A Linked Dataverse Knows Better: Boosting Recommendation Quality Using Semantic Knowledge

Andreas Lommatzsch, Till Plumbaum, and Sahin Albayrak
Technische Universität Berlin, DAI-Labor
Ernst-Reuter-Platz 7, D-10587 Berlin, Germany
{*andreas.lommatzsch, till.plumbaum, sahin.albayrak*}*@dai-labor.de*

***Abstract*—The advent of Linked Open Data (LOD) gave birth to a plethora of open datasets freely available to everyone. Accompanied with LOD, a new research field arises focusing on how to handle and to take advantage of this huge amount of data. In this paper, we introduce a novel approach utilizing and aggregating open datasets to compute the most-related entities for a set of weighted input entities. We optimize different algorithms for large semantic datasets enabling combining data from different semantic open sources and providing high quality results even if only limited resources are available. We evaluate our approach on a large encyclopedic dataset. The evaluation results show that our approach efficiently supports different semantic edge types. The application build on our framework provides highly relevant results and visual explanations helping the user to understand the semantic relationship between the computed entities.**

***Keywords*-linked open data; recommendation; semantic web; user profile enrichment; personalization**

I. Introduction

With the rapidly growing number of large open datasets following the Linked Open Data (LOD) principles [1], semantic recommender systems and applications based on linked datasets become an important research area. Semantic datasets, which represent knowledge as a huge network of nodes and labeled edges, provide the basis for the effective deployment of (natural) language independent knowledge processing. Thus, the approach for processing semantic datasets abstracts from classical text processing tasks (e. g., handling of synonyms, homonyms, typos, multi-lingual content, ambiguous terms), but focuses on deploying the relationship between unique entities. Moreover, the ontology based semantic representation of data simplifies the reuse of existing datasets and the integration of new information sources.

For many domains (such as music, movies, and geographic locations), large semantic encyclopedic datasets are available from Freebase [2] and DBpedia [3]. These encyclopedic datasets provide generally accepted, almost static knowledge. The data is represented as nodes ("vertexes") connected by labeled edges, describing the relationship between the nodes. The entities (such as artists, events, locations, or points of interest) represented as nodes are usually annotated with meta-data (such as images or labels for different languages).

An important question, when working with semantic datasets, is how to discover the entities (of a specific type) most closely related to a set of input entities. The computation of related entities is used for interfering knowledge for enriching profiles or for calculating recommendations. The main questions that have to be answered when calculating related entities are:

1) What types of edges should be considered for computing the semantic similarity between nodes?
2) How to assign weights to labeled edges?
3) How to combine edge weights of paths between the source node and the destination node?
4) How to efficiently compute related items based on huge datasets? Which network models adequately reduce the complexity without spoiling the result quality?

In this paper we discuss and compare several algorithms for computing the most-related entities for a weighted set of input entities. The evaluation is based on a recommender system for the music domain. In contrast to most existing systems that focus on user ratings and user generated tags, our system bases on well accepted encyclopedic data. Thus, we concentrate on computing related entities and not on personalized recommendation (personalized recommendation cannot be found in an encyclopedia). The computation of related entities based on encyclopedic data has the advantage that results are built on a reliable dataset and thus are suitable for enriching sparse user profiles.

The paper is structured as follows: Section II gives an overview of related work; Section III explains the dataset used for evaluating our approach. In Section IV, we introduce our approach in detail. Section V presents a recommender systems implemented based on our approach. The experiments and the evaluation performed for proving the properties of our approach are discussed in Section VI. Finally, a conclusion and an outlook to future work are given in Section VII.

II. Related work

Most of the existing recommender systems apply collaborative filtering (CF) methods [4], [5], [6]. Recommendations are calculated by analyzing the similarity of user profiles (user-based CF) or the similarity of rated items, such as artists, albums, films, books (item-based CF). Some authors [7], [8], [9] combine user-based CF and item-based CF approaches. These hybrid recommender systems often deploy expert-defined, domain-specific rules for a scenario dependent combination of different feature types.

