Identifying candidate datasets for data interlinking

Luiz André P. Paes Leme[1], Giseli Rabello Lopes[2], Bernardo Pereira Nunes[2,3], Marco A. Casanova[2], Stefan Dietze[3]

[1] Computer Science Institute, Fluminense Federal University, Niterói/RJ – Brazil, CEP 24210-240
{lapaesleme}@ic.uff.br

[2] Department of Informatics, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro/RJ – Brazil, CEP 22451-900
{grlopes,bnunes,casanova}@inf.puc-rio.br

[3] L3S Research Center, Leibniz University Hannover, Appelstr. 9a, 30167 Hannover, Germany
{nunes,dietze}@l3s.de

Abstract. *One of the design principles that can stimulate the growth and increase the usefulness of the Web of data is URIs linkage. However, the related URIs are typically in different datasets managed by different publishers. Hence, the designer of a new dataset must be aware of the existing datasets and inspect their content to define sameAs links. This paper proposes a technique based on probabilistic classifiers that, given a datasets S to be published and a set $\boldsymbol{T}$ of known published datasets, ranks each $T_i \in \boldsymbol{T}$ according to the probability that links between S and T_i can be found by inspecting the most relevant datasets. Results from our technique show that the search space can be reduced up to 85%, thereby greatly decreasing the computational effort.*

Keywords: Linked Data, datasets recommendation, Bayesian classifier, data interlinking

1 Introduction

Over the past years there has been a considerable movement toward publishing data on the Web following the Linked Data principles [1]. This huge effort has resulted in the creation of catalogs of Linked Data datasets, such as *the Data Hub*[4], to mainly make data findable and reusable. However, despite the fact that extensive list of open datasets are available in these catalogs, most of the data publishers still connects their datasets to other popular datasets, such as DBpedia[5], Freebase [6] and Geonames[7]. Although the linkage with popular datasets

[4] http://datahub.io/
[5] http://dbpedia.org/
[6] http://www.freebase.com/
[7] http://www.geonames.org/

would allow us to explore external resources, it would fail to cover highly specialized information. Basically, as described in [2], linkage with popular datasets is favoured because of two main reasons: (i) the difficulty in finding related open datasets; and (ii) the strenuous task of discovering instance mappings between different datasets.

Catalogues of linked data describe the content of datasets in terms of the update periodicity, authors, SPARQL endpoints, linksets with other datasets, amongst others, as recommended by *W3C Void Vocabulary* [3]. However, catalogues by themselves do not provide any explicit information to help the URI linkage process. Therefore, due to the lack of information or of an heuristic for selecting datasets, the search for links should be done almost by an exhaustive search of all datasets in the catalogues, which is rather unfeasible. On the other hand, catalogues may provide data for algorithms which would reduce the number of datasets to inspect.

This paper proposes a probabilistic classifier based on Bayesian theory that, given a dataset S to be published and a set $\boldsymbol{T}$ of known published datasets, ranks each $T_i \in \boldsymbol{T}$ according to the probability that it will be possible to define links between URIs of S and T_i, so that most of the links, if not all, could be found by inspecting the most relevant datasets in the ranking. We refer to this technique as *dataset recommendation*.

The rest of the paper is organized as follows. Section 2 presents the most relevant related work in the area. Section 3 introduces our proposed technique based on probabilistic classifiers. Section 4 presents the experiments that we have conducted to test our technique. Section 5 presents some performance analysis. Finally, section 6 presents conclusions and future work.

2 Related Work

The recommendation for the interlinking of datasets in the Linked Data domain is a research area still initial but in expansion. Many recommendation systems have been studied and published, nevertheless most of them have been applied to e-commerce [4], social networks [5], professional jobs [6], amongst others, but they rarely have been applied to linked data recommendation. There are few approaches developed specifically for this purpose. The most related works are described in this section.

In general, the approaches to construct recommendation systems can be classified according to the filtering technique, as collaborative, content-based and hybrid [7, 8]. The first approach collect evidences for recommendation from similar behavior, for instance, if a group of users are interested in buying science fiction books, then recommend buying science fiction books for every one similar to them. The content-based approach is based on the preferences of users, for example, if a user has a collection of classical songs, then songs of the same genre are suggested. The hybrid approach combine the previous two to take advantage of their benefits.

In Open Innovation (OI) scenarios, where companies outsource tasks to a network of collaborators, Damljanovic et al. [9] present a Linked Data-based concept recommendation method for topic discovery that is used to match innovation problems and experts. Their approach exploits reference datasets to find direct or laterally related data from the user and problem descriptions. Although they tackle the problem of recommending experts to open innovation problems, their work is similar to ours since we focus on recommending the most relevant datasets to a data publisher.

Nikolov et al. [10, 2] propose an approach to identify relevant datasets for data linking. Their approach has two main steps: (i) searching for potential relevant entities in other datasets using as keywords a subset of labels in the new published dataset; and (ii) filtering out irrelevant datasets by measuring semantic concept similarities obtained by applying ontology matching techniques. The focus of their work is recommendation for the linking process. Thus, in the filtering step, they consider only the most relevant datasets based on their semantic similarity.

