Combining Learning and Word Sense Disambiguation for Intelligent User Profiling*

Giovanni Semeraro, Marco Degemmis, Pasquale Lops, Pierpaolo Basile
Department of Informatics
University of Bari, Italy
{semeraro, degemmis, lops, basilepp}@di.uniba.it

Abstract

Understanding user interests from text documents can provide support to personalized information recommendation services. Typically, these services automatically infer the user profile, a structured model of the user interests, from documents that were already deemed relevant by the user. Traditional keyword-based approaches are unable to capture the semantics of the user interests. This work proposes the integration of linguistic knowledge in the process of learning semantic user profiles that capture concepts concerning user interests. The proposed strategy consists of two steps. The first one is based on a word sense disambiguation technique that exploits the lexical database WordNet 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 *sense-based* user profiles as binary text classifiers (user-likes and user-dislikes) from disambiguated documents. Experiments have been conducted to compare the performance obtained by keyword-based profiles to that obtained by sense-based profiles. Both the classification accuracy and the effectiveness of the ranking imposed by the two different kinds of profile on the documents to be recommended have been considered. The main outcome is that the classification accuracy is increased with no improvement on the ranking. The conclusion is that the integration of linguistic knowledge in the learning process improves the classification of those documents whose classification score is close to the likes / dislikes threshold (the items for which the classification is highly uncertain).

1 Introduction

Personalized systems adapt their behavior to individual users by learning their preferences during the interaction in order to construct a *user profile*, that can be later exploited in the search process. Traditional keyword-based approaches are primarily driven by a string-matching operation: If a string, or some morphological variant, is found in both the profile and the document, a match occurs and the document is considered relevant. String matching suffers from problems of *polysemy*, the presence of multiple meanings for one word, and *synonymy*, multiple words having the same meaning. The result is that, due to synonymy, relevant information can be missed if the profile does not contain the exact keywords in the documents, while, due to polysemy, wrong documents could be deemed relevant. These problems call for alternative methods to learn more accurate profiles that capture concepts expressing user interests from relevant documents. These *semantic* profiles should contain references to concepts defined in lexicons or ontologies. This paper describes an approach in which user profiles are obtained by machine learning techniques integrated with a word sense disambiguation (WSD) strategy based on the WordNet lexical database [Miller, 1990; Fellbaum, 1998]. The paper is organized as follows: After a brief discussion about the main works related to our research, in Section 3 the WSD strategy proposed to represent documents by using WordNet is described. Section 4 presents the naïve Bayes text categorization method we adopted to build *WordNet-based* user profiles. This method is implemented by the content-based profiling system ITem Recommender (ITR). An experimental sessions has been carried out in order to evaluate the proposed approach in a movie recommending scenario. The main results are presented in Section 5. Conclusions and future work are discussed in Section 6.

2 Related Work

Our research was mainly inspired by the following works. *Syskill & Webert* [Pazzani and Billsus, 1997] learns user profiles as Bayesian classifiers able to recommend web pages, but it represents documents using keywords. *LIBRA* [Mooney and Roy, 2000] adopts a Bayesian classifier to produce content-based book recommendations by exploiting product descriptions obtained from the web pages of the Amazon on-line digital store. Documents are represented by using keywords and are subdivided into slots, each one corresponding to a specific section of the document. Like *Syskill & Webert*, the main limitation of this work is that keywords are used to represent documents. Conversely, *SiteIF* [Magnini and Strapparava, 2001] exploits a *sense-based* document repre-

*This research was partially funded by the European Commission under the $6th$ Framework Programme IST Integrated Project VIKEF, Priority 2.3.1.7 Semantic-based Knowledge Systems.