For the entertainment domain several recommender systems exist, such as the FOAFing-the-music project [10], combining social networks and user ratings. Another active research area is the use of *Linked Open Data* [11]. Comprehensive ontologies have been defined for the semantic storage of knowledge for the music domain. Well-known ontologies are provided by the Music Ontology project [12] and the Music Similarity Ontology project [13]. These ontologies focus on the aggregation of various data sources and on providing fine-grained semantic descriptions of relevant entities.

III. Dataset

We use an encyclopedic dataset retrieved from Freebase as data source for testing our semantic processing framework. For the evaluation we use a rating dataset retrieved from LastFM (http://www.last.fm/). Freebase is a comprehensive data source for semantic data containing information about almost every domain. In our scenario (computing the most-related entities in the music domain) we make use of a subset of the data retrieved from Freebase consisting of the four entity types *Artists, Albums, Tracks*, and *Genres*. The relationship between *Artists* and *Genres* describes the genre in which an artist usually works; the relationship between *Albums* and *Artists* describes the album releases of each artist, and finally the relationship between *Albums* and *Genres* defines a genre assignment for each album. The created dataset is schematically visualized in Figure 1.

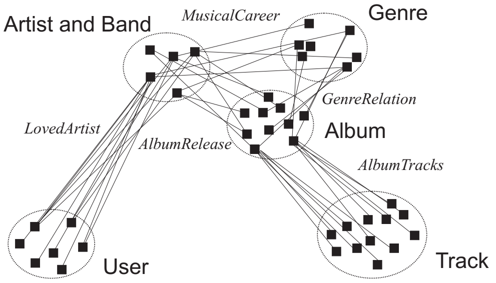


Figure 1. The semantic music dataset.

To compare an encyclopedic "recommender" with a rating-based recommender, we interlink the encyclopedic dataset retrieved from Freebase with a rating dataset retrieved from LastFM (collected in December 2010) consisting of 40,000 user profiles. The linkage of the datasets had been established based on the artist names and the MusicBrainz ID [14]. The size of the respective entity sets and relationship sets is shown in Table I.

Table I
The number of entities and edges in the encyclopedic dataset.

	# entities	**# edges**			
		Artists	**Genre**	**Albums**	**Tracks**
Artists	417217	-	79543	374445	-
Genre	3082	79543	-	90444	-
Albums	438180	374445	90444	-	1048565
Tracks	1048576	-	-	1048565	-

IV. Approach

The necessary steps for computing the most-related entities for a set of input items are: Assign numerical edge weights (describing the similarity between entities) based on the edge labels, and define rules ("an algebra") describing how to combine the edge weights. Additionally, models for coping with the network complexity must be defined, speeding up the computation process and reducing the noise present in real-world datasets.

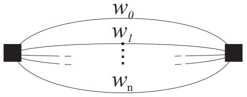
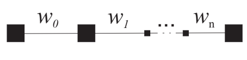
We discuss the challenges and solutions for each step in detail in the following paragraphs. At first, we analyze the task of link prediction in a semantic network. In other words, we infer for a given node the entities strongly related and suggest to add edges to these nodes [15], [16]. In our application scenario, the prediction of new edges means to compute the most-related entities for a given input entity that are not directly connected by an edge in the semantic dataset. We focus on algorithms allowing us to provide explanations for each predicted entity. In many scenarios this is important since good explanations help to increase the user's trust and confidence in the recommendations as wells as in the recommender system itself [17].

How to define relatedness: For computing related items in a large semantic network, we have to define criteria for measuring the semantic *similarity* between two entities. Criteria for defining the similarity between two nodes in a semantic network are:

- Entities connected by a short path are more related to each other than entities connected by a long path.
- Entities connected by several different parallel paths are more closely related than entities connected by one path only.
- The edge labels (and the derived edge weights) of a path between two nodes should influence the computed node relatedness. In general, the edge weight might depend on the path context (in other words, on the other edges of a path).

Path Algebra: Based on the proposed criteria for the relatedness of nodes of a network, an edge algebra is defined. Well-known approaches for combining the edge weights of a path are the *shorted path distance*, the *resistance distance*, and the *weighted path distance* [18]. The rules for calculating the path weight according to the different combination approaches are shown in Table II.

