Content Recommendation through Semantic Annotation of User Reviews and Linked Data

Iacopo Vagliano
ZBW - Leibniz Information Centre for Economics
Kiel, Germany
i.vagliano@zbw.eu

Diego Monti
Politecnico di Torino
Turin, Italy
diego.monti@polito.it

Ansgar Scherp
ZBW - Leibniz Information Centre for Economics
Kiel, Germany
a.scherp@zbw.eu

Maurizio Morisio
Politecnico di Torino
Turin, Italy
maurizio.morisio@polito.it

ABSTRACT

Nowadays, most recommender systems exploit user-provided ratings to infer their preferences. However, the growing popularity of social and e-commerce websites has encouraged users to also share comments and opinions through textual reviews. In this paper, we introduce a new recommendation approach which exploits the semantic annotation of user reviews to extract useful and non-trivial information about the items to recommend. It also relies on the knowledge freely available in the Web of Data, notably in DBpedia and Wikidata, to discover other resources connected with the annotated entities. We evaluated our approach in three domains, using both DBpedia and Wikidata. The results showed that our solution provides a better ranking than another recommendation method based on the Web of Data, while it improves in novelty with respect to traditional techniques based on ratings.

KEYWORDS

Recommender Systems, User Reviews, Semantic Annotation, Linked Data, Web of Data, Semantic Web, DBpedia, Wikidata

1 INTRODUCTION

The Web has evolved from an information space to share textual documents into a medium to distribute structured data. Linked Data[1] is a set of best practices for publishing and interlinking data on the Web and it is the base of the Web of Data, an interconnected global knowledge graph. Because of the increased amount of machine-readable knowledge freely available on the Web, there is a high interest in investigating how such information can be used to improve recommender systems [4].

Currently, most recommender systems exploit ratings to infer user preferences, although the growing popularity of social and e-commerce websites has encouraged users to write reviews. These reviews enable recommender systems to represent the multi-faceted nature of users' opinions and build a fine-grained preference model, which cannot be obtained from overall ratings [2].

We address the issue of mining reviews and show how the extracted information, combined with Linked Data, can be exploited in recommendation tasks. On one side Linked Data can provide a rich representation of the items to be recommended since they include interesting features. On the other side, reviews may reveal additional connections among items. For instance, various reviews of *Interstellar* mention Stanley Kubrick, although in DBpedia there is not a direct link between these two resources.

We propose a new recommendation approach that semantically annotates reviews to extract useful information from them. The annotated entities and the knowledge freely available in the Web of Data are then combined to discover additional resources and generate recommendations. Our method can exploit any dataset available in the Web of Data to provide recommendations, although we rely on DBpedia[2] and Wikidata[3] in our implementation. We performed an offline study in the movie, book, and music domains, to evaluate different properties of recommender systems, i. e. prediction accuracy (both in terms of ratings and ranking), diversity, and novelty. The results showed that our method achieved the highest diversity, provided a better accuracy than the method based on Linked Data, and increased the novelty of recommendations with respect to traditional techniques.

The contribution of this paper is threefold. Firstly, we exploit state-of-the-art semantic annotation techniques to extract, from user reviews, useful and non-trivial information about the items to recommend. The extracted entities are resources in the Web of Data; thus we can discover additional knowledge through their links. Secondly, we rely on the annotated and discovered entities to provide recommendations, taking into account their occurrence in the reviews and their relationships in the Web of Data. Thirdly, we validate our approach by evaluating its effectiveness through an offline study conducted in the movie, book, and music domains. A technical report [13] extensively describes the offline study and provides additional information on our approach.

The remainder of this paper is organized as follows: Section 2 presents our approach; Section 3 describes the evaluation method, while Section 4 shows the obtained results and Section 5 discusses them; Section 6 provides the conclusions.

2 APPROACH

The architecture of SemRevRec consists of two main modules: semantic annotation and discovery, and recommendation. The former is responsible for feeding the recommender system with semantically annotated entities and Linked Data through the knowledge

[1]http://linkeddata.org

K-CAP, Dec. 4 - 6, 2017, Austin, TX, USA
2017. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[2]http://wiki.dbpedia.org
[3]https://www.wikidata.org

base, while the latter provides recommendations to users. Every time a new review is submitted, the system executes the semantic annotation and discovery steps and possibly adds new entities, while the recommendation process can start when the user provides an initial item. The recommendation module works online, while the semantic annotation and discovery are done offline. Initially, some reviews are annotated and the resulting entities are used to discover additional entities through Linked Data.

