Comparing Vocabulary Term Recommendations Using Association Rules and Learning to Rank: A User Study

Johann Schaible[1(✉)], Pedro Szekely[2], and Ansgar Scherp[3]

[1] GESIS – Leibniz Institute for the Social Sciences, Cologne, Germany
johann.schaible@gesis.org
[2] Information Sciences Institute, University of Southern California, Los Angeles, CA, USA
pszekely@isi.edu
[3] ZBW – Leibniz Information Center for Economics, Kiel University, Kiel, Germany
asc@informatik.uni-kiel.de

Abstract. When modeling Linked Open Data (LOD), reusing appropriate vocabulary terms to represent the data is difficult, because there are many vocabularies to choose from. Vocabulary term recommendations could alleviate this situation. We present a user study evaluating a vocabulary term recommendation service that is based on how other data providers have used RDF classes and properties in the LOD cloud. Our study compares the machine learning technique Learning to Rank (L2R), the classical data mining approach Association Rule mining (AR), and a baseline that does not provide any recommendations. Results show that utilizing AR, participants needed less time and less effort to model the data, which in the end resulted in models of better quality.

1 Introduction

Linked Data engineers typically employ Resource Description Framework (RDF) vocabularies to represent data as Linked Open Data (LOD). Reusing vocabulary terms, i.e., RDF types and properties, from existing vocabularies is considered best practice and reduces heterogeneity in data representation [1]. However, this task has challenges: First, even when data engineers are focused on a specific domain, there are many vocabularies to choose from, and second, after choosing a vocabulary, there might be multiple terms in that vocabulary that seem to be correct (considering rdfs:domain and rdfs:range information) for modeling the data in the same way. In detail, data engineers need to select the vocabulary that adequately represents the semantics of the data and maximizes compatibility with existing LOD. For example, when modeling museum data as LOD, one should preferably reuse some vocabulary terms that are also used by other museums for modeling their data. However, this is not trivial. Figure 1(a) shows an example where several vocabularies contain terms that can represent publications and person data, i.e., there are various terms describing that a resource

H. Sack et al. (Eds.): ESWC 2016, LNCS 9678, pp. 214–230, 2016.
DOI: 10.1007/978-3-319-34129-3_14

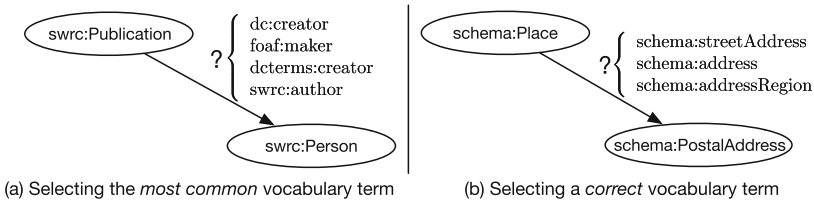


Fig. 1. *Problems when choosing vocabulary terms for reuse.* In (a) the engineer searches for a property that is most commonly used by other data providers to express the intended semantics of the relationship. In (b) the engineer searches for a property that connects two RDF types in a *correct* way considering the rdfs:domain and rdfs:range information.

of type swrc:Publication has a creator, maker, or author that is a resource of type swrc:Person. Figure 1(b) illustrates an example where multiple alternatives remain even after selecting one vocabulary, i.e., the schema.org vocabulary offers various properties which seem useful to describe that a place has a postal address. Without additional guidance, data engineers may not know which term to select. The examples in Fig. 1 illustrate the difficulties when selecting a property, but the full problem is also to select appropriate terms to represent the RDF types of the subject and object of triples in a dataset.

In this paper, we perform a user study comparing two RDF vocabulary term recommendation approaches and a baseline of no recommendations. The recommendation approaches are based on previous work [2], where the recommendations are based on the RDF types and properties used by other data providers in datasets contributed to the LOD cloud. The first approach applies the machine learning method *Learning To Rank* (L2R) to generate a list of recommended vocabulary terms. In addition, we implemented a second approach that calculates the recommendations via the data mining approach *Association Rule* (AR) mining (cf. Sect. 2). In the user study, we asked 20 participants to model three different datasets as LOD (cf. Sect. 3). For one dataset they received recommendations based on AR, for another they received recommendations based on L2R, and for the third dataset the participants received no recommendations. We measured task completion time, the recommendation-acceptance rate (number of times participants chose a recommended term vs. manual search), and the quality of the resulting LOD representation. To assess the quality, we compared the participants' LOD representation to representations generated by five different LOD experts independently, i.e., did the participants choose the same vocabulary terms as the experts. Additionally, we asked the participants to rate the two recommendation approaches and to report their level of satisfaction regarding the recommendations on a 5-point Likert-scale. The main contribution of this paper are the real-life results based on these measures, as there is no gold standard that can be used for an automatic evaluation.