Lóscio et al. [11] propose the recommendation of relevant datasets for specific applications, i.e., sources that contribute to answering queries posed to the applications. The authors argue that a dataset may contribute to answering application queries, but the response may not be according to the user requirements. Thus, they propose the discovery of relevant datasets in a specific domain using information quality (IQ) as multidimensional criteria. Their recommendation function estimates a degree of relevance of a given dataset based on the IQ criteria of correctness, schema completeness and data completeness.

Oliveira et al. [12] use application queries and user feedback to the discover relevant datasets in Linked Data. The applications queries help filter datasets that are potentially strong candidates to be relevant and the user feedback helps analyze the relevance of such candidates. They argue that, by considering both aspects, one obtains better recommendations. While the works by Lóscio and Oliveira aim at recommending datasets with respect to user queries, Nikolov focuses on the recommendation for the linking process, which is closer to our approach.

Finally, Kuznetsov [13] presents a description of a data integration system for the Linked Open Space. In his work, he describes a modular architecture consisting mainly of a "linking system" responsible for (i) discovering relevant datasets for a given dataset and (ii) creating instance level linkage. Relevant datasets are discovered by using the *referer* attribute available in HTTP message header as described in [14] and ontology matching techniques are used to reduce the number of pairwise comparisons for instance matching. However, the work does not present any practical experiment to test the techniques. Although the approach described in this paper addresses the first step of the linking system described, we addressed (ii) in previous works [15–17].

Most of the related work presented in this section use techniques as keyword-based search, schema matching and ontology matching, while others adopt user feedback and information quality as criteria of relevance. By contrast, our ap-

proach considers the interlinking amongst data sources as a "high" level information, and does not perform analysis at the instance or schema levels. We do not explicitly consider a user query, and our recommendation function aims at recommending datasets that are candidates to be interlinked with a new dataset being published in the Web of Data. The inputs of our approach are the previous linkages of the candidates and some known linkage of the new dataset. For the generation of recommendation ranking, we propose a collaborative approach which uses Naive Bayes assumptions. To the best of our knowledge there is no previous work in this sense.

3 Proposed Technique

Instead of providing a restricted list of recommendations, we define the task of recommending datasets as a task of ranking existing datasets according to its relevance to URI linkage. Thus, it is at the user's discretion to decide how far he/she goes into the ranking in search for links. More precisely, the problem we address is:

> *Given a dataset S, calculate a rank score for each dataset T_i $(i = 1, ..., m)$ in a known set $\boldsymbol{T}$ of datasets. The rank score should favor those datasets with the highest chance of containing resources that could be linked to resources of S.*

We used metadata about connections between datasets available in catalogues as the source of evidences of relevance. The interconnection of datasets can be modeled as a directed graph $G = \{V, E\}$ where the nodes V are the datasets in $\boldsymbol{T}$ and there is an edge from A to B in E if and only if there is an RDF triple $t = (s, p, o) \in A$ whose subject s is a resource of A and whose object o is a resource of B; we say that t is a *link* from A to B. Furthermore, if there is an edge from A to B in E then we say that A is *connected* to B. Note that there can be only one edge from A to B, even if there are multiple distinct RDF triples linking A to B.

The actual evidences of relevance are extracted from the correlation between connections. For example, if datasets connected to DBLP, ACM and CiteSeer are very often connected to OAI (Open Archives Initiative) then suggest OAI for those datasets which are connected to DBLP, ACM and CiteSeer but not to OAI. Intuitively, a high degree of correlation between the sets of connections {DBLP, ACM and CiteSeer} and {OAI} may indicate that OAI is relevant for any dataset which is connected to DBLP, ACM and CiteSeer.

One can argue, at this point, that such correlation can be sometimes obvious inside a specific community, for example, datasets such as DBLP, ACM, CiteSeer, IEEE, RAE, PubMed, etc. can be frequently correlated in the bibliographic domain. Moreover, generic datasets, such as DBPedia and Geonames, are correlated with quite a few datasets, as they provide generic resources and act as hubs for most datasets. However, as the Linked Data Web grows, the familiarity with the available datasets of specific domains can decrease and the

generic datasets can become exceptions. Therefore, we believe that the correlation between connections, the basis of our recommendation technique, is an appropriate approach to the problem.

One can define the rank score function as a conditional probability:

$$score(T_i, S) = P(T_i|S). \tag{1}$$

where S is the event of selecting S as the dataset one wants to make recommendations to and T_i is the event of containing URIs in T_i that could be linked to URIs of S. As required, this score function favors those datasets with the highest probabilities of record linkage with S.

