Identifying Linked Data Datasets for sameAs Interlinking Using Recommendation Techniques

Haichi Liu[1(✉)], Ting Wang[1], Jintao Tang[1], Hong Ning[1], Dengping Wei[1], Songxian Xie[1], and Peilei Liu[2]

[1] College of Computer, National University of Defense Technology, Changsha, Hunan Province, China
liuhaichi@nudt.edu.cn
[2] Academy of National Defense Information, Wuhan, Hubei Province, China

Abstract. Due to the outstanding role of owl:sameAs as the most widely used linking predicate, the problem of identifying potential Linked Data datasets for sameAs interlinking was studied in this paper. The problem was regarded as a Recommender systems problem, so serveral classical collaborative filtering techniques were employed. The user-item matrix was constructed with rating values defined depending on the number of owl:sameAs RDF links between datasets from Linked Open Data Cloud 2014 dump. The similarity measure is a key for memory-based collaborative filtering methods, a novel dataset semantic similarity measure was proposed based on the vocabulary information extracted from datasets. We conducted experiments to evaluate the accuracy of both the predicted ratings and recommended datasets lists of these recommenders. The experiments demonstrated that our customized recommenders outperformed the original ones with a great deal, and achieved much better metrics in both evaluations.

Keywords: Linked data datasets · Interlinking · sameAs links · Recommender systems

1 Introduction

In order to be considered as Linked Data, the datasets published on the web have to be connected, or linked, to other datasets [1]. The RDF links such as owl:sameAs between datasets are fundamental for Linked Data as they connect data islands into a global data space so-called Web of Data. Data linking [2] can be formalized as an operation, which takes two Linked Data dataset as input and produces a collection of links between entities of the two datasets as output. When a new dataset was published as Linked Data, the publisher should check all the datasets in the Web of Data to identify the possible links, which is very time-consuming. So if there are some technology can be utilized, being recommended based on known links and focusing on those datasets most likely to link, one can sharply reduce the computational costs if the recommendations are accurate enough.

B. Cui et al. (Eds.): WAIM 2016, Part I, LNCS 9658, pp. 298–309, 2016.
DOI: 10.1007/978-3-319-39937-9_23

In the Web of Data, an increasing number of owl:sameAs[1] statements have been published to support merging distributed descriptions of equivalent RDF resources from different datasets. The owl:sameAs property is part of the Web Ontology Language (OWL) ontology [3], the official semantics of owl:sameAs is: *an owl:sameAs statement indicates that two URI references actually refer to the same thing.* When all of these owl:sameAs statements are taken together, they form a very large directed graph connecting Linked Data datasets to each other. Due to the outstanding role of owl:sameAs as the most widely used linking predicate [4], we focus on recommendation of datasets for sameAs interlinking. Previous works [5–8] mostly did not distinguish RDF link types when identifying datasets for interlinking, and experiments were conducted on the experimental data constructed from RDF links of various types, while the graphs formed from various types of RDF links exhibit different characters [4]. Previous works would be of less help for real application scenarios, as dataset publishers still do not know what kinds of RDF links can be established furthermore how to configure the data linking algorithms. Due to the limitations of previous methods, it is necessary to find better ways.

In this paper we try to tackle the problem of identifying more datasets that can be established owl:sameAs links with, when the publisher's dataset has already linked to a few datasets. This is the scenario that the Recommender systems [11] techniques can be applied. We construct user-item matrix with rating values depending on the number of owl:sameAs RDF link triples between datasets from newly updated LOD Cloud 2014 dump [4]. Several classical collaborative filtering methods of Recommender systems are applied. Utilizing the semantic schema information extracted from Linked Data datasets, we define dataset semantic similarity to replace the original similarity component of memory-based collaborative filtering methods to develop our customized recommenders. To evaluate the recommenders, we conduct two experiments for assessing rating and top n recommendation accuracy. Experimental results demonstrate our customized recommenders perform much better than the original ones. The MAEs are only half of the original ones, the values are low to the range of (0.3, 0.5) on a rating scale of 1 to 7. The F-Measures are almost twice higher, the values are within the range of (0.2, 0.5), which are promising given the large set of datasets to recommend from. This drastic improvement are liable on the peculiar properties of the merging of dataset semantic similarity and memory-based collaborative filtering recommenders. The source codes and experimental data have been uploaded to Github[2].

