

TMR: A Semantic Recommender System using Topic Maps on the Items' Descriptions

Angel L. Garrido and Sergio Ilarri

IIS Department, University of Zaragoza, Spain

Abstract. Recommendation systems have become increasingly popular these days. Their utility has been proved to filter and to suggest items archived at web sites to the users. Even though recommendation systems have been developed for the past two decades, existing recommenders are still inadequate to achieve their objectives and must be enhanced to generate appealing personalized recommendations effectively. In this paper we present TMR, a context-independent tool based on topic maps that works with item's descriptions and reviews to provide suitable recommendations to users. TMR takes advantage of lexical and semantic resources to infer users' preferences and thus the recommender is not restricted by the syntactic constraints imposed on some existing recommenders. We have verified the correctness of TMR using a popular benchmark dataset.

Keywords: topic maps, recommendation systems, NLP, semantics

1 Introduction

Machine learning, information retrieval, data mining, natural language processing, and probabilistic models have been adopted for developing systems that recommend items like books, songs, and movies, for example. Our proposed system, TMR (Topic Map Recommender), is a semantic, ontological, and linguistic enhanced recommendation system, which takes advantage of natural language processing (NLP) and semantic tools to provide personalized item suggestions tailored to the preferences of individual users. Unlike its counterparts, TMR examines the "meaning" of textual item metadata, such as content descriptions and reviews on items to be recommended, considered during the recommendation process, as opposed to simply syntactically analyse the words in the texts. So, it is expected to lead to more suitable recommendations.

There are already some semantic and ontological approaches such as [2, 5]. TMR differs from them in the way the system generates abstractions of themes and subject areas from items and user profiles. For this purpose, the system uses topic maps, a kind of diagram that shows relationships between concepts within a context. As a representation of a conceptualization corresponds to the definition of an ontology [4], we can use techniques and methodologies from ontological engineering to model these representations and work with them. Furthermore, unlike its ontology-based counterparts, TMR does not depend on the availability of a domain ontology, since it is not domain-dependent: ontologies in the form of topic maps are automatically built by the system.

2 Our Proposed Methodology

The main idea is to represent both the user’s likes and dislikes and the items. We will use topic maps to represent all this information, and we will compare the corresponding topic maps in order to evaluate the degree of similarity between the likes/dislikes of the user and the items. The more similar the representation of an element is with respect to the representation of the profile of what a user likes / dislikes, the more likely we are to recommend it / not to recommend it.

We use text descriptions of the items (the items that can be recommended to the user, as well as the items that the user valued positively –likes– and negatively –dislikes–) and other user’s reviews. All this information is contained in natural language texts, so we need an information extraction tool to exploit it. To obtain the relevant data in order to build the topic maps from text, TMR adopts TM-Gen [3], which is a tool that extracts information from any number of texts and represents them in a topic map format.

TM-Gen scans the texts to find the most important keywords and the main named entities [13]. It divides the text into sentences and assigns them a relevance score, in order to find those that are most important in the text. Afterwards, TM-Gen analyzes syntactically the sentences to find the best candidates to be a topic, and then it establishes associations between them, creating the relations. We have adapted this method to analyse the items’ descriptions in TMR.

TMR examines the descriptions using Freeling¹, an NLP tool. The system then proceeds to extract concepts and the corresponding relationships among them using the aforementioned techniques from TM-Gen. The different topic maps obtained are merged into a single one using SIM (Subject Identity Measure) [8], an existing approach that describes the relationships among two subjects or topics. As part of the topic map generation process, TMR performs a semantic analysis of the topic map and simplifies it if the system finds redundancies, incompatible associations, or ambiguities, using for this purpose lexical databases (i.e., WordNet), Linked Data resources like DBPedia, and a disambiguation engine [10] (similar to that used in [11]).

TMR analyzes also each item review to find relevant information, which is used to enrich the topic map of the item. As the language used in the reviews is usually much less formal than the one employed in item descriptions, it is more difficult to use parsers to extract information. For this reason, TMR lemmatizes the texts in the reviews and extracts the most frequent keywords and named entities using the well-known TF-IDF algorithm [12]. These extracted keywords and named entities are incorporated into the topic map as new elements either as topics or as relationships, by using Freeling’s morphological analyzer.

The next step in the TMR’s recommendation process is to construct a profile of the user which captures his/her preferences, by examining the ratings that he/she has previously assigned to other items. In doing so, TMR generates two different topic maps: one for the likes (*TMlikes*), and another one for the dislikes

¹ <http://nlp.lsi.upc.edu/freeling/>

(*TMdislikes*). The texts used to build those topic maps are the ones describing the corresponding data items.

