Semantic Coherence-based User Profile Modeling in the Recommender Systems Context

Roberto Saia, Ludovico Boratto and Salvatore Carta
Dipartimento di Matematica e Informatica, Università di Cagliari, Italy
{*roberto.saia,ludovico.boratto,salvatore*}*@unica.it*

Keywords: Recommender systems, User Profiling.

Abstract: Recommender systems usually produce their results to the users based on the interpretation of the whole historic interactions of these. This canonical approach sometimes could lead to wrong results due to several factors, such as a changes in user taste over time or the use of her/his account by third parties. This work proposes a novel dynamic coherence-based approach that analyzes the information stored in the user profiles based on their coherence. The main aim is to identify and remove from the previously evaluated items those not adherent to the average preferences, in order to make a user profile as close as possible to the user's real tastes. The conducted experiments show the effectiveness of our approach to remove the incoherent items from a user profile, increasing the recommendation accuracy.

1 INTRODUCTION

The exponential growth of companies that sell their goods through the Word Wide Web generates an enormous amount of valuable information that can be exploited to improve the quality and efficiency of the sales criteria (Schafer et al., 1999). This aspect collides with the information overload, which needs an appropriate approach to be exploited to the fullest (Wei et al., 2014). Recommender systems represent an effective response to the so-called information overload problem, in which companies are finding it increasingly difficult to filter the huge amount of information about their customers in order to get useful elements to produce suggestions for them (Vargiu et al., 2013). The denomination *Recommender Systems* (RS) (Ricci et al., 2011) denotes a set of software tools and techniques providing to a user suggestions for items. In this work we address one of the main aspects related to the recommender systems, i.e., how to best exploit the information stored in the user profiles.

The problem is based on the consideration that most of the solutions regarding the *user-profiling* involve the interpretation of the whole set of previous user interactions, which are compared with each item not yet evaluated, in order to measure their similarity and recommend the most similar items (Lops et al., 2011). This is because recommender systems usually assume that users' preferences remain unchanged over time and this can be true in many cases, but it is not the norm due to the existence of temporal dynamics in their preferences (Li et al., 2007; Lam et al., 1996; Widyantoro et al., 2001). Therefore, a static approach of user profiling can lead towards wrong results due to various factors, such as a simple change of tastes over time or the temporary use of a personal account by other people. The primary aim of the approach that we introduce is the measure of the similarity between a single item and the others within the user profile, in order to improve the recommendation process by discarding the items that are highly dissimilar with the rest of the user profile. To perform this task we introduce the *Dynamic Coherence-Based Modeling* (*DCBM*) algorithm, through which we face the problems mentioned before. The *DCBM* algorithm is based on the concept of *Minimum Global Coherence* (*MGC*), a metric that allows us to measure the semantic similarity between a single item with the others within the user profile. The algorithm, however, takes into account two other factors, i.e., the position of each item in the chronology of the user's choices, and the distance from the *mean value* of the *global similarity* (as "global" we mean all the items in a user profile). These metrics allow us to remove in a selective way any item that could make the user profiles non-adherent to their real tastes. In order to evaluate the capability of our approach to produce accurate user profiles, we are going to include the *DCBM* algorithm into a *state-of-the-art* semantic-based recommender system (Capelle et al., 2012) and evaluate the accuracy of the recommendations. Since the task of the recommender system that predicts the interest of the users for the items relies on the information included in a user profile, more accurate user profiles lead to an improved accuracy of the whole recommender system. Experimental results show the capability of our approach to remove the incoherent items from a user profile, increasing the accuracy of recommendations. The main contribution of our proposal is the introduction of a novel approach to improve the quality of suggestions within the recommender systems environment, i.e., a dynamic way to use the information in the user profiles, in order to discover and remove from the user profiles any item that could make the profile non-adherent to real tastes of the users. The rest of the paper is organized as follows: Section 2 presents related work on user profiling; in Section 3 we introduce the background on the concepts and problems handled by our proposal; Section 4 presents the details of the *DCBM* algorithm and its integration into a *state-of-the-art* semantic-based recommender system; Section 5 presents the experimental framework used to evaluate our approach; Section 6 contains conclusions and future work.

