Top-N Book Recommendations Using Wikipedia

Nitish Aggarwal*, Kartik Asooja*, Jyoti Jha°, and Paul Buitelaar*

*Insight Centre for Data Analytics
National University of Ireland
Galway, Ireland
firstname.lastname@insight-centre.org
°IIIT Hyderabad, India

Abstract. This paper presents an approach of recommending a ranked list of books to a user. A user profile is defined by a few liked and disliked books. To recommend a book, we calculate semantic relatedness of the given book to the liked and disliked books by using Wikipedia. Based on the obtained scores, we predict ratings of the book. We evaluate our approach on a dataset that consists of 6,181 users, 8,171 books and 67,990 user-item pairs to predict the rating.

Keywords: Top N Recommendations, Semantic Relatedness, Wikipedia

1 Introduction

Wikipedia provides a valuable source of background knowledge about millions of entities such as movies, actors, places and books. This knowledge can be exploited to build the Top-N recommendation system which deals with finding a set of N items that best match a user profile. The user profile can be defined by liked and disliked items. We assume that user might like the items that are similar or related to his/her liked ones. Therefore, the recommendation task can be re-modeled as to find out a ranked list of items which are more related to the liked items than the disliked ones.

In recent years, there have been several efforts in utilizing external knowledge bases such as Wikipedia and DBpedia [1] for recommendation. Most of the work focuses on boosting collaborative filtering approach or to improve content-based systems [8]. In particular they have shown some benefits in solving cold start and data sparsity issues in conventional collaborative filtering methods. Ostuni et al. [7] have shown the effectiveness of using Linked Open data in boosting collaborative filtering method for Top N movies recommendation.

In this paper, we present Top-N books recommendation system that calculates semantic relatedness of a given book to the liked and disliked books. Wikipedia contains information about thousands of books and their authors. Every book can be seen as a Wikipedia entity. Therefore, in order to compute the semantic relatedness scores between two books, we can calculate the relatedness score

[1] http://wiki.dbpedia.org/

between their corresponding Wikipedia entities (articles). We use Wikipedia-based Distributional Semantics for Entity Relatedness (DiSER) [3] to calculate the relatedness score between two books to perform book recommendations [2]. DiSER calculates the relatedness scores by building distributional vector over Wikipedia articles. However, Aggarwal and Buitelaar [3, 4] have shown a significant improvement over other existing methods of computing entity relatedness such as ESA [1, 5] and KORE [6], we also perform experiments with those other methods for Top N books recommendation task in this paper.

2 Approach

2.1 Top N Recommendation

User profile is defined by a few liked and disliked books. The task is to find out a ranked list of N other books that a user might like. We compute the relatedness scores of a given book with the liked and disliked books. We recommend the book only if the score for like prediction is greater than dislike prediction. Since user can like or dislike more than one book, we need to aggregate the relatedness scores to obtain final confidence scores for like and dislike predictions. Therefore, we use three methods to aggregate the relatedness scores.

Average: We calculate relatedness scores of the given book with all the books liked by the user, and obtained a confidence score by taking an average of these scores. Similarly, we calculate the confidence score for disliked items.

Maximum: We calculate relatedness scores of the given book with all the books liked by the user. Unlike to the Average case, we choose the relatedness score of the most related pair as the confidence score. Similarly, we calculate the confidence score for disliked items.

Random: We randomly select one book from all the books liked by the user, and one from all the disliked ones. We calculate relatedness scores of the given book with the randomly selected liked and disliked books. The obtained relatedness scores are considered as the confidence scores for the corresponding classes.

2.2 Computing Semantic Relatedness

DiSER generates a high dimensional vector by taking every Wikipedia article as a dimension, and considers the associativity weight of an entity with the article as the magnitude of the corresponding dimension. To measure the semantic relatedness between two entities, it computes the cosine score between their corresponding DiSER vectors.

DiSER retrieves a list of relevant Wikipedia articles and rank them according to their relevance scores with the given entity. It considers only the human annotated entities in Wikipedia, thus keeping only the canonical entities that appear with hyperlinks in Wikipedia articles. The tf-idf weight of an entity with every Wikipedia article is calculated and used to build a semantic vector. The semantic vector of an entity is represented by the retrieved Wikipedia concepts

sorted by their tf-idf scores. For instance, there is an entity e, DiSER builds a semantic vector v, where $v = \sum_{i=0}^{N} a_i * c_i$ and c_i is i^{th} concept in the Wikipedia concept space, and a_i is the tf-idf weight of the entity e with the concept c_i. Here, N represents the total number of Wikipedia articles.