The rest of the paper is organized as follows. In Sect. 2, at first we describe the framework which consider the dataset identification problem as a Recommender systems problem and how we construct user-item rating matrix. Then we describe the collaborative filtering technologies we used upon the problem. At last we define a dataset semantic similarity algorithms used for injecting domain-specific information. In Sect. 3, we describe the experiments data,

[1] http://www.w3.org/2002/07/owl#sameAs.
[2] https://github.com/HaichiLiu/Recommending-Datasets-for-Interlinking.

evaluation methodology, and results. In Sect. 4, we present related works. Finally, in Sect. 5 we conclude the paper.

2 Recommender Systems Techniques

We model the problem of identifying target dataset for sameAs interlinking as a Recommender systems problem, and we describe how to construct user-item rating matrix which is necessary for recommendation algorithms in Sect. 2.1. Several representative collaborative filtering algorithms we employed are briefly described in Sect. 2.2. Also we define a dataset semantic similarity algorithm as the similarity computation component of memory-based recommenders in Sect. 2.3.

2.1 Recommendation Framework

Recommender systems are personalized information agents that attempt to predict which items out of a large pool a user may be interested in. The user's interest in an item is expressed through the rating the user gives the item. Generally, the interaction between user and item is represented with a user-item rating matrix. A recommender system has to predict the ratings for items that the user has not yet seen. With these estimated ratings the system can recommend the items that have the highest estimated rating to the target user. Note that item is a general term used to denote what the system recommends to users, and can be of any type, like movies, books, websites, or news articles. In our case, these items are Linked Data datasets available in the Web of Data. We use $U = \{u_1, u_2, u_3, ..., u_n\}$ to denote the set of dataset publishers (users), $D = \{d_1, d_2, d_3, ..., d_m\}$ for the set of datasets (items). We view that each dataset d_i is published by a unique publisher u_i, this makes $n = m$. This may not be hold in real world, but actually u_i is merely an identifier of dataset d_i in the publishers set U, which makes the representation to be understood easily in a Recommender systems scenario. And we denote R as an $n \times n$ matrix of ratings $r_{i,j}$, with $i \epsilon \{1, ..., n\}, j \epsilon \{1, ..., n\}$. Recommender algorithms are used to predicting the rating values of a certain dataset publisher for the datasets he or she has not linked, or recommending a ranked list of datasets he or she might want to link according to the rating values predicted.

We aggregate all owl:sameAs RDF links by dataset, meaning that we consider dataset publisher (user) of dataset a has a rating for dataset b if there exists at least one owl:sameAs RDF link triple from dataset a which contains the subject of the triple to the dataset b which contains the object. We find that some Linked Data dataset publisher did not choose the standard http://www.w3.org/2002/07/owl#sameAs as linking predicates, but use terms from proprietary vocabulary, such as http://www.abes.fr/owlsameAs, even mistakenly used http://www.w3.org/2002/07/owlsameAs, http://www.w3.org/2000/01/rdf-schema#sameAs, we also extract links defined by these predicates. Since we can view that a dataset is sameAs interlinked to itself, the number of RDF

link triples equals to the number of entities defined in the dataset. Rating values are set based on number of owl:sameAs RDF link triples, the rating value equals to the number of digits of link triples count. We illustrate the construction of rating matrix from datasets interlinking with the example as Fig. 1. For example, dataset d_1 has 243 RDF links to dataset d_2, the corresponding matrix entry $r_{1,2}$ equals to 3.

