ESWC-14 Challenge : Linked Open Data-enabled Recommender Systems Participation in the Task 3 : Diversity

Diego Antognini and Hatem Ghorbel

Haute Ecole Arc Ingénerie,
Espace de l'Europe 11, 2000 Neuchâtel, Switzerland
{diego.antognini,hatem.ghorbel}@he-arc.ch
http://ingenierie.he-arc.ch/

Abstract. In this article, we focus on the challenge which concerns Top-N books recommendation with diversity. Books are described using Linked Open Data formalism. As a recommendation approach we propose a hybrid recommender system based on a feature-weight user model which mixes collaborative filtering and content-based approaches. Moreover, we present a feature extraction system which takes into account the abstract, title, author, genre and subject for each book and attribute to these features a specific weight from the perspective of tf-idf approach. The evaluation of the performance are structured in 2 categories : the diversity and the F-measure. For the first one, we have achieved relatively good results; however, for the F-measure we have achieved low performance. For the latter, we plan to improve results by adjusting the feature extraction process in the future work.

Keywords: Hybrid Recommender System, Linked Open Data, Top-N Recommendations, Diversity, Semantic Web, Web of Data.

1 Introduction

Each year since 2005, the *European Semantic Web Conference* (ESWC) proposes some challenges around *Semantic Web*. We focus on the challenge which concerns Top-N books recommendation with diversity, the third task proposed this year. With a specified list of books and user ratings, we have to compute the Top-20 recommended items for every user while taking into account the diversity of the items.

Classical approaches attempt to predict users interests by using their rating history [1] and usually rely on either collaborative filtering [2] or content-based filtering approaches [3]. Collaborative filtering approaches recommend items based on the users past behavior as well as other users choices who purchased similar items (e.g., Amazon and eBay). Content-based filtering approaches utilize a series of discrete characteristics of an item in order to recommend additional items with similar properties.

In our work, we combine content-based and collaborative filtering. The items we have to recommend are books, available in the *Linked Open Data* (LOD) cloud as DBpedia[1], the British Library Bibliography or the Library of Congress. The *Web of Data* (WOD) which oftens referred as *Semantic Web* or *Linked Open Data*, contains billions RDF[2] triples and covers a large number of domains such books, movies etc. All of them are available with unique URIs related by conceptual relations. And finally we have access to a huge graph of data. In our case, we only focus on DBpedia and we extract data with SPARQL[3] queries.

In this paper, we present our proposal for the features extraction with books which exploits LOD datasets in DBpedia. This includes the fields we are interested in, the ontologies we use and the weight attributed (which can be fixed or computed) for each feature. Moreover, we present a hybrid recommender system based on a feature-weight user model [5] which mixes content-based and collaborative filtering in order to have a maximum of diversity while having items which might interest the most of the users.

2 Book Analysis

The book analysis is one of the most important part because we have to describe each book with its own most representative set of features. If a book has a small set of features or a set of non-relevant features, it will be difficult to recommend it correctly, especially if we are in a pure content-based recommender system. The idea is to have a certain number of features which are categorized in different fields (e.g. author, subject, ...).

2.1 Feature Extraction

To have a full description of books, we use DBpedia website which describes 24.9 million things (books, movies, personality etc) and contains 2.46 billion pieces of information (RDF triples). To access this data, we use SPAQRL queries with which we can specify the desired attributes with the corresponding ontologies.

The different textual fields we focus on are the title, abstract, author, subject and genre, in differents ontologies (dbpedia-owl, rdfs, dbpprop, foaf, dcterms) aiming at having a very large information set. This data can be splitted into 2 sets : the first one which needs only word lemmatization, it concerns the author, subject, title and genre. The second one (abstract) needs a more advanced analysis because not all of the lemmas will have the same importance. The language of the data we refer to is essentially English.

The lemmatization process (which is used for both sets) is quite simple. It consists on defining the canonical form of a given word, for example the infinitive form for a verb. First, we focus on nouns only, proper nouns and 4 digits numbers. Proper nouns can represent a location, a character name (firstname

[1] `http://dbpedia.org`
[2] `http://www.w3.org/RDF`
[3] `http://www.w3.org/TR/rdf-sparql-query`

and lastname) or a specific moment of the history. The 4 digits numbers is essentially to detect a specific year and finally, the usage of the noun is trivial. Both sets are lemmatized and only the weight attribution is different.

