Serendipitous Recommendations through Ontology-based Contextual Pre-filtering

Serendipitous Recommendations through Ontology-based Contextual Pre-filtering

Aleksandra Karpus[1], Iacopo Vagliano[2]*, and Krzysztof Goczyła[1]

[1] Faculty of Electronics Telecommunication and Informatics,
Gdańsk University of Technology
ul. Gabriela Narutowicza 11/12, 80-233 Gdańsk-Wrzeszcz, Poland
{aleksandra.karpus, krzysztof.goczyla}@eti.pg.gda.pl
[2] Dept. Control and Computer Engineering,
Politecnico di Torino,
C.so Duca degli Abruzzi, 24, 10129 Turin, Italy
iacopo.vagliano@polito.it

Abstract. Context-aware Recommender Systems aim to provide users with better recommendations for their current situation. Although evaluations of recommender systems often focus on accuracy, it is not the only important aspect. Often recommendations are overspecialized, i.e. all of the same kind. To deal with this problem, other properties can be considered, such as serendipity. In this paper, we study how an ontology-based and context-aware pre-filtering technique which can be combined with existing recommendation algorithm performs in ranking tasks. We also investigate the impact of our method on the serendipity of the recommendations. We evaluated our approach through an offline study which showed that when used with well-known recommendation algorithms it can improve the accuracy and serendipity.

Keywords: Recommender Systems, Ontologies, Context-awareness, Serendipity

1 Introduction

Recommender systems aim at providing suggestions for items to be of use to a user. An item could be a movie, a book or even a friend in some social recommender. Context-Aware Recommender Systems (CARS) are a particular category of recommender systems which exploits contextual information to provide more adequate recommendations. For example, in a temporal context, vacation recommendations in winter should be very different from those provided in summer. Or a restaurant recommendation for a Saturday evening with your friends should be different from that suggested for a workday lunch with co-workers [1].

Typically, recommendation algorithms are evaluated according to some accuracy measure, such as *mean absolute error*, *precision* or *recall*. However, accuracy is not the only important aspect of a recommender system. Overspecialized

* The second author was supported by a fellowship from TIM.

recommendations may be unsatisfactory for a user [2]. Recommendations are overspecialized if they are all of the same kind, for example, all movies of the same genre. To deal with this problem, many other properties can be considered, e.g. *novelty*, *serendipity* and *diversity*. *Novelty* describes how many items unseen by the user appeared in the recommendation list. *Serendipity* measures the number of unexpected and interesting items recommended, while *diversity* assesses how much items in the list differ from each other.

In particular, serendipity is useful because users do not want to receive recommendations about items they already knew or consumed. It also does not make a lot of sense to recommend to a user very popular items, e.g. a bestseller book, which he could discover by seeing a commercial or going to the nearest bookstore. For this reason, it is important to propose items that are interesting and unexpected [2].

In a previous study, we proposed an ontology-based and context-aware technique which can be combined with existing recommendation algorithms [3]. In this paper, we examine how it performs in ranking tasks. We also investigate the impact of our method on the quality of recommendations according to *precision*, *recall* and *serendipity* measures, trying to answer following research question:

– *Can incorporating contextual information in the recommendation process improve not only accuracy but also serendipity of recommendations?*

To answer this question, we performed an off-line study on the `ConcertTweets` data set [4], which describes users interests in musical events. The evaluation of the obtained recommendations confirmed that the use of contextual information can improve the serendipity of recommendations.

The rest of the paper is organized as follows. Section 2 describes related works. Section 3 presents our ontology-based and context-aware approach. Section 4 discusses the evaluation approach, and the results obtained. Conclusions and directions for future work close the paper.

2 Related Work

In this section, we focus on the state-of-the-art of the two main topics related to this paper. Section 2.1 describes work concerning the usage of ontologies in the recommendation processes. Section 2.2 focuses on the meaning of the word *serendipity*, its etymology and different definitions in the field of recommender systems.

