Personalized Searching by Learning WordNet-based User Profiles

Giovanni Semeraro
Department of Informatics
University of Bari
Via E. Orabona, 4
70126 – Bari
Italy
semeraro@di.uniba.it

Journal of Digital Information Management

ABSTRACT: *The amount of information available on the Web and in Digital Libraries is increasing over time. In this context, the role of user modeling and personalized information access is becoming crucial: Users need a personalized support in sifting through large amounts of retrieved information according to their interests. Information filtering and retrieval systems relying on this idea adapt their behavior to individual users by learning their preferences during the interaction in order to construct a profile of the user that can be later exploited in the search process. We propose a novel technique to learn user profiles which exploits word sense disambiguation based on the WordNet lexical database, in an attempt to produce semantic user profiles that might discover topics semantically closer to the user interests. Semantic profiles are used in the definition of a retrieval model that turns the traditional document-query search paradigm into a novel document-query-profile paradigm. As an example of this paradigm, we present an extension of the vector space model in which profiles are used to modify the ranking of search results obtained in response to a query, hopefully putting personally relevant items on the top of the result list. Experimental results in a movie retrieval scenario indicate that the proposed model to personalize Web search is effective.*

Categories and Subject Descriptors
H.3.1 [Content analysis and indexing]; Lingusitics Processing: **H.3.7** [**Digital libraries]**; Web based services **I.2.7 [Natural Language Processing**]

General Terms
Web search, Lexical databases, Information access, Lingusitic analysis

Keywords: User modeling, Personalization and Information Filtering, Machine learning, WordNet

Received 7 August 2006; Revised 11 March 2007; Accepted 12 April 2007

1. Introduction

There are likely to be thousands of documents with content relevant to any particular query, while distinct users issuing the same query usually have different interests and information needs.

Search engines usually adopt a "one fits all" approach, which takes into little account user individual needs and preferences, but there are information access scenarios that cannot be solved through a straightforward matching of queries and documents. For example, a user looking for "interesting movies about criminal minds" or "interesting news about serial killers" cannot easily express this form of information need as a query suitable for search engines. Even if the user can easily formulate the query "criminal minds" or "serial killers", other elements typically influence the relevance of the retrieved results, like the plot of the movie or the nature of the committed crime.

This situation creates many new challenges for Web search. Recent developments at the intersection of Information Retrieval, Information Filtering, Machine Learning, User Modeling and Natural Language Processing offer novel solutions for *personalized information access*. Most of this work focuses on the use of Machine Learning algorithms [Mitchell, 1997] for the automated induction of a structured model of user interests and preferences from text documents, referred to as *user profile*. If a profile accurately reflects the preferences of a user, it is of tremendous advantage for her. For instance, it could be used to filter search results, by deciding whether a user is interested in a specific Web page or not and, in the negative case, preventing it from being displayed. Another way to personalize Web search by means of user profiles is to expand or refine the original query issued by the user by including keywords extracted from her personal profile. The problem with this approach is that traditional keyword-based profiles are unable to capture the semantics of the user interests because they are primarily driven by a string-matching operation. For the query "bat", some users may be interested in documents dealing with "bat" as "nocturnal mammal", while other users may want documents related to baseball. If the word "bat" is found in the profile, a match is made but, due to polysemy, incorrect expansion could be performed or inappropriate filtering could be applied, therefore including not relevant documents in the result list.

To solve these problems, we propose a method to improve retrieval effectiveness which is mainly based on:

- Alternative techniques able to learn more accurate profiles that capture concepts expressing user interests from relevant documents. These *semantic profiles* will contain references to concepts defined in lexicons or ontologies. Although methods for learning semantic profiles clearly require additional knowledge and processing, they potentially have a number of advantages: For example, if a user likes documents about *robotics* and *machine learning*, a method with the ability to identify these concepts (and that has access to the proper concept hierarchy) could infer that the user is interested in *artificial intelligence*;
- An extended search paradigm in which user preferences stored in semantic profiles are included in the computation of query-document similarity. As a consequence, the order of documents in the result list is modified according to the user preferences, thus producing a personalized ranking, in an attempt to improve retrieval effectiveness.

In the following, we summarize the contributions of the paper:

- We provide a strategy which integrates external knowledge sources, such as generalization hierarchies,

in the process of learning semantic user profiles. In our approach, WordNet [Miller, 1990] is employed as a reference lexicon for substituting word forms with the correct word meanings in the document indexing step. The association between word forms and corresponding meanings is performed by a word sense disambiguation procedure which takes advantage of the hierarchical structure of WordNet. The adoption of this semantic indexing approach allows learning algorithms to work on disambiguated documents, thus inducing more accurate WordNet-based user profiles. We made an empirical evaluation which showed that the accuracy of WordNet-based profiles is higher than that of keyword-based profiles;

- According to the results of the experiments about profiles' accuracy, we propose a retrieval model to personalize Web search which relies on both the query and the WordNet-based profile of the user. We called this model "Personalized Synset Similarity Model" because it extends the classical vector space model:
  - by using WordNet concepts, called *synsets*, to index documents rather than keywords;
  - by adopting a similarity function able to deal with synsets, so that a concept matching between query and document is realized, rather than a string matching;
  - by including the user profile in the computation of the query-document similarity score. In this way, the user profile contributes in ranking documents in the result list.
- In order to evaluate our personalized approach to Web search, we select as workbench a scenario in which user preferences really affect the user acceptance of retrieved results: movie retrieval. The Internet Movie DataBase Web Site[1] was crawled in order to obtain a dataset on which a thorough experimental session was performed. We make the following comparisons and show that:
  - The accuracy of synset-based profiles is higher than that of keyword-based profiles;
  - The introduction of synset-based profiles in the vector space model improves retrieval effectiveness.

1.1 Related Work

Many papers have been published in the area of Information Filtering and Intelligent Recommendation Agents.

Syskill & Webert [Pazzani and Billsus, 1997] is an agent that learns a user profile exploited to identify interesting Web pages. The learning process is performed by first converting HTML source into positive and negative examples, represented as keyword vectors, and then using learning algorithms like Bayesian classifiers, a nearest neighbor algorithm and a decision tree learner.

Personal WebWatcher [Mladenic, 1999] is a Web browsing recommendation agent that accompanies the user from page to page and highlights interesting hyperlinks. It generates a user profile based on the content analysis of the requested pages without asking the user to provide any keywords or ratings. Learning is done by a naïve Bayes classifier, documents are represented as weighted keyword vectors, and classes are "interesting" and "not interesting".

[1] The Internet Movie Database, http://www.imdb.com Accessed on December 6, 2006.

Mooney and Roy [2000] adopt a text categorization method in their Libra recommendation agent that performs content-based filtering by exploiting product descriptions obtained from the Web pages of the Amazon on-line digital store. Also in this case, documents are represented by using keywords and a naïve Bayes text classifier is adopted.

A difference between our work and these approaches is that they represent documents by using keywords. In our approach, each document is represented by a vector of weighted WordNet synsets obtained by a word sense disambiguation procedure applied to the words occurring in the document.

Moreover, the above discussed systems construct user profiles, explicitly or implicitly, and use them to recommend documents. The technique we employ is different. While previous methods directly exploit user profiles to filter documents, our aim is to introduce user profiles in the search model to improve retrieval effectiveness by including user preferences in the ranking function.