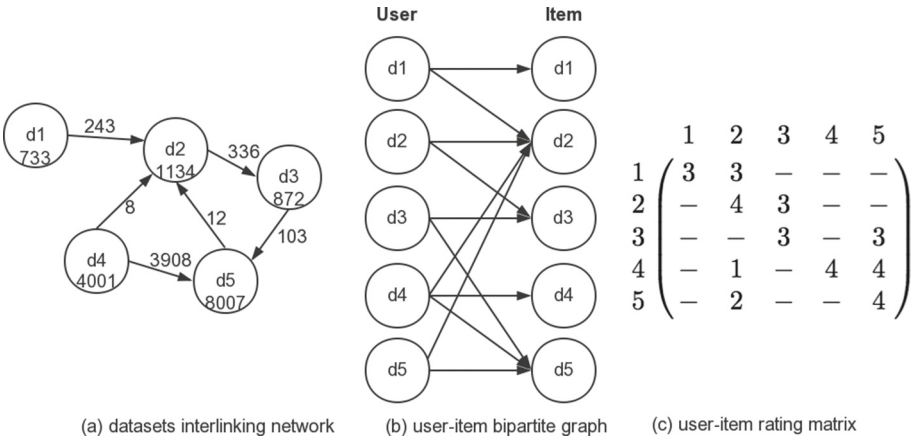

Fig. 1. An example to illustrate how to construct rating matrix from datasets interlinking network. (a) is an example of interlinking network of five datasets, identified by $d1, d2, ..., d5$. The number inside the circle is the entities number of each dataset. The arrows represents owl:sameAs RDF links between datasets with a number of RDF links triples count. (b) is an example of user-item bipartite graph constructed from (a), each user has a link set to itself in item set. Generated rating matrix is shown in (c), and it is a 5×5 matrix.

2.2 Collaborative Filtering Recommendation

Collaborative filtering is widely implemented and the most mature recommendation technique. The concept is to make correlations between users or between items. There are memory-based and model-based techniques [11].

Memory-Based Recommendation. Memory-based recommenders can be divided into: user-based and item-based recommenders. The main idea of user-based algorithms is simply as follows: given a ratings matrix and the ID of the current (active) user as an input, identify nearest users that had similar preferences to those of the active user in the past. Then, for every item i that the active user has not yet seen, a prediction is computed based on the ratings for i made by the nearest users:

$$r_{a,i} = \sum_{b \epsilon N_a} sim(a, b) \times r_{b,i} \tag{1}$$

The similarity measure between user a and b, *sim(a,b)*, is essentially a distance measure and is used as a weight. Different algorithms can be used to compute the similarity, such as cosine, Pearson, Spearman, Euclidean distance, et al.

The main idea of item-based algorithms is to compute predictions using the similarity between items rather than the similarity between users. An item-based algorithm computes a weighted average of these other ratings as the following with *sim(i,j)* is computed similar to what we did in user-based recommender:

$$r_{a,i} = \sum_{j \epsilon S_i} sim(i,j) \times r_{a,j} \tag{2}$$

Matrix Factorization Models. Matrix factorization models [12] is a model-based method which maps both users and items to a joint latent factor space of dimensionality *f*. Accordingly, each item i is associated with a vector $q_i \epsilon R_f$, and each user u is associated with a vector $p_u \epsilon R_f$. For a given item i, the elements of q_i measure the extent to which the item possesses those factors, the same is true for a user. The resulting dot product, ${q_i}^T p_u$, approximates the rating of user u for item i. To learn the factor vectors (p_u and q_i), the system minimizes the regularized squared error on the set of known ratings:

$$\min_{p^*,q^*} = \sum_{u,i \epsilon K} \left(r_{u,i} - {q_i}^T p_u\right)^2 + \lambda(|p_u|^2 + |q_i|^2) \tag{3}$$

