Taking Advantage of Semantics in Recommendation Systems

Victor CODINA[a,1] and Luigi CECCARONI[a]
[a]*Departament de Llenguatges i Sistemes Informàtics (LSI),*
Universitat Politècnica de Catalunya (UPC),
Campus Nord, Edif. Omega, C. Jordi Girona, 1-3, 08034 Barcelona, Spain

Abstract. Recommendation systems leverage product and community information to target products to consumers. Researchers have developed collaborative recommendation systems, content-based recommendation systems and a few hybrid systems. We propose a semantic framework to overcome common limitations of current systems. We present a system whose representations of items and user-profiles are based on concept taxonomies in order to provide personalized recommendation and services. The recommender incorporates semantics to enhance (1) user modeling by applying a domain-based inference method, and (2) recommendation by applying a semantic-similarity method. We show that semantics can often be used to overcome information scarcity. Experiments on movie-data from Netflix show that systems incorporating semantics produce significantly better quality recommendations than content-based ones.

Keywords: Recommendation systems, Semantic Web, Ontology-based representation, Semantically-enhanced reasoning, Content-based filtering.

Introduction

Recommendation systems leverage product and community information to target products to consumers. Most common limitations of current recommendation systems are: *cold-start*, *sparsity*, *overspecialization* and *domain-dependency* [1]. Although some particular combination of recommendation techniques can improve the recommendation's quality in some domains, there is not a general solution to overcome these limitations. The use of *semantic Web* technologies to formally represent data [2] can provide several advantages in the context of personalized recommendation systems: the dynamic contextualization of user's interests in specific domains, the guarantee of inter-operability of system resources and the inference of incomplete information about user's interests. We think that the next generation of recommenders should focus on how their personalization processes can take advantage of semantics as well as social data to improve their recommendations.

The structure of the paper is as follows: in section 1 we present the state of the art of semantic recommenders; in section 2 we describe a new domain-independent recommendation system; and in section 3 we present the experimental evaluation of the semantically-enhanced algorithms of the recommender.

[1] Corresponding author: vcodina@lsi.upc.edu

1. Background and Related Work

Semantic recommendation systems are characterized by the incorporation of semantic knowledge in their processes in order to improve recommendation's quality. Most of them employ a concept-based approach to improve the user-profile representation (*user modeling* stage), and use standard vocabularies and ontology languages like OWL. Two different approaches can be distinguished (See [3] for more details about related work.):

1. Approaches employing *spreading activation* to maintain user interests and treating the user-profile as a semantic network. The interest scores of a set of concepts are propagated to other related concepts based on pre-computed weights of concepts relations. *News@hand* [4] is a news recommender system which employs this method to expand the initial set of long-term user's interests with other concepts related with the runtime context. A search recommender [5] uses this method to incrementally update the short-term user's interests during the user session.
2. Approaches that apply *domain-based inferences*, which consist of making inferences about user's interests based on the hierarchical structure defined by the ontology. The most commonly used is the *upward-propagation*, whose main idea is to assume that the user is interested in a general concept if he is interested in a given percentage of its direct sub-concepts. This kind of mechanisms allows inferring new knowledge about the long-term user's interests and therefore modeling richer user-profiles.

Other recommenders focus on exploiting semantics to improve their recommendation techniques (*content adaptation* stage). Most of them make use of *semantic similarity* methods to enhance the performance of *content-based* (CB) approaches, although there are also recommenders [6] using semantics to enhance the user-profile matching of *collaborative filtering* approaches. *ePaper* [7], a scientific-paper recommender, makes use of the hierarchical relationships of domain concepts to calculate the matching between the concepts describing an item and the concepts forming user's interests. The *FOAFing the music* project [8] is a music recommender that employs the standard FOAF vocabulary[2] to represent user profiles and exploits the semantic descriptions of music, mainly artist relationships, to find music similar to the listening habits of users to be recommended. The only recommender that makes use of semantic reasoning methods in both stages of the personalization process is *AVATAR* [9], a TV recommender that employs *upward-propagation* and *semantic similarity* methods.

2. A New Semantic Recommendation System

In this section we present the main components and characteristics of the semantic recommendation system we developed. The approach we present brings into focus two main aspects: the system is domain-independent and therefore can provide personalization services to semantic Web-applications of different domains (tourism, movies, books…); the recommender makes use of semantically-enhanced algorithms to

[2] See [http://xmlns.com/foaf/spec/], visited 10 May 2010.

enhance both stages of the personalization process, the *user modeling* and the *content adaptation*.