Among the state-of-the-art systems in the area of personalized Web Search, 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 of categories which are employed to expand queries, but it does not have profiles learned from the user interaction. The strategy proposed in [Liu et al., 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 to the user are inferred from her current query and the profile, and are used as a context of 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. A similar idea has also been exploited in the ARCH (Adaptive Retrieval based on Concept Hierarchies) system [Sieg et al., 2004] in which user profiles are used to automatically learn the semantic context of user's information need but, differently from [Liu et al., 2004], a *concept* hierarchy is exploited rather than a common category hierarchy.

Our approach is different from all the above in that we try to embed user profiles directly 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. In more detail, the WordNet hierarchy is exploited in the indexing step by a word sense disambiguation procedure which maps words into 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.

A remarkable attempt to indexing documents according to WordNet senses is reported in [Mihalcea and Moldovan, 2000]. The authors designed an information retrieval system performing a combined word-based and sense-based indexing and retrieval. They added lexical and semantic information to both the query and the documents during a pre-processing step in which the query and the text are disambiguated. Inspired by this work, we propose a strategy to integrate *semantic* synset-based profiles in a Vector Space Model applied to WordNet synsets, which extends the classical model based on a *lexical* space to a *semantic* space, where each dimension is represented by a concept expressed using WordNet synsets.

To the best of our knowledge, none of the available systems described in this section proposes a formal retrieval model based on semantic user profiles.

1.2 Outline of the paper

The paper is organized as follows: In Section 2, after a brief introduction about our vision of the user profiling task, we describe the word sense disambiguation strategy adopted to represent documents by using WordNet synsets. A detailed description of the learning method that allows to infer synset-based profiles is given in Section 3, which provides also an experimental evaluation aiming at comparing synset-based profiles to keyword-based profiles. Section 4 proposes a semantic retrieval model based on WordNet synsets, named the *Synset Similarity Model (SSM),* extended with a possible strategy for including knowledge contained in user profiles (*Personalized Synset Similarity Model*). In Section 5, experimental results about the effectiveness of the proposed personalized search strategy are evaluated and discussed. Some final conclusions are drawn in the last section of the paper.

2. Using WordNet to Represent Documents

In a Machine Learning approach to Text Categorization, a general inductive process automatically builds a text classifier by learning, from a set of training documents (documents labeled with the categories they belong to), the features of the categories. Many inductive approaches have been proposed [Sebastiani, 2002], including numerical learning, such as Bayesian classification [Lewis, 1998], and symbolic learning [Moulinier and Ganascia, 1996; Lewis and Ringuette, 1994].

We consider the problem of learning user profiles as a binary Text Categorization task: 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). There are several ways in which content can be represented in order to be used as a basis for the learning component and there exists a variety of machine learning methods that could be exploited for inferring user profiles. We propose a strategy to learn sense-based profiles that consists of two steps. This section describes the first one, that is, a word sense disambiguation technique that exploits the word senses in WordNet to represent documents. In the second step, described in Section 3, a naïve Bayes approach learns sense-based user profiles as binary text classifiers from disambiguated documents.

2.1 Keyword-based and Synset-based Document Representation

In the case of text categorization, the selection of appropriate document features is usually referred to as *document representation*. Many document representations appeared in the literature [Kehagias et al., 2003; Yang and Pedersen, 1997] and most of them are based on the use of the words occurring in a document. In the classic bag-of-words (BOW) model, each feature used to represent a document corresponds to a single word found in the document.

We propose a document representation that can be exploited as a starting point to build a more accurate profile of the user interests, that we call *semantic user profile* since it is based on the senses of words found in the training documents. Here, *word sense* is used as a synonym of *word meaning*. There are two crucial issues to address: First, a repository for word senses has to be identified; second, any implementation of sense-based document representation must solve the problem that, while words occur in a document, meanings do not, since they are often hidden in the context. Therefore, a procedure is needed for assigning senses to words: The task of word sense disambiguation (WSD) consists in determining which sense of an ambiguous word is invoked in a particular use of the word [Manning and Schütze, 1999]. As for sense repository, we adopted WordNet (version 1.7.1), a large lexical database for English developed and maintained at Princeton University since 1985, which is freely available online[2] and has been extensively used in NLP research [Stevenson, 2003]. WordNet was designed to establish connections between four types of Parts of Speech (POS): *Noun*, *verb*, *adjective*, and *adverb*. The basic building block for WordNet is the *synset* (SYNonym SET), which represents a specific meaning of a word. The specific meaning of one word under one type of POS is called a *sense*. Synsets are equivalent to senses, which are structures containing sets of words with synonymous meanings (words that are interchangeable in some contexts). Each synset has a *gloss* that defines the concept it represents. For example, the words *'night'*, *'nighttime'* and *'dark'* constitute a single synset that has the following gloss: "the time after sunset and before sunrise while it is dark outside". Synsets are connected through a series of relations: *antonymy* (opposites), *hyponymy/hypernymy* (is-a), *meronymy* (part-of), etc. The hyponymy relation serves to organize the lexicon into a hierarchical structure. There are separate hierarchies for nouns, verbs, adjectives and adverbs. We addressed the WSD problem by proposing an algorithm based on semantic similarity between synsets, computed by exploiting the hierarchical structure of WordNet defined by the hyponymy relation. The WSD procedure is fundamental to obtain a synset-based vector space representation that we called bag-of-synsets (BOS). In this model a synset vector, rather than a word vector, corresponds to a document. Another key feature of the approach is that each document is represented by a set of slots. Each slot is a textual field corresponding to a specific feature of the document, in an attempt to take into account also the structure of documents. For example, in our application scenario, in which documents are movie descriptions, we selected five slots to represent movies:

1. *title* - title of the movie;
2. *cast* - list of 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 parts of the story;
5. *keywords* - a list of words describing the main topics of the movie.

The text in each slot is represented in the BOS model 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 bags of synsets, one for each slot. Each bag of synsets d_n^m is defined as follows:

$$d_n^m = \left\langle t_{n1}^m, t_{n2}^m, \ldots, t_{nD_{nm}}^m \right\rangle \quad m = 1, \ldots 5$$

where t_{nk}^m is the k-th synset in the slot of the 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, m, $t_{nk}^m \in V_m$ where V_m is the vocabulary for the slot s_m (the set of all different synsets found in slot s_m). Document d_n is finally represented in the vector space by five synset-frequency vectors. Each synset-frequency vector f_n^m is defined as follows:

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

[2] WordNet: A Lexical Database for the English Language, http://wordnet.princeton.edu/wn1.7.1.shtml Accessed on December 6, 2006.