2 Related Work

When it comes to producing personalized recommendations to users, the first requirement is to understand the needs of the users and to build a user profile that models these needs. There are several approaches to build profiles: some of them focus on *short-term* user profiles that capture features of the user's current search context (Shen et al., 2005; Budzik and Hammond, 2000; Finkelstein et al., 2002), while others accommodate *long-term* profiles that capture the user's preferences over a long period of time (Chirita et al., 2005; Asnicar and Tasso, 1997; Ma et al., 2007). As shown in (Widyantoro et al., 2001), compared with the *short-term* user profiles, the use of a *long-term* user profile generally produces more reliable results, at least when the user preferences are fairly stable over a long time period. Regardless of the type of profiling that is adopted (e.g., *long-term* or *short-term*), there is a common problem that may affect the goodness of the obtained results, i.e., the capability of the information stored in the user profile to lead towards reliable recommendations. In order to face the problem of dealing with unreliable information in a user profile, the state-of-the-art proposes different operative strategies. Several approaches, such as (Lam et al., 1996), take advantage from the Bayesian analysis of the user provided relevance feedback, in order to detect non-stationary user interests. Also exploiting the feedback information provided by the users, other approaches such as (Widyantoro et al., 2001) make use of a *tree-descriptor* model to detect shifts in user interests. Another technique exploits the knowledge captured in an ontology (Schickel-Zuber and Faltings, 2006) to obtain the same result, but in this case it is necessary for the users to express their preferences about items through an explicit rating. There are also other different strategies that try to improve the accuracy of the information in the user profiles by collecting the implicit feedbacks of the users during their natural interactions with the system (reading-time, saving, etc.) (Kelly and Teevan, 2003). However, irrespective of the approach used, it should be pointed out that most of the strategies are usually effective only in specific contexts, such as for instance (Zeb and Fasli, 2011), where a novel approach to model a user profile according to the change in her/his tastes is designed to operate in the context of the articles recommendation. With regard to the analysis of information related to user profiles and items, there are several ways to operate and most of them work by using the *bag-of-words* model, an approach where the words are processed without taking account of the correlation between terms (Lam et al., 1996; Widyantoro et al., 2001). This trivial way to manage the information usually does not lead towards good results, and more sophisticated alternatives, such as the semantic analysis of the content in order to model the preferences of a user (Pedersen et al., 2004), are often adopted.

3 Background

Here, we introduce two key concepts for this work, i.e., the document representation based on the *Vector Space Model*, and the *WordNet* environment.

3.1 Vector Space Model

Many content-based recommender systems use relatively simple retrieval models (Lops et al., 2011), such as the *Vector Space Model* (*VSM*), with the basic *TF-IDF* weighting. *VSM* is a spatial representation of text documents, where each document is represented by a vector in a n-dimensional space, and each dimension is related to a term from the overall vocabulary of a specific document collection. In other words, every document is represented as a vector of term weights, where the weight indicates the degree of association between the document and the term. Let $D = \{d_1, d_2, ..., d_N\}$ indicate a set of documents, and $d = \{t_1, t_2, ..., t_N\}, t \in T$ be the set of terms in a document. The dictionary T is obtained by applying some standard Natural Language Processing (*NLP*) operations, such as tokenization, *stop-words* removal and stemming, and every document d_j is represented as a vector in a n-dimensional vector space, so $d_j = \left\{w_{1j}, w_{2j}, ..., w_{nj}\right\}$, where w_{kj} represents the weight for term t_k in document d_j . The main problems of the document representation with the *VSM* are the weighting of the terms and the evaluation of the similarity of the vectors. The most common way to estimate the term weighting is based on *TF-IDF* weighting, a trivial approach that uses empirical observations of the documents' text (Salton et al., 1975).