2.1. Architectural Design

In order to develop a domain-independent recommender, it is necessary to decouple the recommendation engine from the application domains. For this reason, we designed the system as a service provider following the well-known *service oriented architecture* (SOA) paradigm.

In **Figure 1**, the abstract architectural design is represented. Using this decoupled design, each Web-application or domain has to expose a list of items to be used in the personalization process; items have to be semantically annotated using the hierarchically structured concepts of the domain ontology, which is shared with the recommender. Thus, the recommendation engine can work as a personalization service, providing methods to generate personalized recommendations as well as to collect user feedback while users interact with Web-applications.

In order to facilitate the reuse of user profiles as well as the authentication process we employ the widely used FOAF vocabulary as the basis of our ontologically extended user profiles, which is compatible with the *OpenID* authentication[3]. This method is a decentralized standard that allows a portable identity for users across the Web.

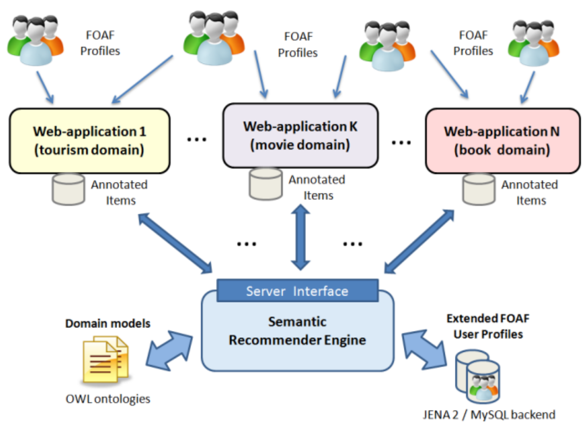


Figure 1. General architecture design

2.2. The User Modeling Process

The objective of the user modeling process is to accurately model long-term user-interests. The recommender employs a *weighted overlay* approach to model user's interests that consists of mapping collected feedback about semantically annotated items to the concepts of the domain ontology; the association is done with a weight, which indicates the degree of interest (*DOI_weight*) of the user. In combination with the weight, we use a measure of how trustworthy the interest prediction of the

---

[3] See [http://openid.net/], visited 10 May 2010.

particular concept (*DOI_confidence*) is. The new concept associations representing the user's interests are stored in the corresponding FOAF-based user profiles.

Because Web-applications of different domains can make use of the system, the recommender has to be able to learn user's interests from a variety of information sources. For this reason, the system's user modeling approach employs a hybridization of user-information collection-techniques. Consequently, the *DOI* values for a particular concept can be inferred from multiple sources of feedback, each one with its own weight prediction and confidence values. In particular, our system uses three types of interest-prediction methods (A more detailed explanation of how the methods are implemented can be found in [1].):

- based on explicit feedback, such as user-ratings and user own predictions;
- based on implicit feedback, using user's browsing-events related with the concept, such as user queries and item selections;
- based on domain generalizations, such as stereotypes and domain inferences.

Once a user starts interacting with the personalization service, the user-profile learning algorithm is responsible for expanding and maintaining the user's interests up-to-date. The learning algorithm is executed offline each time a user logs out of the system and consists of two main steps (see [1] for more details about the learning algorithm):

1. The system processes all user implicit and explicit feedback collected during the session and updates the interest-predictions of the related concepts by using the prediction methods mentioned in Codina (see [1], pag. 33).
2. The system groups the updated interests by level of concept-depth in the hierarchy and, starting from the deepest concepts, it carries out the following actions for each group:
   2.1. It recalculates the *DOI* values for each concept using a weighted linear combination (see [1], pag. 37) of the available interest-predictions, according to how trustworthy each prediction is and priority rules defining an order of relevance among the different types of interest-prediction methods based on previous experimentation. (E.g., predictions based on domain generalizations are considered less relevant than those based on implicit or explicit feedback.)
   2.2. It applies the *domain-based inference* method (see section 2.4.1) in the subfamilies of concepts that have some updated interest. The inferred predictions using this method are applied only when no interest-prediction based on explicit or implicit feedback exists about the concept.