The results show that the recommendations based on AR are effective, whereas the L2R-based recommendations do not seem to satisfy the participants. The task

completion times for AR (4:13 min) were significantly faster than the completion times for L2R (5:41 min) and the baseline (5:26 min). Users accepted the term recommendations from AR most of the time (4.85 out of 7 times), but not from L2R (2.05 out of 7 times). AR also led to high quality results (75 % of the selected terms were also chosen by the experts), whereas the L2R-based recommendations did not perform as good (58 % of the selected terms were chosen by the experts). The user satisfaction survey resembles similar results and indicates a preference for the AR recommendations (cf. Sect. 4 and the discussion in Sect. 5).

2 Apparatus

To perform a user study evaluating the quality of the recommendations based on L2R and AR, we integrated the recommendation service into the data integration tool Karma [3].[1] This way, participants were able to use a graphical UI in order to explore the recommendations and to model data as LOD.

In the following, we describe Karma in more detail, especially the operations that the participants of the user study need to complete the modeling tasks (cf. Sect. 2.1). Furthermore, we provide insights on how the vocabulary term recommendations are calculated with Learning to Rank (L2R) and explain the Association Rule mining (AR) based approach in detail (cf. Sect. 2.2).

2.1 Modeling Data with Karma

Karma is an interactive tool that enables data engineers to model data sources as Linked Data. It provides a visual interface to help data engineers to incrementally select RDF vocabulary terms and to build a model for their data. Figure 2 shows a screenshot of Karma with a partially specified model. The bottom part shows a table with the data being modeled, and the top part shows a graphical representation of the model. The dark ovals represent classes, and the labeled edges represent properties. The edges connecting classes to the columns in the data source are called Semantic Types, as they specify the semantics of the data in a column. For example, in Fig. 2, the semantic type for the third column labeled imageDescription specifies that the contents of the column are notes of a resource of type ecrm:E38_Image, and is represented using the pair (ecrm:P3_has_Note, ecrm:E38_Image). The semantic type "uri" specifies that the corresponding resource is described via a URI, otherwise it is specified via a blank node. The edges connecting the dark ovals represent object properties specifying the relationships between the instances of the corresponding classes. For example, the edge labeled P138i_has_representation between E22_Man-Made_Object and E38_Image specifies that the image represents the object.

To build models and to refine partially specified models, users can click on the edge labels or on the classes to edit a model. When users click on an edge label, Karma presents commands to let users change the property depicted in

[1] http://usc-isi-i2.github.io/karma/, last access 12/12/15.

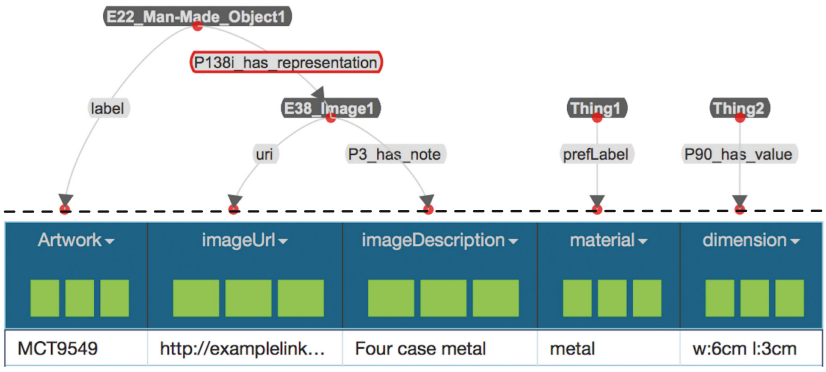


Fig. 2. *Karma User Interface.* The bottom part (below the dotted line) shows the table data being modeled and the top part shows the data's graphical representation including RDF classes (the dark ovals) and properties (the labeled edges). The edges connect classes either to table columns (Semantic Types specifying the semantics of the column data) or to other classes (object properties specifying the relationships between the instances of the corresponding classes)

the label, the source class of the edge, and the destination class of the edge (this last option is only available for object properties). Similarly, when users click on a class, Karma shows commands to change the class depicted in the oval and to add incoming or outgoing edges.

2.2 Vocabulary Term Recommendations

Recommending a vocabulary term means that the recommendation approach suggest the engineer to reuse a specific RDF class or property for describing a data entity or a relationship between data entities. To this end, our approach, which is described in [2] in detail, makes use of so-called *schema-level patterns* (SLPs) that represent the properties commonly used to connect instances of specific classes. The output is a ranked list of recommended vocabulary terms (ordered from most appropriate to least appropriate) which is calculated using the machine learning approach Learning To Rank (L2R) or the data mining approach Association Rule (AR) mining.

In the following, we explain the notion of SLPs as well as provide insights on the recommendations based on L2R and AR. For the interested reader, formalizations of SLPs as well as of the recommendation approach can be found in [2].

Schema-Level Patterns. Schema-level patterns contain three sets of vocabulary terms that describe the connection between two sets of RDF types via a set of properties. For example, the following schema-level pattern specifies that resources of type swrc:Publication are connected to resources of types foaf:Person and swrc:Person via the properties foaf:maker and dc:creator.

$$(\{\text{swrc:Publication}\}, \{\text{foaf:maker}, \text{dc:creator}\}, \{\text{foaf:Person}, \text{swrc:Person}\}) \quad (1)$$

We make use of such SLPs in a two-fold way: First an input SLP (provided by the modeling tool such as Karma) is used to describe the input for the recommendation service based on L2R and AR, and second they simulate how other data providers use vocabulary terms to represent their data. The latter can be calculated from existing data sets on the LOD cloud using a straight-forward counting algorithm.