SemRevRec deals with the annotated or discovered entities and the items to recommend. We consider the items a particular type of entities since SemRevRec recommends items which may be annotated or discovered entities, although an item may not appear as an entity in the system, e. g., a movie is reviewed but was never annotated or discovered. However, this does not mean that an entity corresponding to such film does not exist in the considered knowledge base. Semantic annotation and discovery are explained in Section 2.1, while recommendation is presented in Section 2.2.

2.1 Semantic Annotation and Discovery

Semantic annotation is the process of annotating textual or multimedia contents with semantic tags to add information about their meaning [11]. In written text, this can be done by associating a URI to the recognized entities. We considered two popular semantic annotators that rely on Wikipedia: AIDA [6] and DBpedia Spotlight [3]. They are both capable of disambiguating entities according to the surrounding context: this is useful because users frequently write acronyms and abbreviations. We finally selected AIDA because it is more accurate according to an independent comparison [5].

The module of semantic annotation and discovery analyzes the text of the reviews and stores the identified entities in a relational database. The URI of each annotated entity is associated with the URI of the reviewed item and with the occurrence of that entity in all the reviews of that item. In effect, the same entity may appear again in reviews regarding another item. AIDA is capable of identifying and disambiguating the entities mentioned in the review considering, by default, the ones available in YAGO[4].

The AIDA resources are mapped with the equivalent ones available in DBpedia exploiting the similar structure of the URIs. In contrast, the mapping between DBpedia and Wikidata relies on the `owl:sameAs` predicate available in DBpedia. If the same entity corresponds to more than one in the other knowledge base, it is ignored in order to avoid probable inconsistencies. The same holds if there is no `owl:sameAs` property, although DBpedia is well linked to YAGO and Wikidata.

Semantic annotation allows SemRevRec to exploit Linked Data for retrieving additional entities. This is possible because the annotated entities are also resources in the Web of Data. Thus, the discoverer can find resources which are related to the annotated entities in order to enable our system to recommend more items. Reviews are a source of non-trivial relations: for example, in a movie recommendation scenario, a user can mention a movie which reminds him the reviewed one because of the colors, the setting, or the atmosphere, and these features are hardly available as Linked Data. At the same time, Linked Data can enrich information coming from users.

Given the annotated entities, the discoverer retrieves from the knowledge base other relevant entities through SPARQL queries. It relies on some properties which can be configured and depend on the domain and on the dataset considered. The discovery is not bounded to a particular knowledge base or domain. On the contrary, this approach is fairly general since it relies only on RDF and SPARQL.

More specifically, the discoverer reads the annotated entities stored during the semantic annotation phase. The discoverer is able to obtain all the resources which have the given entities as an object of the selected properties.

The discoverer stores the discovered entities in a relational database for efficiency reasons. The URI of each discovered entity is associated with the URI of the annotated entity through which it was discovered, and, optionally, with the LDSD measure [9] between them. This measure is inversely proportional to the number of links between two resources: more links result in a lower distance. Each discovered entity may be found through more than a single annotated entity. The LDSD can be exploited in the ranking phase, which is described in Section 2.3.

2.2 Recommendation

The recommendation process consists of two main steps: the generation of the candidate recommendations and their ranking. Given an initial item, SemRevRec retrieves all the entities which are related to the initial item and then ranks them.

Firstly, the system selects the annotated entities which were mentioned in the reviews of the initial item. Afterwards, it obtains the entities which mention the initial item, i. e., entities whose reviews generated an annotated entity that corresponds to the initial item. For example, if the initial item is *Interstellar* and a review of *2001: A Space Odyssey* mention *Interstellar*, then *2001: A Space Odyssey* is considered as a candidate recommendation. Then, SemRevRec optionally retrieves the discovered entities. They may include entities discovered through the initial item. For instance, if the initial item is *Interstellar* and *The Dark Knight* was previously discovered because both these movies have been directed by Christopher Nolan, *The Dark Knight* is selected. The same holds if *Interstellar* was discovered from *The Dark Knight*, i. e., Christopher Nolan was annotated in the reviews of the latter. Similarly, the entities discovered through other entities which were annotated in the reviews of the initial item are relevant. For example, if *Interstellar* is the initial item, Stanley Kubrick was annotated in one of its reviews, and *2001: A Space Odyssey* was discovered through Stanley Kubrick, then *2001: A Space Odyssey* is a candidate recommendation. It is possible to configure the generator to include in the candidate recommendations the discovered entities or not. It is also possible to specify the minimum occurrence required for entities to be included in the candidate recommendation set, which is expressed as a percentage of the maximum occurrence of entities in the reviews of the item considered.