Table II
THE TABLE SHOWS THE FORMULAS FOR CALCULATING THE PATH WEIGHTS FOR (A) PARALLEL EDGES AND (B) FOR A SEQUENCE OF EDGES. THE DISCOUNT FACTOR γ ENSURES THAT SHORT PATHS GET A HIGHER WEIGHTING THAN LONG PATHS.

		Weighted Path	Resistance Distance	Shortest Path
(A)		$w = \overset{n}{\underset{i=0}{}} w_i$	$w = \frac{1}{\overset{n}{\underset{i=0}{}} w_i}$	$w = \min\limits_{i=0}^{n} w_i$
(B)		$w = {}^{n} \overset{n}{\underset{i=0}{}} w_i$	$w = \overset{n}{\underset{i=0}{}} w_i$	$w = \overset{n}{\underset{i=0}{}} w_i$

Computing recommendations on semantic datasets: Large semantic datasets usually consist of several node types (often annotated with `rdf:type`) and edge sets connecting exactly two entity sets (*bipartite relationship sets*). Additionally, unipartite relationships, connecting nodes within one entity set, may exist (e. g., to model hierarchies of entities). Each relationship between the entity sets has a semantic meaning that can be used for deriving edge weights. In general, two entity sets can be connected by several different relationship sets, describing different semantic associations.

For computing the most-related items for a set of input entities, we define which relations can be combined to build valid paths. In other words, we identify a set of valid pipes, describing the edge types combined in a path as well as the minimal and maximal path length. This approach allows us to assign edge weights based on the context of an edge. Thus, we do not use a static edge weight, but choose the edge weight dependent on the semantic meaning of an edge in a path. Moreover, for each relationship type specific models can be defined allowing us to consider the special properties of each relationship type.

Memory-based Recommender: To compute related entities for a given set of input items, we determine the entities best connected to the input entities (according to the defined edge algebra). We implement the search based on a Branch and Bound algorithm [19], adapted to handle parallel paths in the search process. The search process takes into account the different semantic edge types and ensures that only paths consisting of valid edge sequences are considered. The advantage of path-based recommenders is that no additional effort is needed for building a dataset model. Thus, updates in the dataset immediately affect the computed results. Another advantage of calculating the most-related nodes directly on the dataset is that the computed paths can be used as explanations for the derived nodes. In most scenarios the path length is limited so that the explanations are not too complex ensuring that users understand these computed explanations. An example for an path-based explanation (taken from the encyclopedic recommender system for the music domain) is shown in Figure 2. Starting from the input node `Kelis`, the path recommender used five genre nodes, to find several parallel paths to the artist `Pink`. Edge weights and edge labels are not shown in the explanation graph in order to keep the explanation simple.

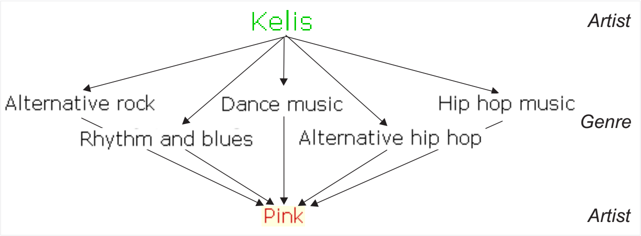


Figure 2. Explanation of a path-based recommendation (used in our music recommendation web application). The user can see the different nodes that are relevant for recommending the artist *Pink*.

Model-based Recommender: While working with real-world data, semantic relationship sets are often huge, noisy, and sparse. Models for simplifying the semantic relationship set are applied to cope with these problems.

Clustering: An efficient approach for reducing the entity set size and the relationship size is aggregating similar entities into clusters. The advantage of this approach is that most users understand the idea of clustering and path-based explanations can be computed (handling clusters as "virtual" entities). Figure 3 shows an example for an explanation containing clustered entity sets.

