2012 ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust

Enhancing Query Expansion through Folksonomies and Semantic Classes

Claudio Biancalana, Fabio Gasparetti, Alessandro Micarelli, and Giuseppe Sansonetti
Department of Computer Science and Automation
Roma Tre University
00146 Rome, Italy
Email: claudio.biancalana,gaspare,micarel,gsansone@dia.uniroma3.it

***Abstract*—Adaptive query expansion (QE) allows users to better define their search domain by supplementing the original query with additional terms related to their preferences and information needs. The system we present is an extension of the traditional QE techniques, which rely on the computation of two-dimensional co-occurrence matrices. Our system 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 such as *delicious*, *Digg*, and *StumbleUpon*. The results of an in-depth experimental evaluation on artificial datasets and real users show that our system outperforms some well-known approaches in the literature, as well as a state-of-the-art search engine.**

I. Introduction

The amount of information published on the World Wide Web is growing at an astonishing rate, thus making it necessary to devise effective methods for helping users find what they are looking for [1], [2] . Query expansion (QE) allows users to expand their search domain by supplementing their original query with additional terms and phrases [3], [4]. The system we present, named Nereau, is a social extension of the traditional QE techniques, which are based on the computation of two-dimensional co-occurrence matrices [5], [6]. Our system 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 [7] such as *delicious* [1], *Digg* [2], and *StumbleUpon* [3]. The whole procedure of adaptation is completely transparent to the user, as it takes place in an implicit way based on his profile. The user profile is created and dynamically updated using the information related to visited pages and corresponding search queries. The system analyzes the input queries and, if they actually reflect the interests already shown by the user in previous searches, it returns different QEs involving different semantic fields. The output of the system is structured in different blocks categorized through keywords, thus helping the user decide which result is most relevant to him [8].

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

This paper is organized as follows. Section II outlines the system architecture, Section III describes in detail the underlying algorithms. The results of experimental evaluations performed on artificial datasets and real users are presented in Section IV. Section V reports our concluding remarks and plans for future work.

II. System Architecture

The roles of modules and the modalities which they actively collaborate through, can be described as follows:

- **Interface:** the system interface is the contact point with the user. It has the main role of readdressing external requests to the specialized modules and processing the results obtained in order to show them in a more understandable form;
- **Expansion:** after the user has submitted his search query, this module is responsible of the QE process. To perform multiple expansions, this module has to access the user interests stored in the user model;
- **Search:** it deals with the actual search, receiving (possibly expanded) queries in input and returning the corresponding results;
- **Persistence:** it retains all the necessary information: 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 consultation of the user model);
- **UserModel:** it is the largest module because it deals with constantly updating the user profile realized as a three-dimensional co-occurrence matrix. The interaction with the persistence module is the first step in order to obtain data (visited URLs and corresponding queries) from which to extrapolate information for the model update. Before carrying out the necessary calculations, this module makes use of two other sub-modules: Parser and TagFinder;
  - **Parser:** the main role of this sub-module is to filter out the unnecessary information concerning the user interests collected by the system, and to provide the user model with a sorted set of terms for computing the three-dimensional matrix. It includes parsing

978-0-7695-4848-7/12 $26.00 © 2012 IEEE
DOI 10.1109/SocialCom-PASSAT.2012.67

- functionalities (i.e., the format filtering in the HTML pages visited by the user), stemming, and stopword removal;
- **TagFinder:** it is the module dedicated to the search of tags to be associated with the pages visited by the user. It interacts with external resources (social bookmarking services) to find complete tags of a relevance index, in order to provide them to the user model.

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 with the pages visited by the user is carried out by analyzing the information provided by main sites of social bookmarking. 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 [9]. 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 using the Hyperspace Analogue to Language approach [10]: once a term is given, its co-occurrence is calculated 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 $co\text{-}oc(t, w_i) = \frac{w_i}{i}$, $i = 1 \ldots, n$. A pair (a, b) is equal to pair (b, a), that is, the co-occurrence matrix is symmetrical. For each one of the training documents 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, so generating one single co-occurrence 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, thus 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 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.