The last step applied by TMR in making suggestions involves predicting the degree to which a user will like (or not) a new item. TMR evaluates the degree of similarity between the topic map of an item and each of the topic maps that capture the likes and dislikes of the user. To calculate the similarity between topic maps, TMR employs an algorithm we developed that evaluates the resemblance between the topics of any two topic maps. This algorithm is based on two measures introduced in [7]: *lexical similarity* and *relation overlap*; while the first measure calculates the lexical overlap between strings, the second one quantifies the degree to which the relations of two concepts in an ontology match. Using Equation 1, TMR yields a score for an item on a [1, 10] range.

$$Rate(Item) = Norm[(Sim(TMlikes) - Sim(TMdislikes))] \quad (1)$$

where *Sim* captures the degree of similarity between the corresponding topic map of likes and dislikes and the one corresponding to the item, and *Norm* is a function that maps the differences in similarity scores from a [-1, 1] range to a [1, 10] range.

3 Experiments

To evaluate the performance of TMR, we have used the BookCrossing dataset as a test case. BookCrossing is a popular benchmark dataset commonly-used to assess the performance of book recommendation systems. We apply the popular five-fold cross validation protocol. For each one of the five repetitions, 85% of the books rated by a user U in a set of users BX were used to model U 's likes/dislikes (i.e., U_{train}) and the remaining 15% (U_{test}) were used for actual testing.

In our empirical study, we quantified the performance of a recommender system R using the Root Mean Squared Error (RMSE), as shown in Equation 2, which is a de facto metric for evaluating predictive recommendation systems.

$$RMSE(R) = \frac{\sum_{U \in BX} \sqrt{\frac{\sum_{b \in U_{test}} |R_{U,b} - r_{U,b}|}{|U_{test}|}}}{|BX|} \quad (2)$$

where $R_{U,b}$ denotes the rating *predicted* by R for a book b ($\in U_{test}$) given the corresponding user U , and $r_{U,b}$ is the *actual* rating given to b by U .

We executed each experiment five times, and the overall RMSE score is the average of the RMSE scores computed for each repetition. In our experiments, the RMSE score generated using TMR is 1.25. Its performance, in terms of RMSE, is much higher than some baseline recommenders like SVD++ [6] (4.67) and Bias-SVD [6] (3.94). If we compare TMR with other state-of-the-art recommenders like fLDA [1] (1.31), RLmf [1] (1.32), and uLDA [1] (1.35), we find that our results are very promising, given the significant difference obtained with respect to its counterparts.

4 Conclusions and Future Work

In this paper we have presented TMR, a domain-independent recommender that combines semantic and ontological techniques with NLP tools and lexical resources to make recommendations suitable to the preferences/interests of each individual user. In principle, TMR can work in any context where a textual description and textual reviews of the data items are available. We conducted an empirical study with the BookCrossing dataset and obtained positive results.

Our intention now is to verify the generality of our solution. For this purpose, we will evaluate the performance of TMR using other datasets to prove that our system is indeed context-independent. Comparing the proposal with other domain-specific recommenders in different contexts is also a relevant task of future work, as we can expect a trade-off between the generality of the proposal and its performance, that needs to be quantified.

Acknowledgment

Supported by CICYT project TIN2010-21387-C02-02 and DGA-FSE.

References

1. D. Agarwal and B. Chen. *fLDA: Matrix Factorization through Latent Dirichlet Allocation*. ACM WSDM, pp. 91-100, 2010.
2. I. Cantador, A. Bellogin, and P.Castells. *A Multilayer Ontology-based Hybrid Recommendation Model*. AI Communications, 21(2-3):203-210, 2008.
3. A. L. Garrido, M. G. Buey, S. Escudero, S. Ilarri, E. Mena, and S. Silveira. *TM-Gen: A Topic Map Generator from Text Documents*. IEEE ICTAI, pp.735-740, 2013.
4. T. Gruber. *A Translation Approach to Portable Ontology Specifications*. Knowledge Acquisition, 5(2):199-220. Academic Press Ltd. 1993.
5. W. IJntema, F. Goossen, F. Frasinca, and F. Hogenboom. *Ontology-based News Recommendation*. In ACM EDBT/ICDT Workshops, Article 16, 6 pages, 2010.
6. Y. Koren. *Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model*. ACM SIGKDD, pp. 426-434, 2008.
7. A. Maedche and S. Staab. *Measuring Similarity Between Ontologies*. Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web. Springer, pp. 251-263, 2002.
8. L. Maicher and H. Witschel. *Merging of Distributed Topic Maps based on the Subject Identity Measure (SIM) approach*. Berliner XML-Tage, 4:301-307, 2004.
9. G. Miller. *WordNet: A Lexical Database for English*. Communications of the ACM, 38(11):39-41, 1995.
10. R. Navigli. *Word Sense Disambiguation: A Survey*, ACM Computing Surveys, 41(2), 10:1-10:69, 2009.
11. M. G. Buey, A. L. Garrido, S. Escudero, S. Ilarri, R. Trillo, E. Mena *SQX-Lib: Developing a Semantic Query Expansion System in a Media Group*, ECIR. Springer, pp. 780-783, 2014.
12. G. Salton and C. Buckley. *Term-weighting Approaches in Automatic Text Retrieval*. Information Processing and Management, 24(5):513-523, 1988.
13. S. Sekine. and E. Ranchhod. *Named Entities: Recognition, Classification and Use*. John Benjamins. 2009.