The *IDF* metric is based on the assumption that infrequent terms are not less important than frequent terms (as shown in Equation 1, where $|D|$ is the number of documents in the corpus and $\{|d \in D : t \in d|\}$ is the number of documents where term t appears).

$$IDF(t,D) = log\frac{|D|}{\{|d \in D : t \in d|\}} \quad (1)$$

For the *TF* assumption, multiple occurrences of a term are not less important than the single occurrences and, in addition, long documents are not preferred to short documents (as shown in Equation 2, where $f(t,d)$ is the number of occurrences of the considered term, and the denominator $max\{f(w,d) : w \in d\}$ is the number of occurrences of all terms).

$$TF(t,d) = \frac{f(t,d)}{max\{f(w,d) : w \in d\}} \quad (2)$$

To sum up, terms with multiple occurrences in a document (*TF*) but with a few of occurrences in the rest of documents collection (*IDF*) are more likely to be important to the topic of the document. The last step of the *TF-IDF* process is to normalize the obtained weight vectors, in order to prevent longer documents to have more chance of being retrieved (as shown in Equation 3).

$$\text{TF-IDF(t,d,D)} = TF(t,d) \cdot IDF(t,D) \quad (3)$$

Since the ratio inside the *IDF* equation is always greater than or equal to 1, the value of *IDF* (and of *TF-IDF*) is greater than or equal to zero.

3.2 WordNet Environment

In order to perform the similarity measures used in this work, we introduce the WordNet environment, since we use its dictionary to calculate the semantic similarity between two words. The main relation among words in WordNet is the synonymy and, in order to represent these relations, the dictionary is based on *synsets*, i.e., unordered sets of grouped words that denote the same concept and are interchangeable in many contexts. Each synset is linked to other synsets through a small number of *conceptual relations*. Words with more meanings are represented by distinct synsets, so that each form-meaning pair in WordNet is unique (e.g., the *fly* insect and the *fly* verb belong to two distinct synsets). Most of the WordNet relations connect words that belong to the same part-of-speech (POS). There are four POSs: *nouns*, *verbs*, *adjectives*, and *adverbs*. Due to the chosen similarity measure, we consider only the *nouns* and the *verbs*. In this work we exploited a *state-of-the-art* semantic-based approach to recommendation based on the WordNet synsets (Pedersen et al., 2004), in order to evaluate the similarity between the items not yet evaluated by a user and those stored in the profile.

4 Our Approach

As previously highlighted, individual profiles need to be as adherent as possible to the tastes of the users, because they are used to predict their future interests. In this section, we propose the novel *Dynamic Coherence-Based Modeling* (*DCBM*) approach that allows us to find and remove the incoherent items in a user profile. The implementation of *DCBM* on a recommender system is performed in four steps:

1. **Data Preprocessing**: preprocessing of the text description of the items in a user profile, as well as of the text description of the items not yet considered, in order to remove the useless elements and the items with a rating lower than the average;
2. **Dynamic Coherence-Based Modeling**: the items dissimilar from the average preferences of a user are identified by measuring the Minimum Global Coherence (MGC) and removed from the profile;
3. **Semantic Similarity**: WordNet features are used to retrieve all the pairs of synsets that have at least an element with the same *part-of-speech*, for which we measure the semantic similarity according to the *Wu and Palmer* metric;
4. **Item Recommendation**: we sort the not evaluated items by their similarity with the user profile, and recommend to the user a subset of those with the highest values of similarity.

Note that steps 1, 3, and 4 are followed by a *state-of-the-art* recommender system based on the semantic similarity (Capelle et al., 2012), in which we integrate our novel Dynamic Coherence-Based Modeling (*DCBM*) algorithm (step 2), in order to improve a user profile and increase the recommendation accuracy.

In the following, we describe in detail each step.