Learning to Rank. Learning to Rank (L2R) refers to a class of supervised machine learning algorithms for inducing a ranking model based on the utilized features [4,5]. We use an L2R algorithm to produce a generalized ranking model that orders vocabulary term recommendation from most appropriate to least appropriate. To this end, we use an SLP that represents a part of the current model as input for the L2R algorithm. For each recommendation candidate x (the candidate is a vocabulary term from the set of all vocabulary terms appearing in data sets on the LOD cloud) the algorithm calculates five features representing the popularity of x, whether the engineer already uses the vocabulary of x, and how many other data sets on the LOD cloud use x in a model that is similar to the engineer's model. In detail, these features are:

Feature 1. Number of datasets on the LOD cloud using the vocabulary term x
Feature 2. Number of datasets on the LOD cloud using the vocabulary of x
Feature 3. Total number of occurrences of vocabulary term x on the LOD cloud
Feature 4. Is term x from a vocabulary that the engineer already uses?
Feature 5. The SLP-feature: Number of SLPs calculated from the LOD cloud that represent the modeled dataset and additionally contain vocabulary term x.

During a training phase, the L2R algorithm uses these features to derive a generalized ranking model. This ranking model can then be used to rank vocabulary term recommendations in new and previously unknown situations. We chose to use the L2R algorithm *Random Forest* [6] for the user study following our prior evaluation (cf. [2]), in which the algorithm achieved the best results (Mean Average Precision (MAP) of vocabulary term recommendations with MAP ≈ 0.8).

Association Rule Mining. Association Rules (ARs) are *if/then* statements that help discover relationships between data in a data repository [7]. We use ARs to identify such relationships between vocabulary terms, i.e., find vocabulary terms that are frequently used together. For example, we can use ARs to find properties that are frequently used as outgoing datatype properties from the RDF class ecrm:E22_Man-Made_Object.

In detail, our data repository for calculating ARs is the set of SLPs calculated from data sets on the LOD cloud, and the quality of the association rules depends on two measures called *support* and *confidence*. These measures depict a threshold for generating an *if/then* statement, i.e., a rule, and for our use-case they specify how often a set of vocabulary terms appear in the set of SLPs calculated from the data on the LOD cloud. Let us assume, we have two

sets of different vocabulary terms X and Y. The support specifies how often the union of X and Y occurs in the set of SLPs calculated from the data on the LOD cloud. The confidence denotes how often the union of X and Y occurs in the set of SLPs compared to the number of occurrences of X in the set of SLPs. If both the support and the confidence are higher than a self-defined threshold, then one can induce the rule $X \Rightarrow Y$ specifying "*if* the vocabulary terms in X are used, *then* the vocabulary terms in Y are used as well". For example, X comprises the vocabulary terms ecrm:E22_Man-Made_Object as well as ecrm:E38_Image and Y comprises the term ecrm:P138i_has_representation. If there are many SLPs containing the object property ecrm:P138i_has_representation together with ecrm:E22_Man-Made_Object and ecrm:E38_Image, then it is very likely that the AR algorithm generates the rule $X \Rightarrow Y$ specifying: *If* ecrm:E22_Man-Made_Object and ecrm:E38_Image are used together, *then* ecrm:P138i_has_representation is used as well. Formally, the support $Supp$ and the confidence $Conf$ for a rule $X \Rightarrow Y$ are defined as follows:

$$Supp(X,Y) = \frac{frq(X,Y)}{N} \qquad Conf(X,Y) = \frac{frq(X,Y)}{frq(X)} \tag{2}$$

where frq denotes the frequency how often the arguments occur together and N is total number of SLPs calculated from the data on the LOD cloud. Generally, the higher the manually defined minimum values, the more precise are the rules. But with higher minimum values, various rules will not be considered relevant, and by setting the minimum values too high, it is likely to produce no rules at all [7]. Thus, in order to generate all possible rules, we specified that the support and confidence must solely be greater than zero ($Supp > 0$ and $Conf > 0$).

To generate the rules, we make use of the state of the art *Apriori* algorithm [8]. It is an iterative approach to find frequent sets of vocabulary terms that are used together based on the assumption that a subset of a frequent set of terms must also be a frequent set of terms. Based on this assumption, the algorithm is able to generate a set of frequent SLPs $\{slp_1, \ldots, slp_n\}$ to calculate the association rules. These rules are defined as

$$s_i \rightarrow (slp_i - s_i)$$

where s_i is a non-empty subset of slp_i with $1 \leq i \leq n$. This means that every non-empty subset of slp_i except s_i contains vocabulary terms that can be recommended, if the terms in s_i are provided as input SLP for the recommender.