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

2.2 Word Sense Disambiguation

Each document to be disambiguated is processed by *META (Multi Language Text Analyzer*), a natural language processing tool able to index textual documents in English or Italian by applying different operations: Language recognition, tokenization, stopword elimination, stemming, part-of-speech tagging, word sense disambiguation. In our experiments, documents to be disambiguated are first processed by two basic phases: 1) part-of-speech tagging; 2) synset identification through WSD. For the part-of-speech tagging, we use the POS component for English included in META. It is based on the *t3* algorithm (a trigram tagger based on Markov models) included in ACOPOST[3], a collection of part-of-speech tagging algorithms, each originating from a different Machine Learning paradigm. In the first phase the text is first tokenized, then for each word the possible lemmas as well as their morpho-syntactic features are collected. Finally, part of speech ambiguities are solved. This is the input for the synset identification phase, which is mainly based on the WSD algorithm depicted in Figure 1. The idea behind the algorithm is that semantic similarity between synsets *a* and *b* is inversely proportional to the distance between them in the WordNet IS-A hierarchy [Leacock and Chodorow, 1998], measured by the number of nodes in the path from a to b. The path length similarity, computed by the function *SinSim* in Figure 1, is used to associate the proper synset to a polysemous word *w*. Let *S* denote the set of all candidate synsets for *w* and C the context of *w*, that is the window of all words that surround w with a fixed radius. The proposed WSD algorithm first builds *T*, the set of all synsets of the word forms in *C* with the same part-of-speech as *w*, and then computes the semantic similarity $score_{ih}$ between each synset s_i in *S* and each synset s_h in *T*. The synset s associated to *w* is the s_i with the highest similarity $score_{ih}$. For example, let us consider the sentence "*The white cat is hunting the mouse*". Let *w* be the word "*cat*" to be disambiguated. First, the algorithm selects the words with the same POS as *w*: in this case the only noun in the sentence, that is "*mouse*". Next, the two sets of synsets, *S* and *T*, corresponding respectively to the words "cat" and "mouse", are built. `S = {01789046: "feline mammal", 00683044: "computerized axial tomography",…}`, and `T = {01993048: "small rodents", 03304722: "a hand-operated electronic device that controls the coordinates of a cursor",…}`. Then, for each pair of synsets (*s*,*t*) in *S X T*, *SinSim*(*s*,*t*) is computed. In this case, *SinSim*(*01789046*, *01993048*) = 0.727 is the highest similarity score, thus *w* is interpreted as "*feline mammal*".

Each document is mapped into a list of WordNet synsets following three steps:

1. Each monosemous word ***w*** in a slot of a document *d* is mapped into the corresponding WordNet synset;
2. For each pair of words <*noun*,*noun*> or <*adjective*,*noun*>, a search in WordNet is made to verify if at least one synset exists for the bigram <w_1;w_2>. In the positive case, the WSD algorithm is applied to the bigram, otherwise it is applied separately to w_1 and w_2, using all words in the slot as the context *C* of *w*;
3. Each polysemous unigram w is disambiguated using all words in the slot as the context *C* of *w*.

As an example, Figure 3 shows a fragment of the BOS representation for the document presented in Figure 2. For readability reasons, we show the natural language description of the synsets provided by WordNet, in addition to the synset unique identifier used in the actual implementation and the number of occurrences of the synset.

[3] ACOPOST: A Collection Of POS Taggers, http://acopost.sourceforge.net/ Accessed on December 6, 2006.

Algorithm 1 The WordNet-based WSD algorithm

```
1: procedure WSD(w, d) ▷ finds the appropriate synset of a polysemous word w in
       document d; w may be also a bigram
2:     C ← {w_1, ..., w_n}                                              ▷ C is
           the context of w and it is defined as the window of all words that surround
           w with a fixed radius. For example, C = {w_1, w_2, w_3, w_4} is a window with
           radius=2 if the sequence of words {w_1, w_2, w, w_3, w_4} appears in d
3:     S ← {s_1, ...s_k}      ▷ S is sense inventory for w, that is the set of all candidate
           synsets for w returned by WordNet
4:     s ← null                                                  ▷ s is the synset to be returned
5:     score ← 0     ▷ score is the similarity score assigned to s wrt to the context C
6:     T ← ∅                  ▷ T is the set of all candidate synsets for all words in C
7:     for all w_j ∈ C do
8:         if POS(w_j) = POS(w) then                    ▷ POS(x) is the part-of-speech of x
9:             S_j ← {s_j1, ...s_jm}                  ▷ S_j is the set of m possible senses for w_j
10:            T ← T ∪ S_j
11:        end if
12:    end for
13:    for all s_i ∈ S do
14:        for all s_h ∈ T do
15:            score_ih ← SINSIM(s_i, s_h)     ▷ computing similarity scores between s_i
                   and every synset s_h ∈ T
16:            if score_ih ≥ score then
17:                score ← score_ih
18:                s ← s_i     ▷ s is the synset s_i ∈ S with the highest similarity score
                       wrt the synsets in T
19:            end if
20:        end for
21:    end for
22:    return s
23: end procedure
24: function SINSIM(a, b)                          ▷ The similarity of the synsets a and b
25:    N_p ← the number of nodes in path p from a to b
26:    D ← maximum depth of the taxonomy                    ▷ D = 16 in WordNet 1.7.1
27:    r ← −log(N_p/2 · D)
28:    return r
29: end function
```

Figure 1. The WordNet-based WSD algorithm

Title: 2001: A Space Odyssey

Cast: Keir Dullea, Gary Lockwood, William Sylvester, Daniel Richter, Leonard Rossiter, Margaret Tyzack, Robert Beatty, Sean Sullivan, Douglas Rain, Frank Miller, Bill Weston, Ed Bishop, Glenn Beck, Alan Gifford, Ann Gillis

Director: Stanley Kubrick

Keywords: astronaut, artificial-intelligence, nasa, cryogenics, space, …

Summary: This movie is concerned with intelligence as the division between animal and human, then asks a question; what is the next division? Technology is treated as irrelevant to the quest - literally serving as mere vehicles for the human crew, and as a shell for the immature HAL entity. Story told as a montage of impressions, music and impressive and careful attention to subliminal detail. A very influential film and still a class act, even after 25 years…

Figure 2. The movie "2001: A Space Odyssey" represented using the bag-of-words (BOW) model

Title: {00023929 space — (the unlimited expanse in which everything is located): 1, 00294497 odyssey — (a long wandering and eventful journey):1**}**

Cast: {}

Director: {10389739 Kubrick, Stanley Kubrick — (United States filmmaker born in 1928):1**}**

Keywords: {09192375 astronaut, spaceman, cosmonaut — (a person trained to travel in a spacecraft):1, 05766061 artificial intelligence, AI — (the branch of computer science that deal with writing computer programs that can solve problems creatively):1, 07635097 National Aeronautics and Space Administration, NASA — (an independent agency of the United States government responsible for aviation and spaceflight):1, 05734401 cryogenics, cryogeny — (the branch of physics that studies the phenomena that occur at very low temperatures):1, 00023929 space — (the unlimited expanse in which everything is located): 1, **…}**

Summary: {06205452 movie, film, picture, moving picture, moving-picture show, motion picture, motion-picture show, picture show, pic, flick — (a form of entertainment that enacts a story by a sequence of images giving the illusion of continuous movement):2, 05296893 intelligence — (the ability to comprehend; to understand and profit from experience):1, 07715643 division — (an army unit large enough to sustain combat):2, 00012748 animal, animate being, beast, brute, creature, fauna — (a living organism characterized by voluntary movement):1, 00006026 person, individual, someone, somebody, mortal, human, soul — (a human being):2, 00892915 technology, engineering — (the practical application of science to commerce or industry):1, **…}**

Figure 3. The movie "2001: A Space Odyssey" represented using the bag-of-synsets (BOS) model

Our hypothesis is that the proposed indexing procedure helps to obtain profiles able to catch documents semantically closer to the user interests. The difference with respect to keyword-based profiles is that synset unique identifiers are used instead of words. A more recent and improved version of the WSD procedure is described in [Semeraro et al., 2007].

In the next section, we will describe how a naïve Bayes learning algorithm [Mitchell, 1997] has been adapted for the task of acquiring user profiles. A system prototype named *ITem Recommender (ITR)* has been developed in order to implement the proposed approach. The final goal of our investigation is to evaluate the effectiveness of the above mentioned method in learning intelligible profiles of user interests from documents represented by WordNet synsets.

