A Retrieval Model for Personalized Searching Relying on Content-based User Profiles

Marco de Gemmis, Giovanni Semeraro, Pasquale Lops and Pierpaolo Basile
University of Bari - Department of Computer Science
Via E. Orabona, 4 - 70125 Bari (ITALY)
{degemmis, semeraro, lops, basilepp}@di.uniba.it

Abstract

Canonical Information Retrieval systems perform a ranked keyword search strategy: Given a user's one-off information need (query), a list of documents, ordered by *relevance*, is returned.

The main limitation of that "one fits all" approach is that long-term user interests are neglected in the search process, implicitly assuming that they are completely independent of the user query. Actually, there are information access scenarios that cannot be solved through a straightforward matching of queries and documents, since other elements influence the relevance of the retrieved results. In these scenarios, a smart search engine could exploit information about topics of interest, stored in the user profile, to automatically tailor ranking functions to a particular user.

The main contribution of this paper is an extension of the vector space retrieval model in which user profiles learned by a content-based recommender system are taken into account to modify the ranking of search results.

Experimental results in a movie retrieval scenario show how promising is the approach.

Motivations

Distinct users issuing the same query may have different information needs, different preferences as well as different linguistic competencies. Thus, the "one-fits-all" approach adopted by most search engines can turn out to be inadequate in several information access scenarios. For example, an Italian native speaker looking for interesting movies about "criminal minds" or "serial killers" cannot easily express this form of information need as a query suitable for movie retrieval systems and, even if she can formulate the query "criminal minds" or "serial killers", other elements typically influence the *relevance* of the retrieved results, such as the plot of the movie, the nature of the committed crime or the actors in the cast. In this case, *personal tastes* might be considered to change the order of the retrieved results. With an heterogeneous user population, growing most rapidly in non-English-speaking countries (Chung May 2008), *personalization* is crucial for helping search go beyond a one-ranking-fits-all approach (Teevan, Dumais, and Horvitz 2007).

Recent advances in areas such as user profiling and personalization suggest potential solution strategies capable of delivering more meaningful and personalized search experiences.

Search engines able to learn from implicit feedback or past click history have been proposed by several authors (Joachims and Radlinski 2007; Joachims et al. 2007; Agichtein, Brill, and Dumais 2006; Qiu and Cho 2006), as well as techniques that infer rich models of user interests by analyzing previously issued queries and visited Web pages (Teevan, Dumais, and Horvitz 2005).

Leading commercial search engines Google[1] and Yahoo![2] have undertaken initiatives related to Web personalization, which have offered their own particular demonstration of personalized search.

Our research focuses on the use of machine learning algorithms for the automated induction of a structured model of user interests and preferences from text documents, referred to as *user profile*, that could be used to filter search results or to refine the original query issued by the user.

This paper proposes a search paradigm in which user profiles are included in the computation of query-document similarity in order to achieve *personalized ranking* of results. The retrieval model adopts techniques to learn *semantic* profiles which, differently from keyword-based profiles (Pazzani and Billsus 1997), are able to capture concepts expressing user interests from relevant documents. Semantic profiles contain references to concepts defined in lexicons like WORDNET (Miller 1995) or domain ontologies.

The main contribution of this paper is the "Personalized Synset Similarity Model" (PSSM), that extends the classical Vector Space Model (VSM) by including the user profile in the computation of the query-document similarity score. In this way, the user profile contributes to rank documents in the result list. In the PSSM, WORDNET concepts, called *synsets*, are adopted to index documents, rather than keywords, and a similarity function able to deal with synsets is used to realize a *concept matching* between query and documents.

[1] `www.google.com/psearch`

[2] `myweb2.search.yahoo.com`

As a workbench for the evaluation, we selected the task of *movie retrieval*, in which user preferences really affect the acceptance of results. The proposed personalized retrieval model is suitable for search scenarios in which personal tastes might affect the ranking of results, besides the user query. For example, the advanced search for books at Amazon.com allows users to set several ways to sort results: query relevance, price, average customer review (Figure 1). In this case, another option might be "personalized relevance", which takes into account user profiles.

The paper is organized as follows: First, we describe both the indexing strategy based on WORDNET synsets and the learning method to build semantic profiles from synset-indexed documents. Then, a detailed description of the Personalized Synset Similarity Model is provided together with experimental results about its effectiveness. A brief discussion of related work in personalized retrieval precedes some conclusions and directions for future research, which close the paper.

Learning Semantic User Profiles

The problem of learning user profiles is here considered as a binary Text Categorization task (Sebastiani 2002) since each document has to be classified as interesting or not with respect to the user preferences. Therefore, the set of categories is restricted to c_+, that represents the positive class (*user-likes*), and c_- the negative one (*user-dislikes*).

The proposed strategy to learn semantic profiles consists of two steps. The first one is based on a Word Sense Disambiguation (WSD) technique that exploits the WORDNET lexical database to select, among all the possible meanings (senses) of a polysemous word, the correct one. In the second step, a naïve Bayes approach learns semantic synset-based user profiles from disambiguated documents.

Semantic Indexing of Documents

In order to build semantic user profiles based on the senses (meanings) of words found in a training set of documents, a suitable representation of those documents should be adopted.

A concept-based document representation appears to be the right choice, but then (at least) two main problems must be solved: First, a repository for word senses has to be chosen and integrated; second, an automated procedure for assigning the proper sense to each word occurring in a document has to be designed, developed and integrated. As regards the first problem, WORDNET version 2.0 has been embodied in the semantic indexing module. The basic building block for WORDNET is the synset (SYNonym SET), a structure containing sets of words with synonymous meanings, which represents a specific meaning of a word. As regards the second problem, in Natural Language Processing the task of Word Sense Disambiguation (WSD) consists exactly in determining which sense of an ambiguous word is suitable for a specific occurrence of that word in a document (Manning and Schütze 1999).