3 Experiment Setup

3.1 Evaluation Procedure

To compare term recommendations based on L2R and AR, we integrated the recommendation algorithms in Karma. We configured Karma so that the experimenter can select the recommendation approach for a participant and for

a modeling task. In our setup, Karma can be configured to offer no recommendation, or recommendations using L2R or AR in its menus for selecting vocabulary terms. We performed a within-subject [9] design user study with 20 participants (cf. Sect. 3.5). The subjects were invited to the lab to conduct some LOD modeling tasks. Each participant first performed a training modeling task to become familiar with the Karma user interface and the menus for selecting vocabulary terms. After completing the training (about 10 min), the participants worked on modeling three different data sets (6 min each) (cf. Sect. 3.2). We applied a Latin-square design to arrange the tasks and the recommendation approaches, i.e., we configured Karma so that we would be able to select one of the three recommendation conditions for each data set (no recommendation, L2R and AR). The no-recommendation condition is the baseline for comparison. We measured task completion time, the number of times participants chose a term from the recommendations vs. searching for the desired term manually, as well as the quality of the resulting LOD representation. Finally, the subjects filled in a user satisfaction questionnaire (Sect. 3.4).[2]

3.2 Modeling Assignments (Tasks)

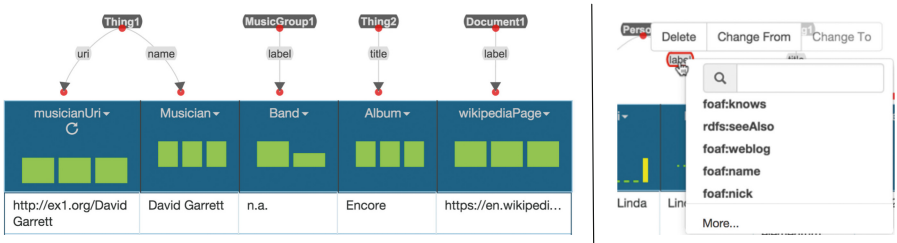
In our study, we used three datasets from three different domains, namely music, museum objects, and product offers. We chose disparate domains to reduce the overlap in the classes and properties used to model them, thereby reducing learning effects across the modeling tasks. The music dataset includes information about musicians, their recordings, and their wikipedia page. The dataset on museum objects includes information about artworks, materials, dimensions, and images of them. The product offers dataset comprises information about the seller, location, and offered items.

One concern with the user evaluation is that modeling three full datasets takes a long time. To mitigate this problem, we gave participants partially-defined models, and asked them to complete these models. For example, Fig. 3(a) illustrates the modeling assignment for the music dataset. The dataset contains information about the musician's name, their band's name, albums, wikipedia page, as well as a column containing the URI for each musician. The elements above the table represent a partially-defined model of the data: for example, the Musician column is modeled as the name of a resource of type owl:Thing, whereas the desired model is the name of a resource of type mo:MusicArtist. The instructions to the participants explained that the models are incomplete, and asked them to complete the model:

- replace the owl:Thing classes and the rdfs:label properties, and
- define object properties specifying that a musician is a member of a band, that a musician recorded an album, and that a musician has a wikipedia page.

The instructions for the other datasets are similar. To complete the modeling assignments, participants needed to perform six or seven Karma operations for

[2] Accompanying material and the modeling results can be found at https://github.com/WanjaSchaible/termpicker_karmaeval_material, last access 3/6/16.



(a) Modeling Task from the Music Domain

(b) Menu with Recommendations

Fig. 3. *Example Modeling Task and Menu with Recommended Terms.* (a) depicts the modeling task from the Music Domain. In order to fulfill the modeling task, the engineers have to: (i) replace the owl:Thing classes and the rdfs:label properties, as well as (ii) define object properties specifying that a musician is a member of a band, that a musician recorded an album, and that a musician has a wikipedia page. (b) illustrates an example menu the participants can use to select a recommended term (bold) or search for another term manually using the search box.

each dataset. When users perform editing operations to change or add properties or classes, Karma uses the pre-configured recommendation approach (including no recommendations) to populate the menus that users see for selecting vocabulary terms (cf. Fig. 3(b)). These menus have two sections, one showing the recommended terms and one showing all the terms (all properties or all classes), if the user clicks on "more". In addition, Karma also provides a search box that filters terms containing the query string.

3.3 Dataset for Recommendations

To compute the vocabulary term recommendations, we calculated the L2R features and the association rules using the BTC 2014 dataset [10]. In order to reduce the memory requirements for SLP computation, we use approximately the first 34 million triples out of the 1.4 billion triples in BTC 2014. Our subset includes data from 3,493 pay-level domains[3], including dbpedia.org and bbc.co.uk. In total, it contains about 5.5 million distinct RDF classes and properties from 1,530 different vocabularies, which were sufficient to calculate 227,010 different SLPs. In order to adequately support the user evaluation, we verified that the 34 million triples covered all the types and properties needed to model the datasets in the evaluation.

[3] A pay-level domain (PLD) is a sub-domain of a top-level domain, such as *.org* or *.com*, or of a second-level country domain, such as *.de* or *.uk*. To calculate the PLD, we use the Google guava library: https://code.google.com/p/guava-libraries/, last access 12/12/15.

3.4 Measurements

To evaluate the recommendation approaches, we measured the participants' *effort* needed to complete an assignment, as well as the *quality* of the resulting LOD model.