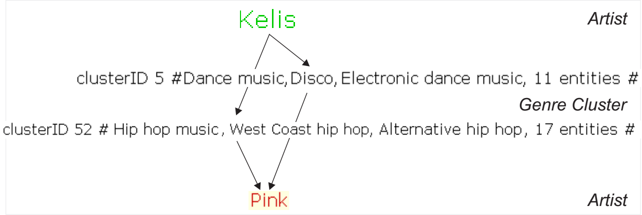

Kelis
Artist
clusterID 5 #Dance music, Disco, Electronic dance music, 11 entities #
Genre Cluster
clusterID 52 # Hip hop music, West Coast hip hop, Alternative hip hop, 17 entities #
Pink
Artist

Figure 3. Explanation of a path-based recommendation using a clustered entity set. By aggregating similar entities into clusters, the graph complexity and thus the complexity of the provided explanation are reduced.

Analyzed approaches for clustering: We focus on Hierarchical Agglomerative Clustering (HAC) [20]. The advantage of HAC is that the desired number of clusters does not need to be chosen in advance. The distance measure used for clustering may take into account several different entity properties. In our music recommendation scenario, we used a similarity measure based on a weighted combination of the

ISBN: 978-1-61208-175-5

genre name similarity and user-defined genre hierarchy data (retrieved from Freebase) for clustering the music genres.

Low Rank Approximation: An alternative approach for reducing the complexity is to compute a low rank approximation of the adjacency matrix for a relationship set. For this purpose we calculate the singular value decomposition (SVD) of the normalized adjacency matrix A and consider only the first k latent dimensions.

$$A = USV^T \approxeq U_k S_k V_k^T$$

The adjacency matrix A is decomposed into a diagonal matrix S, containing the singular values of A in descending order. The matrices U and V consist of the left-singular and right-singular vectors for S. The low rank approximation of A considers only the largest k singular values of A and the respective eigenvectors (U_k, V_k^T).

The advantage of this approach is, that it allows us an efficient reduction of the matrix size. Moreover, the low rank approximation has been proven to be a good model for large sparse matrices [21]. Disadvantages of this approach are on the one hand that no easily understandable explanations can be provided and on the other hand that the singular value decompositions is resource-demanding. Dataset updates require a recalculation of the matrix decomposition.

Conclusion: In this section we discussed the problem of computing related items for a given set of entities considering the node and edge semantics. In contrast to most of the existing systems which consider only one edge type (typically "like" or "is similar to") our system focuses on analyzing the edge semantics. The combination of heterogeneous edges takes into account the semantics of respective paths. We explained different approaches for combining edge sequences and parallel paths (*edge algebra*) dependent on the respective node types and edge labels. A promising approach consists of expert-defined rules, reflecting the specific properties of the respective domain, and optimized parameter settings computed using machine learning methods based on the available training data.

Additionally, we discussed the advantages and disadvantages of memory-based and model-based approaches for efficiently computing related entities. The analysis showed that memory-based approaches allow providing user-understandable explanations without precomputing sophisticated models. Model-based approaches allow reducing the complexity and taking into account the noise in real-world datasets.

V. Implementing a System for Encyclopedic Music Recommendations

Based on the developed framework for semantic data processing, we implemented a web application for suggesting entities semantically related to the entities present in the user profile. As the knowledge base for our recommender system, we use a semantic dataset for the music domain retrieved from Freebase (see Table I).

User profile: The user preferences are stored as a set of weighted entities. The entities define artists, genres, tracks and albums the user "likes". User preferences are collected implicitly (by analyzing the user behavior) and explicitly (allowing the user to enter entities she is interested in). A disambiguation component computes potentially matching entities to the user's input ensuring that only valid entities are added to the user profile. The disambiguation component is needed due to the fact that a user-entered name may represent different entities. For instance, the name `Madonna` may stand for an American singer, her first album or the second studio album from the American band `...And You Will Know Us by the Trail of Dead`.

The analyzed edge combinations: For computing the recommendations based on the encyclopedic dataset, we tested which semantic relationship sets should by combined to provide good results. We focus on path of limited length (maximal 4 edges) consisting of edges from only one edge set, since the meaning of those paths is understood best by the users. While calculating the most-related entities for a set of user profile entities several different relationship sets are taken into account. Figure 4 shows an example for computing related items for the entities `Dr. Dre` and `50 Cent`. The entity `Eminem` is related to the input entities because `Eminem` has four music genres in common with `Dr. Dre` and `50 Cent`. Moreover, he worked together at the albums `Welcome To The Dogg House`, `The Slim Shady LP` and `Up In Smoke Tour`.