4.1 Data Preprocessing

Before comparing the similarity between the items in a user profile, we need to follow several preprocessing steps. The first step is to detect the correct *part-of-speech* (*POS*) for each word in the text; in order to perform this task, we have used the *Stanford Log-linear Part-Of-Speech Tagger* (Toutanova et al., 2003). In the second step, we remove punctuation marks and *stop-words*, i.e., words such as adjectives, conjunctions, etc., which represent noise in the semantic analysis. Several *stop-words* lists can be found on the Internet, and we have used a list of 429 *stop-words* made available with the *Onix Text Retrieval Toolkit*[1]. In the third step, after we have determined the lemma of each word using the Java API implementation for WordNet Searching *JAWS*[2], we perform the so-called word sense disambiguation, a process where the correct sense of each word is determined, which permits us to accurately evaluate the semantic similarity. The best sense of each word in a sentence was found using the Java implementation of the adapted Lesk algorithm provided by the *Denmark Technical University* (*DTU*) similarity application (Salton et al., 1975). After these preprocessing steps, we use *JAWS* to compute the semantic similarity between each user profile (the descriptions of the items evaluated with a score above the average[3]) and the description of the items not rated by the user.

4.2 Dynamic Coherence-Based Modeling

For the purpose of being able to make effective recommendations to users, their profiles need to store only the descriptions of the items that really reflect their tastes. In order to identify which items of a profile do not really reflect the user taste, the Dynamic Coherence-Based Modeling *DCBM* algorithm measures the *Minimum Global Coherence* (*MGC*) of each single item description with the set of other item descriptions stored in the user profile. In other words, through *MGC*, the most dissimilar item with respect to the other items is identified. Although the most used semantic similarity measures are five, i.e. *Leacock and Chodorow* (Leacock and Chodorow, 1998), *Jiang and Conrath* (Jiang and Conrath, 1997), *Resnik* (Resnik, 1995), *Lin* (Lin, 1998) and *Wu and Palmer* (Wu and Palmer, 1994), and each of them evaluates the semantic similarity through the Wordnet environment, we calculate the semantic similarity by using the *Wu and Palmer* (Wu and Palmer, 1994) measure, a method based on the path lengths between a pair of concepts (WordNet synsets), which in the literature is considered to be the most accurate when generating the similarities (Dennai and Benslimane, 2013; Capelle et al., 2012). It is a measure between concepts in an ontology restricted to taxonomic links (as shown in Equation 4).

$$sim_{WP}(x,y) = \frac{2 \cdot A}{B + C + (2 \cdot A)} \quad (4)$$

Assuming that the *Least Common Subsumer* (*LCS*) of two concepts x and y is *the most specific concept that is an ancestor of both x and y*, where the concept tree is defined by the *is-a* relation, in Equation 4 we have that: A=*depth(LCS(x,y))*, B=*length(x,LCS(x,y))*, C=*length(y,LCS(x,y))*. We can note that $B+C$ represents the path length from x and y, while A indicates the global depth of the path in the taxonomy.

The metric can be used to calculate the *MGC*, as shown in Equation 5.

$$MGC = min\left(sim_{WP}(y_n, \textstyle\sum y \in Y \setminus y_n), \forall y \in Y\right) \quad (5)$$

The idea is to isolate each individual item y_n in a user profile, and then measure the similarity with respect to the remaining items (i.e., the merging of the synsets of the rest of the items), in order to obtain a measure of its coherence within the overall context of the profile.

In other words, in order to detect the most distant element from the evaluated items, we exploit a basic principle of the differential calculus, since the *MGC* value shown upon is nothing else than the *maximum negative slope*, which is calculated by finding the ratio between the changing on the y axis and the changing on the x axis. Placing on the x axis the user interactions in chronological order, and on the y axis the corresponding values of *GS* (Global Similarity) calculated as $sim_{WP}(y_n, \sum y \in Y \setminus y_n), \forall y \in Y$, we can trivially calculate the slope value, denoted by the letter m, as shown in Equation 6 (where $y = f(x)$ since y is a function of x, thus as x varies, y varies also).

$$m = \frac{\triangle y}{\triangle x} = \frac{f(x + \triangle x) - f(x)}{\triangle x} \quad (6)$$