2.3. The Content Adaptation Process

The recommender makes use of semantics to generate recommendations following a CB approach. In order to allow the recommender to make use of semantics, item descriptions have to be based on semantic annotations referring to concrete ontology concepts. These concepts should be hierarchically structured by feature type. Thus, the root node of each hierarchy defines the type of a feature.

The algorithm used to calculate the matching score between an item 'I' and a user profile 'U' consists of calculating a weighted average of the individual matching scores of the concepts annotating the item, the *iConcepts(I)*, according to the relevance of their feature types (expressed as *REL(j)*), as can be observed in Eq. (1). The relevance of each feature type (root node of concept taxonomy) is previously computed for each user using an ad-hoc feature-weighting method based on the explicit feedback on recommended items.

$$ItemScore(I,U) = \frac{\sum_j^{iConcepts(I)} ConceptScore(j,U) * REL(j)}{\sum_j^{iConcepts(I)} REL(j)} \quad (1)$$

The matching score of each concept 'j' is calculated as the weighted average of *DOI_weight* of the user's interests belonging to 'U' that match the concept 'j' according to their semantic similarity (expressed as *SIM(i,j)* and explained in more detail in section 2.4.2) and their *DOI_confidence* values, as can be observed in Eq. (2). We consider that an interest 'i' of 'U' matches the concept 'j' if either 'i' is the same concept as 'j' or 'i' is an ancestor of 'j'. *mConcepts(j,U)* is the set of interests of 'U' matching the concept 'j'. Note that we use only the ancestors of the concept as possible matching concepts, being the objective of the algorithm to enrich the specific interest-predictions with contextual information about the same topic. This does not exclude computing the semantic similarity using other relations (which we will research in the future).

$$ConceptScore(j,U) = \frac{\sum_i^{mConcepts(j,U)} \mathrm{DOI_{weight(i)}} * \mathrm{DOI_{confidence(i)}} * \mathrm{SIM(i,j)}}{\sum_i^{mConcepts(j,U)} \mathrm{DOI_{confidence(i)}} * \mathrm{SIM(i,j)}} \quad (2)$$

The recommendation algorithm can be used to obtain the matching score of a single item for a particular user as well as to obtain a *top-n* recommendation, i.e. a list of *n* items ranked in descent order by matching score.

2.4. Reasoning Methods

The recommender makes use of semantically-enhanced algorithms in the two stages of the personalization process: the user-profile learning algorithm employs a *domain-based inference* method in combination with other relevance feedback methods to populate more quickly the user profile; the recommendation algorithm makes use of a *semantic similarity* method based on the hierarchical structure of the ontology to refine the item-user matching score calculation, expressed as *ItemScore(I,U)* in the previous section. In the following subsections we present these two methods in more detail.

2.4.1. The Domain-based Inference Method

The domain-based inference method we used is an adaptation of the approach presented in [10] and consists of inferring the degree of interest in a concept using subclass or sibling relations (upward or sideward propagation) when the user is also interested in a minimum percentage (the inference threshold) of direct sub-concepts or sibling concepts (see example in **Figure 2**). In general, different thresholds are used for each type of inference, being them stricter in the case of sideward propagation.

The predicted weight is calculated as the weighted (according to the *DOI_confidence* values) average of the *DOI_weight* of the sub-concepts or sibling concepts the user is interested in. The confidence value of a prediction P, *ConfidenceValue(P)*, is computed as the average of two parameters, as can be observed in Eq. (3). The first parameter represents the difference between the percentage of sub-concepts used in the inference, *iSubConcepts(P)*, and the inference threshold (*IT*). The second parameter is the *DOI_confidence* average of the sub-concepts.

$$ConfidenceValue(P) = \frac{\frac{pct(iSubConcepts(P))-IT}{1-IT} + \frac{\sum_{j}^{iSubConcepts(P)} \mathrm{DOI}_{\mathrm{confidence(j)}}}{\#iSubConcepts(P)}}{2} \qquad (3)$$

In **Figure 2**, we present an example of how this method works. The user is interested in four sub-concepts of the *Sport* class (Baseball, Basketball, Football and Tennis). The proportion of sub-concepts the user is interested in (4 out of 5, i.e., 0.8) is greater than both inference thresholds, therefore both can be applied. Thus, the system infers that the user is interested in *Sport* and *Golf* with the same predicted weight (0.62). The difference between the two types of inference is that the confidence value of sideward-propagation is lower than the one of upward-propagation (0.50 vs. 0.66) because the sideward inference-threshold is greater (0.75 vs. 0.60).