2.1 Ontology based Recommender System

A number of ontology-based and context-aware recommender system have been proposed. AMAYA allows management of contextual preferences and contextual recommendations [5]. AMAYA also uses an ontology-based content categorization scheme to map user preferences to entities to recommend. News@hand [6] is a hybrid personalized and context-aware recommender system, which retrieves news via RSS feed and annotates by using system domain ontologies. User context is represented by a weighted set of classes from the domain ontology. Rodriguez et al. [7] proposed a CARS which recommends Web services. They use a multi-dimensional ontology model to describe Web services, a user context, and an application domain. The multi-dimensional ontology model consists of a three independent ontologies: a user context ontology, a Web service ontology, and an application domain ontology, which are combined into one ontology by some properties between classes from different ontologies. The recommendation process consists in assigning a weight to the items based on a list of interests in the user ontology. Our work is somehow similar to this approach, because we also use more than one ontology and one of them represents the context dimensions. However, all those works focus on a specific domain and an ad-hoc algorithm, while our approach for representing user preferences is cross-domain and can be applied to different recommendation algorithm.

Hawalah and Fasli [8] suggest that each context dimension should be described by its own taxonomy. Time, date, location, and device are considered as default context parameters in the movie domain. It is possible to add other domain specific context variables as long as they have clear hierarchical representations. Besides context taxonomies, this approach uses a reference ontology to build contextual personalized ontological profiles. The key feature of this profile is the possibility of assigning user interests in groups, if these interests are directly associated with each other by a direct relation, sharing the same superclass, or sharing the same property. Similarly, we model context-dependent user preferences using ontology. They are kept in the form of modules, which represent specific context situations, so we actually also group user interests. However, we have one ontology for all context dimensions, in contradiction to one taxonomy per each context dimension, which is a crucial conceptual simplification.

Other works use ontologies and taxonomy to improve the quality of recommendations. Su et al. proved that ontological user profile improves recommendation accuracy and diversity [9]. Middleton et al. [10] use an ontological user profile to recommend research papers. Both research papers and user profiles are represented through a taxonomy of topics, and the recommendations are generated considering topics of interest for the user and papers classified in those topics. Mobasher et al. [11] proposed a measure which combines semantic knowledge about items and user-item rating, while Anand et al. [12] inferred user preferences from rating data using an item ontology. Their approach recommends items using the ontology and inferred preferences while computing similarities. A more detailed description of ontology-based techniques is available in [13] and [14].

2.2 Serendipity

According to the Oxford dictionary[3], *serendipity* is "the occurrence and development of events by chance in a happy or beneficial way". It was coined in 1754 by

[3] `https://en.oxforddictionaries.com/definition/serendipity`

the English author Horace Walpole in one of his letters, in which he describes his unexpected discovery by referring to "a silly fairy tale, called *The Three Princes of Serendip*: as their highnesses travelled, they were always making discoveries, by accidents and sagacity, of things which they were not in quest of" [15].

The common definition of serendipity in recommender systems does not exist yet, since it is challenging to say which items are serendipitous and why [16].

Ziegler et al. described serendipitous items as those with a low popularity [17]. Results obtained by Maksai et al. confirmed this intuition. They have proved that the most popular items have serendipity equal to zero. However, the definition of the serendipity by Maksai et al. differs from the previous one. "Serendipity is the quality of being both unexpected and useful" [18].

Iaquinta et al. [16] extended previous definitions of the serendipity. Serendipitous items are novel, unexpected and interesting to a user. Adamopoulos and Tuzhilin also require that items have to be novel and unexpected to the user, but they add a third feature: a positive emotional response. "Serendipity, the most closely related concept to unexpectedness, involves a positive emotional response of the user about a previously unknown (novel) item and measures how surprising these recommendations are" [19].

Simpler definition was proposed by Zhang et al. "Serendipity represents the *unusualness* or *surprise* of recommendations" [20].

Kotkov et al. emphasized the problem of technical understanding of concepts used in the prior definitions, i.e. novelty and unexpectedness. "Publications dedicated to serendipity in recommender systems do not often elaborate the components of serendipity [...]. It is not entirely clear in what sense items should be novel and unexpected to a user" [21].

3 Recommendation Approach

Our recommendation approach is based on two ontologies: Recommender System Context (RSCtx)[4] which represents the context, and Contextual Ontological User Profile (COUP), which represents user preferences. In the following, we firstly introduce each ontology, and then we describe the recommendation process.

3.1 The Recommender System Context Ontology

Recommender System Context (RSCtx) extends PRISSMA[5], a vocabulary based on Dey's definition of context [22]. PRISSMA relies on the W3C Model-Based User Interface Incubator Group proposal[6], which describes mobile context as an encompassing term, defined as the sum of three different dimensions: user model and preferences, device features, and the environment in which the action is performed. A graph-based representation of PRISSMA is provided in Figure 1.