We collected two measures of effort. One measure is the amount of time a participant spent on each modeling task. We limited participants to six minutes for each modeling assignment. The second measure is the recommendation-acceptance rate, i.e., the number of times the participants chose a recommended vocabulary term from the menu of recommendations, as opposed to searching for terms in the menu containing the full list of vocabulary terms. Our expectation is that participants would search the full list of terms when the recommended terms were inadequate, so that faster completion times would indicate adequacy of the recommendations. As each assignment comprises seven operations, choosing a recommended vocabulary term seven times in one assignment is the maximum score, whereas choosing a recommendation zero times is the minimum score.

To measure quality, we assembled a panel of five ontology engineering experts to construct a gold-standard model for each dataset. We sent each of the modeling assignments to each expert, together with a proposed model that we created. The experts provided feedback suggesting additional vocabulary terms to specify a data entity or a relation between data entities, and suggested replacing some of the vocabulary terms we proposed with better ones. Thus, the gold standard consists of a set of vocabulary terms for each modeling operation. A participant's selection of a vocabulary term for a modeling operation is correct, if this vocabulary term is in the set of vocabulary terms defined in the gold standard for that operation. The quality score for each dataset is the fraction of correct modeling operations.

3.5 Participants of the Study

Overall, $n = 20$ participants (5 female) took part in the user study, all working with LOD in academia and 2 also in industry. The participants' profession ranges from master students (2) over research associates (14) to post doctoral researchers (3) and professors (1) with an average age range between 30–35 years. On average the participants have worked for 3.05 years with LOD. However, they rated their own expertise consuming LOD as moderate ($M = 2.8, SD = 1.1$) and publishing LOD as little ($M = 2.1, SD = 1.6$) on a 5-point-Likert scale from 1 (*none at all experienced*) to 5 (*expert*). The same applies to the participants' knowledge about the domains of the data from the three modeling tasks ($M = 2.1, SD = 1.1$ across the three domains). In addition, 13 participants had never used Karma, whereas 7 specified they are experts or have a high knowledge about Karma. Summarizing, we observe that the participants of our study had a moderate knowledge of Linked Open Data, but they were quite inexperienced regarding LOD publishing and the domain of the modeling tasks' data.

We recruited the participants from the Information Integration group at the University of Southern California. These participants were familiar with Karma

but not with the recommendation systems being evaluated in this work. We also recruited participants from GESIS - Leibniz Institute for the Social Sciences, who did not use Karma before. The participants were acquired in person or via personal email inviting them to participate within a private surrounding (laboratory or private office space), in which solely the principal investigator was present. Participants signed a consent form prior to their participation and received no compensation for their efforts.

4 Results

We illustrate and discuss the results based on the measurements considering the efficiency (Sect. 4.1), quality (Sect. 4.2), and level of satisfaction (Sect. 4.3).[4]

4.1 Effort

To calculate the average time the participants needed to complete a modeling task, we added up the times from participants who used the same recommendation approach for the modeling task and divided the sum by the number of these participants. For example, if seven out of the 20 participants modeled the music data using AR, we summed up the times these seven participants needed to complete modeling the music data, and divided it by the seven participants. Table 1 illustrates these calculated average times for each of the three data models and the combined total average time.

The results show that AR led to the fastest completion times (4:13 min), over a minute faster than the baseline of having no recommendations (5:28 min) and

Table 1. *Average Task Completion Time in Minutes.* The table shows the average time the participants needed to complete the modeling tasks followed by the standard deviation and the number of participants modeling the data with a recommendation approach

Recommendations	Music domain	Museum domain	Product Offer domain	Average in total
L2R	5:31 (0:41) - 6	5:48 (0:24) - 7	5:42 (0:25) - 7	5:41 (0:30) - 20
AR	4:50 (1:10) - 6	4:37 (1:02) - 7	3:16 (1:02) - 7	4:13 (1:03) - 20
No Rec	5:22 (0:59) - 8	5:28 (0:37) - 6	5:34 (0:42) - 6	5:28 (0:47) - 20

Table 2. *Recommendation-Acceptance Rate.* The table shows the average number of times the participants selected a recommended vocabulary term out of the number of operations and the standard deviation within the brackets.

Recommendations	Music domain	Museum domain	Product Offer domain	Average in total
L2R	2.00/7 (.89)	2.37/7 (.91)	1.67/6 (.81)	2.05/6.67 (.88)
AR	4.33/7 (.51)	4.85/7 (.69)	5.28/6 (.95)	4.85/6.67 (.95)

[4] Research data is available at http://dx.doi.org/10.7802/1206, last access 3/3/16.