Our WSD algorithm, called JIGSAW, takes as input a document $d = [w_1, w_2, \ldots, w_h]$ encoded as a list of words in order of their appearance, and returns a list of WORDNET synsets $X = [s_1, s_2, \ldots, s_k]$ $(k \leq h)$, in which each element s_j is obtained by disambiguating the *target word* w_i based on the *semantic similarity* of w_i with the words in its context. Notice that $k \leq h$ because some words, such as most proper names, might not be found in WORDNET, or because of bigram recognition.

Semantic similarity computes the relatedness of two words. We adopted the Leacock-Chodorow measure (Leacock, Chodorow, and Miller 1998), which is based on the length of the path between concepts in a IS-A hierarchy. Since WSD is not the focus of the paper, the complete description of the adopted strategy is not described here. More details are reported in (Semeraro et al. 2007; Basile et al. 2007a; 2007b). What it is worth to point out here is that the WSD procedure allows to obtain a synset-based vector space representation, called bag-of-synsets (BOS), that is an extension of the classical bag-of-words (BOW) model. In the BOS model a synset vector, rather than a word vector, corresponds to a document.

Moreover the structure of documents is taken into account since each document is represented by a set of *slots*, where each slot denotes a specific feature of the document, and takes a text fragment as its value.

This choice is motivated by the fact that specialized search engines often provide users with advanced search options to query specific portions of documents. For example, the Amazon advanced search on books gives the opportunity to issue separate queries on authors, title, publisher, etc. The adoption of slots does not jeopardize the generality of the approach, since the case of documents not structured into slots corresponds to have just a single slot in our document representation strategy.

In our application scenario, in which documents are movie descriptions, five slots have been selected to represent movies:

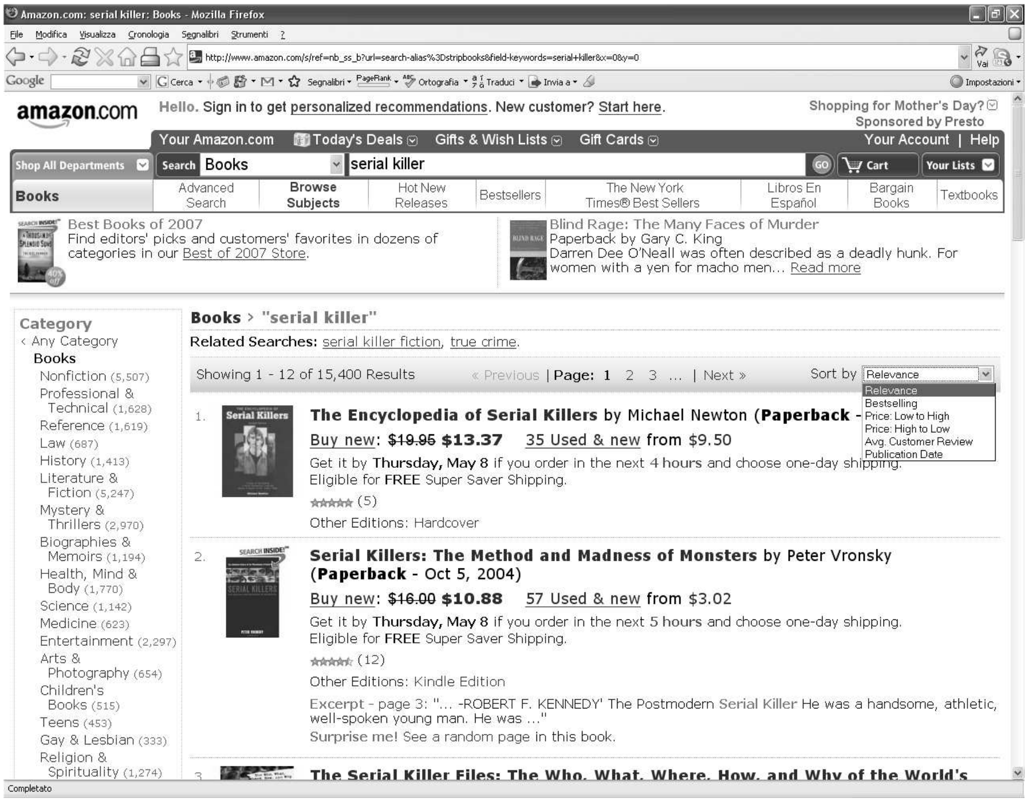
1. *title*, the title of the movie;
2. *cast*, the list of the names of the actors appearing in the movie;
3. *director*, name(s) of the director(s) of the movie;
4. *summary*, a short text that presents the main points of the narration;
5. *keywords*, a list of words describing the main topics of the movie.

In the BOS model, the text in each slot is represented by counting separately the occurrences of a synset in the slots in which it appears.

More formally, assume that we have a collection of N documents. Let *m* be the index of the slot, for $n = 1, 2, ..., N$, the n-th document is reduced to five bag of synsets, one for each slot:

$$d_n^m = \langle t_{n1}^m, t_{n2}^m, \ldots, t_{nD_{nm}}^m \rangle$$

where t_{nk}^m is the k-th synset in slot s_m of document d_n and D_{nm} is the total number of synsets appearing in the m-th slot of document d_n. For all n, k and m, $t_{nk}^m \in V_m$, which is the vocabulary for the slot s_m (the set of all different synsets



http://www.amazon.com/s/ref=nb_ss_b?url=search-alias%3Dstripbooks&field-keywords=serial+killer&x=0&y=0

Hello. Sign in to get personalized recommendations. New customer? Start here.