The differential calculus defines the slope of a curve at a point as the slope of the tangent line at that point. Since we are working with a series of points, the slope can be calculated not at a single point but between two points. Considering that for each user interaction $\triangle x$ is equal to 1 (i.e., for N user interactions: $1 - 0 = 1$, $2 - 1 = 1$, ..., $N - (N-1) = 1$), the slope m is always equal to $f(x + \triangle x) - f(x)$. As Equation 7 shows, the

[1] http://www.lextek.com/manuals/onix/stopwords.html

[2] http://lyle.smu.edu/ tspell/jaws/index.html

[3] The assumption is that users do not like the items that have been rated with values under the average, but only those whose rating is equal or higher to the average.

maximum negative slope is equal to the MGC[4].

$$min\left(\frac{\triangle y}{\triangle x}\right) = min\left(\frac{sim_{WP}(Y)}{1}\right) = MGC \qquad (7)$$

Figure 1, which displays the data reported in Table 1, illustrates this concept in a graphical way.

Table 1: User profile sample data

x	y	m	x	y	m
1	0.2884	+0.2884	7	0.2708	-0.0178
2	0.2967	+0.0083	8	0.3066	+0.0358
3	0.2772	-0.0195	9	0.3188	+0.0122
4	0.3202	+0.0430	10	0.2691	-0.0497
5	0.2724	-0.0478	11	0.2878	+0.0187
6	0.2886	+0.0162			

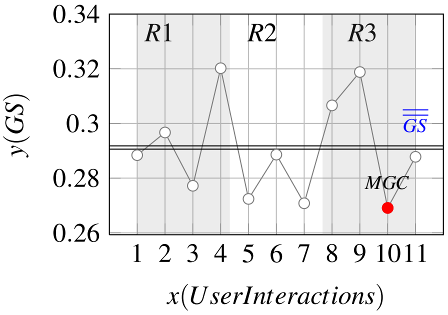

Figure 1: The maximum negative slope is equal to the *MGC*

In order to avoid the removal of an item that might correspond to a recent change in the tastes of the user or not semantically distant enough from the context of the remaining items, the *DCBM* algorithm removes an item only if meets the following conditions:

1. it is located in the first part of the user interaction history. Therefore, an item is considered far from the user's tastes only if it is in the first part of the interactions. This condition is checked thanks to a parameter r, which defines the *removal area*, i.e., the percentage of a user profile where an item can be removed. Note that $0 \leq r \leq 1$, so in the example in Figure 1, $r = \frac{2}{3} = 0.66$ (i.e., the element related to MGC value is located in the region $R3$, so it does not meet the requirement);
2. the value of MGC must be within a tolerance range, which takes into account the *mean value* of the *global similarity* (as global we mean the environment of the items in the user profile).

With respect to the second requirement, we prevent the removing of items when they do not have a *significant* semantic distance with the remaining items. In order to do so, we first need to calculate the value of the mean similarity in the context of the user profile and for this reason we need to define a threshold value that determines when an item must be considered incoherent with respect to the current context. Equation 8 measures the mean similarity, denoted by $\overline{GS}$, by calculating the average of the *Global Similarity* (*GS*) values, which are obtained as $sim_{WP}(y_j, \sum y \in Y \setminus y_j), \forall y \in Y$.

$$\overline{GS} = \frac{1}{N} \cdot \sum (sim_{WP}(y_n, \sum y \in Y \setminus y_n), \forall y \in Y) \qquad (8)$$

where N is the total number of user interactions, i.e., the number of items y_n in the profile (in the case of data shown in Table 1, $\overline{GS} = 0.2906$). Obtained this average value, we can define the condition ρ, used to decide whether an item has to be removed ($\rho = 1$) or not ($\rho = 0$), based on a threshold value α, added to the average value $\overline{GS}$ to define a certain tolerance (as shown in Equation 9.

$$\rho = \begin{cases} 1, & \text{if } MGC < (\overline{GS} - \alpha) \\ 0, & \text{otherwise} \end{cases} \qquad (9)$$

We can now define Algorithm 1, used to remove the semantically incoherent items from a user profile. The algorithm requires as input a user profile Y, a parameter α used to define the accepted distance of an item from the average, and a removal area r used to define in which part of the profile an item should be removed. Steps 3-5 compute the similarity between each couple of synsets that belong to the user profile. In step 6, the average of the similarities is computed, so that steps 7-14 can evaluate if an item has to be removed from a user profile or not. In particular, once an item y_i is removed from a profile in Step 11, its associated similarity s is removed from the list S (step 12), so that m in step 8 can be set as the minimum similarity value after the item removal. In step 15, the algorithm returns the user profile with the items not removed.