sentation to build a user profile as a semantic network whose nodes represent senses of the words in documents requested by the user. The semantic network is built by assigning each node with a score that is inversely proportional to its frequency over all the corpus. Thus, the score is higher for less frequent senses, and this prevents very common meanings from becoming too prevailing in the user model. In our approach, a probability distribution of the senses, found in the corpus of the documents rated by the user, is learned. *OntoSeek* [Guarino *et al.*, 1999] is a system designed for content-based information retrieval from online yellow pages and product catalogs, which explored the role of linguistic ontologies in knowledge-retrieval systems. That approach has shown that structured content representations, coupled with linguistic ontologies, can increase both recall and precision of content-based retrieval systems. By taking into account the lessons learned by the aforementioned works, ITR has been conceived as a text classifier able: 1) To deal with a sense-based document representation obtained by exploiting a linguistic ontology; 2) To learn a Bayesian profile from documents subdivided into slots. The strategy we devise in order to shift from a keyword-based document representation to a sense-based one, is *to integrate lexical knowledge in the indexing step of training documents*. Several methods have been proposed to accomplish this task. Scott and Matwin [1998] included WordNet information at the feature level by expanding each word in the training set with *all* its synonyms in WordNet in order to avoid a WSD process. This approach has shown a decrease of effectiveness in the obtained classifier, mostly due to the word ambiguity problem. Therefore, it suggests that some kind of disambiguation is required. Bloedhorn and Hotho [2004] experiment with various settings for mapping words to senses: No WSD, most frequent sense as provided by WordNet, WSD based on context. They found positive results on the Reuters 25178[1], the OHSUMED[2] and the FAODOC[3] corpora. None of the previous approaches for embedding WSD in classification has taken into account the fact that WordNet is a hierarchical thesaurus. A distinctive feature of our work is the adoption of a similarity measure that takes into account the hierarchical structure of WordNet.

3 Using WordNet to Represent Documents

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. The set of categories is $C = \{c_+, c_-\}$, where c_+ is 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 WSD technique that exploits the word senses in WordNet to represent documents. In the second step, described in Section 4, a naïve Bayes approach learns *sense-based* user profiles as binary text classifiers (user-likes and user-dislikes) from disambiguated documents. A thorough experimental evaluation of that idea in the context of a hybrid (content-based / collaborative) recommender system has been carried out in [Degemmis *et al.*, 2007].

3.1 The JIGSAW Algorithm for Word Sense Disambiguation

Textual documents cannot be directly interpreted by learning algorithms. An indexing procedure that maps a document d_i into a compact representation of its content must be applied. A typical choice for document indexing is the classical *bag-of-words* (BOW) approach, where each document is represented as a feature vector counting the number of occurrences of different words as features [Sebastiani, 2002]. We extend the BOW model to a model in which each document is represented by the senses corresponding to the words in its content and their respective occurrences. This sense-based document representation is exploited by the learning algorithm to build semantic user profiles. Here, "sense" is used as a synonym of "meaning". Any implementation of sense-based document indexing 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. This task, known as word sense disambiguation, consists in determining which of the senses of an ambiguous word is invoked in a particular use of that word [Manning and Schütze, 1999].

The goal of a WSD algorithm is to associate a word w_i occurring in a document d with its appropriate meaning or sense s, by exploiting the *context* C in which w_i is found, commonly defined as a set of words that precede and follow w_i. The sense s is selected from a predefined set of possibilities, usually known as *sense inventory*. In the proposed algorithm, the sense inventory is obtained from WordNet (version 1.7.1). 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. Each synset has a gloss, a short textual description that defines the concept represented by the synset. 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. JIGSAW is a WSD algorithm based on the idea of combining three different strategies to disambiguate nouns, verbs, adjectives and adverbs. The motivation behind our approach is that the effectiveness of the WSD algorithms is strongly influenced by the POS tag of the target word. An adaptation of Lesk dictionary-based WSD algorithm has been used to disambiguate adjectives and adverbs [Banerjee and Pedersen, 2002], an adaptation of the Resnik algorithm has been used to