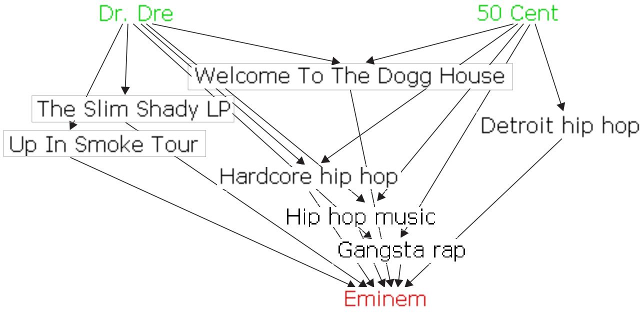


Figure 4. The figure visualizes the considered path of length 2 between user profile entities and the derived entity `Eminem`. Each path consists of edges from only one relationship set.

Since a joint album usually implies a close similarity between two artists, in our web application the paths based on the Artist-Album relationship have a higher weight than paths based on the Artist-Genre relationship. Only in the case that no related entities can be computed, neither based on the Artist-Album relationship nor based on the Artist-Genre relationship, more complex paths (such as Artist → Genre → Album → Artist) are taken into account.

Preliminary experiences: The first evaluation results of the developed encyclopedic “recommender” system showed, that the entities calculated to be related to the user profile are useful to the user. The huge number of nodes enables the system to compute results even for only locally known artists. In contrast to systems focused on individual ratings, the suggested entities are related to the user profile (according to the encyclopedic knowledge base) and not based on the user’s taste. Most users liked the idea of providing explanations for the results, especially if a recommendation is not obviously related to the user interest. The presentation of explanations as a graph seems to be an acceptable solution as long as the explanations are not too complex. Hence, we simplify complex explanation graphs keeping only the edges with the highest weights.

VI. Experiments and Evaluation

To evaluate the implemented algorithms, we analyze different scenarios.

A. Link prediction on encyclopedic data

We analyze the task of predicting links on the encyclopedic dataset retrieved from Freebase. We focus on the scenario of computing related artists for a given set of artists (e. g., for the entities from a user’s preference profile). Following the idea of cross-validation, we split the edge set of our dataset into a training set and a test set. Entities connected with less than two edges are not considered in the evaluation. Based on the edges of the training set, the recommender component predicts edges to the most-strongly connected entities and provides a list of edges ordered by the semantic similarity between the connected nodes. The prediction precision is evaluated with the test set. Since the number of entities related to the input entity set varies over the user profiles, the Mean Average Precision (MAP) [22] is used as performance measure. The MAP for a set of user profiles $P = \{p_1, p_2, \ldots, p_n\}$ is calculated as follows:
Let $\mathrm{Prec@}i(L_p)$ be the Precision of the first i items in the predicted result list L for the profile $p \in P$, and $\mathrm{rel@}i(R_p)$ be a Boolean function returning 1 if the ith item in the list L is relevant, then

$$\mathrm{MAP}(P) = \frac{1}{|P|} * \sum_{p \in P} \sum_{i=1}^{m} \mathrm{Prec@}i(L_p) \cdot \mathrm{rel@}i(L_p)$$

Memory-based Recommenders: We analyze the task of predicting related entities directly on the semantic graph retrieved from Freebase (see Figure 1). For the evaluation we performed the following steps: (1) We randomly select a node. (2) The set of edges connected to this node is split into a training set and a test set (50% / 50%). (3) Based on the training set we compute the most-related nodes limiting the maximal considered path length. (4) The predicted nodes are evaluated with the test set. (5) We calculate the average over all the evaluation results for 10,000 nodes. Figure 5 shows the observed prediction precision for the two baseline strategies (predict edges to randomly chosen entities, and predict edges to the entities having the highest number of edges) and for the path-based recommender considering a maximal path length of two or four respectively. The results show that our approach provides highly relevant prediction results. A higher search depth (4 instead of 2) leads to slightly improved results as more nodes are taken into account. The high prediction precision can be explained by the fact, that in the deployed music dataset several parallel paths for connecting two entities exist. Moreover, the artists in the music dataset seem to form “clusters” whose nodes are well connected but have only a small number of connections to other entities.