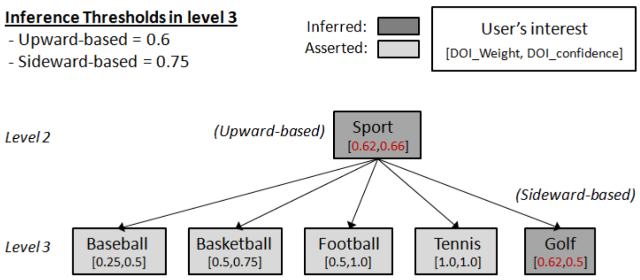


Figure 2. An example of how new interests are inferred

2.4.2. The Semantic Similarity Method

The basic idea of this method is to measure the relevance of the matching between a particular concept the user is interested in and a concept annotating the item. (In **Figure 3**, two examples are shown, in which the user's interest is the parent of the item concept.) We can distinguish two types of matching:

1. The item concept is one of the user's interests, so the matching is perfect.
2. An ancestor of the item concept (e.g., the direct parent) is one of the user's interests. In this case the similarity is calculated using the recursive function of Eq. (4) whose result is always a positive real number (lower than 1).

$$\mathrm{SIM}_n = \mathrm{SIM}_{n-1} - \mathrm{K} * \mathrm{SIM}_{n-1} * \mathrm{n}; \qquad \mathrm{SIM}_0 = 1; \qquad (4)$$

Where:

- 'n' is the distance between the item concept and the user's interest (e.g., when it is the direct parent, n = 1);
- K is the factor that marks the rate at which the similarity decreases (the higher 'n', the higher the decrement). K is calculated using the depth of the item

concept in the hierarchy, following the assumption that semantic differences among upper-level concepts are bigger than among lower-level concepts. As this measure depends on the density and maximum depth of the taxonomy, the calculation of K has to be adjusted to the concept taxonomy. In our experiments we empirically adjusted it (see **Table 2**), but automatic methods may be researched for it based on information-theoretic approaches [11][12].

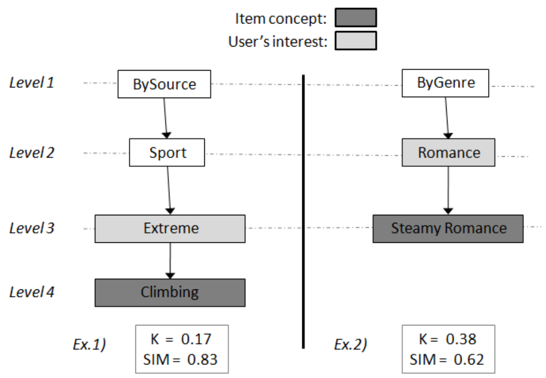


Figure 3. How the similarity method works

In **Figure 3**, we present two examples showing how the decreasing factor K influences the similarity of the matching. Both examples represent partial matches in which the user is interested in the parent of the item concept (n = 1). As can be observed, the K value of the deeper-concept example (Ex. 1) is lower (0.17 vs. 0.38); therefore the resulting similarity is higher (0.83 vs. 0.62).

3. Results and Evaluation

In this section the undertaken experimental evaluation of the recommender is presented. The main goal of the experiments is to demonstrate how the recommendation's quality is improved when semantically-enhanced algorithms, presented in the previous section, are employed.

3.1. Method and Dataset Specifications

To evaluate recommendation's quality, we decided to employ the well-known Netflix-prize movie dataset. This decision was made because of its popularity among the research community in the last years as well as the very large amount of data it contains. The Netflix dataset consists of 480,000 users, 17,700 movies and a total of 100,480,507 user's ratings ranging between 1 and 5. A strong limitation of this dataset is that only rating feedback is provided as user information, which implies the learned user-profiles can only be based on explicit feedback. In the experiments, we employ, to evaluate the quality of the rating prediction, the same metrics used in the Netflix contest: the *root mean square error* (RMSE). This error calculates the difference between the real rating of the user and the predicted one.

3.2. Experiments and Results

To evaluate how semantically-enhanced algorithms contribute to the improvement of the recommendation's quality in terms of accuracy, in a first experiment we compare the prediction results obtained from a configuration using semantics with the ones of a traditional CB approach; in a second experiment we compare two configurations using the same semantically-enhanced algorithms but with different concept taxonomies.