Shopping for Mother's Day? Sponsored by Presto

Search: Books — serial killer

Best Books of 2007
Find editors' picks and customers' favorites in dozens of categories in our Best of 2007 Store.

Blind Rage: The Many Faces of Murder
Paperback by Gary C. King
Darren Dee O'Neall was often described as a deadly hunk. For women with a yen for macho men... Read more

Category
‹ Any Category
Books
Nonfiction (5,507)
Professional & Technical (1,628)
Reference (1,619)
Law (687)
History (1,413)
Literature & Fiction (5,247)
Mystery & Thrillers (2,970)
Biographies & Memoirs (1,194)
Health, Mind & Body (1,770)
Science (1,142)
Medicine (623)
Entertainment (2,297)
Arts & Photography (654)
Children's Books (515)
Teens (453)
Gay & Lesbian (333)
Religion & Spirituality (1,274)

Books › "serial killer"
Related Searches: serial killer fiction, true crime.

Showing 1 - 12 of 15,400 Results « Previous | Page: 1 2 3 ... | Next » Sort by Relevance

1. **The Encyclopedia of Serial Killers** by Michael Newton (**Paperback** -
Buy new: ~~$19.95~~ **$13.37** 35 Used & new from $9.50
Get it by **Thursday, May 8** if you order in the next **4 hours** and choose one-day shipping.
Eligible for **FREE** Super Saver Shipping.
★★★★★ (5)
Other Editions: Hardcover

2. **Serial Killers: The Method and Madness of Monsters** by Peter Vronsky (**Paperback** - Oct 5, 2004)
Buy new: ~~$16.00~~ **$10.88** 57 Used & new from $3.02
Get it by **Thursday, May 8** if you order in the next **5 hours** and choose one-day shipping.
Eligible for **FREE** Super Saver Shipping.
★★★★★ (12)
Other Editions: Kindle Edition
Excerpt - page 3: "... -ROBERT F. KENNEDY' The Postmodern Serial Killer He was a handsome, athletic, well-spoken young man. He was ..."
Surprise me! See a random page in this book.

3. **The Serial Killer Files: The Who, What, Where, How, and Why of the World's**

Figure 1: Advanced search for books at Amazon.com

found in slot s_m). Document d_n is finally represented in the vector space by five synset-frequency vectors:

$$f_n^m = \langle w_{n1}^m, w_{n2}^m, \ldots, w_{nD_{nm}}^m \rangle$$

where w_{nk}^m is the weight of the synset t_k in the slot s_m of document d_n and can be computed in different ways: it can be simply the number of times synset t_k appears in slot s_m or a more complex TF-IDF score.

A Naïve Bayes Method for Learning WordNet-based User Profiles

A naïve Bayes text categorization algorithm has been developed to build user profiles as binary classifiers (*user-likes* or c_+ vs. *user-dislikes* or c_-). This strategy is implemented by the ITem Recommender (ITR) system (Degemmis, Lops, and Semeraro 2007; Degemmis et al. 2006).

The induced probabilistic model estimates the *a posteriori* probability, $P(c_j|d_i)$, of document d_i belonging to class c_j as follows:

$$P(c_j|d_i) = \frac{P(c_j)}{P(d_i)} \prod_{w \in d_i} P(t_k|c_j)^{N(d_i,t_k)} \qquad (1)$$

where $N(d_i, t_k)$ is the number of times token t_k occurs in document d_i. In ITR, each document is encoded as a vector of BOS in the synset-based representation, or as a vector of BOW in the keyword-based representation, one BOS (or BOW) for each slot. Therefore, equation (1) becomes:

$$P(c_j|d_i) = \frac{P(c_j)}{P(d_i)} \prod_{m=1}^{|S|} \prod_{k=1}^{|b_{im}|} P(t_k|c_j, s_m)^{n_{kim}} \qquad (2)$$

where $S = \{s_1, s_2, \ldots, s_{|S|}\}$ is the set of slots, b_{im} is the BOS or the BOW in the slot s_m of d_i, n_{kim} is the number of occurrences of token t_k in b_{im}. When the system is trained on BOW-represented documents, tokens t_k in b_{im} are words, and the induced categorization model relies on word frequencies. Conversely, when training is performed on BOS-represented documents, tokens are synsets, and the induced model relies on synset frequencies. To calculate (2), the system has to estimate $P(c_j)$ and $P(t_k|c_j, s_m)$ in the training phase. The documents used to train the system are rated on a discrete scale from 1 to MAX, where MAX is the maximum rating that can be assigned to a document. According to an idea proposed in (Mooney and Roy 2000), each training document d_i is labeled with two scores, a "user-likes" score w_+^i and a "user-dislikes" score w_-^i, obtained from the original rating r:

$$w_+^i = \frac{r-1}{MAX-1}; \qquad w_-^i = 1 - w_+^i \qquad (3)$$

The scores in (3) are exploited for weighting the occurrences of tokens in the documents and to estimate their probabilities from the training set TR. The prior probabilities of the classes are computed according to the following equation:

$$\hat{P}(c_j) = \frac{\sum_{i=1}^{|TR|} w_j^i + 1}{|TR| + 2} \qquad (4)$$

Witten-Bell smoothing () is adopted to compute $P(t_k|c_j, s_m)$, by taking into account that documents are structured into slots and that token occurrences are weighted using scores in equation (3):