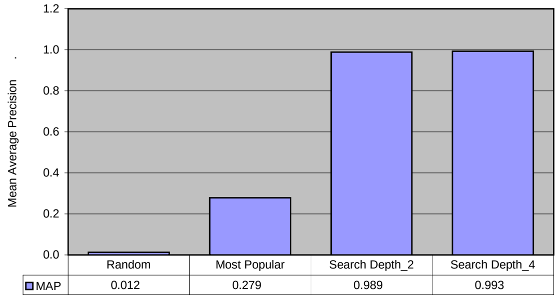


Figure 5. The evaluation of link prediction for artists based on the Freebase dataset.

Link prediction on the clustered entity sets: Due to the large number of music genres in the used Freebase dataset we apply a clustering algorithm for aggregating similar genres. We analyze how the edge prediction precision depends on the number of clusters. The clusters are computed based on a hierarchical agglomerative clustering algorithm. For calculating the distance between two music genres we determine the number of artists and albums directly connected with theses genres. Additionally, we consider the metadata from Freebase describing relations between the music genres.

In our evaluation we compute clusters for the genre entity set and apply a path-based search algorithm with a search depth of two. The measured results (see Figure 6) show that aggregation of the 15% most similar genres into clusters leads to only a minimal decrease of the precision. In the case of a small number of clusters the precision decreases appreciably. For the analyzed scenario the use of $\approx$ 900 clusters provides reasonable results while reducing the considered genre entity set size by $\approx$ 15%, and thus reducing the complexity of the dataset.

B. Profile enrichment based on encyclopedic data

We interlink the encyclopedic music dataset from Freebase with LastFM user profiles and analyze how our recommender improves the collaborative computation of recommendation results by enriching small user profiles.

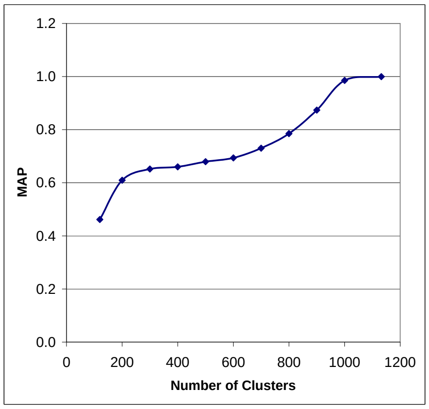


Figure 6. The evaluation of link prediction for artist based on the Freebase dataset using clustered music genres.

For the evaluation, we use 10,000 LastFM user profiles having at least 30 (to have enough information for a proper evaluation) and at most 50 preferred artists. We split each user profile into a training set containing $n \quad (1 \leq n \leq 10)$ artists and a test set containing the remaining artists. As a baseline for our evaluation, we use a standard collaborative filtering (CF) algorithm, computing the similarity between two users based on the number of common entities. CF computes the 100 most similar users (based on the number of common artists) and predicts the entities most frequently present in these profiles. While determining similar users, only the training set for the user (for which the recommendations are computed) is taken into account. The recommendation precision is calculated based on the test set.

We analyze how the recommendation performance changes, if we enrich user profiles using the encyclopedic data retrieved from Freebase. For the recommender on the encyclopedic dataset we consider the artist-genre relation and search depths of two and four. Figure 7 shows that profile enrichment improves the recommendation precision for small user profiles. For users having more than $\approx$ 7 profile entries the profile enhancement leads to less precise results. Thus, encyclopedic knowledge helps to improve the recommendation results for new user. If the user profile consists of an adequate number of entities a profile enrichment based on encyclopedic data should not be applied.