3. A Naïve Bayes Method for User Profiling

We adopt a naïve Bayes text categorization algorithm to build user profiles as binary classifiers (user-likes vs. user-dislikes). This strategy is implemented by our ITem Recommender (ITR) system [Degemmis et al., 2007]. 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:

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

where $N(t_k, d_i)$ is the number of times token t_k occurs in document d_i.

Since, for any given document, $P(d_i)$ is a constant with respect to c_j, this factor can be ignored in calculating Eq. (1), because all we need is to find the hypothesis with the highest posterior probability - *maximum a posteriori hypothesis* - rather than a probability estimate.

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:

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

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 labeled with a discrete rating, from 1 to *MAX*, provided by a user according to her degree of interest in the item. Following an idea proposed in [Mooney and Roy, 2000], each training document d_i is labeled with two scores, a "user-likes" score and a "user-dislikes" score , obtained from the original rating r_i:

$$\omega_i^+ = \frac{r_i - 1}{MAX - 1} \qquad \omega_i^- = 1 - \omega_i^+ \qquad (3)$$

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:

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

Witten-Bell smoothing [1991] is adopted to estimate $P(t_k \mid c_j, s_m)$, by taking into account that documents are structured into slots and that token occurrences are weighted using scores in equation (3):

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

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

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

In Eq. (6), n_{kim} is the count of occurrences of the token t_k in the slot s_m of the 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 the class c_j. In other words, $\hat{P}(t_k|c_j,s_m)$ is estimated as a ratio between the weighted occurrences of token 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).

In our approach, we do not use directly the classification scores computed by Eq. (2) to select documents to be recommended. In fact, in Section 5 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. Figure 4 shows a fragment of the profile of user #1.

In a movie retrieval scenario, documents are usually grouped by genre (e.g. *Comedy*). ITR learns a profile of the movies preferred by a user in a specific category or genre *G*. Therefore, given a user *u* and a set of rated movies in a specific genre or category, the system learns a profile able to recognize movies liked by *u* in that category. In the proposed example, the profile is related to genre *romance*. *Class priors* P(YES), corresponding to $P(c_+)$, and P(NO), corresponding to $P(c_-)$, are reported on the top of the profile. For each slot, the user profile contains a list of WordNet synsets (represented by their identifiers in the lexical database) ranked according to a strength measure computed by using the conditional probabilities in Eq. (5) as follows:

$$strength(t_k, s_m) = \log \frac{P(t_k \mid c_+, s_m)}{P(t_k \mid c_-, s_m)} \qquad (7)$$

The strength of token t_k in slot s_m represents the informative power of t_k (occurring in slot s_m) for classifying a new document.

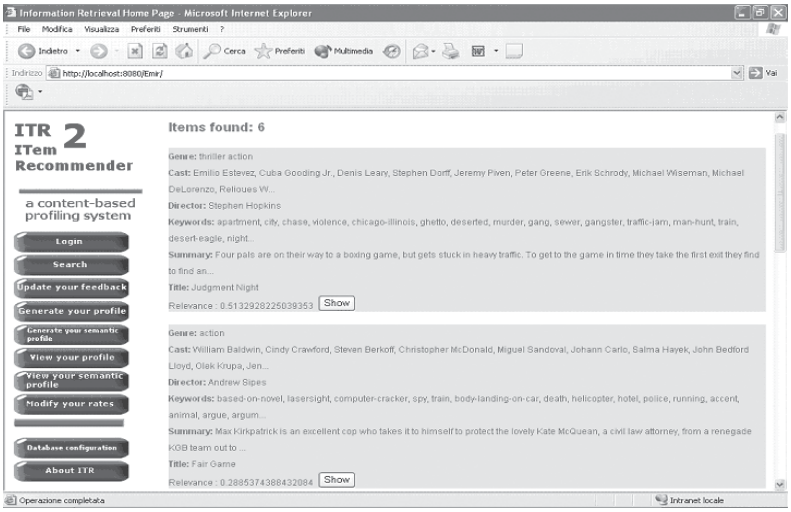

Figure 4. A fragment of the profile of user #1 for genre romance

3.1. Experimental Evaluation

The goal of the experiments was to compare the performance of synset-based user profiles to that of keyword-based profiles. This is a fundamental preliminary step before designing a retrieval model based on profiles containing synsets. The experiment was carried out on a content-based extension of the EachMovie dataset[4], a collection of 1,628 textual descriptions of movies rated on a 6-point scale (1-6) by 72,916 real users. The original EachMovie dataset does not contain any information about the content of the movies. The content information for each movie was collected from the Internet Movie Database (IMDb) using a simple crawler that, following the IMDb link provided in the original dataset, collects information from the various links of the main URL. In particular the crawler gathers the *title*, the *director*, the *genre* (the category of the movie), the list of *keywords*, the *summary* and the *cast*.

Movies are subdivided into 10 genres (categories): *Action*, *Animation*, *Art_Foreign*, *Classic*, *Comedy*, *Drama*, *Family*, *Horror*, *Romance*, and *Thriller*. For each genre, a set of 100 users has been randomly selected among users that rated *n* items, $30 \leq n \leq 100$, in that movie category (only for 'Animation', the number of users that rated *n* movies was 33, due to the low number of movies - 43 - in that genre, as reported in Table 1). In this way, for each category, a dataset of at least 3000 triples (user, movie, rating) was obtained (at least 990 for 'Animation'). Table 1 summarizes the data used for the experiments.

The main aim of our study was to point out the advantages and limitations of adopting a *pure* lexical knowledge approach in the process of learning user profiles. This is the main reason why we do not use any other external domain knowledge source in the process of shifting from words to concepts, even if the proposed WSD procedure suffers from the problem of unsuccessful recognition of domain entities, since names of most directors and actors do not occur in WordNet.

Tokenization, stopword elimination and stemming have been applied to index the documents according to the BOW model. The content of slots *title*, *director* and *cast* was only tokenized because the elimination of the stopwords produced some unexpected results. For example, slots containing exclusively stopwords, such as "*It*" or "*E.T.*", became empty. Moreover, it does not make sense to apply stemming and stopword elimination to proper names. Preprocessing operations are listed in Table 2.

[4] EachMovie dataset no longer available for download, see the GroupLens home page for a new version named MovieLens, originally based on this dataset: http://www.cs.umn.edu/Research/GroupLens/ Accessed on December 6, 2006.

Genre Id	Genre	Total number of movies in the original EachMovie dataset	Number of ratings	% POS	% NEG
1	Action	198	4,474	72	28
2	Animation	43	1,103	57	43
3	Art Foreign	299	4,246	76	24
4	Classic	201	5,026	92	8
5	Comedy	400	4,714	63	37
6	Drama	536	4,880	76	24
7	Family	145	3,808	64	36
8	Horror	87	3,631	60	40
9	Romance	137	3,707	73	27
10	Thriller	177	3,709	72	28
			39,298	72	28

Table 1. 10 'Genre' datasets obtained from the original EachMovie dataset

Slot	Tokenization	Stopword elimination	Stemming
Title	X		
Cast	X		
Director	X		
Summary	X	X	X
Keywords	X	X	X

Table 2. Preprocessing operations performed on the EachMovie dataset

Documents have been processed by the algorithm in Figure 1 and indexed according to the BOS model, obtaining a 38% feature reduction. This is mainly due to the fact that synonym words are represented by the same synset. Keyword-based profiles were learned from BOW-represented documents, while synset-based profiles were inferred from BOS-represented documents.