$$\hat{P}(t_k|c_j, s_m) = \begin{cases} \frac{N(t_k,c_j,s_m)}{V_{c_j}+\sum_i N(t_i,c_j,s_m)} & \text{if } N(t_k, c_j, s_m) \neq 0 \\ \frac{V_{c_j}}{V_{c_j}+\sum_i N(t_i,c_j,s_m)} \frac{1}{V-V_{c_j}} & \text{otherwise} \end{cases} \qquad (5)$$

where $N(t_k, c_j, s_m)$ is the count of the weighted occurrences of token t_k in the slot s_m in the training data for class c_j, V_{c_j} is the total number of unique tokens in class c_j, and V is the total number of unique tokens across all classes. $N(t_k, c_j, s_m)$ is computed as follows:

$$N(t_k, c_j, s_m) = \sum_{i=1}^{|TR|} w_j^i n_{kim} \qquad (6)$$

In (6), n_{kim} is the number of occurrences of token t_k in slot s_m of document d_i. The sum of all $N(t_k, c_j, s_m)$ in the denominator of equation (5) denotes the total weighted length of the slot s_m in class c_j. In other words, $\hat{P}(t_k|c_j, s_m)$ is estimated as the ratio between the weighted occurrences of t_k in slot s_m of class c_j and the total weighted length of the slot. The final outcome of the learning process is a probabilistic model used to classify a new document in the class c_+ or c_-. This model is the user profile, which includes those tokens that turn out to be most indicative of the user preferences, according to the value of the conditional probabilities in (5).

Given a new document d_j, the profile computes the a-posteriori classification scores $P(c_+|d_j)$ and $P(c_-|d_j)$ by using probabilities of synsets contained in the user profile and estimated in the training step.

In order to compare the accuracy of WORDNET-based profiles with that of keyword-based profiles, we performed an experimental evaluation of a content-based extension of the well known EACHMOVIE dataset[3]. The main result of the experiments is that synset-based profiles outperformed keyword-based ones in suggesting relevant movies to users (average accuracy 81% vs. 73%). More details are reported in (Degemmis, Lops, and Semeraro 2007).

In the retrieval scenario, we do not use directly the classification scores to select documents to be recommended. In fact, in the following section we will describe a formal model that exploits the classification score for the class c_+ to modify the ranking of documents in the result list obtained in response to a user query.

A Model for Personalized Searching

This section describes a personalized searching strategy by proposing a retrieval model in which user profiles learned by ITR are exploited to extend the traditional query-document retrieval paradigm. First, we introduce a semantic retrieval model based on WordNet synsets, the Synset Similarity Model (SSM), in which the similarity between a document and a query, represented through the BOS model, is computed according to a synset similarity function. Then, a strategy that extends the SSM to a Personalized SSM (Semeraro 2007), by including synset-based user profiles in computing the ranking function, is described.

Synset Similarity Model

According to (Baeza-Yates and Ribeiro-Neto 1999), an information retrieval model must define:

- a proper representation for documents and user queries;
- a relevance function $R(q_i, d_j)$ which computes the *degree of similarity* between each document d_j in the collection and the user query q_i. $R(q_i, d_j)$ may define an ordering among the documents with respect to the query.

In (Gonzalo et al. 1998), the authors performed a shift of representation from a lexical space, where each dimension is represented by a term, towards a semantic space, where each dimension is represented by a WORDNET synset. Then, they adapted the VSM to WordNet synsets. The implementation of the semantic tf-idf model was rather simple because it indexes documents and queries by using strings representing synset identifiers. The retrieval phase was similar to the classic tf-idf model, with the only difference that matching was carried out between synset identifiers. An exact matching between synsets is not enough to understand how similar the meanings of the concepts are. Thus, it is necessary to redefine the similarity between a document and a query (Corley and Mihalcea 2005). Computing the degree of relevance of a document with respect to a query means computing the similarity among the synsets of the document and the synsets of the query.

The SSM proposed in the paper computes the semantic similarity between the set of synsets of the query and that of the document by extending the approach in (Smeaton and Quigley 1996), which computes the maximum similarity score for each synset in the query by comparing it to each synset in the document. The sum of all maximum similarity measures obtained for each synset in the query is then divided by the number of synsets in the query:

$$R(q_i, d_j) = \frac{\sum_{i=1}^{m} max_{j=1,\ldots,n}[\textsc{SynSim}(q_{ik}, s_{jh})]}{m} \qquad (7)$$

where q_{ik} is the k-th synset in q_i, s_{jh} is the h-th synset in d_j, m is the number of synsets in q_i and n is the number of synsets in d_j. Notice that Eq. 7 does not take into account the importance of s_{jh} in d_j; on the contrary, the semantic tf-idf model proposed in (Gonzalo et al. 1998) was mainly based on this crucial aspect. For this reason, we decided to take into account the importance of the synsets in the document by multiplying the semantic similarity between the

[3] EACHMOVIE dataset no longer available for download: `http://www.cs.umn.edu/Research/GroupLens/`

pair of synsets (q_{ik}, s_{jh}) by the weight w_{jh} in the synset-frequency vector for d_j. SYNSIM is redefined as:

$$\text{SYNSIM}_{SSM}(q_{ik}, s_{jh}) = w_{jh} \cdot \text{SYNSIM}(q_{ik}, s_{jh}) \quad (8)$$

The relevance in Eq. (7) is modified by replacing $\text{SYNSIM}(q_{ik}, s_{jh})$ with $\text{SYNSIM}_{SSM}(q_{ik}, s_{jh})$. In the SSM, w_{jh} is the same tf-idf for synsets in (Gonzalo et al. 1998), and $\text{SYNSIM}(q_{ik}, s_{jh})$ is the Leacock-Chodorow measure adopted in the WSD step. As a proof of concept, we developed a movie retrieval system based on the SSM.