[1] http://about.reuters.com/researchandstandards/corpus/

[2] http://www.ltg.ed.ac.uk/disp/resources/

[3] http://www4.fao.org/faobib/index.html

disambiguate nouns [Resnik, 1995], while the algorithm we developed for disambiguating verbs exploits the nouns in the context of the verb as well as the nouns both in the glosses and in the phrases that WordNet utilizes to describe the usage of the verb. The algorithm disambiguates only words which belong to at least one synset. JIGSAW takes as input a document $d = \{w_1, w_2, \ldots, w_h\}$ and will output a list of WordNet synsets $X = \{s_1, s_2, \ldots, s_k\}$ $(k \leq h)$ in which each element s_i is obtained by disambiguating the *target word* w_i based on the information obtained from WordNet about a few immediately surrounding words. We define the *context* C of the target word to be a window of n words to the left and another n words to the right, for a total of $2n$ surrounding words. The algorithm is based on three different procedures for nouns, verbs, adverbs and adjectives, called $JIGSAW_{nouns}$, $JIGSAW_{verbs}$, $JIGSAW_{others}$, respectively. The POS tag of each word is computed by the HMM-based tagger ACOPOST t3[4]. JIGSAW proceeds in several iterations by using the disambiguation results of the previous iteration to reduce the complexity of the next one. First, JIGSAW performs the $JIGSAW_{nouns}$ procedure. Then, verbs are disambiguated by $JIGSAW_{verbs}$ by exploiting the words already disambiguated by $JIGSAW_{nouns}$. Finally, the $JIGSAW_{others}$ procedure is executed. More details for each one of the above mentioned procedures follow.

$JIGSAW_{nouns}$

The algorithm assigns to w_i the most appropriate synset s_{ih} among the sense inventory W_i for w_i. It computes the similarity between each s_{ik} in the sense inventory and the context for w_i. The method differs from the original algorithm by Resnik [1995] in the use of the similarity measure. We adopted the Leacock-Chodorow measure [1998], which is based on the length of the path between concepts in an IS-A hierarchy. The idea behind this measure is that similarity between synsets a and b is inversely proportional to their distance in the WordNet IS-A hierarchy. The distance is computed by counting the number of nodes in the shortest path joining a with b (by passing through their most specific subsumer). The similarity function is: $\texttt{SinSim}(a, b) = -\log(N_p/2D)$, where N_p is the number of nodes in the shortest path p from a to b, and D is the maximum depth of the taxonomy ($D = 16$, in WordNet 1.7.1). The procedure starts by defining the context C of w_i as the set of words having the same POS tag and found in the same sentence as w_i. Next, the algorithm identifies both the sense inventory for w_i and the sense inventory W_j, for each word w_j in C. The sense inventory T for the whole context C is given by the union of all W_j. $JIGSAW_{nouns}$ measures the similarity between each candidate sense $s_{ik} \in W_i$ and each sense $s_h \in T$. The sense assigned to w_i is the one with the highest similarity score.

$JIGSAW_{verbs}$

Before describing the $JIGSAW_{verbs}$ procedure, the *description* of a synset must be defined. It is the string obtained by concatenating the gloss and the sentences that WordNet uses to explain the usage of a word. For example, the gloss for the synset corresponding to the sense n.2 of the verb *look* ($\{look, appear, seem\}$) is "*give a certain impression or have a certain outward aspect*", while some examples of usage of the verb are: "*She seems to be sleeping*"; "*This appears to be a very difficult problem*". The description of the synset is "*give a certain impression or have a certain outward aspect She seems to be sleeping This appears to be a very difficult problem*". First, the $JIGSAW_{verbs}$ includes in the context C for the target verb w_i all the nouns in the window of $2n$ words surrounding w_i. For each candidate synset s_{ik} of w_i, the algorithm computes $nouns(i, k)$, that is the set of nouns in the description for s_{ik}. In the above example, $nouns(look, 2)$={*impression*, *aspect*, *problem*}. Then, for each w_j in C and each synset s_{ik}, the following value is computed:

$$max_{jk} = max_{w_l \in nouns(i,k)} \{\texttt{SinSim}(w_j, w_l)\} \qquad (1)$$

In other words, max_{jk} is the highest similarity value for w_j, with respect to the nouns related to the k-th sense for w_i. Finally, a score for each s_{ik} is computed:

$$\varphi(i,k) = R(k) \cdot \frac{\sum_{w_j \in C} G(pos_j) \cdot max_{jk}}{\sum_h G(pos_h)} \qquad (2)$$

where $R(k)$ is the ranking of s_{ik} (synsets in WordNet are ranked according to their frequency of usage) and $G(pos_j)$ is a gaussian factor related to the position of w_j with respect to w_i in the original text that gives a higher weight to words near the target word. The synset assigned to w_i is the one with the highest φ value.