As ITR is conceived as a text classifier, its effectiveness can be evaluated by the well-known classification accuracy measures *precision* and *recall* [Sebastiani, 2002]. *F-measure*, a combination of precision and recall, has also been used. In addition, we adopted the *Normalized Distance-based Performance Measure (NDPM)* [Yao, 1995] to evaluate the distance between the ranking imposed on documents by the user ratings and the ranking predicted by ITR, which ranks documents according to the a-posteriori probability of the class c_+. Values range from 0 (agreement) to 1 (disagreement). The adoption of both classification accuracy and rank accuracy metrics gives us the possibility of evaluating both whether profiles are able to select relevant documents and how these documents are ranked. In all the experiments, a movie description d_j is considered relevant by a user if her rating *r* is greater than *(MAX+1)/2*, while ITR considers an item as relevant if the a-posteriori probability of the class c_+ is greater than *0.5*. We executed one run of the experiment for each user in the dataset.

Each run consisted in:

1. Selecting the documents and the corresponding ratings given by the user;
2. Splitting the selected data into a training set Tr and a test set Ts;
3. Using Tr for learning the corresponding user profile;
4. Evaluating the predictive accuracy of the learned profile on Ts, using the aforementioned measures.

The methodology adopted for obtaining *Tr* and *Ts* was the 5-fold cross-validation [Kohavi, 1995].

3.2 Discussion of results

Results of the comparison between BOS-generated profiles and BOW-generated profiles are reported in Table 3.

We can notice a significant improvement both in precision (+8%, from 0.67 to 0.75) and recall (+10%, from 0.78 to 0.88). Specifically, the BOS model outperforms the BOW one on datasets 5 (+11% of precision, +14% of recall), 7 (+15% of precision, +16% of recall), and 8 (+19% of precision, +24% of recall). Only on dataset 4 we did not observe any improvement, probably because precision and recall are already very high thus there is not much room for improvement. NDPM has not been improved, but it remains acceptable.

A Wilcoxon signed ranked test, requiring a significance level $p < 0.05$, has been performed in order to validate these results. We considered each dataset as a single trial for the test. The test confirmed that there is a statistically significant difference in favor of the BOS model with respect to the BOW model as regards precision, recall and F-measure, and that the two models are equivalent in defining the ranking of the preferred movies with respect to the score for the class c_+.

The main outcome of the experiments is that profiles obtained using the BOS model have a better classification accuracy compared to BOW-generated ones, but both kinds of profiles have the same "ranking power", if we use as ranking score the classification score for the class c_+.

This result might appear contradictory, but a deeper analysis of the classification scores and corresponding positions of documents in the ranking revealed that contradiction is only apparent: The BOS-generated profiles improved classification for those items for which classification score is close to the relevant/not relevant threshold, thus changes in the ranking were so slight that they did not affect NDPM values. A more detailed analysis of this result is given in [Semeraro et al., 2007].

Anyway, NDPM values indicate that the adoption of the classification score exclusively for the class c_+ to rank documents does not produce positive effects. Thus, a more advanced function should be studied to improve ranking effectiveness, which is the main aim of the next section.

4. Intelligent Personalized Searching

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

Genre Id	Precision		Recall		F-measure		NDPM	
	BOW	BOS	BOW	BOS	BOW	BOS	BOW	BOS
1	0.70	0.74	0.83	0.89	0.76	0.80	0.45	0.45
2	0.51	0.57	0.62	0.70	0.54	0.61	0.41	0.39
3	0.76	0.86	0.84	0.96	0.79	0.91	0.45	0.45
4	0.92	0.93	0.99	0.99	0.96	0.96	0.48	0.48
5	0.56	0.67	0.66	0.80	0.59	0.72	0.46	0.46
6	0.75	0.78	0.89	0.92	0.81	0.84	0.46	0.45
7	0.58	0.73	0.67	0.83	0.71	0.79	0.42	0.42
8	0.53	0.72	0.65	0.89	0.58	0.79	0.41	0.43
9	0.70	0.77	0.83	0.91	0.75	0.83	0.49	0.49
10	0.71	0.75	0.86	0.91	0.77	0.81	0.48	0.48
Mean	0.67	0.75	0.78	0.88	0.73	0.81	0.45	0.45

Table 3. Performance of ITR on 10 different datasets using BOW and BOS model

4.1 The Synset Similarity Model

According to [Baeza-Yates and Ribeiro-Neto, 1999], an *information retrieval* model is a 4-tuple:

$$< D, Q, F, R(q_i,d_j) >$$

where:

- *D* is a set composed of logical views (or representations) for the *documents* in the collection;
- *Q* is a set composed of logical views (or representations) for user information needs. Such representations are called *queries*;
- *F* is a *framework* for modeling document representations, queries, and their relationships;
- $R(q_i,d_j)$ is a *ranking function* which associates a real number with a query $q_i \in Q$ and a document representation $d_j \in D$. Such a ranking defines an ordering among the documents with respect to the query q_i.

The classic Vector Space Model [Salton and McGill, 1983] applied to synsets rather than words has good performances: Results reported in [Gonzalo et al., 1998] showed the advantages of using synsets instead of words. In that work, 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 concept expressed using WordNet synsets. Then, they adopted the Vector Space Model applied to WordNet synsets. The realization of the *semantic tf-idf model* was rather simple, because it was sufficient to index the documents or the user-query by using strings representing synsets. The retrieval phase is similar to the classic tf-idf model [Salton and McGill, 1983], with the only difference that matching is carried out between synsets. However, this approach has some limits due to the shifting from the lexical to the semantic space. The semantic tf-idf needs some improvements: The matching between terms with the same 'stem' in the lexical space is able to group all the terms in the document, by guessing that all these terms are related to the same concept. In the semantic space, each string in the document becomes a synset, represented by the 'offset' in the WordNet hierarchy. The matching is carried out not only between terms with the same stem, but also between different terms that can be referred to the same concept (synset). Nevertheless, an exact matching between synsets is not enough: If the user submits a query by searching the synset associated to the concept of "carnivore", the result set would consist solely of documents that contain the concept "carnivore". All the documents related to subsumed concepts, such as "dogs", would not be retrieved. The synsets "dog" and "carnivore" are different, but they have a semantic correlation that should be considered by the semantic tf-idf model. Thus, it is necessary to extend the model by introducing semantic similarity measures able to redefine the similarity between a document and a user query [Corley and Mihalcea, 2005; Jiang and Conrath, 1997; Resnik, 1995 and 1999]. The semantic similarity between concepts is useful to understand how similar are the meanings of the concepts, in other words to what extent the WordNet subtree rooted in a concept is similar to the corresponding subtree rooted in the other concept. 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 user 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 described 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.

In [Scott and Matwin, 1998] a method to improve the matching between a query and a document is described. It includes not only the concepts contained in both the documents, but also similar concepts.

According to this approach, the document indexing phase is carried out by including not only the synsets related to the concepts in the text, but also the synsets related to the concepts that subsume them.