III. NEREAU

The system is based on two main algorithms: the first refers to the user model creation and update (discussed in Section III-A), the second to the query expansion (discussed in Section 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. Maps of maps are organized around topics, which represent subjects of discourse; associations, which express relationships between the subjects; and occurrences, which connect the subjects to pertinent information resources.

A. 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 used for the search. Our system performs the analysis of the visited URLs in incremental way, according to the following algorithm:

- 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;
- 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 stop-words(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 by means of the Porter's algorithm [11]. At the same time the system records 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 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;
- 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 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}, relative 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 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.

B. Query Expansion

Query expansion 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.The algorithm of multiple expansion is the following:

- 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.

IV. Evaluation

In this section, we present a comparative performance analysis between Nereau, the proposed social-based search engine, and other query expansion and personalized search approaches. 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 (*Nereau*). The comparative analysis consists in the following three evaluations:

- TREC corpus-based evaluation;
- ODP corpus-based evaluation;
- Web user-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. Nevertheless, as stated previously, experimental settings to real scenarios provide undoubted insights into the performance of the retrieval engines.

A. TREC corpus-based evaluation

In the first evaluation, we consider the TREC [4] 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*, *Google*, and *Nereau*. The closed nature of this corpus has not allowed us to include *PersGoogle* as well in this comparative analysis.

The precision at 20 (P@20) and the Mean Average Precision (MAP) measure the performance of the retrieval. The former evaluates the fraction of the retrieved documents that are relevant to the user information needs, the latter is useful to average various precisions when there are sets of distinct queries to be submitted to the search engine. The average number of result pages viewed by a typical user for a query is 2.35 [12], and a more recent study [13] 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. *Google* approach shows the worst outcomes with a low average precision and MAP. 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 *Nereau* approaches.

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

It must be noted how several Web references included in the corpus do not find a correspondence in the *delicious* social service. For this reason, *Nereau* 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. IV-B). In spite of that, *Nereau* is still able to obtain a better MAP and upper quartile on the obtained precision values.

B. 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 [5] (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 chosen at random, 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 I.

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 *Nereau* 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 II 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. *Nereau* outperforms the other approaches, even though several links in the training set do not have any reference in the *delicious* service. Part of the training documents are indeed very old or not very popular, therefore it is not likely that users attach metadata to these resources on *delicious*.

[5] www.dmoz.org

C. Web user-based evaluation

We have discussed a system evaluation through a test collection and the results of evaluation metrics to calculate the effectiveness score of the system.

Personalized search engines, such as *Nereau*, need to collect and analyze large amount of usage data related to the current and past user interests and needs in order to provide better recommendations in comparison with traditional approaches. For this reason, the evaluation also involves a group of people that have evaluated the effectiveness of the search engines in real scenarios. A total of 42 people were recruited to participate in the user evaluation, mostly students of Computer Science courses. All participants hold a bachelor's degree. A vast majority of males (36) outnumbers females (6). All of them are aged below 30. This choice allowed us to have people deemed comfortable with using search engines in their activities. Some of the recruited people (8%) use search engines once a week on average, while the others use these tools at least once a day. A substantial number of people (70%) are to be considered experts, namely, they know the basic notions of boolean matching between words and page contents, and they are familiar with some advanced search techniques (e.g., boolean operators and phrase search).

Each user is asked to choose two general domains of interest with the recommendation that the awareness and familiarity of the topic is adequate for analyzing contents retrieved on the web. For each of these topics, the user performs five search sessions, each one related to some specific sub-topic of the chosen domain. The prototype monitors the pages the user decides to visit in the top ten results page. There is no time limit to be observed during the evaluation.

After training, the user is asked to perform and evaluate a search session related to one information need in the chosen domains. In particular, the user has 40 results made up of the three lists of ten results obtained by four engines: *Google*, *PersGoogle*, *CoOcc*, and *Nereau*. The final lists are randomized. *Google* search engine is chosen for its popularity, high effectiveness, and the state-of-the-art of ranking algorithms in Web information retrieval. Moreover, by asking users to create a personal account, Google is able to provide personalized ranks based on the users Web history. Users with a Google account were asked to clear their Web history or otherwise create a new one. *Google* evaluation is performed by asking the users to log out from the search engine before retrieving any search result. Users express a judgment for each result with a five-point Likert-type scale of values. The performance of the recommendation process was assessed by evaluating the normalized version of Discounted Cumulative Gain (nDCG) [14]. It is a well-known measure for evaluating a graded relevance scale of documents in a search engine result set. Rather than