[4] `http://softeng.polito.it/rsctx/`
[5] `http://ns.inria.fr/prissma`
[6] `http://www.w3.org/2005/Incubator/model-based-ui/XGR-mbui/`

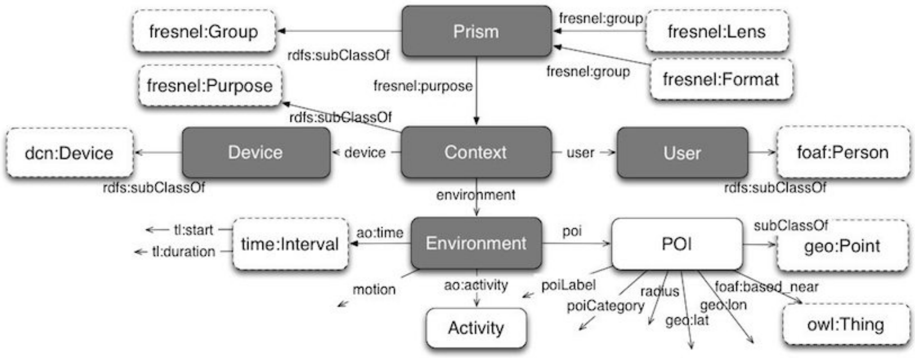


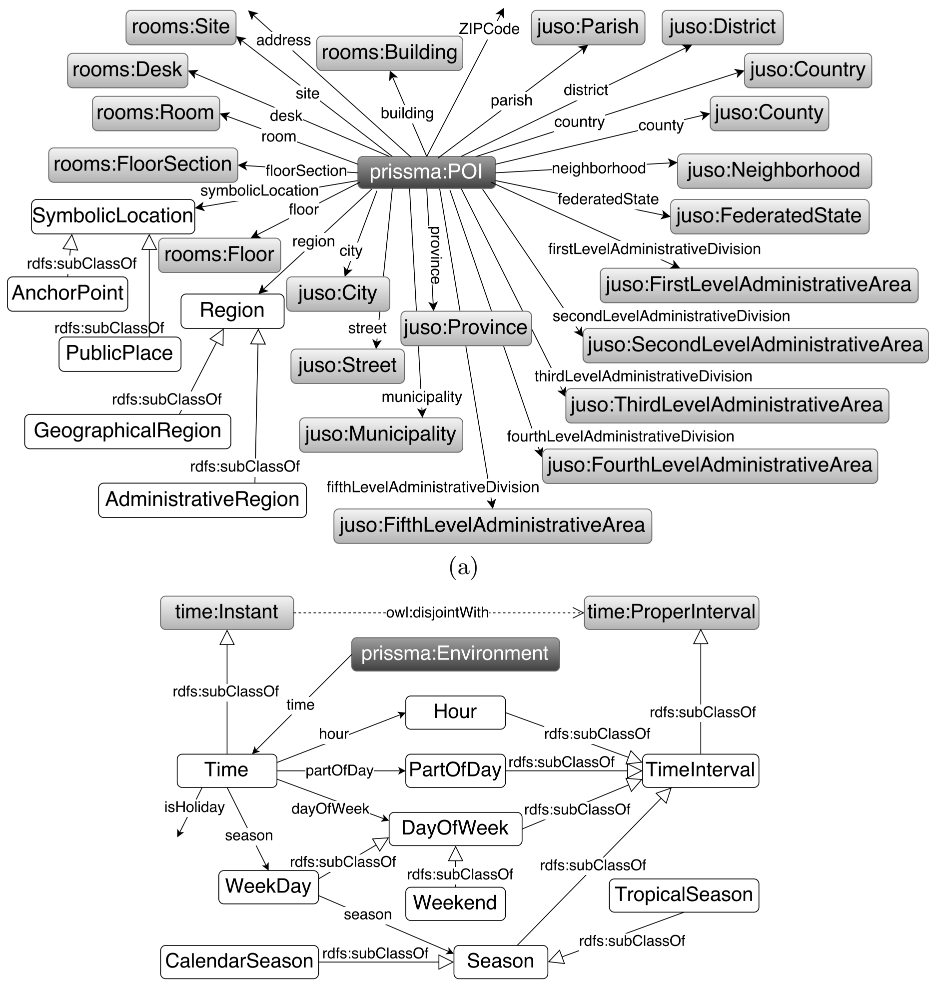
Fig. 1. The PRISSMA vocabulary [23].