The results can be explained by the fact that similar users cannot be reliably computed for users with a very small profile. Thus, enriching the profile with related entities improves the calculation of similar users and the computation of predictions. Due to the fact that encyclopedic knowledge does not consider the individual user taste, the profile enrichment adds fuzziness to the profile. For large user profiles the items (added by the enrichment) adulterate the user profiles resulting in less precise recommendation results.

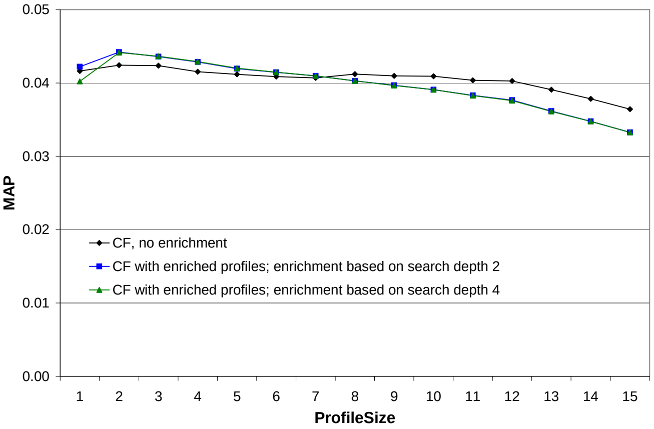


Figure 7. The evaluation shows that profile enrichment based on encyclopedic knowledge improves the precision of collaborative filtering for users with a small profile. For users with more than six entries the profile enrichment reduces the recommendation precision.

VII. Conclusion and future work

In this paper, we introduced a new semantic recommender framework and discussed different algorithms for the efficient processing of large semantic datasets. We explained our graph-based recommendation approach utilizing model- and memory-based link prediction methods. We showed how to provide explanations to increase the trust in the computed recommendations. With the aggregation (“clustering”) of similar entities we could reduce the computational complexity with the trade-off of a small loss of precision. The evaluation of the link prediction approach shows that our recommender provides precise link prediction results on the encyclopedic dataset. The analyzed algorithms require only limited resources and provide comprehensible explanation for the recommendations.

We also demonstrated the application of our recommender to enrich user profiles and explained how the enhanced profiles can be used to improve collaborative filtering. The results showed that encyclopedic data helps only in the case of very small user profiles. This can be explained by the fact that for a user having a small user profile users with similar interests cannot be reliably computed. A profile enrichment based on encyclopedic data improves the computation of similar users and leads to better recommendations. Thus, the profile enrichment helps to overcome the cold-start problem [23]. For users with a big profile encyclopedic data does not improve the recommendation precision. A reason for this is that our encyclopedic data neither considers individual user preferences nor the “quality” of albums or musicians.

Future Work: As future work, we want to analyze and integrate additional recommender models based on matrix decomposition [24], [25] and graph kernels [26]. Preliminary tests with these methods show promising results in effectively reducing the dataset complexity and reducing the noise in the datasets. Furthermore, it is intended to extend the dataset with additional entities and meta-information. First, we want to extend the scope of the music recommendation scenario by adding information such as movies and actors to test our approach in a cross-domain recommendation scenario. Second, we want to add meta-information to the encyclopedic dataset like “quality” of a node to extend the recommendation approach with methods that do not only take into account the graph structure but also the type and quality of a node. Such quality information can be the popularity of an artist or the commercial success. Ongoing work is the preparation of a user study where we want to get real user feedback about the recommendation and explanation quality in order to validate our results based on the automatic evaluation.

REFERENCES

[1] Bizer, C., Heath, T., Idehen, K., Berners-Lee, T.: Linked data on the web (ldow2008). In: Proceedings of the 17th International Conference on World Wide Web. WWW ’08, New York, NY, USA, ACM (2008) 1265–1266

[2] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proc. of the 2008 ACM SIGMOD intl. conf. on Management of data. SIGMOD ’08, New York, NY, USA, ACM (2008) 1247–1250

[3] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In Aberer, K., Choi, K.S., Noy, N., Allemang, D., Lee, K.I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudr-Mauroux, P., eds.: The Semantic Web. Volume 4825 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2007) 722–735