2.2 Weight attribution

For the first category of data (author, title, subject, name), we have defined fixed weights for their respective features. This strategy allows us to highlight specific kind of features.

The second has a more advanced definition of the weight based on a tf-idf (Term Frequency-Inverse Document Frequency) approach. Tf-idf is a statistical method used in information retrieval [4] which allows to calculate the importance of a word inside a specific document within a corpus. It is composed of two parts: the frequency calculs of a term and this for the inverse document frequency. Finally, the tf-idf is the multiplication between those intermediate results. The corpus is composed of all book's abstracts. Then we compute the tf-idf for each terms in the book's abstract and project it in a vector space model.

3 Recommendation

The purpose of the recommendation is to find the fittest books for each user and have a maximum of diversity so to avoid over-specialization. We use a hybrid recommender system using collaborative filtering and content-based. It is based on a feature-weight user model proposed by Panagiotis et al. [5]. The recommendation is decomposed in 5 steps : 1) to create the basic matrices for the user's ratings, and book's features. 2) to compute the matrix user-feature. 3) to realize a tf-idf approach and to compute the weighted user-feature matrix. 4) To apply the cosine similarity and finally, 5) To find the Top-N recommended books. Let's first define $\mathcal{U}$ the set of users, $\mathcal{B}$ the set of books and$\mathcal{F}$ the set of features.

1) We define the rating matrix R of dimension $|\mathcal{U}| \times |\mathcal{B}| : \forall(u,b), u \in \mathcal{U}, b \in \mathcal{B}, R(u,b) \in \{-1,0,1\}$. $R(u,b)$ is the rating value of user u for book b, -1 is an unrated book and 0,1 the rating where the highest value is 1. We also define the feature-book matrice F of dimension $|\mathcal{B}| \times |\mathcal{F}| : \forall(b,f), b \in \mathcal{B}, f \in \mathcal{F}, F(b,f) \in \mathbb{R}^+$. $F(b,f)$ denotes the weight of feature f for book b. If $F(b,f) = 0$, the feature doesn't describe the book.

2) We construct a content-based user profile P which represents the matrix user-feature, of dimension $|\mathcal{U}| \times |\mathcal{F}|$. We take only into accounts all ratings with a minimum value. $P(u,f)$ denotes the correlation between a user u and a feature f. Those values are computed with the Equation 1 [5] :

$$P(u,f) = \sum_{\forall R(u,b) > P_t} F(b,f) \tag{1}$$

where $u \in \mathcal{U}, b \in \mathcal{B}$ and $P_t \in \mathbb{R}^+$.

3) The purpose is to weight the features of matrix P and find the features which better describe a user u and those which better distinguish him from the

others. We define the frequency-features matrix FF which is equivalent in our case to the matrix P. FF represents the number of time in which a feature f occurs in the profile of user u. We compute then the user-frequency matrix UF of dimension $1 \times |\mathcal{F}|$ where $UF(f)$ is the number of users in which the feature f occurs at least once. Then we compute the inverse user frequency matrix IUF with the Equation 2 [5] :

$$IUF(f) = \log_{10} \frac{|\mathcal{U}|}{UF(f)} \tag{2}$$

where $|\mathcal{U}|$ is the total number of users and $f \in \mathcal{F}$.

Finally, we can compute the weighted matrix (which is the tf-idf) W of dimension $|\mathcal{U}| \times |\mathcal{F}|$. $W(u, f)$ denotes the importance of a feature f for a user u. If the feature f occurs in many users' profiles, it will be considered less important than those which occurs rarely. We compute the matrix W with the Equation 3 [5] :

$$W(u, f) = FF(u, f) * IUF(f) \tag{3}$$

where $u \in \mathcal{U}, f \in \mathcal{F}$.