$JIGSAW_{others}$

This procedure is based on the WSD algorithm proposed in [Banerjee and Pedersen, 2002]. The idea is to compare the glosses of each candidate sense for the target word to the glosses of all the words in its context. Let W_i be the sense inventory for the target word w_i. For each $s_{ik} \in W_i$, $JIGSAW_{others}$ computes the string $targetGloss_{ik}$ that contains the words in the gloss of s_{ik}. Then, the procedure computes the string $contextGloss_i$, which contains the words in the glosses of all the synsets corresponding to each word in the context for w_i. Finally, the procedure computes the *overlap* between $contextGloss_i$ and $targetGloss_{ik}$, and assigns the synset with the highest overlap score to w_i. This score is computed by counting the words that occur both in $targetGloss_{ik}$ and in $contextGloss_i$. The JIGSAW algorithm was evaluated according to the parameters of the *Senseval* initiative[5], that provides a forum where the WSD systems are assessed against disambiguated datasets. In order to measure the capability of disambiguating a complete text, the "All Words Task" for English was chosen. JIGSAW reaches the fourth position in that task, by achieving precision and recall equal to 50%. This result assures that our WSD algorithm can be configured to have high precision, and thus would add very little noise in the training set. Due to space limitations, the details of the experiments are not reported.

[4] http://acopost.sourceforge.net/

[5] http://www.senseval.org.

3.2 Keyword-based and Synset-based Document Representation

The WSD procedure described in the previous section is adopted to obtain a synset-based vector space representation that we called *bag-of-synsets* (BOS). In this model, a synset vector instead of a word vector represents a document. Another key feature of the approach is that each document is represented by a set of M *slots*, where each slot is a textual field corresponding to a specific feature of the document, in an attempt to take also into account the document structure. According to the BOS model, the text in each slot is represented by counting separately the occurrences of a synset in the slots in which it occurs. More formally, assume that we have a collection of N documents. Let *m* be the index of the slot, for $n = 1, 2, \dots, N$, the n-th document d_n is reduced to M bags of synsets, one for each slot:

$$d_n^m = \langle t_{n1}^m, t_{n2}^m, \dots, t_{nD_{nm}}^m \rangle,\ \text{m=1, 2, \dots, M}$$

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

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

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, as we used in our experiments, or a more complex TF-IDF score. Our hypothesis is that the proposed document representation helps to obtain profiles able to recommend documents semantically closer to the user interests. The difference with respect to keyword-based profiles is that synset unique identifiers replace words.

4 A Naïve Bayes Method for User Profiling

ITem Recommender (ITR) uses a Naïve Bayes text categorization algorithm to build profiles as binary classifiers (*user-likes* vs *user-dislikes*). 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:

$$P(c_j|d_i) = P(c_j) \prod_{w \in d_i} P(t_k|c_j)^{N(d_i,t_k)} \tag{3}$$

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 (3) becomes:

$$P(c_j|d_i) = \frac{P(c_j)}{P(d_i)} \prod_{m=1}^{|S|} \prod_{k=1}^{|b_{im}|} P(t_k|c_j, s_m)^{n_{kim}} \tag{4}$$

where $S = \{s_1, s_2, \dots, s_{|S|}\}$ 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 (4), 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:

$$w_+^i = \frac{r-1}{MAX-1}; \qquad w_-^i = 1 - w_+^i \tag{5}$$

The scores in (5) 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:

$$\hat{P}(c_j) = \frac{\sum_{i=1}^{|TR|} w_j^i + 1}{|TR| + 2} \tag{6}$$

Witten-Bell smoothing [1991] 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 (5):

$$\hat{P}(t_k|c_j, s_m) = \begin{cases} \frac{N(t_k,c_j,s_m)}{V_{c_j} + \sum_i N(t_i,c_j,s_m)} & \text{if } N(t_k, c_j, s_m) \neq 0 \\ \frac{V_{c_j}}{V_{c_j} + \sum_i N(t_i,c_j,s_m)} \frac{1}{V - V_{c_j}} & \text{otherwise} \end{cases} \tag{7}$$

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:

$$N(t_k, c_j, s_m) = \sum_{i=1}^{|TR|} w_j^i n_{kim} \tag{8}$$

In (8), 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 (7) 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 (7).