We extended the time and location representations. We needed a more expressive model of these two dimensions, since asking for recommendations which have the same time stamp and the coordinates of the actual context is too restrictive and the recommender system may not have enough data. On the contrary, by generalizing the context (for example distinguishing among weekend and working day, or considering the city or neighborhood instead of the actual user position) may enable the recommender system to provide recommendations. The concept `prissma:POI` has been extended with various properties to represent the location in the context of a specific site by integrating the Buildings and Rooms vocabulary[7]. Furthermore, other properties related to the hierarchical organization of the location (such as the neighborhood, the city and the province of the current user position) have been added, and some concepts from the Juso ontology[8] have been reused. Figure 2(a) depicts relations and attributes which characterize a location. Gray rectangles indicate concepts from Juso and rooms vocabulary. The representation of time augments `time:Instant` defined in the Time ontology[9]. Some time intervals have been defined: the hours and the parts of the day (morning, afternoon, etc.). In addition, days of the week are classified in weekdays or weekend and seasons are represented. Figure 2(b) illustrates how time is represented and the relations with PRISSMA and the Time ontology.

3.2 The Contextual User Profile Ontology

To model user profiles, we used the Structured-Interpretation Model (SIM) [24, 25], which consists of two types of ontological modules, i.e. *context types* and *context instances*. Context types describe the terminological part of an ontology (TBox) and are arranged in a hierarchy of inheritance. Context instances

[7] `http://vocab.deri.ie/rooms`
[8] `rdfs.co/juso/latest/html`
[9] `https://www.w3.org/2006/time`



(a)

(b)

Fig. 2. Concepts and relations of RSCtx representing the location dimension (a) and the time dimension. (b)

describe assertional part of an ontology (ABox) and are connected with corresponding context types through a relation of instantiation. There is another kind of relation, i.e. aggregation, which links context instances of more specific context types to a context instance of a more general context type. In the class hierarchy in a classical ontology there always exists a top concept, i.e. `Thing`. In a SIM ontology there is a top context type and a top context instance connected by instantiation. It is possible to add multiple context instances to one context type and aggregate multiple context instances into one context instance. Details can be found in [26–28].

We allow storage of many user profiles in one SIM ontology. We also support a storage of preferences from multiple domains by adding context types related to different domains. We add context types and context instances related to contextual parameters in a dynamic way. As a consequence, we can use as many variables as needed in our approach. An example of a contextual profile for one user is shown in Figure 3.

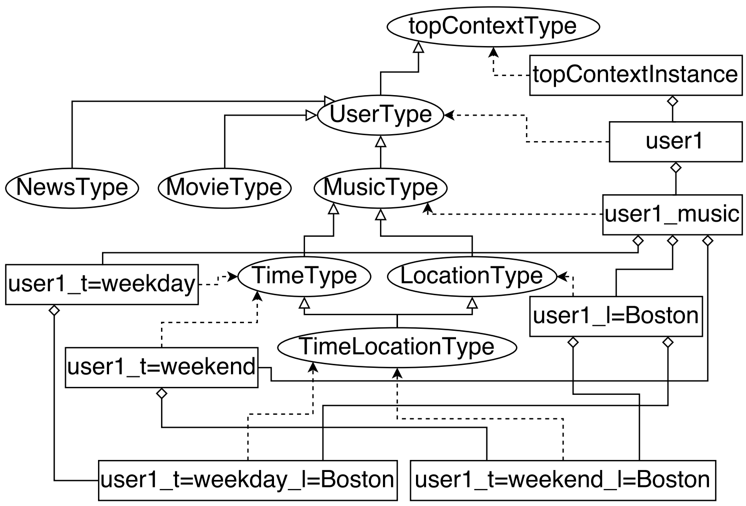


Fig. 3. An example of the Contextual User Profile Ontology

Three modules in the example illustrated in Fig. 3 are fixed: `UserType`, `topContextInstance` and `topContextType`. All others are configurable or can be added in a dynamic way. In `topContextType` we defined the concept `Rating` and its corresponding roles, e.g. `isRatedWith` and `hasValue`. `UserType` is artificial and is present in the SIM ontology because it enables to add many user profiles to the ontology. In the next level of the hierarchy, there are context types that describe domains of interests related to a recommender system which will use the profile. In the next levels, all context types and instances are added to the contextual user profile during the learning phase or later, when a new context situation occurs.