Here, K is the set of (u,i) pairs for which $r_{u,i}$ is known (the training set). Simon Funk popularized a stochastic gradient descent optimization of Eq. 3 wherein the algorithm loops through all ratings in the training set. It modifies the parameters by a magnitude proportional to γ in the opposite direction of the gradient, yielding:

$$e_{u,i} = r_{u,i} - {q_i}^T p_u \tag{4}$$

$$q_i \leftarrow q_i + \gamma \cdot (e_{u,i} \cdot p_u - \lambda \cdot q_i) \tag{5}$$

$$p_u \leftarrow p_u + \gamma \cdot (e_{u,i} \cdot q_i - \lambda \cdot p_u) \tag{6}$$

2.3 Injecting Domain-Specific Information

Both user-based and item-based recommender rely on similarity component. In user-based recommender the similarity between two users is based on their ratings of items that both users have rated, likewise in item-based recommender the similarity between two items is based on ratings of users that rated both items. As a Linked Data dataset is a collection of RDF triples describing entities with RDFS vocabularies or OWL ontologies, this motivates us to extract its semantic schema features to define dataset semantic similarity, and make it as the similarity component of memory-based recommenders to develop our customized recommenders. In this section, we describe how we model a Linked Data dataset with vector space model (VSM) using semantic features, and how to calculate similarity between datasets based on the model further.

Vector Space Model for Linked Data Dataset. Vector space model is an algebraic model for representing text documents (and any objects, in general) as vectors of identifiers, such as, for example, index terms. In VSM each document is represented by a vector in a m-dimensional space, where each dimension corresponds to a term from the overall vocabulary of a given document collection. VSM was adapted to model the dataset in this paper. A Linked Data dataset uses one or more RDFS vocabularies or OWL ontologies. The vocabulary provides the terms (classes and properties) for expressing the data. A vocabulary URI, a class URI or a property URI used in triples of the dataset can be seen as *semantic features* of the dataset, they are called *vocabulary feature*, *class feature* and *property feature* respectively. Let $T = \{t_1, t_2, ..., t_m\}$ be the dictionary, that is the set of semantic features of datasets in the corpus. Formally, each dataset $d_i = \{w_{1i}, w_{2i}, ..., w_{mi}\}$, where w_{kj} is the weight for feature t_k in dataset d_i.

Dataset representation in the VSM raises two issues: weighting the terms and measuring the feature vector similarity. The most commonly used term weighting scheme is TF-IDF (Term Frequency-Inverse Document Frequency) weighting:

$$TF - IDF(t_k, d_j) = \frac{f(t_k, d_j)}{|\{t_i \epsilon d_j\}|} \cdot log \frac{n}{|\{d_j \epsilon D : t_k \epsilon d_j\}|} \tag{7}$$

$f(t_k, d_j)$ is the number of times that feature t_k occurs in dataset d_j.

Dataset Semantic Similarity. As stated earlier, a similarity measure is required to determine the closeness between two datasets. Many similarity measures have been derived to describe the proximity of two vectors; among those measures, cosine similarity is most widely used:

$$sim(d_i, d_j) = \frac{\sum_k w_{ki} \cdot w_{kj}}{\sqrt{\sum_k {w_{ki}}^2} \cdot \sqrt{\sum_k {w_{kj}}^2}} \tag{8}$$

As we assume each dataset is published by a unique publisher, the dataset semantic similarity can be used as user similarity component in user-based recommender as well as item similarity component in item-based recommender. Using dataset semantic similarity in memory-based methods also helps to relieve the cold start problem of recommender systems, which is common in our scenario, as the number and the variety of Linked Data datasets are increasing rapidly. A "new" dataset with few interlinkings to other datasets cannot easily be recommended in pure memory-based recommenders, because the similarity was computed on past ratings. While dataset semantic similarity is computed utilizing only the information of datasets themselves, recommendations can also be made for new datasets.