5 Experimental Evaluation

The goal of the experiments was to compare the performance of synset-based user profiles to that of keyword-based profiles. Experiments were carried out on a content-based extension of the EachMovie dataset[6], a collection of $1,628$ textual descriptions of movies rated by $72,916$ users on a 6-point scale $(1-6)$. The content information for each movie was collected from the Internet Movie Database[7] by using a crawler that gathered the *Title*, the *Director*, the *Genre*, that is the category of the movie, the *Keywords*, the *Summary* and the *Cast*. Movies are subdivided into different genres: *Action*, *Animation*, *Classic*, *Art_Foreign*, *Comedy*, *Drama*, *Family*, *Horror*, *Romance*, *Thriller*. For each genre or category, a set of 100 users was randomly selected among users that rated *n* items, $30 \leq n \leq 100$ in that movie category (only for genre 'Animation', the number of users that rated *n* movies was 33, due to the low number of movies in that genre). In this way, for each category, a dataset of at least $3,000$ triples (user, movie, rating) was obtained (at least 990 for 'Animation'). Table 1 summarizes the data used for the experiments.

Id	Genre	Number ratings	% POS	% NEG
1	Action	4,474	72	28
2	Animation	1,103	57	43
3	Art_Foreign	4,246	76	24
4	Classic	5,026	92	8
5	Comedy	4,714	63	37
6	Drama	4,880	76	24
7	Family	3,808	64	36
8	Horror	3,631	60	40
9	Romance	3,707	73	27
10	Thriller	3,709	72	28
		39,298	72	28

Table 1: 10 'Genre' datasets obtained from EachMovie

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 on proper names. Documents have been processed by the JIGSAW algorithm 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 inferred by learning from BOW-represented documents, whilst synset-based profiles were obtained from BOS-represented documents. As ITR is conceived as a text classifier, its effectiveness is evaluated by the well-known classification accuracy measures precision and recall [Sebastiani, 2002]. Also used is F1 measure, a combination of precision and recall. We adopted the Normalized Distance-based Performance Measure (NDPM) [Yao, 1995] to measure the distance between the ranking imposed on documents by the user ratings and the ranking predicted by ITR, that ranks documents according to the a-posteriori probability of the class *likes*. 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 the system is able to recommend relevant documents and how these documents are ranked. In all the experiments, a movie description d_i is considered *relevant* by a user if the rating is greater or equal to 4, while ITR considers a description relevant if $P(c_+|d_i) > 0.5$, computed as in equation (4). 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 induced profile on *Ts*, using the aforementioned measures. The methodology adopted for obtaining *Tr* and *Ts* was the 5-fold cross validation. Table 2 shows the results reported over all 10 genres by ITR.

	Precision		**Recall**		**F1**		**NDPM**	
Id	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
	0.67	0.75	0.78	0.88	0.73	0.81	0.45	0.45

Table 2: Performance of ITR on 10 different datasets

A significant improvement of BOS over BOW both in precision (+8%) and recall (+10%) can be noticed. The BOS model outperforms the BOW model specifically on datasets 5 (+11% of precision, +14% of recall), 7 (+15% of precision, +16% of recall), 8 (+19% of precision, +24% of recall). Only on dataset 4 no improvement can be observed, probably because precision and recall are already very high. It could be noticed from the NDPM values that the relevant / not relevant classification accuracy is increased without improving the ranking. This result can be explained by the example in Table 3, in which each column reports the ratings or scores of the items and the corresponding positions in the ranking.

Let R_u be the ranking imposed by the user *u* on a set of 10 items, R_A the ranking computed by A, and R_B the ranking computed by method B (ratings ranging between 1 and 6 - classification scores ranging between 0 and 1). An item is considered relevant if the rating $r > 3$ (symmetrically, if the ranking score $s \geq 0.5$). Method A has a better classification accuracy compared to method B (Recall=4/5, Precision=$4/5$ vs. Recall=3/5, Precision=$3/4$). NDPM is almost the same for both methods because the two rankings R_A and R_B are very similar. The difference is that I4 is ranked above I6 in R_A, whilst I6 is ranked above I4 in R_B. The general conclusion is that method A (BOS model) has improved the classification of items whose score (and ratings) are close to the relevant / not relevant threshold (items for which the classi-