about 90 s faster than using recommendations based on L2R (5:41 min). It is surprising that users took less time to complete tasks with the baseline system than with the L2R recommendations. The survey reveals participants were dissatisfied with the L2R recommendations (cf. Sect. 4.3). The results suggest that participants spent time evaluating the L2R recommendations (they used the L2R recommendations for about 2 out of 7 modeling operations), and proceeded to search for terms as they would have done in the baseline system, thus taking more time overall. The L2R recommendations use multiple features, including the popularity of a term and whether or not it is from an already used vocabulary. Thus, the recommendations sometimes include terms that are popular or from a commonly used vocabulary, but that do not fit the desired semantics. For example, instead of suggesting the datatype property foaf:name to specify the name of a person, it suggests the property foaf:knows. This occurred regardless of the domain of the data. Applying a Friedman test, we could show that the differences between the task completion time is significant ($\mathcal{X}^2 = 15.083,\ p = .001$) However, the Wilcoxon signed-rank test (with a Bonferroni correction applied) shows that recommendations based on L2R and having no recommendations at all is not significant ($Z = -1.256$, **n.s.**, $p = .209$), whereas having recommendations based on AR is significantly better (compared to L2R: $Z = -3.220,\ p = .001$; compared to the baseline of no recommendations: $Z = -3.018,\ p = .003$).

The average number of selected recommendations, which is displayed in Table 2, underpins the observations from Table 1. The participants needed more time to complete an assignment while given L2R recommendations, and did not use the recommended terms very often (on average 2.05 times out of 6.67). Compared to the number of selected recommendations based on AR (on average 4.85 selected recommendations out of 6.67 possibilities) it is significantly lower ($Z = -3.848, p < .001$).

4.2 Quality

The modeling tasks had either six or seven operations. Table 3 shows the average number of correctly selected vocabulary terms these six or seven operations. One can observe that the recommendations based on Association Rules (in total 4.97/6.67) helped the participants to select more correct vocabulary terms than the recommendations based on L2R (3.88/6.67) or not using any recommendations at all (3.57/6.67). This especially applies to modeling the data from the Product Offers domain (5.25/6), which uses mainly the schema.org vocabulary.

When searching for an appropriate vocabulary term, participants tried to find a term that matched the description in the assignment. For example, a task from the Product Offers domain states "Find a better data type property specifying that the table column contains the price of an item". The participants were instantly looking for a term that contained the string "price" in its specification, such as hasPrice. Even if the recommendations did not contain such a term, the participants easily found the property schema:price using Karma's string search and incorporated it into the model. However, not every assignment was as easy. Another example from the task with the Product Offers data states "Find a

Table 3. *Quality of the Resulting RDF Representation.* The table shows the average number of vocabulary terms that were also chosen by the LOD experts out of the number of operations and the standard deviation within the brackets.

Recommendations	Music domain	Museum domain	Product Offers domain	Average in total
L2R	4.02/7 (0.64)	3.75/7 (1.05)	3.87/6 (0.83)	3.88/6.67 (.14)
AR	4.87/7 (0.88)	4.75/7 (1.03)	5.25/6 (0.64)	4.95/6.67 (.26)
No Rec	3.84/7 (0.83)	3.37/7 (0.91)	3.52/6 (0.70)	3.57/6.67 (.24)

better RDF class specifying that the table column contains the address of a place". Only in the conditions with AR recommendations, participants were able to choose the correct vocabulary term (schema:PostalAddress). Recommendations based on L2R did not provide such good results. The quality of the resulting data is almost the same as modeling the data without any recommendations. A Friedman test shows that the differences are significant ($\mathcal{X}^2 = 13.125, p = .001$), but only the ARs seem to provide significantly better recommendations ($Z = -3.384, p = .001$).

4.3 Satisfaction

The satisfaction of the recommendation approaches was obtained using a questionnaire. We asked the participants various questions to identify their level of satisfaction considering the recommendations based on AR and L2R. First, we asked the participants to provide their general level of satisfaction on a 5-point Likert scale from "1 - Very Dissatisfied" to "5 - Very Satisfied", with a neutral position "3 - Unsure". They were rather unsure ($M = 3.00$, $SD = 1.1$) whether they were satisfied with the recommendations based on L2R. The recommendations based on AR were considered rather satisfying ($M = 4.23$, $SD = 0.7$). A Friedman test showed that this difference is significant ($\mathcal{X}^2 = 6.231, p = .013$).

In addition, we asked the participants to compare both recommendation approaches to each other. First, they were asked to compare the AR-based recommendation approach on a 5-point Likert scale from "1 - much worse than L2R" to "5 - much better than L2R". A total of 19 participant stated that the AR recommendation are either somewhat better or much better than the L2R recommendations (10 participants stated "much better than L2R" and only one stated "3 - About the same"). These results are supported by our second question to rank the two recommendation approaches (and the no recommendations baseline) from "best" to "worst" considering the help for reusing vocabulary terms. Recommendations based on AR were unanimously considered to be most helpful (each participants ranked AR at the first position). A Friedman showed that the difference in the ranking positions is significant ($\mathcal{X}^2 = 32.500, p < .001$). A post-hoc analysis using the Wilcoxon signed-rank test with Bonferroni correction illustrates that the difference between the recommendation approaches AR and L2R and between the AR-based approach and no recommendations are significant ($Z = -4.134, p < .001$ and $Z = -4.251, p < .001$, respectively).

However, the differences between getting recommendation based on L2R and getting no recommendations at all are not significant ($Z = -2.236$, **n.s.**, $p = .025$).