3 Experiments

3.1 Experimental Data

We construct the experimental data from the LOD Cloud 2014 dataset published in [4]. The data is a crawl of the Web of Linked Data conducted in

April 2014, which contains 8,038,396 resources crawled from 900,129 documents. The crawled data is provided for download as a single N-Quads formatted 2.6 GB zipped dump file. Using the dataset URIs published by the authors, we managed to divide the dump into 990 separated dataset dumps, in which the quads whose subject' URI defined under the same dataset URI are grouped together. For each dataset dump, we extract semantic features of property, class and vocabulary types. The property features of a dataset are obtained by grouping all the predicate URIs of RDF triples in the dataset dump. The class features are obtained by grouping the object URIs of RDF triples with (s rdfs:type o) pattern. Since the namespace of class or property URI are the URI of the vocabulary where the class and property were defined, by grouping all the namespace of class and property URIs of a dataset, we can get the vocabulary features of the dataset. Using the method we describe in Sect. 2.1, We manage to construct our experimental user-item matrix with 990 users, 990 items and 1641 ratings from users to items.

3.2 Evaluation Methodology

Evaluating Rating Accuracy. Mean Absolute Error (MAE) is used to measure the closeness of predicted ratings to the true ratings. It is defined as the average absolute difference between the n pairs $< p_h, r_h >$ of predicted ratings and real ratings:

$$MAE = \frac{\sum_{h=1}^{n} |p_h - r_h|}{n} \tag{9}$$

In our experiments, for each user, we take a certain percentage of the ratings as "training data" to produce recommendations, and the rest of the ratings is compared against estimated rating values to compute MAE. The results may differ as the data set is split randomly, hence for each algorithm, we run the test for 10 times and take the average score for final presentation.

Evaluating Top N Recommendations. It's not always essential to present estimated rating values to users. In many cases, an ordered list of recommendations, from best to worst, is sufficient. So we could apply classic information retrieval metrics F1-Measure to evaluate recommenders. We adopt *leave-one-out* strategy, for a user we remove his top n ratings, and use his left ratings and all the other users' ratings as training set. The final scores are calculated by averaging all the users' test results. As the test rating records are selected in descending order of its value rather than randomly, for memory-based algorithms we do not need to repeat the experiments. For matrix factorization algorithm in which calculation was started from randomly initialized vectors, we run the test 10 times and present the average results.

3.3 Results and Discussion

The recommendation algorithms and evaluation methods are implemented with the help of a Java machine learning library called Mahout [13]. To inject

domain-specific information as stated in Sect. 2.3, we implement a customized class that extends ItemSimilarity and UserSimilarity of Mahout.

To the best of our knowledge, there are rare works applying Recommender systems techniques to the problem of identifying target dataset for sameAs interlinking. For comparision, we have chosen three simple recommenders: Random, ItemAverage and ItemUserAverage and three original collaborative filtering recommenders: Item-based, User-based and Rating SGD. Random recommender produces random recommendations and preference estimates. ItemAverage recommender is a simple recommender that always estimates rating for an item to be the average of all known preference values for that item. ItemUserAverage recommender is like ItemAverage recommender, except that its estimated ratings are adjusted for the users' average rating value. Item-based recommender is the original one implemented in Mahout, Item-{Vocabulary, Class, Property} recommenders are our customized recommenders in which the similarity components of original item-based algorithm are replaced by our dataset semantic similarity components with vocabulary, class and property features respectively. This is the same for User-based recommenders. For User-based recommenders, there are two ways for choosing neighborhoods: fixed-size neighborhoods (noted with n as the neighborhoods size parameter) and threshold-based neighborhoods (noted with t as the threshold parameter). We explored a range of possible choices of both parameters for both evaluation. For fixed-size neighborhoods, n is in the range of [1,10] with 1 as step size, for threshold-based neighborhoods, t is in the range of [0.1,0.9] with 0.1 as step size. The best results are shown in the Tables 1 and 2, and the optimized parameters are noted in cells. There are three parameters can be tuned for RatingSGD recommender, f is the number of factors used to compute this factorization, γ is the learning rate, and i is the number of iterations. These parameters also have been tuned for optimum performance.