[6] 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/

[7] IMDb, http://www.imdb.com

Item	R_u	R_A	R_B
I1	6 (1)	0.65 (2)	0.65 (2)
I2	5 (2)	0.62 (3)	0.60 (3)
I3	5 (3)	0.75 (1)	0.70 (1)
I4	4 (4)	0.60 (4)	0.45 (5)
I5	4 (5)	0.43 (6)	0.42 (6)
I6	3 (6)	0.55 (5)	0.55 (4)
I7	3 (7)	0.40 (7)	0.40 (7)
I8	2 (8)	0.30 (8)	0.30 (8)
I9	1 (9)	0.25 (9)	0.25 (9)
I10	1 (10)	0.20 (10)	0.20 (10)

Table 3: Example of situation in which classification accuracy is increased without improving ranking

fication is highly uncertain). A Wilcoxon signed ranked test ($p < 0.05$) has been performed to validate the results. Each genre dataset has been considered as a single trial for the test. Results confirmed that there is a statistically significant difference in favor of the BOS model compared to the BOW model as regards precision, recall and F1-measure. Conversely, the two models are equivalent in defining the ranking of the preferred movies according to the score for the class "likes".

6 Conclusions and Future Work

We presented a system that exploits a Bayesian learning method to induce *semantic* user profiles from documents represented by WordNet synsets suggested by the WSD algorithm JIGSAW. Our hypothesis is that, replacing words with synsets in the indexing phase, produces a more effective document representation, which can help learning algorithms to infer more accurate user profiles. Our approach has been evaluated in a movie recommending scenario. Results showed that the integration of the WordNet linguistic knowledge in the learning process improves the classification of those documents for which the classification is highly uncertain. As a future work, we plan to exploit not only the WordNet hierarchy, but also domain ontologies in order to realize a more powerful document indexing.

References

[Banerjee and Pedersen, 2002] S. Banerjee and T. Pedersen. An Adapted Lesk Algorithm for Word Sense Disambiguation using WordNet. In *CICLing '02: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, pages 136–145, London, UK, 2002. Springer-Verlag.

[Bloedhorn and Hotho, 2004] S. Bloedhorn and A. Hotho. Boosting for text classification with semantic features. In *Proceedings of 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Mining for and from the Semantic Web Workshop*, pages 70–87, 2004.

[Degemmis *et al.*, 2007] M. Degemmis, P. Lops, and G. Semeraro. A Content-Collaborative Recommender that Exploits WordNet-based User Profiles for Neighborhood Formation. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*, 2007. Forthcoming.

[Fellbaum, 1998] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[Guarino *et al.*, 1999] N. Guarino, C. Masolo, and G. Vetere. Content-Based Access to the Web. *IEEE Intelligent Systems*, 14(3):70–80, 1999.

[Leacock and Chodorow, 1998] C. Leacock and M. Chodorow. Combining Local Context and WordNet Similarity for Word Sense Identification. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 266–283. MIT Press, 1998.

[Magnini and Strapparava, 2001] B. Magnini and C. Strapparava. Improving User Modelling with Content-based Techniques. In *Proceedings of the Eighth International Conference on User Modeling*, pages 74–83. Springer, 2001.

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

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

[Mooney and Roy, 2000] R. J. Mooney and L. Roy. Content-Based Book Recommending Using Learning for Text Categorization. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, pages 195–204, San Antonio, US, 2000. ACM Press, New York, US.

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

[Resnik, 1995] P. Resnik. Disambiguating Noun Groupings with respect to WordNet Senses. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 54–68. Association for Computational Linguistics, 1995.

[Scott and Matwin, 1998] S. Scott and S. Matwin. Text Classification Using WordNet Hypernyms. In S. Harabagiu, editor, *Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference*, pages 38–44. Association for Computational Linguistics, Somerset, New Jersey, 1998.

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

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

[Yao, 1995] Y. Y. Yao. Measuring Retrieval Effectiveness Based on User Preference of Documents. *Journal of the American Society for Information Science*, 46(2):133–145, 1995.