[4] Herlocker, J., Konstan, J., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the International Conference on Research and Development in Information Retrieval. (1999) 230–237

[5] Sahoo, N., Krishnan, R., Duncan, G., Callan, J.: Collaborative filtering with multi-component rating for recommender systems. In: Proceedings of the Workshop on Information Technologies and Systems. (2006)

[6] Sun, J., Zeng, H., Liu, H., Lu, Y., Chen, Z.: CubeSVD: A novel approach to personalized web search. In: Proc. of the Intl. World Wide Web Conference. (2005) 382–390

[7] Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Transactions on Knowledge and Data Engineering **17** (2005) 734–749

[8] Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. ACM Transactions Information Systems **23**(1) (2005) 103–145

[9] Zhou, D., Orshanskiy, S., Zha, H., Giles, C.: Co-ranking authors and documents in a heterogeneous network. In: Proc. of the Intl. Conf. on Data Mining. (2007) 739–744

[10] Celma, O.: Foafing the music: A music recommendation system based on rss feeds and user preferences. In: Proc. of the 6th Intl. Conf. on Music Information Retrieval 2005. (2005) 464–467

[11] Hausenblas, M.: Exploiting linked data to build web applications. IEEE Internet Computing **13** (2009) 68–73

[12] Yves, R., Samer, A., Mark, S., Frederick, G.: The music ontology. In: Proc. of the Intl. Conf. on Music Information Retrieval. ISMIR 2007 (2007) 417–422

[13] Jacobson, K., Raimond, Y., Gängler, T.: The similarity ontology - musim. Technical report, School of EECS, Queen Mary, University of London (2010) http://kakapo.dcs.qmul.ac.uk/ontology/musim/0.2/musim.html.

[14] Swartz, A.: Musicbrainz: a semantic web service. Intelligent Systems, IEEE **17**(1) (jan/feb 2002) 76 – 77

[15] Popescul, A., Ungar, L.H.: Statistical relational learning for link prediction. In: Proceedings of the Workshop on Learning Statistical Models from Relational Data. (2003)

[16] Taskar, B., Wong, M.F., Abbeel, P., Koller, D.: Link prediction in relational data. In: Proceedings of Neural Information Processing Systems. (2004)

[17] Swearingen, K., Sinha, R.: Beyond algorithms: An hci perspective on recommender systems. In: ACM SIGIR Workshop on Recommender Systems, New Orleans, LA, USA (2001)

[18] Tizghadam, A., Leon-Garcia, A.: Betweenness centrality and resistance distance in communication networks. Network, IEEE **24**(6) (november-december 2010) 10 –16

[19] Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. 2 edn. Pearson Education (2003)

[20] Zhao, Y., Karypis, G.: Evaluation of hierarchical clustering algorithms for document datasets. In: Proc. of the 11th Intl. Conf. on Information and Knowledge Management. CIKM ’02, New York, NY, USA, ACM (2002) 515–524

[21] Kunegis, J., Lommatzsch, A.: Learning spectral graph transformations for link prediction. In: ICML ’09: Proceedings of the 26th Annual Intl. Conf. on Machine Learning, New York, NY, USA, ACM (2009) 1–8

[22] Voorhees, E.M., Harman, D.K., eds.: TREC: Experiment and Evaluation in Information Retrieval. MIT Press (2005)

[23] Zhang, Z.K., Liu, C., Zhang, Y.C., Zhou, T.: Solving the cold-start problem in recommender systems with social tags. EPL (Europhysics Letters) **92**(2) (2010) 28002

[24] Saul, L.K., Weinberger, K.Q., Sha, F., Ham, J., Lee, D.D.: Spectral Methods for Dimensionality Reduction. In: Semi-supervised Learning. MIT Press (2006)

[25] Lathauwer, L.D., Moor, B.D., Vandewalle, J.: A multilinear singular value decomposition. Matrix Analysis and Applications **21**(4) (2000) 1253–1278

[26] Kunegis, J., Lommatzsch, A., Bauckhage, C., Albayrak, S.: On the scalability of graph kernels applied to collaborative recommenders. In: Proceedings ECAI 2008, Workshop on Recommender Systems. (2008) 35–38