2.3 Ranking Functions

Finally, SemRevRec ranks the candidate recommendations. We defined three different ranking functions. The first is presented in Equation 1 and takes into account only the occurrence $occur(i)$ of the entities available in the reviews. $occur(i)$ is equal to the number of reviews of an initial item i_{in} where an entity i is annotated, plus the number of reviews of i where i_{in} is annotated (if any). However, the entity i can be annotated or discovered. For the latter, the

[4]http://www.yago-knowledge.org

occurrence of the entity through which it was discovered is used. The α coefficient is 1 if i is an annotated entity. Otherwise, it can be configured to a custom value (the default is 0.5) to weight the contribution of a discovered entity to the ranking. To obtain a value between 0 and 1, R1 is normalized to the maximum occurrence of entities j which belong to the candidate recommendation set CR.

$$R1(i) = \frac{\alpha \cdot occur(i, i_{in})}{max_{j \in CR}(occur(j, i_{in}))} \tag{1}$$

The second ranking function (Equation 2) also considers the LDSD measure between each discovered entity and the entity through which it was discovered. This avoids assigning the same value to all the entities discovered through the same annotated entity as R1 does. As for R1, the entity i can be annotated or discovered. The β coefficient is 1 if i is an annotated entity, 0.5 otherwise. The γ coefficient is 0.5 for discovered entities, 0 otherwise. In this way, R2 returns a number between 0 and 1, which is equal to R1 for the annotated entities, while, for the discovered entities, it is the average of R1 and $LDSD(i, i_o)$, where i_o is the entity through which i was discovered.

$$R2(i) = \beta \cdot R1(i) + \gamma \cdot (1 - LDSD(i, i_o)) \tag{2}$$

The third ranking function (Equation 3) considers the LDSD measure between an entity i and the initial item i_{in}. The coefficients η and κ can be set to custom values and they allow the ranker to weight differently the contribution of the occurrence in the review (given by R2) and Linked Data (through the LDSD measure).

$$R3(i) = \eta \cdot R2(i) + \kappa \cdot (1 - LDSD(i, i_{in})) \tag{3}$$

LDSD measures between discovered entities and the entities through which they were discovered need to be precomputed at discovery time (see Section 2.1) to enable SemRevRec to exploit R2, LDSD measures between entities in CR and the initial item need to be computed while ranking. In the latter case, the ranking time is increased.

3 EVALUATION PROCEDURE

We evaluated the performance of SemRevRec with two offline experiments conducted in the movie, book, and music domains. The purpose of the first experiment is to understand the impact of the ranking function, the discovery, the occurrence threshold, and the coefficients of R3. Furthermore, we performed the first experiment two times, first relying on DBpedia and then on Wikidata, to assess the effect of the exploited knowledge base on the quality of the recommended items. This experiment and its results are described in the technical report [13]. The aim of the second experiment is to compare our proposal with traditional recommendation techniques that rely on ratings and a state-of-the-art recommender system based on Linked Data.

In order to conduct both experiments, we obtained from IMDb, LibraryThing, and Amazon the user reviews regarding all the items included in the MovieLens 1M[5], the LibraryThing[6] and the HotRec 2011 LastFM[7] datasets of user ratings.

The items of such rating datasets were mapped with the corresponding entities available in DBpedia relying on the work of Di Noia et al. [8]. Moreover, their equivalent entities in Wikidata were obtained from DBpedia itself, as described in Section 2.1. For the purpose of retrieving the user reviews, Wikidata was exploited in order to discover the IMDb identifiers of the movies available in the MovieLens 1M dataset. On the contrary, the LibraryThing dataset already contained the references useful for obtaining the reviews. Regarding the musical artists present in the HotRec 2011 LastFM dataset, we relied on the search feature of Amazon for identifying their most reviewed musical work.

In order to perform the evaluations, a 5-fold cross-validation was executed. Exploiting the lists of the top-10 recommendations for each user, we computed the measures of precision, recall, nDCG, Entropy Based Novelty (EBN) [1], and diversity [14].

For the implementation, we rely on the LibRec library[8]. It computes measures according to the *all unrated items* protocol [12]. More specifically, it creates a top-N recommendation list for each user by predicting a score for every item not rated by that particular user, whether that item appears in the user test set or not. All the non-rated items are considered to be irrelevant for the user. This explains the low values for the measures (e. g., precision and recall) as the quality of recommendations tend to be underestimated. However, Steck [12] suggests to rely on this protocol rather than the *rated test-items*, which includes only rated test items in the top-N list, as the user satisfaction regarding top-N recommendations depends on the ranking of all items.