5 Discussion

Discussion of Results. The results indicate a quite clear picture: vocabulary term recommendations based on Association Rule (AR) mining are perceived to aid the participants more in reusing vocabulary terms than the recommendations based on the machine learning approach Learning To Rank (L2R). Both approaches use the SLPs calculated from datasets on the LOD cloud, in order to identify how other data providers model their data. However, L2R uses additional features describing the popularity of a recommendation candidate and whether it is from an already used vocabulary. Based on all these features, the list of recommendations can also contain very popular terms from a vocabulary that is already used, but that are not used by others in a similar model. Such recommendations are very likely irrelevant, and if the engineer is not sure whether the recommended terms are appropriate for reuse, she is more likely to overlook the relevant recommendations. This was also observed in the user study. In about 70 % of all assignments across the three modeling tasks, L2R recommended a correct vocabulary term at the fifth or tenth position of the list of all recommendations. The participants however either skipped it, or were not sure whether to select it, as the list of recommendations contained various terms at a better rank that seemed rather inappropriate. This uncertainty stopped most participants from selecting the correct vocabulary term from the recommendation list. On the contrary, AR does not rely on the popularity of a vocabulary term, or whether it is from an already used vocabulary. It solely exploits the information which terms other data sets on the LOD cloud used in conjunction with the terms in the modeling task. We observed that the recommendation lists generated by AR supported the participants select the correct terms. Even when participants were not aware of the existence of the correct term, they felt confident to select a term from the list. For example, the assignment to model an outgoing object property from mo:MusicArtist to mo:MusicGroup from the modeling task on music data, most participants were not aware of the existence of the property mo:member_of. Having no recommendations, they searched for "part_of" or "is_member", but with recommendation based on AR, they saw this property very quickly and selected it.

In summary, most participants stated that the L2R recommendations were useful in general, but they were unsure whether the recommended terms were appropriate for reuse. Recommendations based on AR alleviated this situation and supported the participants in selecting the correct terms. Therefore, Association Rules seem to provide the better vocabulary term recommendations, and they can be seen as the better option to reduce the heterogeneity in data representation on the LOD cloud for similar data.

Thread to Validity. Our user study was based on a within-subject design. The advantage is that it requires fewer participants. However, we needed to design three different modeling tasks that contained data from different domains, but were approximately equally difficult to model. One could argue that the modeling tasks we chose did not have the same complexity, and that the participants were able to model one task with more ease than the others. That is why we constantly changed which recommendation approach is used with which modeling task (Latin-square), and the results demonstrate that the AR recommendations performed best for all of the three tasks.

Another disadvantage of within-subject design is the problem of carry-over effects, where the first modeling task adversely influences the other. First, participants who were not familiar with Karma might have needed a longer time to complete the first modeling task, as they needed to get accustomed to Karma, but then the practice effect might make them more confident for the second and third modeling task. To address this issue, we let the participants first model a data set from the publications domain for practicing to use Karma and its menus containing the recommended vocabulary terms. The user study started only, as soon as the participants felt ready to begin with the actual modeling tasks. In addition, we changed the order of the modeling tasks and the recommendation approach for a task (based on Latin-square), such that no tasks or none of the two recommendation approaches would benefit from the carry-over effects. Second, the participants might get tired during the user study, such that it could decrease their performance on the last modeling task. To address this problem, we limited the time for each modeling task in the user study to six minutes. This way, each participant was able to complete all three modeling tasks within 18 min, such that it is less likely that their performance decreased during the last modeling task.

6 Related Work

To the best of our knowledge, no comparable user study has been conducted yet. Thus, we discuss the related word on other vocabulary term recommendation approaches that aid the engineer in reusing vocabulary terms.

Existing services that recommend RDF types and properties are generally based on syntactic and semantic similarity measures. Prominent examples are CORE (short for: Collaborative Ontology Reuse and Evaluation) [11] and the Watson [12] plug-in for the NeOn ontology engineering toolkit.[5] CORE determines a ranked list of domain-specific ontologies considerable for reuse whereas the Watson plug-in uses semantic information from a number of ontologies and other semantic documents published on the Web to recommend appropriate vocabulary terms. Whereas these services provide recommendations based on string analyses (as the input is a string value), they do not exploit any structural information on how vocabulary terms are connected to each other. In contrast,

[5] http://www.neon-project.org/, last access 12/18/15.

Falcons' Ontology Search [13] provides the engineer with such information. However, it is mainly designed to establish a general relatedness between vocabularies specifying that different vocabularies contain terms that describe similar data.

The "data2Ontology" module of the Datalift platform [14] provides suggestions to match data entities to a vocabulary term based linguistic proximity between the data entity and the vocabulary term as well as on the quality of the vocabulary using criteria from LOV [15]. LOV itself also offers a SPARQL endpoint that can be used to search for vocabulary terms that are equivalent or a sub-class-of another vocabulary term, or for properties that have a specific RDF type as rdfs:domain and/or rdfs:range. But such information needs to be encoded in the vocabulary specification, whereas using SLPs it can be directly derived from datasets on the LOD cloud. Karma itself also comprises two different types to recommend vocabulary terms. Recently, Karma comprises new algorithms that use previously created models as well as the axioms in the ontology to automate model construction [16,17]. A key difference between that work and the work presented in this paper is that the work presented in [16,17] learns from previous Karma models, while our recommendation approach learns from LOD data, irrespective of how it was created. The latest Karma work investigates algorithms to exploit patterns in the LOD cloud to automate model creation [18]. An important difference to our approach is that the data engineer first selects a set of ontologies for modeling the data. Karma then searches the LOD cloud for patterns of how the classes and properties in the selected ontologies are used in published LOD cloud, and then attempts to combine the patterns to construct a complete model for the source. The work presented in this paper does not require data engineers to select the ontologies, and its focus is on presenting recommendations interactively rather than on computing a model automatically.