3.3 The Recommendation Process

We use the ontologies previously presented for pre-filtering in the recommendation process. *Pre-filtering* approaches use the current context to select a relevant subset of historical data on which a recommendation algorithm is applied [29]. The aim is to provide a general ontology-based pre-filtering approach, which can be used with existing algorithms.

The system consists of three main functional parts: context detection and generalization, user profile and pre-filtering, and recommendation. In the first part, we used the RSCtx ontology to identify the user context from raw data and generalize it in the desired granularity level. The second part is responsible for building the user profile, finding a context instances that fits the actual user context, and returning only relevant preferences. Because of the lack of similarity measure for SIM ontologies, we get the data from all users who have rated an item in currently considered context (i.e. all context instances for the same values of contextual parameters from different users). However, taking only k nearest neighbors would be more efficient.

The last part uses well-known algorithms, e.g. Item k-Nearest Neighbors (kNN), for providing recommendations. For this task we exploit implementations from the LibRec[10] library.

The general recommendation process proceeds as follows. Given a user and his current situation, a proper generalization of his context is generated by using the RSCtx ontology. Then, an appropriate context instances from COUP is identified by using the generalized context. If a context instance is not found in the user profile, the generalization step is repeated to search for a module that corresponds to the new context. If it is found, preferences from considered user and all other users who have the context instance with the same value for the same contextual parameters are prepared to be used with a recommendation algorithm.

4 Evaluation

We conducted offline experiments in order to evaluate the impact of contextual information on the serendipity of recommendations. We selected a number of algorithms, and we compared the accuracy and serendipity of each algorithm when used as is and when combined with the proposed pre-filtering technique. We aimed to answer the following question: *Does the context improve serendipity of items in the recommendation list?*

We relied on ConcertTweets[11] data set, which combines implicit and explicit user ratings with rich content as well as spatiotemporal contextual dimensions. It contains ratings that refer to musical shows and concerts of various artists and bands. Since the data set was generated automatically, there are two rating scales: one numerical scale with ratings in the range [0.0, 5.0] and one descriptive scale with possible values equal to *yes*, *maybe* and *no*, although *no* never occurred. We decided to split the dataset into two separate sets according to the scale type and we mapped the descriptive values *yes*, *maybe* and *no* with the numerical values 2, 1 and 0. Table 1 presents some statistics about the data by considering the whole data set and each of the sets generated when splitting by scale type. We prepared two pairs (one for each scale) of training and test sets for hold-out validation. In each test set, we put 20% of the newest ratings

[10] `http://www.librec.net/`

[11] `https://github.com/padamop/ConcertTweets`

of each user. All other ratings were placed in each training set. The split was performed based on rating time value.

Table 1. Statistics on the data contained in ConcertTweets data set at the time of the experiment

	Ratings		
	All	**Descriptive**	**Numeric**
Number of users	61803	56519	16479
Number of musical events	116320	110207	21366
Number of pairs artist and musical events	137382	129989	23383
Number of ratings	250000	219967	30033
Maximum number of ratings per user	1423	1419	92
Minimum number of ratings per user	1	1	1
Average number of ratings per user	4.045	3.892	1.823
Maximum number of ratings per item	218	216	38
Minimum number of ratings per item	1	1	1
Average number of ratings per item	2.149	1.996	1.406
Minimum popularity of an artist	1	1	1
Average popularity of an artist	84.317	62.421	13.768
Maximum popularity of an artist	1670	1337	244
Sparsity	0.999971	0.999970	0.999922

We computed an accuracy value by means of the classical information retrieval measures: *precision* and *recall*. The corresponding formulas are as follows:

$$\text{precision} = \frac{|\{\text{relevant items}\} \cap \{\text{recommended items}\}|}{|\{\text{recommended items}\}|} \quad , \tag{1}$$

$$\text{recall} = \frac{|\{\text{relevant items}\} \cap \{\text{recommended items}\}|}{|\{\text{relevant items}\}|} \quad . \tag{2}$$

To measure serendipity of recommendations we use a simple metric presented in [17] that we called *expectedness* and showed in Equation 3.

$$\text{expectedness} = \frac{1}{N}\sum_{i=1}^{N} \text{popularity}\left(\text{item}_i\right) \quad . \tag{3}$$

According to the meaning of the serendipity, the lower the value of the formula (3) is, the bigger the serendipity of the top-N recommendations. In contrast, for (1) and (2) higher values means better precision accuracy.