3.2.1. Comparing Prediction Accuracy

In a *first experiment* we compare the prediction results executing the recommender in two different configurations:

CB. It represents the traditional CB approach; therefore the methods that take advantage of the ontology representation are disabled. In this case, the item-user matching only takes into account the concepts that perfectly match and therefore works as a pure keyword-based approach.

Sem-CB. It represents the CB approach enhanced with the algorithms presented in section 2 using, as domain ontology and movie indexation, the same taxonomy of three levels of depth used by Netflix and publicly available[4.]

The error of the predictions generated by the system (see **Table 1**) demonstrates that, when semantics is used, the recommendation's accuracy improves with respect to the *CB* configuration.

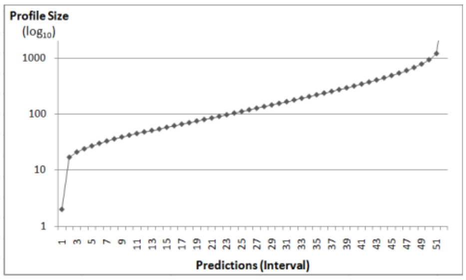
Table 1. Global prediction-error (RMSE) results

Configuration	RMSE
CB	1.060
Sem-CB	1.039

To better understand these results, we compare the prediction-RMSE results of *Sem-CB* and *CB* configurations based on the user profile-size, which is calculated as the *total number of user's ratings*. The test dataset contains 2,817,131 predictions made by the system for each configuration. As can be observed in **Figure 4**, these predictions are grouped into 51 intervals (each one of approximately 55,000 predictions) according to the size of the profile of the user for whom they were made. The RMSE of the two configurations calculated per each interval are presented in **Figure 5**. As can be observed, *Sem-CB* is better than *CB* for all intervals, but the improvement is more notable in intervals 2 to 16 (corresponding to users with profile-size between 17 and 65). This means that semantically-enhanced algorithms are more effective when there is scarcity of user data.

Another point of view from which the results are analyzed is taking into account the number of item concepts that perfectly match user's interests when the recommender makes a prediction. In **Figure 6**, the proportion of predictions for each case is shown. In **Figure 7**, the RMSE of the two configurations is compared for each case of matching. The results again show that the *Sem-CB* approach is always more accurate, above all when there are few perfectly matching concepts (from 1 to 3).

[4] See [http://www.netflix.com/AllGenresList], visited 10 May 2010.



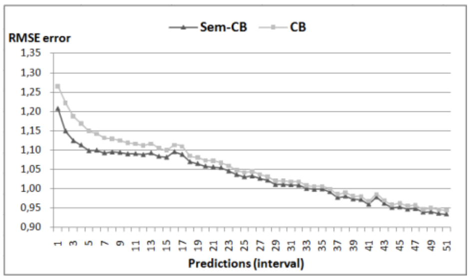
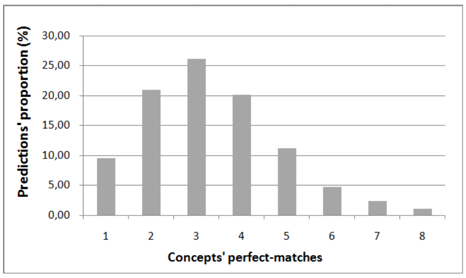


Figure 4. Predictions grouped by profile size **Figure 5**. Comparing errors by profile size



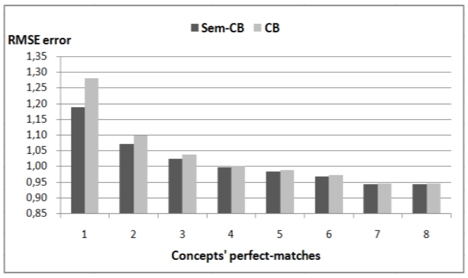


Figure 6. Predictions grouped by concept matches **Figure 7**. Comparing errors by concept matches

In a *second experiment* we compare the results of the *Sem-CB* configuration (used in previous experiment) with another one (*Sem-CB+*), which uses the same algorithms but a different movie domain ontology. In particular, we use an adaptation of the Netflix taxonomy, with a concepts hierarchy of four levels of depth and four features. The indexation of the movies was adapted to the new taxonomy.