When evaluating rating accuracy, we vary the percent of rating records used for training from 50 % to 90 %. In Table 1 we can see that the MAEs are around 2.5 for Random recommender. With some simple intuitions, the MAEs are lower to about 1.0 or 1.2 for ItemUserAverage and ItemAverage recommenders. Original item-based recommender in Mahout has further lower MAEs about 0.8. Our Item-{Vocabulary, Class, Property} recommender shows better performance in MAEs at the training percent 50 %, 60 % and 70 %, but worse at 80 % and 90 %. The MAEs of original user-based recommender with fixed-size neighborhoods are in the range of (0.9, 1.0), the MAEs of original user-based recommender with threshold-based neighborhoods are also in the range of (0.9, 1.0) but lower. Both User-based recommenders have better performance than Item-based recommender. RatingSGD recommender is generally better than original item and user based recommenders, but not as good as our customized recommenders. User-{Vocabulary, Class, Property} recommenders with fixed-size neighborhoods mostly have better performance than the original User-based recommender with fixed-size neighborhoods. User-{Vocabulary, Class, Property} recommenders with threshold-based neighborhoods have much better performance than the original User-based recommender with threshold-based neighborhoods, actually the MAEs of User-Class

recommender with threshold-based neighborhoods are the lowest of all tested recommenders at all training percent. The values are around (0.3, 0.5), which are only half of the MAEs achieved by the best original recommender, i.e., the Item-based recommender.

Table 1. MAE comparison of various recommenders

	50 %	60 %	70 %	80 %	90 %
Random	2.4591	2.5758	2.4587	2.5194	2.5293
ItemAverage	1.0517	1.0702	1.0793	1.0867	1.0351
ItemUserAverage	1.2407	1.2429	1.2380	1.2671	1.2265
Item-based	0.8611	0.8698	0.8361	0.7700	0.8148
Item-Vocabulary	0.7754	0.8049	0.7978	0.8359	0.8454
Item-Class	0.7329	0.7757	0.7882	0.8021	0.8246
Item-Property	0.7775	0.8191	0.7981	0.8678	0.8863
User-based $n = 8$	1.0151	0.9890	1.0951	0.9683	0.9138
User-Vocabulary $n = 4$	1.0043	0.9930	0.9077	0.9399	0.8584
User-Class $n = 2$	0.8796	0.9125	1.0679	1.1103	0.8546
User-Property $n = 10$	0.9720	1.0057	1.0761	1.1061	0.9074
User-based $t = 0.6$	0.9620	0.9403	0.9995	0.9660	0.9268
User-Vocabulary $t = 0.6$	0.7934	0.7177	0.7537	0.7002	0.6627
User-Class $t = 0.9$	**0.3669**	**0.4607**	**0.4153**	**0.4102**	**0.3904**
User-Property $t = 0.9$	0.6149	0.5794	0.5709	0.8153	0.5667
RatingSGD $f = 20$ $i = 50$ $\gamma = 0.01$	0.8649	0.8252	0.8419	0.8518	0.8607

For evaluating the Top N recommendation, we evaluate top 1 to top 10 recommendation performance of various recommenders for comprehensive comparisons. The results are shown only for top 1 to 5 in Table 2 due to the space limit, the trend of results for top 6 to 10 tests are similar. Random recommender failed completely in this test, since randomly recommending a few number of datasets out of 990 datasets can hardly hit the right answers. The other two simple recommenders also performed very poorly. Original item and user based recommenders performed better. Item-based recommender achieved F1-Measures larger than 0.1 for top {3, 5, 6, 7, 8} test cases. The F1-Measures of User-based recommender with fixed-size neighborhoods are higher and within the range of (0.2, 0.5). The F1-Measures of User-based recommender with threshold-based neighborhoods are within the range of (0.1, 0.2). RatingSGD recommender performed worse than original item and user based recommenders, the F1-Measures are always below 0.05. Our customized Item-{Vocabulary, Class, Property} recommenders have close performance compared with Item-based recommender. While User-{Vocabulary, Class, Property} recommenders all achieve much better performance compared with the original user-based ones. The User-Vocabulary

recommender with fixed-size neighborhoods achieves the best F1-Measures in all the top n test cases except top 1, the F1-Measures are within the range of (0.2, 0.5), almost twice higher than the best original recommender, i.e., User-based recommender with fixed-size neighborhoods.