3 Evaluation

3.1 Dataset

In order to evaluate our approach, we perform experiments on a dataset provided in "Linked Open Data-enabled Recommender Systems"[2] challenge. There were three different tasks, where task 2 was "Top-N recommendation from binary user feedback". The dataset consists of 6,181 users, 8,171 books and 67,990 user-item pairs to predict the rating. All the books contain their corresponding DBpedia and Wikipedia links.

	Entity Relatedness	**Precision**	**Recall**	**F1**
Average	ESA	0.6132	0.4573	0.5239
	Context-VSM	0.6149	0.4562	0.5237
	DiSER	**0.6254**	**0.4718**	**0.5378**
Maximum	ESA	0.6115	0.4587	0.5242
	Context-VSM	0.6152	0.4551	0.5231
	DiSER	**0.6241**	**0.4683**	**0.5312**
Random	ESA	0.6147	0.4599	0.5262
	Context-VSM	0.6165	0.4588	0.5261
	DiSER	**0.6271**	**0.4703**	**0.5375**

Table 1. Top N books recommendation

3.2 Experiment

We performed experiments with three different relatedness measures: ESA, Context-VSM and DiSER. Similar to DiSER, ESA computes the relatedness score by taking distance between two high dimensional vectors built over Wikipedia. However, unlike DiSER, it does not perform any specific feature selection for entity relatedness. Thus, it considers only the surface form of an entity and do not differentiate between two entities with the same surface forms. For instance, ESA builds the same vector for "Harry Potter (film series)" and "Harry Potter (book series)" as it generates the vector for their surface form i.e. "Harry Potter". We compute ESA score between the book titles. Context-VSM is similar to KORE [6] that computes key-phrase overlap between the contents of the

[2] http://challenges.2014.eswc-conferences.org/index.php/RecSys

corresponding Wikipedia articles. We performed experiments with these three relatedness measures for our recommendation approach by using the above mentioned three aggregation methods: Average, Maximum and Random.

3.3 Results

Table 1 shows the results. The top 5 recommendations are considered to measure the accuracy by calculating Precision@5, Recall@5 and F1@5. Results show that DiSER outperforms other two relatedness measures. All three approaches of finding top N recommendations are comparable and do not show a major difference in the scores. It demonstrates that our approach "Random" can work well with sparse data as it requires only one liked and one disliked items.

4 Conclusion

We presented our approach of Top N books recommendation. We reported the results of three different relatedness measures. DiSER outperformed other two methods of computing relatedness scores. Further, we showed that random aggregation achieves comparable scores. Thus, we can conclude that Wikipedia is a valuable resource for obtaining the recommendation of popular books in a cold-start scenario. Future work will include the investigation of other relatedness measures to boost the conventional recommendation methods like collaborative filtering.

References

1. N. Aggarwal, K. Asooja, G. Bordea, and P. Buitelaar. Non-orthogonal explicit semantic analysis. *Lexical and Computational Semantics (* SEM 2015)*, 2015.
2. N. Aggarwal, K. Asooja, H. Ziad, and P. Buitelaar. Who are the american vegans related to brad pitt?: Exploring related entities. In *Proceedings of the 24th International Conference on World Wide Web Companion*, 2015.
3. N. Aggarwal and P. Buitelaar. Wikipedia-based distributional semantics for entity relatedness. In *2014 AAAI Fall Symposium Series*, 2014.
4. N. Aggarwal, P. Mika, R. Blanco, and P. Buitelaar. Insights into entity recommendation in web search. In *Proceedings of the Intelligent Exploration of Semantic Data, ISWC*, 2015.
5. E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI'07*, pages 1606–1611, 2007.
6. J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum. Kore: Keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 545–554. ACM, 2012.
7. V. C. Ostuni, T. Di Noia, E. Di Sciascio, and R. Mirizzi. Top-n recommendations from implicit feedback leveraging linked open data. In *7th ACM RecSys*, pages 85–92, 2013.
8. G. Semeraro, P. Lops, P. Basile, and M. de Gemmis. Knowledge infusion into content-based recommender systems. In *3rd ACM RecSys*, pages 301–304, 2009.