Personalized Synset Similarity Model

This section proposes a possible strategy to extend an information retrieval model by introducing the user profile P_u, representing the long-term preferences of user u, in the relevance computation.

$R(q_i, d_j, P_u)$ associates a real number with a query q_i, a document d_j, and the profile of user u, thus defining a *personalized* ordering among documents with respect to the information needs of user u, expressed both by q_i and P_u. In the extended model, called Personalized Synset Similarity Model (PSSM), $R(q_i, d_j, P_u)$ starts from the ranking computed by $R(q_i, d_j)$ in the SSM, and defines a new ranking which also takes into account the classification score $P(c_+|d_j)$ computed by using P_u.

Let w_k be the relevance score assigned by the function $R(q_i, d_j)$ to the k-th document in the ranking, and p_k the probability $P(c_+|d_j)$, assigned by the user profile, that the k-th document in the ranking is liked by the user. The definition of the re-ranking function is based on the following two heuristics:

1. The impact of the probability p_k on $R(q_i, d_j, P_u)$ should be non-linear: If the probability of interest in an item is close to 0.5, w_k should remain nearly the same, because this indicates a high level of uncertainty on the prediction. In such a situation the best choice is to trust the SSM decision. Conversely, when the value of p_k is close to 0 or 1, w_k should be strongly modified;
2. It is reasonable that w_k should be updated proportionally to its value.

The first heuristic is realized through a function f, which computes a *preference impact score* for each item in the result set. This score measures how much that item should be moved up or down in the ranked list of results, in reason of the degree of user interest.

More specifically, function f has the following properties:

- defined in the interval $[0, 1]$ that represents the range of the probability of interest in an item d_j, $P(c_+|d_j)$;
- items with a higher degree of interest ($P(c_+|d_j) > 0.5$) receive a *positive* score which amplifies $R(q_i, d_j)$ so that d_j is *moved up* in the ranking;
- items with a lower degree of interest ($P(c_+|d_j) < 0.5$) receive a negative score which decreases $R(q_i, d_j)$ so that d_j is *moved down* in the ranking;
- the codomain is the interval $[-0.5, 0.5]$, therefore $f(p) : [0, 1] \rightarrow [-0.5, 0.5]$;
- f is a growing function and its values in the interval $[0.4, 0.6]$ indicate an absence of precise preferences. The maximum uncertainty is reached when $P(c_+|d_j) = 0.5$, therefore $f(0.5) = 0$;
- Out of the interval $[0.4, 0.6]$, the function grows to reach values that have a heavier impact on the final ranking. In particular, in the intervals $[0, 0.2]$ and $[0.8, 1]$ the function becomes almost constant by achieving values close to its minimum and maximum respectively.

Function f was built by interpolating some specific data points and taking into account the above listed properties, obtaining the following equation:

$$f(p) = \begin{cases} -\frac{5}{2}p^2 + \frac{19}{4}p - \frac{7}{4} & \text{if } p > 0.5 \\ \frac{5}{2}p^2 - \frac{1}{4}p - \frac{1}{2} & \text{if } p \leq 0.5 \end{cases} \quad (9)$$

Figure 2 depicts the graph of the function f.

The second heuristic is realized by another function, which relates the probability $P(c_+|d_j)$ with the weight w_k. The main aim of this function is to amplify or decrease the relevance score w_k for an item in direct proportion of the probability p_k.

We denote with g the function which computes this new value for an item to be re-ranked:

$$g(w_k, p_k) = w_k \cdot (p_k - 0.5) \quad (10)$$

$g(w_k, p_k)$ has positive values for $p_k > 0.5$ and negative ones for $p_k < 0.5$. This means that, if an item is liked, the value associated to the re-ranking function increases; if an item is not liked, the value decreases again in a proportional way with respect to the initial weight w_k.

The final re-ranking function of the PSSM is:

$$\begin{aligned} R(q_i, d_j, P_u) &= R(q_i, d_j) + f(P(c_+|d_j)) \\ &+ g(R(q_i, d_j), P(c_+|d_j)) \quad (11) \end{aligned}$$

In this way, the value computed by the system for each item, and used for ordering results to be presented to the user, varies not only on the ground of the probability given by the profile, but also in a way proportional to the relevance defined by the SSM.

Table 1 shows an example of ranking 12 items obtained by a user submitting the query "love comedy" to the movie retrieval system. The user previously rated a set of 30 items and the corresponding profile was learned by ITR. After a manual query disambiguation, items are ordered in a descending order according to $R(q_i, d_j)$. The first and the second column report the position in the SSM and the value $R(q_i, d_j)$ respectively. The next three columns indicate the value of the intermediate functions and the last two ones report the final value of the re-ranking function and the resulting position in the PSSM.