4.3 Semantic Similarity

In accordance with the state-of-the-art related to the recommendations produced by performing a semantic analysis based on WordNet (Capelle et al., 2013), we perform the measurements of the similarity between items in this way: given a set X of i WordNet synsets $x_1, x_2, \ldots, x_i$ related to the description of an item not yet evaluated by a user, and a set Y of j WordNet synsets $y_1, y_2, \ldots, y_j$ related to the description of the items in a user profile, we define a set I, which contains all the possible pairs formed with synsets of X and Y, as in Equation 10.

$$I = (\langle x_1, y_1 \rangle, \langle x_1, y_2 \rangle, \ldots, \langle x_i, y_j \rangle)\ \forall x \in X, y \in Y \qquad (10)$$

[4] $sim_{WP}(Y)$ denotes $sim_{WP}(y_n, \sum y \in Y \setminus y_n), \forall y \in Y$

Algorithm 1 DCBM Algorithm

```
Require: Y=set of items in the user profile, α=threshold value, r=removal area
procedure PROCESS($Y$)
    $N = |Y|$
    for each Pair p=($y_i, \sum y \setminus y_i$) in $Y$ do
        $S \leftarrow sim_{WP}(p)$
    end for
    $a = Average(S)$
    for each $s$ in $S$ do
        $MGC = Min(S)$
        $i = index(MGC)$
        if $i < r * n$ AND $MGC < (a + \alpha)$ then
            Remove($y_i$)
            Remove($s$)
        end if
    end for
    Return $Y$
end procedure
```

Next, we create a subset Z of the pairs in I that have at least an element with the same *POS* (Equation 11).

$$Z \subseteq I, \forall (x_i, y_j) \in Z : \exists POS(x_i) = POS(y_j) \quad (11)$$

The similarity between an item not evaluated by a user and the user profile (descriptions of the evaluated items with a rating equal or higher than the average) is defined as the sum of the similarity scores for all pairs, divided by its cardinality (the subset Z of synsets with a common part-of-speech), as shown in Equation 12.

$$sim_{WP}(X,Y) = \frac{\sum_{(x,y) \in Z}^{n} sim_{WP}(x,y)}{|Z|} \quad (12)$$

4.4 Item Recommendation

After the user profile has been processed with the Algorithm 1 and its semantic similarity with all the items not evaluated has been computed, this step recommends to the user a subset of those with the highest similarity.

5 Experimental Framework

The experimental environment is based on the Java language, with the support of Java API implementation for WordNet Searching (*JAWS*) previously mentioned. In order to perform the evaluation, we estimated the $F_1 - measure$ increment (or decrement) of our novel *DCBM* approach, compared with a *state-of-the-art* recommender system based on the semantic similarity (Capelle et al., 2012). As highlighted throughout the paper, the system presented in Section 4 performs the same steps as the reference one, with the introduction of the *DCBM* algorithm. Since all the steps in common between the two recommender systems are performed with the same algorithms, the comparison of the F_1-measure obtained by the two systems highlights the capability of *DCBM* to improve the quality of the user profile and of the accuracy of a recommender system. Regarding the first condition to meet (see Section 4) in order to remove the items from a user profile, in our experiments we divided the user interaction history into 10 parts, considering valid for the removal only the first 9 (i.e., $r = 0.9$). The reference dataset was generated by using the Yahoo! Webscope Movie dataset (R4)[5], which contains a large amount of data related to users preferences rated on a scale from 1 to 5. The original dataset is already split into a training and a test set. From this source of data we have extracted two subsets related to 10 users. For each movie, we considered its description and title. Since the algorithm considers only the items with a rating above the average, we selected only the movies with a rating ≥ 3. The subsets involve a total of 568 items (movies), 386 in the training subset and 182 in the test subset. The experimentation result was obtained by comparing the recommendations with the real users choices stored in the test set.