TABLE I
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 II
COMPARISON IN TERMS OF F1 SCORE

Topic	PersGoogle	RF	Nereau
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

MAP, nDCG is much more focused on the top of the ranked list.

nDCG is usually truncated at a particular rank level to emphasize the importance of the documents retrieved first. To focus on the top-ranked items, we considered the DCG@n by analyzing the ranking of the top *n* items in the recommended list with $n \in \{1, 5, 10\}$. The measure is defined as follows:

$$nDCG@n = \frac{DCG@n}{IDCG@n} \tag{1}$$

and the Discounted Cumulative Gain (DCG) is defined as:

$$DCG@n = rel_1 + \sum_{i=2}^{n} \frac{rel_i}{\log_2 i} \tag{2}$$

where rel_i is the graded relevance of the $i-$th result (i.e., from 0=*non-significant* to 4=*very significant*), and the Ideal DCG ($IDCG$) for a query corresponds with the DCG measure where scores are re-sorted monotonically decreasing, that is, the maximum possible DCG value over that query. nDCG is often used to evaluate search engine algorithms and other techniques whose goal is to order a subset of items in such a way that highly relevant documents are placed on top of the list, while less important ones are moved further down. Basically, higher values of nDCG mean that the system output gets closer to the ideally ranked output.

In order to evaluate the reliability of such comparisons, all results were tested for statistical significance using t-test. In each case, we obtained a p-value < 0.05. Therefore, the *null hypothesis* that values are drawn from the same population (i.e., the outputs of two search engines are virtually equivalent) can be rejected.

Table III summarizes the evaluation results. In terms of best performance, *Nereau* wins on the ideal ranking of users,

TABLE III
COMPARISON IN TERMS OF nDCG@n MEASURES

	nDCG@1	nDCG@5	nDCG@10
Google	0.13	0.28	0.32
PersGoogle	0.17	0.33	0.39
CoOcc	0.44	0.51	0.68
Nereau	0.33	0.55	0.71

especially when the user sifts through five or more results. The worst performance is obtained by the non personalized *Google* approach. More precisely, both *CoOcc* and *Nereau* obtain higher results. The contextual information that is included during the query expansion helps reduce ambiguity and makes the retrieval more accurate. *CoOcc* query expansion performs slightly better if the task is to recommend only one document (i.e., the more relevant), while *Nereau* outperforms the other approaches if the task is to retrieve five or ten results in absolute terms. Figure 1 better explains the results with same medians for nDCG@1, while for nDCG@5 and nDCG@10 *Nereau* behaves more accurately. The difference between the two approaches is also observable by the number of terms used during the expansion of the query. *Nereau* adds 2.96 terms to the original query on average, while *CoOcc* uses 2.57 terms. Basically, *Nereau* alters the query with more words than the co-occurrence based retrieval.

V. CONCLUSIONS

In this paper we have proposed a personalized query expansion approach that relies on the definition of semantic classes (i.e., categories comprising all the terms that share a semantic property) related to the folksonomy extracted from social bookmarking services such as *delicious* and *StumbleUpon*. The expansion process takes place by analyzing multiple

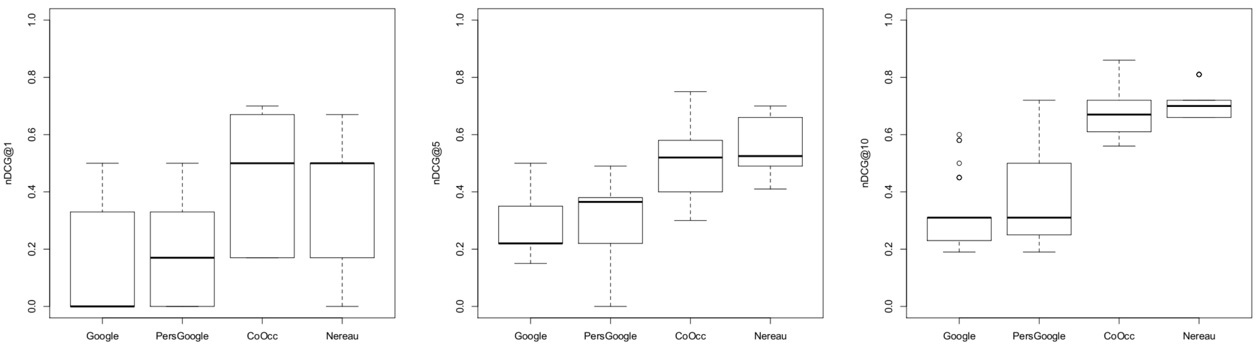

Fig. 1. Box plots of nDCG@1, nDCG@5 and nDCG@10 values.