4) We realize now the collaborative filtering part where we have to find the similar users. We use the cosine similarity in the weighted user-feature W matrix. We take into account only related features between 2 users. We compute the user-user matrix U of dimension $|\mathcal{U}| \times |\mathcal{U}|$ with the Equation 4 [5] :

$$UU(u, v) = \frac{\sum\limits_{\forall f \in \mathcal{X}} W(u, f)W(v, f)}{\sqrt{\sum\limits_{\forall f \in \mathcal{X}} W(u, f)^2}\sqrt{\sum\limits_{\forall f \in \mathcal{X}} W(v, f)^2}}, \mathcal{X} = \mathcal{F}_u \cap \mathcal{F}_v \tag{4}$$

where $\mathcal{F}_u = \{f \in \mathcal{F} | P(u, f) > 0\}$, $\mathcal{F}_v = \{f \in \mathcal{F} | P(v, f) > 0\}$ and $u, v \in \mathcal{U}$.

5) The last step consists in generating the Top-N recommended books. First we have to define the size of the neighborhood for the collaborative part, $k \in \mathbb{N}^*$. We define the nearest neighbors of user u. Then we fetch the books in the neighborhood where the user u hasn't rated it yet. Afterwards we get the features of each book and we compute their frequency in the neighborrhood. Finally for each books, we compute its cumulative feature weight and the k highest will be the recommended books.

4 Performance study

In this section, we study the performance of our hybrid recommender system for the ESWC challenge. In our case $|\mathcal{U}| = 6181, |\mathcal{B}| = 8170$. Matrix R contains 72732 ratings for 6181 users and 6733 rated books. After the feature extraction, we have $|\mathcal{F}| = 29810$ unique features. The matrix F contains 269374 relations between a feature and a book. We define the threshold value for the rating $P_t = -1$ (so we take into account all rated books) and the size of the neighborhood

$k = 300$. The weight values of the title, author, genre and subject was set to 1.0 and for the abstract, a factor 1 to the weight of each tf-idf.

With those values, we obtain our best score with the precision (P@20) at 0.0115, the recall (R@20) at 0.0321, the f-mesaure(F1) at 0.017 and finally the intra-list diversity (IDL) at 0.4801[4].

The date where the challenge was supposed to end, 07th March 2014 23:59 CET, on 11 participants, for the diversity ranking, we were third and for the F-measure ranking, we were seventh. The final ranking was the average between the both positions and finally, we ended up fourth.

5 Conclusion

We proposed a feature extraction and adapted a hybrid user model recommender systems as defined in [5]. We perform different versions of our proposal to the challenge and we can see that for the diversity, we have a good ranking contrary to the F-measure. To improve our score, it will be necessary to change the factors for the different weights in order to highlight certain kind of features (e.g. the author more than the title). Moreover, we can change the size of the neighborhood. However, we cannot change the value of P_t to 0 because some users will not have ratings and it would be impossible to recommend correctly some books for them. Another alternative for improvement would be to enrich the extracted data from DBpedia by including or excluding other textual descriptions.

References

1. Adomavicius, Gediminas and Tuzhilin, Alexander : Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. In: IEEE Trans. on Knowl. and Data Eng, pp.734–749. (2005), `http://dx.doi.org/10.1109/TKDE.2005.99`
2. R. Keshavan, A. Motanari and S.Oh : Matrix completion from noisy entries. In: Journal of Machine Learning 11, pp. 2057–2078. (2010), `http://arxiv.org/pdf/0906.2027v2.pdf`
3. R. J. Mooney and L. Roy : Content-based book recommendation using learning for text categorization. In: Recom.Sys.: Algo. and Evaluation. (1999), `http://www.cs.utexas.edu/~ml/papers/libra-sigir-wkshp-99.pdf`
4. Gerard Salton and Michael J. McGill : Introduction to Modern Information Retrieval. (1986), `http://dl.acm.org/citation.cfm?id=576628`
5. Panagiotis Symeonidis, Alexandros Nanopoulos and Yannis Manolopoulos : Feature-weighted User Model for Recommender Systems. In: Proceeding UM '07 Proceedings of the 11th international conference on User Modeling, pp. 97–106. (2007), `http://dl.acm.org/citation.cfm?id=1419416.1419433`

[4] More details on how to calculate those values available on `http://sisinflab.poliba.it/semanticweb/lod/recsys/2014challenge/eswc2014-lodrecsys-metrics_evaluationservice.pdf`