We can notice that the value $R(q_i, d_j)$ of the first item is close to 1; at the same time it is deemed relevant according to the profile of the user that issued the query. These two factors together make stronger the item's leadership thanks

Figure 2: The graph of the function defined in Equation (9)

Table 1: Ranking of items in the result set of a query submitted to SSM and PSSM

SSM	$R(q_i, d_j)$	$P(c_+\|d_j)$	$f(P(c_+\|d_j))$	$g(R(q_i, d_j), P(c_+\|d_j))$	$R(q_i, d_j, P_u)$	**PSSM**
1	0.96	0.89	0.498	0.378	1.836	1
2	0.89	0.34	-0.292	-0.139	0.455	7
3	0.85	0.32	-0.325	-0.154	0.373	8
4	0.82	0.91	0.501	0.331	1.649	2
5	0.81	0.04	-0.506	-0.375	-0.072	12
6	0.77	0.29	-0.359	-0.159	0.248	9
7	0.55	0.94	0.506	0.238	1.292	3
8	0.30	0.91	0.502	0.122	0.921	4
9	0.20	0.68	0.327	0.037	0.569	6
10	0.20	0.37	-0.248	-0.026	-0.071	11
11	0.17	0.80	0.453	0.052	0.675	5
12	0.02	0.54	0.078	0.001	0.099	10

to the way in which the function (11) was defined. It is interesting to observe items at positions 7, 8, 9, 11 of the SSM ranking: They are ranked down in the list because of the low degree of matching with the query in the SSM. Nevertheless, since each of them has a high probability of interest, they are respectively at positions 3, 4, 6, 5 in the ranking computed in the PSSM. Symmetrically, items at positions 2, 3, 5, 6, which are very relevant with respect to the query, are ranked down in the PSSM list at positions 7, 8, 12, 9, because they are not too relevant with respect to the user profile.

Experimental Evaluation

The main aim of the experimental session is to compare the effectiveness of the proposed PSSM with that of the SSM. We investigated whether the introduction of long-term preferences in the search process has a positive effect on the accuracy of retrieved results. In particular, experiments are devoted to verify whether the adoption of content-based user profiles produces a hopefully better ranking than the one obtained by using just the query.

Experiments were performed on a collection of $1,628$ documents corresponding to movie descriptions obtained by crawling the Internet Movie Database[4]. The crawler gathers the title, the director, the genre (category of the movie), the list of keywords, the plot and the cast. Documents in the collection have been semantically indexed by using the WSD procedure described before. The number of synsets in the collection was $107,990$ (against $172,296$ words). Eight real users were involved in the experiment. Each user submitted a number of queries to the SSM search engine and rated a number of documents in the result list, in order to collect training examples for ITR. After the training, the synset-based profile of each user was generated. Then, each user was requested to submit 3 different queries, according to her information needs, to both the SSM and the PSSM search engines. Queries are manually disambiguated by the user herself by selecting appropriate senses from WordNet. For each result list, top 10 documents were examined by the user and the relevance of each document was judged according to a 1-6 rating scale. Therefore, after collecting relevance feedback, two pairs of rankings were available: SSM ranking with the corresponding ideal ranking set by the user

[4] The Internet Movie Database, http://www.imdb.com. Accessed on February 7, 2008

feedback, and PSSM ranking with the corresponding ideal ranking. Movies rated for training ITR were withheld in this phase, in order to prevent ranking from being affected by documents already used in the training step.

Rank accuracy metrics measure the ability of a system to suggest an ordering of items that matches how the user would have ordered the same items. In our study, we adopted the Normalized Distance-based Performance Measure (NDPM) (Yao 1995), because our aim was to compare the ability of the two models in producing effective document ranking. Values range from 0 (agreement) to 1 (disagreement). Two NDPM values are produced for each query q_i submitted by a user. The first value comes from the comparison between the SSM ranking and the user ranking on the top 10 documents in the result list for q_i. The second value comes from the comparison between PSSM ranking and user ranking for q_i. Table 2 reports, for each user, NDPM values for the 3 queries submitted.

We observed that for 18 out of the 24 queries, PSSM outperforms SSM. An interesting remark is that only for User 5 it happens that 2 queries out of 3 produce a better ranking in the SSM than in the PSSM, thus revealing that the user profile introduced some noise in the search process. We analyzed the training documents provided by that user and we found that a few number of training examples (only 10, while other users provided up to 20 examples) was given. Moreover, the rating style of this user was very confusing because he was inclined to assign ratings standing for “I like it, but not too much” or “I dislike it, I could even like it”. Other users, like User 2, 3, 7 and 8, had a very clean rating style, that is, they are inclined to assign the score 1 to not interesting documents, and the score 6 to interesting ones. We can conclude that this negative result for the PSSM depends on the noise in the training set used as input to ITR. Anyway, the main observation that can be drawn is that the adoption of synset-based user profiles in the SSM gives a better performance than using the SSM alone. This tends to imply that it is worthwhile to perform personalized search. In order to validate this feeling, we performed a Wilcoxon signed ranked test, requiring a significance level $p < 0.05$. The set of 3 queries submitted by a user was considered as a single trial and the averaged NDPM values were used for the test. The test confirmed that there is a statistically significant difference in favor of the PSSM compared to the SSM.

Related Work

The main idea underlying most of the works in the area of information filtering (Belkin and Croft 1992; Hanani, Shapira, and Shoval 2001) and intelligent recommendation agents (Pazzani and Billsus 1997; Mladenic 1999; Joachims, Freitag, and Mitchell 1997; Bollacker, Lawrence, and Giles 1999) is to construct user profiles, either explicitly or implicitly, by using machine learning techniques, as in our work, and then *to recommend documents directly on the ground of the user profiles*.

The technique we employ is different since our aim is to personalize search results by including user preferences in the ranking function of the retrieval model. Therefore, we do not view learned profiles as filters or long-term interests queries, but we embed them in the search model to rank documents according to both the user query and the profile information.

Among the state-of-the-art systems for personalized retrieval, WebMate (Chen and Sycara 1998) exploits user profiles to perform search refinement by keywords expansion and relevance feedback, while Inquirus 2 (Glover et al. 2001) requires the users to provide *explicit* preferences on categories, which are employed to expand queries, even though it does not learn profiles from the user interaction.