Table 2. F1-measure comparison for Top N Recommendation

	top 1	top 2	top 3	top 4	top 5
Random	NaN	NaN	NaN	NaN	NaN
ItemAverage	0.0156	0.0072	NaN	NaN	NaN
ItemUserAverage	0.0156	0.0072	NaN	NaN	NaN
Item-based	0.0272	0.0874	0.1310	0.0667	0.1400
Item-Vocabulary	0.0066	0.0504	0.1071	0.2000	0.1600
Item-Class	0.0132	0.0360	0.0952	0.1000	0.1200
Item-Property	NaN	0.0504	0.0952	0.1500	0.1000
User-based $n = 3$	**0.0468**	0.2009	0.2130	0.2171	0.2188
User-Vocabulary $n = 10$	0.0353	**0.2437**	**0.3709**	**0.4710**	**0.4316**
User-Class $n = 8$	0.0400	0.2102	0.3538	0.4437	0.3716
User-Property $n = 10$	0.0266	0.2336	0.2811	0.2648	0.2387
User-based $t = 0.9$	0.0054	0.0942	0.131	0.2000	0.1800
User-Vocabulary $t = 0.5$	0.0203	0.1956	0.2519	0.3984	0.3568
User-Class $t = 0.5$	0.0301	0.1892	0.2736	0.4193	0.3124
User-Property $t = 0.5$	0.0203	0.1956	0.2519	0.3984	0.3568
RatingSGD $f = 20$ $i = 50$ $\gamma = 0.01$	0.0156	0.0072	0.0119	0.0500	0.0200

4 Related Works

Identifying relevant datasets for interlinking is a novel research area. There are a few approaches developed specifically for this purpose, which can be categorized into two groups.

In the first category, the problem is tackled in a retrieval way, these methods try to retrieve datasets that can be interlinked with for a given dataset. Nikolov et al. [9] proposed a method that depends on an third-party semantic web search service. They use labels of randomly selected individuals from a dataset to query the search service and aggregated the results by datasets. They conducted experiments on three datasets as examples. Also not all datasets have instances with labels, Ell et al. [10] show that only 38.2 % of the non-information resources have a label. Lopes et al. [6] proposed a probabilistic approach based on Bayesian theory, they defined rank score functions that exploit vocabulary features of dataset

and the known dataset links. Liu et al. [5] modeled the problem in an Information Retrieval way, they have developed a ranking function called collaborative dataset similarity which is proven to be quite effective. Using learning to rank algorithm to incorporate these ranking functions, they can further improve the performance and achieve the best MAP (Mean Average Precision) compared with previous works.

In the second category, the problem is tackled using link prediction approaches. The graphs of datasets and interlinking between them are constructed and link prediction measures are used to rank potential dataset pairs. Lopes et al. [6] represented the data space as a directed graph, and used the Preferential Attachment and the Resource Allocation to measure the likelihood that two datasets can be connected. The linear combine of these two score is used to rank the dataset pair. But when computing Preferential Attachment score, instead of using out-degree of source dataset, they used the size of similarity set of source dataset. Similarity set is defined as the set of datasets which have vocabulary features in common with source dataset. Mera et al. [7] developed a dataset interlinking recommendation tool called TRT. They implemented most of state of art local and quasi-local similarity indices, but these indices are not combined in any way. They also developed a tool called TRTML [8]. In TRTML, the interlinking of datasets was represented as an undirected graph, and four link prediction measures were implemented and three supervised classification algorithms were used. They balanced the percentage of unlinked tripleset pairs considered for better performance when comparing various algorithms, in this way the testing settings can no longer reflect the real challenges as the real-world distribution is extremely imbalanced.