We had to choose some existing recommendation technique to evaluate our approach since it is designed to work with existing algorithm. We used six algorithms from LibRec library appropriate for the ranking task, i.e. Item kNN, User kNN, BPR[30], FISM[31], Latent Dirichlet Allocation (LDA)[32] and WRMF [33, 34]. We applied them on both splits of the ConcertTweets data set twice: once as is, and the second time on the data generated by our ontology-based contextual pre-filtering technique. We were unable to finish computations for two algorithms, i.e. Item kNN and User kNN, on the subset with a descriptive rating scale, because of their computational complexity and a size of the data set. We computed values of the accuracy and serendipity measures on two lists, i.e. top 5 and top 10 recommendations. The results are collected in Tables 2 and 3.

Table 2. Results obtained for ConcertTweets subset with a numerical rating scale. The prefix *onto-* denotes an algorithm applied combined with our apporach.

	top 5			top 10		
algorithm	precision	recall	expectedness	precision	recall	expectedness
BPR	0.00135	0.00676	38.37	0.00135	0.01353	36.40
ontoBPR	**0.03876**	**0.19382**	**32.36**	**0.03006**	**0.30056**	**28.84**
FISM	0.00000	0.00000	75.68	0.00045	0.00225	118.52
ontoFISM	**0.04103**	**0.20513**	**54.39**	**0.03615**	**0.36154**	**44.33**
ItemKNN	0.00068	0.00338	**26.22**	0.00034	0.00338	**18.32**
ontoItemKNN	**0.01729**	**0.08644**	30.63	**0.01370**	**0.13697**	29.90
LDA	0.00000	0.00000	**65.40**	0.00045	0.00338	118.58
ontoLDA	**0.02456**	**0.12281**	76.72	**0.02281**	**0.22807**	**69.49**
UserKNN	0.00068	0.00113	**26.43**	0.00034	0.00113	**18.51**
ontoUserKNN	**0.01655**	**0.08275**	33.20	**0.01259**	**0.12587**	30.41
WRMF	0.00113	0.00338	58.55	0.00101	0.00564	47.66
ontoWRMF	**0.03520**	**0.17598**	**24.56**	**0.02486**	**0.24860**	**23.46**

As expected, adding contextual information into the recommendation process increases the precision and recall values for all algorithms, data sets, and ranking lists. Moreover, results show that our approach also improves the serendipity of the selected algorithms. Serendipity increases for all algorithms not based on kNN with one exception for the LDA algorithm for the top 5 recommendation list on data set with a numerical rating scale. Nonetheless, the same algorithm gives almost two times better serendipity value in the top 10 list on the same data set. The situation with kNN algorithms is quite different. We observed the deterioration of serendipity for all of the cases, for which we receive results. Though, it could be simply justified. The kNN algorithms are based on similarity. Thus, popular items will be similar to each other with higher probability than less popular items, and when we decrease the recommendation space by constraining it to some particular context, the probability that the less popular item would be considered is even smaller. The same applies to users.

We should further investigate whether we could measure the serendipity of recommendations made by traditional algorithms and context-aware ones in the same way, or if we should incorporate the context also in the serendipity formula. It is impossible to rely on an offline study to address this issue since it is hard to distinguish unexpected items from others without knowing the user's opinion. To this aim, some online experiments are necessary.

Table 3. Results obtained for ConcertTweets subset with a descriptive rating scale. The prefix *onto-* denotes an algorithm applied combined with our apporach.