7 Conclusion

This paper presented a user study comparing vocabulary term recommendations using Association Rules (AR) mining vs. using Learning To Rank (L2R). We illustrated the experiment set-up, which had the outcome that recommendations based on AR perform best. The participants were able to complete the modeling tasks with less effort, the quality of the resulting RDF representation was higher than using L2R recommendation or no recommendations, and the user satisfaction survey showed a clear favor towards the AR recommendations.

As future work, we plan to extend the implementation of the recommendation service into Karma and combine the Karma recommendation algorithms that reason about the axioms in the ontology together with the AR recommendations that don't require data engineers to pre-select a set of ontologies for modeling sources.

Acknowledgements. We would like to thank Laura Hollink, Benjamin Zapilko, Ruben Verborgh, Jérôme Euzenat, and Oscar Corcho for providing the essential contribution to generate the gold standard RDF representation of the assignment datasets.

We also thank Malte Knauf and Thomas Gottron for providing the basis of our Association Rule generation algorithm for calculating the vocabulary term recommendations. Naturally, we also thank the participants of the user study for helping us with our research.

References

1. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, San Rafael (2011)
2. Schaible, J., Gottron, T., Scherp, A.: TermPicker: enabling the reuse of vocabulary terms by exploiting data from the linked open data cloud - an extended technical report. ArXiv e-prints, December 2015
3. Knoblock, C.A., Szekely, P., Ambite, J.L., Goel, A., Gupta, S., Lerman, K., Muslea, M., Taheriyan, M., Mallick, P.: Semi-automatically mapping structured sources into the semantic web. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 375–390. Springer, Heidelberg (2012)
4. Liu, T.Y.: Learning to rank for information retrieval. Found. Trends Inf. Retr. **3**(3), 225–331 (2009)
5. Hang, L.: A short introduction to learning to rank. IEICE Trans. Inf. Syst. **94**(10), 1854–1862 (2011)
6. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
7. Zhang, C., Zhang, S. (eds.): Association Rule Mining: Models and Algorithms. LNCS (LNAI), vol. 2307. Springer, Heidelberg (2002)
8. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: ACM SIGMOD Record, vol. 22, pp. 207–216. ACM (1993)
9. Charness, G., Gneezy, U., Kuhn, M.A.: Experimental methods: between-subject and within-subject design. J. Econ. Behav. Organ. **81**(1), 1–8 (2012)
10. Käfer, T., Harth, A.: Billion Triples Challenge data set (2014). http://km.aifb.kit.edu/projects/btc-2014/
11. Fernandez, M., Cantador, I., Castells, P.: Core: a tool for collaborative ontology reuse and evaluation. In: 4th International Workshop on Evaluation of Ontologies for the Web (2006)
12. d'Aquin, M., Baldassarre, C., Gridinoc, L., Sabou, M., Angeletou, S., Motta, E.: Watson: supporting next generation semantic web applications. In: Proceedings of the IADIS International Conference WWW/Internet 2007, pp. 363–371 (2007)
13. Cheng, G., Gong, S., Qu, Y.: An empirical study of vocabulary relatedness and its application to recommender systems. 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. 98–113. Springer, Heidelberg (2011)
14. Scharffe, F., Atemezing, G., Troncy, R., Gandon, F., et al.: Enabling linked-data publication with the datalift platform. In: AAAI 2012, 26th Conference on Artificial Intelligence - Semantic Cities. (2012)
15. Vandenbussche, P.Y., Atemezing, G.A., Poveda-Villalón, M., Vatant, B.: Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web. Semant. Web J. (to appear). http://www.semantic-web-journal.net/

16. Taheriyan, M., Knoblock, C.A., Szekely, P., Ambite, J.L.: A graph-based approach to learn semantic descriptions of data sources. In: Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J.X., Aroyo, L., Noy, N., Welty, C., Janowicz, K. (eds.) ISWC 2013, Part I. LNCS, vol. 8218, pp. 607–623. Springer, Heidelberg (2013)
17. Taheriyan, M., Knoblock, C., Szekely, P., Ambite, J.L., et al.: A scalable approach to learn semantic models of structured sources. In: 2014 IEEE International Conference on Semantic Computing (ICSC), pp. 183–190. IEEE (2014)
18. Taheriyan, M., Knoblock, C.A., Szekely, P., Ambite, J.L., Chen, Y.: Leveraging linked data to infer semantic relations within structured sources. In: Proceedings of the 6th International Workshop on Consuming Linked Data (COLD 2015) (2015)