4 EVALUATION RESULTS

We compared our technique to the Most Popular, Random Guess, Item KNN, and Bayesian Personalized Ranking (BPR) [10] algorithms, as implemented in LibRec, and with SPrank [8], a state-of-the-art Linked Data-based recommender. We set the neighborhood size for Item KNN to 80, while we used 100 factors for BPR, as done by Musto et al. [7]. We configured SPrank to exploit LambdaMart as the ranking method and to follow in the DBpedia graph the same properties that we selected for our algorithm.

Table 1, Table 2, and Table 3 list the results obtained in the movie, book, and music domain, respectively. The best values are highlighted with a bold font.[9] For SemRevRec, we reported both the configuration with the best trade-off among the various measures and the best scores achieved for each measure. Its optimization is extensively described in the technical report [13].In all the experimental trails, SemRevRec provided the best diversity and a better accuracy (both in rating prediction and ranking) than SPrank, while it improved in novelty with respect to traditional techniques. BPR accounted for the highest precision, recall, and nDCG. In general the diversity of algorithms is rather low for movies, while for music and books is above 0.6, apart for Item KNN.

5 DISCUSSION

SemRevRec showed the best diversity in all the domains. Notably, in the sparse dataset of books, it achieved precision, recall, and nDCG comparable to Item KNN with a much higher diversity, although both are content based methods. However, collaborative filtering techniques are know to suffer less of the overspecilization problem

[5]http://grouplens.org/datasets/movielens/1m/
[6]http://www.macle.nl/tud/LT/
[7]http://ir.ii.uam.es/hetrec2011/datasets/lastfm/readme.txt

[8]https://www.librec.net
[9]More values are highlighted for the same measure if the differences among them are not statistically significant. In the case of EBN and diversity, when Random Guess was the best, we also highlighted the second best because its precision, recall, and nDCG were close to zero. This means that the recommendations provided are completely unrelated and their novelty and diversity is not relevant.

Table 1: Comparison using the MovieLens dataset

Algorithm	Precis.	Recall	nDCG	EBN	Divers.
SemRevRec	0.0857	0.0561	0.0686	1.4188	0.1513
− Best Scores	0.0857	0.0561	0.0686	**0.7820**	**0.2431**
SPrank	0.0445	0.0254	0.0280	0.8813	0.1612
Item KNN	0.1626	0.1105	0.1302	2.6846	0.0696
BPR	**0.2347**	**0.1737**	**0.1930**	1.8358	0.1769
Popular	0.1325	0.0840	0.0969	2.7439	0.1412
Random	0.0055	0.0028	0.0031	**0.3018**	0.1679

Table 2: Comparison using the LibraryThing dataset

Algorithm	Precis.	Recall	nDCG	EBN	Divers.
SemRevRec	0.0530	0.0530	0.0536	0.2318	0.8846
− Best Scores	0.0530	0.0530	0.0536	0.1946	**0.9118**
SPrank	0.0379	0.0346	0.0337	**0.1562**	0.8037
Item KNN	0.0620	0.0564	0.0662	1.4956	0.2259
BPR	**0.0862**	**0.0817**	**0.0895**	0.6043	0.7177
Popular	0.0423	0.0343	0.0447	1.6034	0.6483
Random	0.0004	0.0002	0.0003	**0.0382**	**0.9879**

Table 3: Comparison using the LastFM dataset

Algorithm	Precis.	Recall	nDCG	EBN	Divers.
SemRevRec	0.0536	0.0549	0.0502	0.6319	0.6168
− Best Scores	0.0536	0.0549	0.0502	0.2411	**0.9329**
SPrank	0.0156	0.0158	0.0176	**0.1834**	0.9077
Item KNN	0.1392	0.1428	0.1720	1.6023	0.4730
BPR	**0.1545**	**0.1583**	**0.1808**	0.9404	0.6547
Popular	0.0686	0.0703	0.0791	2.0360	0.6519
Random	0.0005	0.0005	0.0004	**0.0442**	**0.9946**

and provide better rating prediction and ranking than content based ones as SemRevRec. For this reason, although collaborative filtering is very popular, we decided to include in the baseline only one technique among many, i. e. BPR, which is one of the newest and most promising. Nevertheless, it showed a lower diversity than our algorithm. Not surprinsigly, it also accounted for the best rating prediction and ranking.