	top 5			top 10		
algorithm	precision	recall	expectedness	precision	recall	expectedness
BPR	0.00058	0.00208	200.90	0.00054	0.00376	203.06
ontoBPR	**0.01800**	**0.09000**	**157.05**	**0.01588**	**0.15875**	**143.26**
FISM	0.00022	0.00040	480.00	0.00030	0.00168	648.38
ontoFISM	**0.01507**	**0.07537**	**206.84**	**0.01326**	**0.13257**	**198.00**
ItemKNN	-	-	-	-	-	-
ontoItemKNN	**0.01160**	**0.05801**	**123.23**	**0.01078**	**0.10779**	**105.17**
LDA	0.00021	0.00040	475.27	0.00020	0.00088	482.24
ontoLDA	**0.02151**	**0.10755**	**372.20**	**0.01644**	**0.16442**	**305.28**
UserKNN	-	-	-	-	-	-
ontoUserKNN	**0.00521**	**0.02606**	**158.48**	**0.00484**	**0.04839**	**133.78**
WRMF	0.00102	0.00264	255.38	0.00075	0.00368	214.09
ontoWRMF	**0.02029**	**0.10150**	**161.16**	**0.01583**	**0.15834**	**139.00**

5 Conclusions and Further Work

In this paper, we presented some experiments on the use of the ontology-based contextual pre-filtering technique together with existing algorithms for the ranking task on the ConcertTweets data set. We showed that incorporating contextual information in the recommendation process can significantly increase the precision and recall values for all the algorithms used for testing, i.e. Item kNN, User kNN, BPR, FISM, Latent Dirichlet Allocation and WRMF. Moreover, we observed improvement in serendipity for all the algorithms not based on the kNN approach. This suggests that the use of context in the recommendation process increases (desired) unexpectedness of recommended items.

Undoubtedly, some further research is needed. Firstly, we need to investigate how serendipity should be measured in context-aware recommendation systems. This requires a series of online experiments performed with trusted users. Secondly, we lack of a similarity measure which could be used to compare two contextual ontologies (built according to Structured Interpretation Model). As a result, the pre-filtering process is slow and cannot be used in real-life systems. These issues will be addressed in further research.

References

1. Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In Ricci, F., Rokach, L., Shapira, B., Kantor, P.B., eds.: Recommender Systems Handbook. Springer US (2011) 1–35
2. Herlocker, J., Konstan, J., Terveen, L., Riedl, J.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems **22**(1) (1 2004) 5–53

3. Karpus, A., Vagliano, I., Goczyła, K., Morisio, M.: An ontology-based contextual pre-filtering technique for recommender systems. In: 2016 Federated Conference on Computer Science and Information Systems (FedCSIS). (Sept 2016) 411–420
4. Adamopoulos, P., Tuzhilin, A.: Estimating the Value of Multi-Dimensional Data Sets in Context-based Recommender Systems. In: 8th ACM Conf. on Recommender Systems (RecSys 2014). (2014)
5. Rack, C., Arbanowski, S., Steglich, S.: Context-aware, Ontology-based Recommendations. In: SAINT-W '06: Proc. of the Int. Symposium on Applications on Internet Workshops, Washington, DC, USA, IEEE Computer Society (2006) 98–104
6. Cantador, I., Bellogín, A., Castells, P.: Ontology-based personalised and context-aware recommendations of news items. In: Proc. of the 2008 IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology - Volume 01. WI-IAT '08, Washington, DC, USA, IEEE Computer Society (2008) 562–565
7. Rodríguez, J., Bravo, M., Guzmán, R.: Multidimensional ontology model to support context-aware systems (2013)
8. Hawalah, A., Fasli, M.: Utilizing contextual ontological user profiles for personalized recommendations. Expert Systems with Applications **41**(10) (2014) 4777 – 4797
9. Su, Z., Yan, J., Ling, H., Chen, H.: Research on personalized recommendation algorithm based on ontological user interest model. J. of Computational Information Systems **8**(1) (January 2012) 169–181
10. Middleton, S.E., Shadbolt, N.R., De Roure, D.C.: Ontological user profiling in recommender systems. ACM Trans. Inf. Syst. **22**(1) (January 2004) 54–88
11. Mobasher, B., Jin, X., Zhou, Y.: Semantically Enhanced Collaborative Filtering on the Web. In: Web Mining: From Web to Semantic Web: First European Web Mining Forum, EWMF 2003, Invited and Selected Revised Papers. Springer Berlin Heidelberg, Berlin, Heidelberg (2004) 57–76
12. Anand, S.S., Kearney, P., Shapcott, M.: Generating semantically enriched user profiles for web personalization. ACM Trans. Internet Technol. **7**(4) (October 2007)
13. Lops, P., de Gemmis, M., Semeraro, G.: Content-based Recommender Systems: State of the Art and Trends. In: Recommender Systems Handbook. Springer US, Boston, MA (2011) 73–105
14. Di Noia, T., Ostuni, V.C.: Recommender Systems and Linked Open Data. In: Reasoning Web. Web Logic Rules: 11th Int. Summer School 2015, Berlin, Germany, July 31- August 4, 2015, Tutorial Lectures. Springer International Publishing, Cham (2015) 88–113
15. Weijnen, M., Drinkenburg, A., eds.: Precision Process Technology: Perspectives for Pollution Prevention. Springer Science & Business Media, Dordrecht (2012)
16. Iaquinta, L., de Gemmis, M., Lops, P., Semeraro, G., Molino, P.: Can a recommender system induce serendipitous encounters? In: E-Commerce. IN-TECH, Vienna (2009) 1–17
17. Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th International Conference on World Wide Web. WWW '05, New York, NY, USA, ACM (2005) 22–32
18. Maksai, A., Garcin, F., Faltings, B.: Predicting online performance of news recommender systems through richer evaluation metrics. In: Proceedings of the 9th ACM Conference on Recommender Systems. RecSys '15, New York, NY, USA, ACM (2015) 179–186