occurrences divided into categories related to semantic classes, which are analyzed in the folksonomy. We have presented the results of an in-depth experimental evaluation and a comparative analysis, which confirm the correlation with user interests and the effective coherence and utility of their categorization in semantic classes.

There are several research thrusts that we intend to pursue in the future. First of all, we intend to study ways of integrating natural language processing knowledge and procedures in our approach. Moreover, we want to introduce the temporal component in order to interpret the user information needs as his searches change over time.

A further research challenge is to consider alternative ways of tag categorization to be added to tag search through social bookmarking sites, for example, those based on automatic document categorization. Finally, we would like to enhance our system with new functions, such as (i) to make tag suggestions, thus encouraging the discovery of potentially related topics, (ii) to fully use social aspects by considering friend networks, and (iii) to take into account contextual factors related to the user environment on mobile platforms (e.g., smartphones and tablets).

REFERENCES

[1] A. Micarelli, F. Gasparetti, and C. Biancalana, "Intelligent search on the internet," in *Reasoning, Action and Interaction in AI Theories and Systems*, 2006, pp. 247–264.

[2] P. Lops, M. Degemmis, and G. Semeraro, "Improving social filtering techniques through wordnet-based user profiles," in *Proceedings of the 11th international conference on User Modeling*, ser. UM '07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 268–277.

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

[4] F. Gasparetti and A. Micarelli, "Adaptive web search based on a colony of cooperative distributed agents," in *Cooperative Information Agents* M. Klusch, S. Ossowski, A. Omicini, and H. Laamanen, Eds., vol. 2782: Springer-Verlag, 2003, pp. 168–183.

[5] C. Biancalana, A. Lapolla, and A. Micarelli, "Personalized web search using correlation matrix for query expansion," in *WEBIST (Selected Papers)*, 2008, pp. 186–198.

[6] F. Gasparetti and A. Micarelli, "User profile generation based on a memory retrieval theory," in *Proc. 1st International Workshop on Web Personalization, Recommender Systems and Intelligent User Interfaces (WPRSIUI'05)*, 2005, pp. 59–68.

[7] C. Musto, F. Narducci, P. Lops, M. Gemmis, and G. Semeraro, "Content-based personalization services integrating folksonomies," in *Proceedings of the 10th International Conference on E-Commerce and Web Technologies*, ser. EC-Web 2009. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 217–228.

[8] G. Acampora, V. Loia, and M. Gaeta, "Exploring e-learning knowledge through ontological memetic agents," *IEEE Comp. Int. Mag.*, vol. 5, no. 2, pp. 66–77, 2010.

[9] M.-C. Kim and K.-S. Choi, "A comparison of collocation-based similarity measures in query expansion," *Information Processing and Management*, vol. 35, pp. 19–30, 1999.

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

[11] M. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[12] B. J. Jansen, A. Spink, and T. Saracevic, "Real life, real users, and real needs: a study and analysis of user queries on the web," *Inf. Process. Manage.*, vol. 36, no. 2, pp. 207–227, Jan. 2000. [Online]. Available: http://dx.doi.org/10.1016/S0306-4573(99)00056-4

[13] B. J. Jansen, A. Spink, and J. Pedersen, "A temporal comparison of altavista web searching: Research articles," *J. Am. Soc. Inf. Sci. Technol.*, vol. 56, no. 6, pp. 559–570, Apr. 2005. [Online]. Available: http://dx.doi.org/10.1002/asi.v56:6

[14] K. Järvelin and J. Kekäläinen, "IR evaluation methods for retrieving highly relevant documents," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM Press, 2000, pp. 41–48.