with other state of the art query expansion techniques using personalization.

VI. Acknowledgements

The authors would like to express their sincere gratitude to LAit Technical Scientific Committee members, Claudio Pisu, Francesco Saverio Profiti and **Fabio Raimondi** for the opportunity to explore our social bookmarking and tagging techniques for personalization of web search and its evaluation in a real-world scenario.

References

[1] 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 *CIKM*, pages 688–695.

[2] Bao, S., Xue, G., Wu, X., You, Y. (2007). Optimizing web search using social annotations. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 501–510.

[3] Burgess, C., Livesay, K., and Lund, K. (1999). Exploration in Context Space: Words, Sentences, Discourse. *Discourse Processes*, 25(2&3):211–257.

[4] Burgess, C., Lund, K. (1995) Hyperspace analog to language (hal): A general model of semantic representation. *Proceedings of the annual meeting of the Psychonomic Society*, 12, 177-210.

[5] Gao, J., Nie, J.-Y., Wu, G., and Cao, G. (2004). Dependence language model for information retrieval. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 170–177, New York, NY, USA. ACM Press.

[6] Gasparetti, F. and Micarelli, A. (2007). Personalized search based on a memory retrieval theory. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI): Special Issue on Personalization Techniques for Recommender Systems and Intelligent User Interfaces*, 21(2):207–224.

[7] Jaime Teevan, Susan T. Dumais, and Eric Horvitz, 'Personalizing search via automated analysis of interests and activities', in *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 449–456, New York, NY, USA, (2005). ACM Press.

[8] 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. In *Information Processing and Management*, 36(2):207–227.

[9] Hend, S., Al-Khalifa and Davis, H. , (2007). Towards better understanding of folksonomic patterns, In *HT '07: Proceedings of the 18th conference on Hypertext and hypermedia*, pages 163–166

[10] Harry Halpin, Valentin Robu, and Hana Shepherd, 'The complex dynamics of collaborative tagging', in *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pp. 211–220, (2007).

[11] Radlinski, F. and Joachims, T. (2005). Query chains: Learning to rank from implicit feedback. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 239–248.

[12] Salton, G. and Buckley, C. (1997). Improving retrieval performance by relevance feedback. *Morgan Kaufmann Publishers Inc*, pages 355–364.

[13] Yusuke Yanbe, Adam Jatowt, Satoshi Nakamura, and Katsumi Tanaka, 'Can social bookmarking enhance search in the web?', in *JCDL '07: Proceedings of the 2007 conference on Digital libraries*, pp. 107–116, (2007).