Different approaches based on *implicit* feedback have been proposed (Joachims and Radlinski 2007; Joachims et al. 2007; Agichtein, Brill, and Dumais 2006). The main idea is to understand how users interact with a search engine and how this relates to their preferences. It has been shown that a search engine can reliably infer ranking functions tailored to a particular user group or collection from the user’s clicks. One common approach to personalized retrieval exploits documents that a user creates, copies, or employs on a client machine to build a client-side index treated as a personal profile. The profile is used to disambiguate the query terms and to improve results by re-ranking relevant documents within search results (Teevan, Dumais, and Horvitz 2005). Other researchers, who investigate how to learn from implicit feedback, use search selection histories to choose a *topic-sensitive PageRank* value for each returned search result, which is then used to rank those results (previously selected search results serve as biased indicators of user interests). The strategy proposed in (Liu, Yu, and Meng 2004) learns a user profile based on both the search history of the user and a common category hierarchy, typically used by search engines to help users to specify their intentions. The categories that are likely to be of interest for the user are inferred from her current query and profile, and are used as a context for the query to improve retrieval effectiveness. The user profile consists of a set of categories and, for each category, a set of keywords with corresponding weights.

Similarly, the ARCH (Adaptive Retrieval based on Concept Hierarchies) system (Sieg et al. 2003) exploits user profiles to automatically learn the semantic context of user’s information need but, differently from (Liu, Yu, and Meng 2004), a concept hierarchy is exploited rather than a common category hierarchy.

Our approach is different since we directly embed user profiles in the retrieval model, by including them in the computation of the similarity score, rather than acting on the user query. Moreover, a distinctive feature of our approach is that the construction of user profiles is based on the WORDNET IS-A hierarchy, which is exploited in the indexing step by a WSD algorithm that maps words to synsets. User profiles are learned in form of text classifiers from semantically indexed training documents, thus obtaining synset-based profiles which can effectively support the user in the retrieval step. To the best of our knowledge, none of the systems described proposes a formal retrieval model based on semantic user profiles.

Table 2: Performance of SSM and PSSM in a movie retrieval scenario

	SSM				**PSSM**			
$User$	Q_1	Q_2	Q_3	$Avg.$	Q_1	Q_2	Q_3	$Avg.$
1	0.44	0.29	0.34	0.36	0.44	0.29	0.17	0.30
2	0.71	0.66	0.60	0.66	0.43	0.50	0.44	0.46
3	0.30	0.20	0.34	0.28	0.20	0.14	0.17	0.17
4	0.39	0.27	0.27	0.31	0.24	0.54	0.07	0.28
5	0.43	0.32	0.38	0.38	0.58	0.40	0.17	0.38
6	0.56	0.43	0.34	0.44	0.60	0.34	0.17	0.37
7	0.71	0.66	0.50	0.62	0.43	0.50	0.43	0.45
8	0.61	0.66	0.50	0.59	0.34	0.50	0.37	0.40

Conclusions and Future Work

This paper described a methodology for including user profiles in retrieval scenarios where the role of user preferences strongly affects the acceptance of the results. The main element the proposed strategy is based upon is the Personalized Synset Similarity Model, that extends the semantic retrieval model, called Synset Similarity Model, in which the similarity between a document and a query is computed according to a synset similarity function, by including synset-based user profiles in the computation of query-document similarity.

Experimental results indicate that PSSM retrieval effectiveness is higher than the SSM one, thus the general conclusion is that a personalized semantic space is better than a semantic space which does not take into proper consideration user preferences.

As a future work, domain-dependent knowledge sources will be integrated into the synset-based linguistic approach in order to obtain a more powerful retrieval model. Moreover, we will investigate on how to include user generated content (such as tags), which users might choose to freely annotate relevant documents, in the profile generation process. Another planned extension is the development of a full-fledged multilingual information seeking resource based on SENSE (Basile et al. 2008), a recently developed semantic search engine. Experiments on a larger dataset will be carried out.

Acknowledgments

This research was partially funded by MIUR (Ministero dell'Università e della Ricerca) under the contract Fondo per le Agevolazioni alla Ricerca, DM19410 "Laboratorio di Bioinformatica per la Biodiversità Molecolare" (2007-2011), by Regione Puglia under the contract "Distretto digitale a supporto della filiera produttiva del tessile e abbigliamento", and by MIUR (Ministero dell'Università e della Ricerca) under the contract Legge 297/99, Prot.691 CHAT "Cultural Heritage fruition & e-Learning applications of new Advanced (multimodal) Technologies" (2006-08).

References

Agichtein, E.; Brill, E.; and Dumais, S. T. 2006. Improving web search ranking by incorporating user behavior information. In Efthimiadis, E. N.; Dumais, S. T.; Hawking, D.; and Järvelin, K., eds., *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA*, 19–26. ACM Press.

Baeza-Yates, R., and Ribeiro-Neto, B. 1999. *Modern Information Retrieval*. Addison-Wesley.

Basile, P.; de Gemmis, M.; Gentile, A.; Lops, P.; and Semeraro, G. 2007a. JIGSAW algorithm for word sense disambiguation. In *SemEval-2007: 4th Int. Workshop on Semantic Evaluations*, 398–401. ACL press.

Basile, P.; Degemmis, M.; Gentile, A. L.; Lops, P.; and Semeraro, G. 2007b. The JIGSAW algorithm for word sense disambiguation and semantic indexing of documents. In Basili, R., and Pazienza, M. T., eds., *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing, 10th Congress of the Italian Association for Artificial Intelligence, Rome, Italy, September 10-13, 2007, Proceedings*, volume 4733 of *Lecture Notes in Computer Science*, 314–325. Springer.