19. Adamopoulos, P., Tuzhilin, A.: On unexpectedness in recommender systems: Or how to better expect the unexpected. ACM Trans. Intell. Syst. Technol. **5**(4) (December 2014) 54:1–54:32
20. Zhang, Y.C., Séaghdha, D.O., Quercia, D., Jambor, T.: Auralist: Introducing serendipity into music recommendation. In: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining. WSDM '12, New York, NY, USA, ACM (2012) 13–22
21. Kotkov, D., Veijalainen, J., Wang, S.: Challenges of serendipity in recommender systems. In: Proceedings of the 12th International Conference on Web Information Systems and Technologies - Volume 2: WEBIST,. (2016) 251–256
22. Dey, A.K.: Understanding and using context. Personal Ubiquitous Comput. **5**(1) (January 2001) 4–7
23. Costabello, L.: A Declarative Model for Mobile Context. In: Context-Aware Access Control and Presentation for Linked Data. (2013) 21–32
24. Goczyła, K., Waloszek, A., Waloszek, W., Zawadzka, T.: Modularized Knowledge Bases Using Contexts, Conglomerates and a Query Language. In: Intelligent Tools for Building a Scientific Information Platform. Springer Berlin Heidelberg, Berlin, Heidelberg (2012) 179–201
25. Goczyla, K., Waloszek, A., Waloszek, W.: Towards context-semantic knowledge bases. In Ganzha, M., Maciaszek, L.A., Paprzycki, M., eds.: Federated Conf. on Computer Science and Information Systems - FedCSIS 2012, Wroclaw, Poland, 9-12 September 2012, Proc. (2012) 475–482
26. Goczyla, K., Waloszek, W., Waloszek, A.: Contextualization of a DL knowledge base. In: Proc. of the 2007 Int. Workshop on Description Logics (DL2007). (2007)
27. Goczyła, K., Waloszek, A., Waloszek, W.: Hierarchiczny podzia przestrzeni ontologii na kontekst. In: Bazy Danych: Nowe Technologie. WKŁ, Gliwice (2007) 247–260
28. Goczyła, K., Waloszek, A., Waloszek, W.: Techniki modularyzacji ontologi. In: Bazy Danych: Rozwj metod i technologii. WKŁ, Gliwice (2008) 311–324
29. Adomavicius, G., Tuzhilin, A.: Context-Aware Recommender Systems. In: Recommender Systems Handbook. Springer US, Boston, MA (2011) 217–253
30. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. UAI '09, Arlington, Virginia, United States, AUAI Press (2009) 452–461
31. Kabbur, S., Ning, X., Karypis, G.: Fism: Factored item similarity models for top-n recommender systems. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '13, New York, NY, USA, ACM (2013) 659–667
32. Griffiths, T.: Gibbs sampling in the generative model of Latent Dirichlet Allocation. Technical report, Stanford University (2002)
33. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining. ICDM '08, Washington, DC, USA, IEEE Computer Society (2008) 263–272
34. Pan, R., Zhou, Y., Cao, B., Liu, N.N., Lukose, R., Scholz, M., Yang, Q.: One-class collaborative filtering. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining. ICDM '08, Washington, DC, USA, IEEE Computer Society (2008) 502–511