The final text representation is a list of the synsets related to the concepts recognized in the text and also their hypernyms. This results in an increased similarity of documents containing similar synsets, even if they are not identical. The idea proposed in that approach is introduced in the SSM by adopting the similarity measure between synsets used in the WSD algorithm (*SinSim* function in Figure 1), that bases the computation on the *most specific subsumer (mss)* of the synsets to be compared. The *mss* is found by climbing the WordNet IS-A synset hierarchy. This procedure is carried out for each pair of synsets for which the degree of similarity must be computed. Therefore, the relevance of a document *d* with respect to a query *q* is computed by the formula:

$$R(q,d) = \frac{\sum_{i=1}^{N} \max_j [SinSim(q_i, s_j)]}{N} \qquad (8)$$

where q_i is the *i*-th synset in *q*, s_j is the *j*-th synset in *d*, and *N* is the number of synsets in *q*. Notice that Equation (8) does not take into account the importance of s_j in *d*; on the contrary, the semantic tf-idf [Gonzalo et al., 1998] model was mainly based on this crucial aspect. For this reason, we decided to take into account the importance of the synsets in the document when computing the semantic similarity between two synsets. This was realized by multiplying the semantic similarity between the pair of synsets $<q_i,s_j>$ by the tf-idf weight of s_j in *d*. Therefore, the new semantic similarity measure is described by the following formula:

$$SinSim_{tf\text{-}idf}(q_i,s_j) = Tf\text{-}Idf(s_j,d) * SinSim(q_i,s_j) \qquad (9)$$

where $Tf\text{-}Idf(s_j,d)$ is the *tf-idf* weight of the synset s_j in the document *d*. The SSM was implemented to develop a movie retrieval system. Figure 5 shows a scenario in which a user submits the query "dark" to the system. The retrieval function analyzes the query and suggests a list of possible senses that the user can assign to the keywords in the query (we are planning to improve this manual disambiguation procedure). Figure 6 shows the result set obtained in response to the query when using only keywords and the classic Vector Space Model: The movie occurring in the first position in the ranking is "Under Siege 2: Dark"; Figure 7 shows the movie in the first position in the ranking obtained when assigning the sense "*the time after sunset and before sunrise while it is dark outside*," to the keyword "dark". Notice that the number of movies in the result set obtained in this case is reduced due to the different retrieval model adopted (6 items versus 40 items) and the movie on the top of the list is "Judgment Night".

4.2 Personalized Synset Similarity Model

This section proposes a possible strategy to extend an information retrieval model, as defined in the previous section, to a 5-tuple:

$$< D , Q , F , P_u, R(q_i,d_j,P_u) >$$

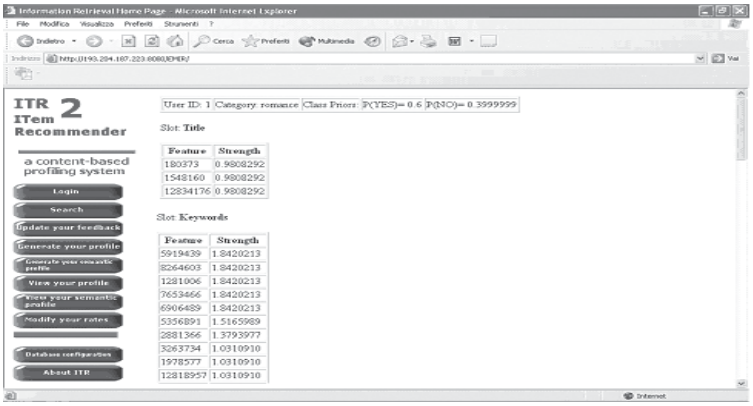

Figure 5. A list of possible senses for the keyword 'dark' in the query

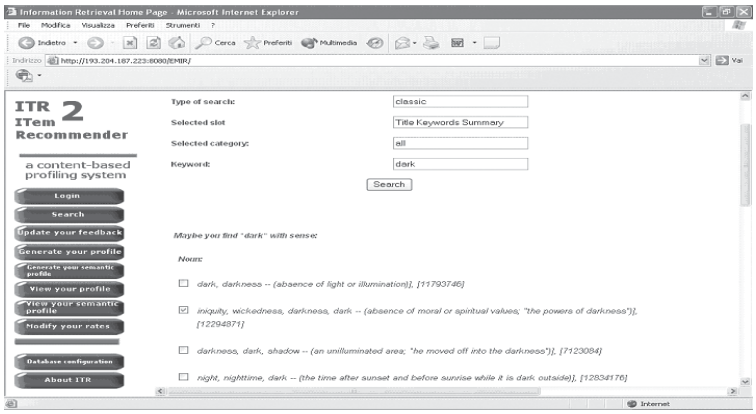

Figure 6. Result set obtained in response to a query using keywords and the Vector Space Model

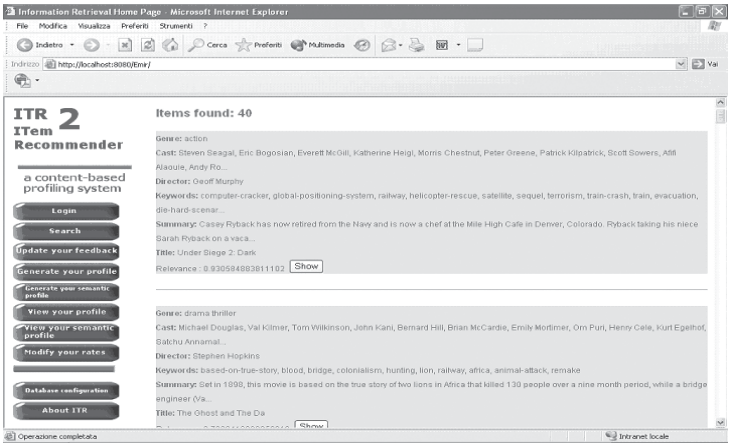

Figure 7. Result set obtained assigning a specific sense to the keywords in the query

where:

- *D, Q, F* are exactly the same as in the model defined in Section 4.1;
- P_u is the user profile, representing the long-term preferences of user *u*;
- $R(q_i,d_j,P_u)$ is a ranking function which associates a real number with a query $q_i \in Q$, a document representation $d_j \in D$, and the profile of user *u*. Such a ranking defines a personalized ordering among documents with respect to the information needs of user *u*, expressed both by q_i and P_u.

The idea is to introduce the information about long-term interests of a user in the retrieval process: The query by itself, which represents the user's short-term interests, in some application domains is not suitable to represent the actual user information needs. For example, a user who wishes to retrieve "interesting horror movies" cannot easily express this form of information need as a query suitable for search engines. In the following, we describe a possible strategy to personalize the search process. The goal is to improve the ranking proposed by the Synset Similarity Model by integrating information contained in the user profile. In Section 3, we described the classification model behind the user profile and how it is possible to use that model to obtain the *a posteriori* probability of liking/disliking an item. The extended retrieval model introduced in this section,

called *Personalized Synset Similarity Model (PSSM)*, proposes a ranking function $R(q_i,d_j,P_u)$ that, starting from the ranking computed by $R(q_i,d_j)$, defines a new ranking which also takes into account the classification score for each item computed by the user profile P_u. The goal is to increase the ranking of the items that are more interesting for a specific user; the core of the proposed model is the definition of a new formula that integrates in a single score both $R(q_i,d_j)$ and the *a posteriori* probability of the class c_+ given by the Bayesian classifier. Let w_k be the weight 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:

- The impact of the probability p_k on the final weight (starting from w_k) should be non-linear; in fact, if the probability of interest in an item is close to 50%, the weight 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 the limits (0%, 100%), w_k should be strongly modified;
- It is reasonable that the update of the initial weight w_k is proportional to the value of the weight itself. It is useful to introduce in the re-ranking function a term that synthesizes both the feedback of the user profile and the weight w_k.