5.1 Metrics

In order to evaluate the performance of our approach with this dataset, we use the performance measures precision and recall, which we combine to calculate the F_1*–measure* (Baeza-Yates and Ribeiro-Neto, 1999). The F_1*–measure* is a combined *Harmonic Mean* of the *precision* and *recall* measures, used to evaluate the accuracy of a recommender system.

5.2 Strategy

For the experiments, it is necessary to set the value of α in Algorithm 1, which controls when an item is too distant from the average value $\overline{GS}$. We have tested some values positioned around the average value of the *Global Similarity* $\overline{GS}$. The tested values interval is the half of the $\overline{GS}$ value (e.g., if $\overline{GS} = 0.4$, the excursion of the values is from -0.2 to +0.2, centered in $\overline{GS}$, so between 0.2 and 0.6). The interval of values is divided into 10 equal parts, labeled from -5 to 5.

5.3 Results

Figure 2 shows the per-cent increasing of F_1*–measure* of our solution compared with the *state-of-the-art* recommender system. From the results, we can observe

[5] http://webscope.sandbox.yahoo.com

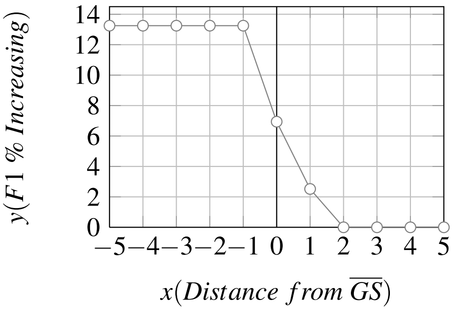

Figure 2: F_1*–measure* Percentage Increasing

how the average value of coherence (i.e., $\overline{GS}$, represented by the zero on the x axis) represents the borderline between the improvement and worsening in terms of quality of the carried out recommendations. That is because we obtain the maximum improvement in correspondence with the -5 value on the x axis, which represents the maximum distance from the mean value of coherence $\overline{GS}$ (i.e., the value corresponding to the most incoherent items stored in the user profile). This improvement is progressively reduced as we approach the value of $\overline{GS}$, becoming zero almost immediately after this, because in this case we are removing from the user profile some items that are coherent with her/his global choices, which are essential to perform reliable recommendations. To sum up, Figure 2 shows that the F_1*–measure* percentage increases, until it becomes stable above certain values and presents no gain below others: this happens because we obtain an improvement only when the exclusion process involves items with a high level of semantic incoherence with respect to the others.

6 Conclusions and Future Work

In this paper we proposed a novel approach to improve the quality of the user profiling, a strategy that takes into account the items related by a user, with the aim of removing those that not reflect her/his real tastes. Future work will aim at discovering the semantic interconnections between different classes of items, in order to evaluate their semantic coherence during the user profiling activity.

ACKNOWLEDGEMENTS

This work is partially funded by Regione Sardegna under project SocialGlue, through PIA - Pacchetti Integrati di Agevolazione “Industria Artigianato e Servizi” (annualità 2010).

REFERENCES

Asnicar, F. A. and Tasso, C. (1997). ifweb: a prototype of user model-based intelligent agent for document filtering and navigation in the world wide web. In *Proceedings of Workshop Adaptive Systems and User Modeling on the World Wide Web'at 6th International Conference on User Modeling, UM97, Chia Laguna, Sardinia, Italy*, pages 3–11.

Baeza-Yates, R. A. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Budzik, J. and Hammond, K. J. (2000). User interactions with everyday applications as context for just-in-time information access. In *Proceedings of the 5th International Conference on Intelligent User Interfaces*, IUI '00, pages 44–51, New York, NY, USA. ACM.