Basile, P.; Caputo, A.; Gentile, A.; de Gemmis, M.; Lops, P.; and Semeraro, G. 2008. Enhancing semantic search using n-levels document representation. In *Proceedings of the ESWC 2008 Workshop on Semantic Search (SemSearch 2008)*. To appear.

Belkin, N., and Croft, B. 1992. Information filtering and information retrieval. *Communications of the ACM* 35(12):29–37.

Bollacker, K. D.; Lawrence, S.; and Giles, C. L. 1999. A system for automatic personalized tracking of scientific literature on the web. In *Proceedings of the Fourth ACM conference on Digital Libraries*, 105–113. ACM Press.

Chen, L., and Sycara, K. P. 1998. Webmate: A personal agent for browsing and searching. In *Proceedings of the Second International Conference on Autonomous Agents*, 132–139. ACM Press.

Chung, W. May 2008. Web searching in a multilingual world. *Communications of the ACM* 51(5):32–40.

Corley, C., and Mihalcea, R. 2005. Measures of text semantic similarity. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence*, 13–18.

Degemmis, M.; Lops, P.; Ferilli, S.; Di Mauro, N.; Basile, T.; and Semeraro, G. 2006. Text learning for user profiling in e-commerce. *International Journal of Systems Science* 37(13):905–918.

Degemmis, M.; Lops, P.; and Semeraro, G. 2007. A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research (UMUAI)* 17(3):217–255.

Glover, E. J.; Flake, G. W.; Lawrence, S.; Birmingham, W. P.; Kruger, A.; Giles, C. L.; and Pennock, D. M. 2001. Improving category specific web search by learning query modifications. In *Symposium on Applications and the Internet (SAINT)*, 23–32.

Gonzalo, J.; Verdejo, F.; Chugur, I.; and Cigarrán, J. M. 1998. Indexing with wordnet synsets can improve text retrieval. In *Proceedings of the COLING/ACL Workshop on Usage of WordNet for NLP*, 38–44.

Hanani, U.; Shapira, B.; and Shoval, P. 2001. Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research (UMUAI)* 11(3):203–259.

Joachims, T., and Radlinski, F. 2007. Search engines that learn from implicit feedback. *IEEE Computer* 40(8):34–40.

Joachims, T.; Granka, L. A.; Pan, B.; Hembrooke, H.; Radlinski, F.; and Gay, G. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems* 25(2):article 7.

Joachims, T.; Freitag, D.; and Mitchell, T. 1997. Web watcher: A tour guide for the world wide web. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, IJCAI-97*, 770–777. Morgan Kaufmann.

Leacock, C.; Chodorow, M.; and Miller, G. 1998. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics* 24(1):147–165.

Liu, F.; Yu, C. T.; and Meng, W. 2004. Personalized web search for improving retrieval effectiveness. *IEEE Transactions on Knowledge and Data Engineering* 16(1):28–40.

Manning, C., and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press. chapter 7: Word Sense Disambiguation, 229–264.

Miller, G. A. 1995. WordNet: a lexical database for English. *Communications of the ACM* 38(11):39–41.

Mladenic, D. 1999. Text-learning and related intelligent agents: a survey. *IEEE Intelligent Systems* 14(4):44–54.

Mooney, R. J., and Roy, L. 2000. Content-based book recommending using learning for text categorization. In *Proceedings of the 5th ACM Conference on Digital Libraries*, 195–204. ACM Press.

Pazzani, M., and Billsus, D. 1997. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning* 27(3):313–331.

Qiu, F., and Cho, J. 2006. Automatic identification of user interest for personalized search. In Carr, L.; Roure, D. D.; Iyengar, A.; Goble, C. A.; and Dahlin, M., eds., *Proceedings of the 15th International Conference on World Wide Web (WWW)*, 727–736.

Sebastiani, F. 2002. Machine learning in automated text categorization. *ACM Computing Surveys* 34(1):1–47.

Semeraro, G.; Degemmis, M.; Lops, P.; and Basile, P. 2007. Combining learning and word sense disambiguation for intelligent user profiling. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence IJCAI-07*, 2856–2861.

Semeraro, G. 2007. Personalized searching by learning wordnet-based user profiles. *Journal of Digital Information Management* 5(5):309–322.

Sieg, A.; Mobasher, B.; Lytinen, S. L.; and Burke, R. D. 2003. Concept based query enhancement in the ARCH search agent. In Arabnia, H. R., and Mun, Y., eds., *International Conference on Internet Computing*, 613–619. CSREA Press.

Smeaton, A. F., and Quigley, I. 1996. Experiments on using semantic distances between words in image caption retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, 174–180. ACM Press.

Teevan, J.; Dumais, S. T.; and Horvitz, E. 2005. Personalizing search via automated analysis of interests and activities. In Baeza-Yates, R. A.; Ziviani, N.; Marchionini, G.; Moffat, A.; and Tait, J., eds., *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 449–456. ACM Press.

Teevan, J.; Dumais, S. T.; and Horvitz, E. 2007. Characterizing the value of personalizing search. In Kraaij, W.; de Vries, A. P.; Clarke, C. L. A.; Fuhr, N.; and Kando, N., eds., *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 757–758. ACM Press.

Witten, I., and Bell, T. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory* 37(4):1085 – 1094.

Yao, Y. Y. 1995. Measuring retrieval effectiveness based on user preference of documents. *Journal of the American Society for Information Science* 46(2):133–145.