The first heuristic is realized through the function $f(p)$:

$$f:[0,1] \rightarrow [-0.5, 0.5]$$

f is defined in the interval [0, 1] that represents the range of the probability of interest in an item d_j, $P(c_+|d_j)$. f transforms this (always) positive argument in a positive value if $P(c_+|d_j)>0.5$, and in a negative value otherwise. The function is negative in [0, 0.5[, and positive in [0.5, 1]. It is built by locating two parabolas with a certain concavity and crossing specific points (for example $f(0.5) = 0$):

$$f(p) = \begin{cases} -\frac{5}{2}p^2 + \frac{19}{4}p - \frac{7}{4} & \text{if} \quad p \geq 0.5 \\ \frac{5}{2}p^2 - \frac{1}{4}p - \frac{1}{2} & \text{if} \quad p < 0.5 \end{cases} \qquad (10)$$

As depicted in Figure 8, f is a growing function and its values in the interval [0.4, 0.6] indicate an absence of precise preferences. Out of this interval 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 maximum and minimum.

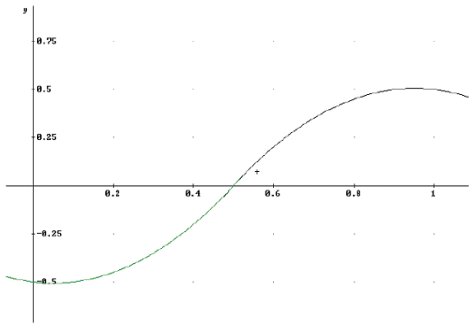

Figure 8. The graph of the function described by Equation (10)

The second heuristic is realized by adding a further term to w_k, which relates the probability $P(c_+|d_j)$ with the weight w_k. 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. We denote with $g()$ the function which computes this new value useful for re-classifying an item:

$$g(w_k, p_k) = w_k \cdot (p_k - 0.5) \qquad (11)$$

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	**9**
3	0.85	0.32	-0.325	-0.154	0.373	**10**
4	0.82	0.91	0.501	0.331	1.649	**2**
5	0.81	0.04	-0.506	-0.375	-0.072	**18**
6	0.77	0.29	-0.359	-0.159	0.248	**11**
7	0.64	0.78	0.432	0.177	1.247	**5**
8	0.63	0.87	0.490	0.233	1.354	**3**
9	0.60	0.23	-0.424	-0.161	0.015	**15**
10	0.55	0.94	0.506	0.238	1.292	**4**
11	0.36	0.36	-0.263	-0.050	0.049	**14**
12	0.36	0.32	-0.320	-0.063	-0.026	**16**
13	0.33	0.42	-0.156	-0.025	0.146	**12**
14	0.30	0.91	0.502	0.122	0.921	**6**
15	0.20	0.68	0.327	0.037	0.569	**8**
16	0.20	0.37	-0.248	-0.026	-0.071	**17**
17	0.17	0.80	0.453	0.052	0.675	**7**
18	0.16	0.14	-0.487	-0.058	-0.385	**19**
19	0.15	0.05	-0.506	-0.068	-0.424	**21**
20	0.12	0.13	-0.492	-0.045	-0.416	**20**
21	0.02	0.54	0.078	0.001	0.099	**13**

Table 4. Ranking of items in the result set of a query according to the ranking function of SSM and PSSM

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

$$R(q_i,d_j,P_u)=R(q_i,d_j) + f(P(c_+|d_j)) + g(R(q_i,d_j), P(c_+|d_j)) \quad (12)$$

Table 4 shows an example of ranking 21 items obtained by a user submitting the query "love comedy" to the movie dataset. The user previously rated a set of 30 items and the corresponding profile was learned by the ITR system. Items are ordered in a descending order on the ground of the function $R(q_i,d_j)$. The first and the second column report the position in the SSM and the value $R(q_i,d_j)$ respectively (in this case the profile was not used for computing the ranking). 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 to the way in which the function (12) was defined. It is interesting to observe items at positions 10, 14, 15, 17, 21 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 probability of interest greater than 0.5, they are respectively at positions 4, 6, 8, 7, 13 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 9, 10, 18, 11, because they are not too relevant with respect to the user profile.

5. Experiments on Personalized Synset Similarity Model

The main aim of the experimental session described in this section is to compare the effectiveness of the proposed Personalized Synset Similarity Model (PSSM) with that of the Synset Similarity Model (SSM).

Experiments were performed on the dataset described in Section 3.1 and were conducted by 8 real users in two phases. In the first phase, 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 the ITR system. A synset-based profile was generated for each genre for which training examples were available. For example, if the user rated 15 *horror* movies and *10 action* movies, two profiles were generated, one for each genre.

In the second phase, each user was requested to issue 3 different queries, according to her information needs, submitted to both the SSM and the PSSM search engines. 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 are available: *SSM ranking* with the corresponding ideal ranking set by the user feedback, and *PSSM ranking* with the corresponding ideal ranking.

The movies rated in the first phase by the user are withheld in this phase, in order to prevent ranking from being affected by documents already used in the training step. Moreover, if a movie in the list of the retrieved results belongs to more than one genre, the profile used in the PSSM search engine is the one with the highest prior probability for the class c_+.

The performance measure adopted is the NDPM because our aim was to compare the ability of the two models in producing effective document ranking.

In more detail, 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 5 reports the results obtained on each query issued by the 8 users. Averaged NDPM values for the 3 queries are also reported.

We observed that for 18 out of the 24 queries (75%), PSSM outperforms SSM. Moreover, queries Q_1 and Q_2 issued by User 1 produce the same rankings, due to the fact that no user profile was exploited in the search because all the documents in the result lists belong to genres for which no training example was provided by the user (and consequently no profiles were generated for those genres). This result shows that when no user profile is available in the PSSM, the ranking of the SSM is preserved. Another 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

User Id	SSM				PSSM			
	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

Table 5. NDPM values for SSM and PSSM

ratings standing for "I like it, but not too much" or "I dislike it, I could even like it". Other users, like User 2, User 3, User 7 and User 8, had a very clean rating style, that is, they tended to assign the score 1 to not interesting documents, and the score 6 to interesting ones. Therefore, we can conclude that this negative result for the PSSM depends on the noise in the training set used as input to the ITR system.

Anyway, the main observation that can be drawn from Table 5 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.

6. Conclusions

In this paper, we described a strategy for personalization of Web search in domains where the role of user preferences strongly affects the acceptance of the results. As an example, we proposed a movie retrieval scenario.

The main elements on which the proposed strategy is based are:

- The bag-of-synsets (BOS) model adopted to represent documents by using WordNet synsets, rather than words, as in the classical bag-of-words approach;
- Synset-based profiles, induced from documents represented according to the BOS model by a word sense disambiguation procedure which maps words into synsets;
- The Synset Similarity Model (SSM), a semantic retrieval model based on WordNet synsets, in which the similarity between a document, represented through the BOS model, and a query is computed according to a synset similarity function;
- The Personalized Synset Similarity Model (PSSM), which extends the SSM by including synset-based user profiles in the computation of query-document similarity. Profiles are used to re-rank search results by moving documents relevant for a user to the top of the result list in order to produce a personalized ranking and to improve retrieval effectiveness.

Experimental results indicate that PSSM retrieval effectiveness is higher than 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 individual preferences.