In **Table 2**, the results of both configurations are compared using different executions that have been divided in two phases: the *training phase*, which is when the system executes the user-profile learning algorithm, and the *test phase*, which is when the system generates the rating predictions. According to the RMSE results, the accuracy of *Sem-CB+* is better than *Sem-CB*. Taking into account the columns showing the average upward- and sideward-propagations per user, the improvement of accuracy is strongly related with the upward-inference threshold (the higher the number of upward-propagations, the better the RMSE results, e.g. *Sem-CB+*: 1.044–1.042–1.039.

Table 2. Comparison of semantics-based configurations.

	Config.	**Training Phase execution**			**Test Phase execution**	
		Inference thresholds (*Upward - Sideward*)	**Avg. Upward propagations**	**Avg. Sideward propagations**	**K factor per level (K_{level})**	**RMSE**
Ex.1	*Sem-CB*	0.60 – 0.75	4.32	2.87	K_2=0.3; K_3=0.1	1.048
	Sem-CB+		6.01	3.83	K_2=0.5;K_3=0.4;K_4=0.3	1.044
Ex.2	*Sem-CB*	0.40 – 0.75	8.89	3.85	K_2=0.3; K_3=0.1	1.044
	Sem-CB+		9.99	3.89	K_2=0.5;K_3=0.4;K_4=0.3	1.042
Ex.3	*Sem-CB*	0.20 – 0.85	13.84	2.88	K_2=0.3; K_3=0.1	1.039
	Sem-CB+		17.73	3.30	K_2=0.5;K_3=0.4;K_4=0.3	1.039

4. Conclusions

We presented two algorithm configurations for recommending items based on an existing and a new ontology for the movie domain. Incorporating semantics into a content-based system can increase the flexibility and quality of the recommender. In our approach, a domain-based method makes inferences about user's interests and a semantic similarity method is used to refine the item-user matching algorithm. Results show that there is an improvement in accuracy and that it is more notable in recommendation for users with relatively small profiles.

The recommender proposed is domain-independent, is implemented as a Web service, and employs a hybridization of user-information collection-techniques to obtain information on user's interests. The use of a FOAF-based user-model linked with concepts of domain ontologies allows an easy integration of the recommender into Web-applications in any domain.

We believe that the recommendation system presented can further reduce the prediction error if the movie descriptions are complemented using other information sources, like the *Internet Movie Database* (IMDb), since approximately ten percent of the movie descriptions provided by Netflix are incomplete or very simplistic, in the sense that only some abstract concept is used to annotate the movie, and this strongly limits the quality that can be reached by our approach.

References

[1] Codina, V. 2009 Design, development and deployment of an intelligent, personalized recommendation system. Master Thesis. Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya. 101 pp.

[2] Berners-Lee T., J. Hendler, and O. Lassila. 2001. The Semantic Web. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American* 284 (5), 34-43.

[3] Codina, V. and Ceccaroni, L. 2010 A recommendation system for the semantic web. To be published in *Proc. of International Symposium on Distributed Computing and Artificial Intelligence 2010 (DCAI'10)*.

[4] Cantador, I., Bellogín, A., and Castells, P. 2008. Ontology-based personalised and context-aware recommendations of news items. *Proc. of IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 562–565.

[5] Sieg, A., Mobasher, B., Burke, R. 2007. Ontological user profiles for personalized Web search. *AAAI Workshop - Technical Report WS-07-08*, 84-91.

[6] Liu, P., Guihua, N., and Chen, D. 2007. Exploiting Semantic Descriptions of Products and User Profiles for Recommender Systems. *Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007)*, 179-185.

[7] Maidel, V., Shoval, P., Shapira, B., Taieb-Maimon, M. 2008. Evaluation of an ontology-content based filtering method for a personalized newspaper. *RecSys'08: Proceedings of the 2008*, 91-98.

[8] Celma, O. and Serra, X. 2008. FOAFing the music: Bridging the semantic gap in music recommendation. *Web Semantics: Science, Services and Agents on the World Wide Web* 6 (4), 250-256.

[9] Blanco-Fernández, Y. et al. 2008. A flexible semantic inference methodology to reason about user preferences in knowledge-based recommender systems. *Knowledge-Based Systems* 21 (4), 305-320.

[10] Fink, J. and Kobsa, A. 2002. User Modeling for Personalized City Tours. *Artificial Intelligence Review* 18 (1), 33-74.

[11] 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.

[12] Li, Y., Bandar, Z.A., McLean, D. 2003. An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources. *IEEE Transactions on Knowledge and Data Engineering* 15 (4), 871 - 882