5 Conclusion

The Web of Data is constantly growing, in order to be considered as Linked Data, the datasets published on the web have to be interlinked to other datasets. Data linking between two given datasets is a time-consuming process, if there are some techiques can bepublisher, it will substantially reduce the need to perform exploratory search. As the ubiquitous owl:sameAs property is used to connect these datasets, we focus on this type of link, and try to solve the problem of identifying target dataset for sameAs interlinking, when the publishers dataset has linked to a few datasets. This is the scenario that the Recommender systems techniques can be applied. We construct user-item matrix with rating values depending on the number of RDF link triples between datasets. We extract vocabulary features of dataset, and define a dataset semantic similarity algorithm as the similarity component of memory-based recommenders. The experiments show that Recommender systems techniques is effective for the problem and our customized recommenders perform better than original collaborative filtering recommenders. For future work, we plan to exploit more advanced recommendation techniques and develop more effective features focusing on the topical aspect of datasets.

Acknowledgements. This material is based on work supported by the National Natural Science Foundation of China (61200337, 61202118, 61472436)

References

1. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. Int. J. Semantic Web Inf. Syst. **5**, 1–22 (2009)
2. Ferrara, A., Nikolov, A., Scharffe, F.: Data linking for the semantic web. Int. J. Semantic Web Inf. Syst. **7**, 46–76 (2011)
3. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, P., Stein, L.A.: OWL web ontology language reference. W3C Recommendation (2004). www.w3.org/TR/owl-ref
4. Schmachtenberg, M., Bizer, C., Paulheim, H.: Adoption of the linked data best practices in different topical domains. In: Mika, P., et al. (eds.) ISWC 2014, Part I. LNCS, vol. 8796, pp. 245–260. Springer, Heidelberg (2014)
5. Liu, H., Tang, J., Wei, D., Liu, P., Ning, H., Wang, T.: Collaborative datasets retrieval for interlinking on web of data. In: Presented at the Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, 18–22 May 2015, Companion Volume (2015)
6. Lopes, G.R., Leme, L.A.P.P., Nunes, B.P., Casanova, M.A., Dietze, S.: Two approaches to the dataset interlinking recommendation problem. In: Benatallah, B., Bestavros, A., Manolopoulos, Y., Vakali, A., Zhang, Y. (eds.) WISE 2014, Part I. LNCS, vol. 8786, pp. 324–339. Springer, Heidelberg (2014)
7. Caraballo, A.A.M., Nunes, B.P., Lopes, G.R., Leme, L., Casanova, M.A., Dietze, S.: TRT - a tripleset recommendation tool. In: Presented at the Proceedings of the ISWC 2013 Posters & Demonstrations Track, Sydney, Australia, 23 October 2013
8. Caraballo, A.A.M., Arruda Jr., N.M., Nunes, B.P., Lopes, G.R., Casanova, M.A.: TRTML - a tripleset recommendation tool based on supervised learning algorithms. In: Presutti, V., Blomqvist, E., Troncy, R., Sack, H., Papadakis, I., Tordai, A. (eds.) ESWC Satellite Events 2014. LNCS, vol. 8798, pp. 413–417. Springer, Heidelberg (2014)
9. Nikolov, A., d'Aquin, M., Motta, E.: What should I link to? identifying relevant sources and classes for data linking. In: Pan, J.Z., Chen, H., Kim, H.-G., Li, J., Horrocks, I., Mizoguchi, R., Wu, Z., Wu, Z. (eds.) JIST 2011. LNCS, vol. 7185, pp. 284–299. Springer, Heidelberg (2012)
10. Ell, B., Vrandečić, D., Simperl, E.: Labels in the web of data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 162–176. Springer, Heidelberg (2011)
11. Adomavicius, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**, 734–749 (2005)
12. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Comput. Soc. **42**, 30–37 (2009)
13. Owen, S., Anil, R., Dunning, T., Friedman, E.: Mahout in Action. Manning Publications Co., Shelter Island (2011)