As a future work, domain-dependent knowledge sources, such as domain ontologies, will be integrated into the synset-based linguistic approach in order to obtain a more powerful retrieval model. A larger scale of experiments is also planned.

References

[1] Baeza-Yates, R., Ribeiro-Neto,B (1999). Modern Information Retrieval. New York: Addison-Wesley.

[2] Chen, L., Sycara, K (1998). WebMate: A Personal Agent for Browsing and Searching. *In: Proceedings of the Second International Conference on Autonomous Agents*, p. 132-139. ACM Press, May.

[3] Corley, C., Mihalcea, R (2005). Measures of Text Semantic Similarity. *In: Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence*, Ann Arbor, June.

[4] Degemmis, M., Lops, P., 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* 17 (3) 217-255.

[5] Glover, E. J., Flake, G. W., Lawrence, S, Birmingham, W. P, Kruger, A., Giles C. L., Pennock, D. M (2001). Improving Category Specific Web Search by Learning Query Modifications. *In: Proceedings of the 2001 Symposium on Applications and the internet (SAINT 2001)*, p. 23-34. *IEEE Computer Society*, Jan.

[6] Gonzalo, J., Verdejo, F., Chugur, I., Cigarran, J (1998). Indexing with WordNet synsets can Improve Text Retrieval. *In: Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP*, p. 38-44. Montreal, Canada, Aug.

[7] Jiang, J.J., Conrath, D.W (1997). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. *In: Proceedings of ROCLING 10th International Conference on Research in Computational Linguistics*, Taiwan, Aug.

[8] Kehagias, A., Petridis, V., Kaburlasos, V. G., Fragkou, P (2003). A Comparison of Word- and Sense-Based Text Categorization Using Several Classification Algorithms. *Journal of Intelligent Information Systems* 21 (3) 227-247.

[9] Kohavi, R (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *In: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI'95)*, p.1137-1145. Morgan Kaufmann, Aug. 1995.

[10] Leacock C., Chodorow , M (1998). Combining Local Context and WordNet Similarity for Word Sense Identification. *In: Christiane Fellbaum (ed.), WordNet: An Electronic Lexical Database.* Cambridge: MIT Press, 265–283.

[11] Lewis, D. D (1998). Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. *In: Proceedings of the 10th European Conference on Machine Learning (ECML'98)*, p. 4–15. Springer, Apr.

[12] Lewis, D. D., Ringuette, M (1994). A Comparison of Two Learning Algorithms for Text Categorization. *In: Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94)*, p. 81–93. Las Vegas.

[13] Liu, F., Yu, C., Meng, W (2004). Personalized Web Search For Improving Retrieval Effectiveness. *IEEE Transactions on Knowledge and Data Engineering* 16 (1) 28-40.

[14] Manning, C., Schütze, H (1999). Foundations of Statistical Natural Language Processing. Chapter 7: Word Sense Disambiguation. Cambridge: The MIT Press.

[15] Mihalcea, R., Moldovan, D (2000). Semantic Indexing using Wordnet Senses. *In: Proceedings of ACL Workshop on Recent Advances in NLP and IR*, p. 35-45. Hong Kong, Oct.

[16] Miller, G (1990). WordNet: An On-Line Lexical Database. *International Journal of Lexicography* 3 (4) Oxford University Press.

[17] Mitchell, T (1997). Machine Learning. New York: McGraw-Hill.

[18] Mladenic, D (1999). Text-Learning and Related Intelligent Agents: A Survey. *IEEE Intelligent Systems* 14 (4) 44-54.

[19] Mooney, R. J., Roy, L (2000). Content-Based Book Recommending Using Learning for Text Categorization. *In: Proceedings of the 5th ACM Conference on Digital Libraries*, p. 195-204. ACM Press, June.

[20] Moulinier, I., Ganascia, J. G (1996). Applying an Existing Machine Learning Algorithm to Text Categorization. *In: Stefan Wermter, Ellen Riloff, Gabriele Scheler (Eds.): Connectionist, Statistical, and Symbolic Approaches to Learning for Natural*

Language Processing. Lecture Notes in Computer Science 1040, Berlin: Springer, 343-354.

[21] Pazzani, M., Billsus, D (1997). Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning* 27 (3) 313–331.

[22] Resnik, P (1995). Using Information Content to Evaluate Semantic Similarity in a Taxonomy. *In: Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, p. 448-453. Morgan Kaufmann, Aug.

[23] Resnik, P (1999). Semantic Similarity in a Taxonomy: An Information-based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research* 11 95-130.

[24] Salton, G., McGill, M (1983). Introduction to Modern Information Retrieval. New York: McGraw-Hill.

[25] Scott, S., Matwin, S (1998). Text Classification Using WordNet Hypernyms. *In: Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP,* p. 45-51. Montreal, Canada, Aug.

[26] Sebastiani, F (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys* 34 (1) 1-47.

[27] Semeraro, G., Degemmis, M., Lops, P., Basile P (2007). Combining Learning and Word Sense Disambiguation for Intelligent User Profiling. *In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07)*, p. 2856-2861. Morgan Kaufmann, Jan.

[28] Sieg, A., Mobasher, B., Burke, R., Lytinen, S (2004). Using Concept Hierarchies to Enhance User Queries in Web-Based Information Retrieval. *In: Proceedings of the IASTED International Conference on Artificial Intelligence and Applications*, p. 226-234. Innsbruck, Feb.

[29] Smeaton, A. F., 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 Information Retrieval (SIGIR'96)*, p. 174-180. ACM Press, Aug.

[30] Stevenson, M (2003). Word Sense Disambiguation: The Case for Combinations of Knowledge Sources. Stanford: CSLI Publications.

[31] Witten, I., Bell, T (1991). The Zero-frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression. *IEEE Transactions on Information Theory* 37 (4) 1085-1094.

[32] Yang, Y., Pedersen, J. O (1997). A Comparative Study on Feature Selection in Text Categorization. *In*: *Proceedings of the 14th International Conference on Machine Learning (ICML'97)*, p. 412-420. Morgan Kaufmann, July.

[33] 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.

Giovanni Semeraro is associate professor at the Department of Informatics of the University of Bari, where he teaches in the courses of "Programming Languages", "Formal Languages and Compilers" and "Enterprise Knowledge Management". In 1993 he was visiting research assistant specialist at the Institute of Computer Science of the University of California, Irvine. His research activity mainly concerns machine learning, semantic web and personalization.

His research interests include logical and algebraic foundations of machine learning and semantic web for inductive reasoning, extraction of dynamic user profiles, web and usage mining, revision of logical theories and application of machine learning techniques to user modelling, digital libraries, electronic commerce, hybrid intelligent systems and intelligent user interfaces.

He is responsible of the research group of the University of Bari for the VI Framework Programme Integrated Project VIKEF (2004-07) and DELOS – A Network of Excellence on Digital Libraries (2004-07), for the Regional projects @sso_net (for the development of services for small and medium enterprises and community of craftsmen), JUMP and for the research contracts with APAT (Agency for the Environment Protection and technical services), Cézanne Software and Luxhora.

He was responsible of the research group of the University of Bari for the V Framework Programme project COGITO (2000-02) and DELOS (2000-2003) and for the research contracts with ENEA, Tecnopolis Csata and IBM. He was involved in several national and European projects.

He is author of more than 100 papers published in international journals, books and conference proceedings, and of a book on formal language theory. He served as co-chair of several international conferences and workshops. He is member of IEEE Computer Society, ACM, AI*IA, AICA, GRIN.