Our approach also provided a higher novelty than traditional techniques and a better rating prediction and ranking than SPrank. In the movie domain, SemRevRec accounted for the best novelty, while with music and books for the second best, with results close to SPrank. Additionally, when optimized for this measure, SemRevRec had similar (for books) or higher (for music) rating prediction and ranking than SPrank. On the contrary, when the former is optimized for rating prediction and ranking, it could be preferred to the latter to increase the novelty of recommendations, but also limiting the loss in rating prediction and ranking. Additionally, SemRevRec was evaluated considering the recommendations generated for all the previous movies a user liked since its generation approach is rather naive and takes into account only an initial item. Combining it with a machine learning technique could significantly improve its performance, but further experiments are required to prove this.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a novel approach, based on the semantic annotation of user reviews and Linked Data. We conducted an offline study of a recommender system in the movie, book, and music domains, which showed that our approach provides the best diversity. It also improved rating prediction and ranking compared to another method based on Linked Data, while it increased the novelty of recommendations with respect to traditional techniques. We also tested our method with different knowledge bases and Wikidata systematically achieved better results than DBpedia. Although the reviews available for the book and music domains seem to contain a smaller amount of useful information, the results of the offline study suggest that our algorithm can provide more diverse recommendations and reach an interesting compromise between the accuracy and the novelty of the suggested items.

ACKNOWLEDGMENTS

This work was supported by the EU's Horizon 2020 programme under grant agreement H2020-693092 MOVING.

REFERENCES

[1] Alejandro Bellogìn, Ivàn Cantador, and Pablo Castells. 2010. A Study of Heterogeneity in Recommendations for a Social Music Service. In *HetRec '10*. ACM, 1–8. https://doi.org/10.1145/1869446.1869447

[2] Li Chen, Guanliang Chen, and Feng Wang. 2015. Recommender systems based on user reviews: The state of the art. *User Modeling and User-Adapted Interaction* 25, 2 (2015), 99–154. https://doi.org/10.1007/s11257-015-9155-5

[3] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. Improving Efficiency and Accuracy in Multilingual Entity Extraction. In *I-SEMANTICS '13*. ACM. https://doi.org/10.1145/2506182.2506198

[4] Cristhian Figueroa, Iacopo Vagliano, Oscar Rodríguez Rocha, and Maurizio Morisio. 2015. A systematic literature review of Linked Data-based recommender systems. *Concurrency and Computation: Practice and Experience* 27, 17 (2015), 4659–4684. https://doi.org/10.1002/cpe.3449

[5] Aldo Gangemi. 2013. A Comparison of Knowledge Extraction Tools for the Semantic Web. In *The Semantic Web: Semantics and Big Data (ESWC 2013)*. Springer Berlin Heidelberg, 351–366. https://doi.org/10.1007/978-3-642-38288-8_24

[6] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust Disambiguation of Named Entities in Text. In *EMNLP 2011*. ACL, 782–792. http://dl.acm.org/citation.cfm?id=2145432.2145521

[7] Cataldo Musto, Pasquale Lops, Pierpaolo Basile, Marco de Gemmis, and Giovanni Semeraro. 2016. Semantics-aware Graph-based Recommender Systems Exploiting Linked Open Data. In *UMAP '16*. ACM, 229–237. https://doi.org/10.1145/2930238.2930249

[8] Tommaso Di Noia, Vito Claudio Ostuni, Paolo Tomeo, and Eugenio Di Sciascio. 2016. SPrank: Semantic Path-Based Ranking for Top-N Recommendations Using Linked Open Data. *ACM Trans. Intell. Syst. Technol.* 8, 1 (2016), 9:1–9:34. https://doi.org/10.1145/2899005

[9] Alexandre Passant. 2010. dbrec — Music Recommendations Using DBpedia. In *The Semantic Web (ISWC 2010)*, Vol. 2. Springer Berlin Heidelberg, 209–224. https://doi.org/10.1007/978-3-642-17749-1_14

[10] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 452–461. http://dl.acm.org/citation.cfm?id=1795114.1795167

[11] Carsten Saathoff and Ansgar Scherp. 2010. Unlocking the Semantics of Multimedia Presentations in the Web with the Multimedia Metadata Ontology. In *WWW '10*. ACM, 831–840. https://doi.org/10.1145/1772690.1772775

[12] Harald Steck. 2013. Evaluation of Recommendations: Rating-prediction and Ranking. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*. ACM, 213–220. https://doi.org/10.1145/2507157.2507160

[13] Iacopo Vagliano, Diego Monti, Ansgar Scherp, and Maurizio Morisio. 2017. Content Recommendation through Semantic Annotation of User Reviews and Linked Data – An Extended Technical Report. (2017). https://arxiv.org/abs/1709.09973

[14] Mi Zhang and Neil Hurley. 2008. Avoiding Monotony: Improving the Diversity of Recommendation Lists. In *Proc. of the 2008 ACM Conference on Recommender Systems (RecSys '08)*. ACM, 123–130. https://doi.org/10.1145/1454008.1454030