Capelle, M., Frasincar, F., Moerland, M., and Hogenboom, F. (2012). Semantics-based news recommendation. In *Proceedings of the 2Nd International Conference on Web Intelligence, Mining and Semantics*, WIMS '12, pages 27:1–27:9, New York, NY, USA. ACM.

Capelle, M., Hogenboom, F., Hogenboom, A., and Frasincar, F. (2013). Semantic news recommendation using wordnet and bing similarities. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 296–302, New York, NY, USA. ACM.

Chirita, P. A., Nejdl, W., Paiu, R., and Kohlschütter, C. (2005). Using odp metadata to personalize search. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 178–185, New York, NY, USA. ACM.

Dennai, A. and Benslimane, S. M. (2013). Toward an update of a similarity measurement for a better calculation of the semantic distance between ontology concepts. In *The Second International Conference on Informatics Engineering & Information Science (ICIEIS2013)*, pages 197–207. The Society of Digital Information and Wireless Communication.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2002). Placing search in context: The concept revisited. *ACM Trans. Inf. Syst.*, 20(1):116–131.

Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*.

Kelly, D. and Teevan, J. (2003). Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28.

Lam, W., Mukhopadhyay, S., Mostafa, J., and Palakal, M. J. (1996). Detection of shifts in user interests for personalized information filtering. In *SIGIR*, pages 317–325.

Leacock, C. and Chodorow, M. (1998). Combining local context and wordnet similarity for word sense identification. In Fellbaum, C., editor, *WordNet: An Electronic Lexical Database*, pages 305–332. MIT Press.

Li, L., Yang, Z., Wang, B., and Kitsuregawa, M. (2007). Dynamic adaptation strategies for long-term and short-term user profile to personalize search. In Dong, G., Lin, X., Wang, W., Yang, Y., and Yu, J. X., editors, *Advances in Data and Web Management, Joint 9th Asia-Pacific Web Conference, APWeb 2007, and 8th International Conference, on Web-Age Information Management, WAIM 2007, Huang Shan, China, June 16-18, 2007, Proceedings*, volume 4505 of *Lecture Notes in Computer Science*, pages 228–240. Springer.

Lin, D. (1998). An information-theoretic definition of similarity. In Shavlik, J. W., editor, *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconsin, USA, July 24-27, 1998*, pages 296–304. Morgan Kaufmann.

Lops, P., de Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 73–105. Springer.

Ma, Z., Pant, G., and Sheng, O. R. L. (2007). Interest-based personalized search. *ACM Trans. Inf. Syst.*, 25(1).

Pedersen, T., Patwardhan, S., and Michelizzi, J. (2004). Wordnet::similarity: Measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL–Demonstrations '04, pages 38–41, Stroudsburg, PA, USA. Association for Computational Linguistics.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to recommender systems handbook. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 1–35. Springer.

Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.

Schafer, J. B., Konstan, J. A., and Riedl, J. (1999). Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166.

Schickel-Zuber, V. and Faltings, B. (2006). Inferring user's preferences using ontologies. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 1413–1418. AAAI Press.

Shen, X., Tan, B., and Zhai, C. (2005). Implicit user modeling for personalized search. In Herzog, O., Schek, H.-J., Fuhr, N., Chowdhury, A., and Teiken, W., editors, *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, October 31 - November 5, 2005*, pages 824–831. ACM.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Vargiu, E., Giuliani, A., and Armano, G. (2013). Improving contextual advertising by adopting collaborative filtering. *ACM Trans. Web*, 7(3):13:1–13:22.

Wei, C., Khoury, R., and Fong, S. (2014). Recommendation systems for web 2.0 marketing. In Yada, K., editor, *Data Mining for Service*, volume 3 of *Studies in Big Data*, pages 171–196. Springer Berlin Heidelberg.

Widyantoro, D. H., Ioerger, T. R., and Yen, J. (2001). Learning user interest dynamics with a three-descriptor representation. *JASIST*, 52(3):212–225.

Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zeb, M. and Fasli, M. (2011). Adaptive user profiling for deviating user interests. In *Computer Science and Electronic Engineering Conference (CEEC), 2011 3rd*, pages 65–70.