One can rewrite the above expression using Bayes's rule as follows:

$$score(T_i, S) = \frac{P(S|T_i)}{P(S)} P(T_i). \tag{2}$$

As in Bayesian classifiers [18, 19], one can represent S as a bag of features $F = \{f_1, ..., f_n\}$ and rewrite once more the above expression:

$$score(T_i, S) = \frac{P(\{f_1, ..., f_n\}|T_i)}{P(\{f_1, ..., f_n\})} P(T_i). \tag{3}$$

By the naive Bayes assumptions [18, 19] $P(\{f_1, f_2, ..., , f_n\}|T_i)$ can be calculated by multiplying probabilities. Moreover, because $P(S)$ is the same for every T_i and to make the computation simpler, the score function can be rewritten again:

$$score(T_i, S) = \left(\prod_{j=1..n} P(f_j|T_i) \right) P(T_i). \tag{4}$$

$$score(T_i, S) = \left(\sum_{j=1..n} log(P(f_j|T_i)) \right) + log(P(T_i)). \tag{5}$$

where we define that $\sum_{j=1..n} log(P(f_j|T_i)) = 0$, for $n = 0$, i.e., when S does not have any feature, the score function takes into account only the probability of connections to T_i. In this case, the most popular datasets , such as DBPedia, Geonames, etc. will be favored by the score at the expense of the more highly appropriate datasets. We are aware that the recommendation may not be quite accurate in such borderline cases, but we believe that a popularity-based ranking is preferable to no ranking at all, when nothing is known about S.

Equation 5, therefore, defines the final score function that induces the ranking of the datasets.

By using the maximum likelihood estimate of the probabilities [19] in a training dataset, the above probabilities can be calculated by the following ratios.

$$P(f_j|T_i) = \frac{count(f_j, T_i)}{\sum_{j=1}^{n} count(f_j, T_i)}. \quad (6)$$

$$P(T_i) = \frac{count(T_i)}{\sum_{i=1}^{m} count(T_i)}. \quad (7)$$

where $count(f_j, T_i)$ is the number of occurrences in the training set where datasets containing feature f_j are connected to a dataset T_i, $count(T_i)$ is the number of datasets connected to T_i in $\boldsymbol{T}$ disregarding the feature set. So, for any new dataset S represented by a set of features F, possibly empty, the rank position of each one of the existing datasets can be computed by equation (5).

So far we have used a generic set of features $F = \{f_1, ..., f_n\}$ of S without indicating how to apply it to the intuition that correlated datasets provide evidences on the degree of relevance of a dataset T_i to S. In the experiments of section 4, we used known connections of S as the feature set. In section 4, we also avoided the borderline case where no feature is known in order to analyze the effects of knowing some connections of S on the recommendations.

The maximum likelihood estimate can be computed in a training dataset as follows. Let,

- $Conn$ be a set of ordered pairs (T_j, T_i) indicating that a dataset T_j is connected to a dataset T_i in a training dataset.
- $Corr$ be a set of ordered triples (w, f_j, T_i) indicating that if a dataset w is connected to f then it is connected to T_i as well in the training dataset.

Fragments of $Conn$ and $Corr$ are depicted in Table 1.a and 1.b. Note that $Corr$ can be created from $Conn$ by making all possible combinations two by two of the connections of each distinct T_j.

Note that $count(f_j, T_i)$ in equation (6) can be computed from $Corr$ by counting distinct occurrences of pairs (f_j, T_i) and that $count(T_i)$ in equation (7) can be computed by counting distinct pairs (w, T_i). Equations (6) and (7) are then straightforward computed from these values.

4 Experiments

We tested the recommendation method with data available in the Data Hub catalogue[8], a repository of metadata of open datasets, in the style of Wikipedia. The Data Hub catalogue stores metadata of the datasets present in the Linking Open Data (LOD) cloud diagram [20]. It is openly editable and is running a

[8] `http://datahub.io`

Table 1. Fragment of the existing connections of the Association for Computing Machinery (ACM) dataset in the Data Hub catalogue (*left side*) and simultaneous connections of ACM based on $Conn$ and $Corr$.

$Conn$	
T_j	T_i
acm	dblp
acm	citeseer
acm	ieee

(a)

$Corr$		
w	f_j	T_i
acm	dblp	citeseer
acm	ieee	citeseer
acm	citeseer	dblp
acm	ieee	dblp
acm	dblp	ieee
acm	citeseer	ieee

(b)

data cataloguing software (CKAN)[9] maintained by the Open Knowledge Foundation[10].

A multivalued property named *relationships*, available in the catalogue vocabulary and exposed by the REST API[11] of the catalogue, whose domain is the complete set of catalogued datasets, allows one to assert that a dataset T_j is connected to a dataset T_i by adding the assertions $T_j[relationships] = \_node$ and $\_node[object] = T_i$ to the catalogue data. We used the property *relationships* to extract the relation $Conn = (T_j, T_i)$.

To evaluate the technique, we adopted the 10-fold cross validation approach. The $Conn$ relation is split into training and testing sets in ten different ways. Testing partitions contain datasets with known connections which are used as feature sets and ground truth connections for assessment of the ranking. Training partitions contain datasets to compute the probabilities in equations (6) and (7) . The overall performance is taken as the average of the performances in the testing partitions. We stress that the references between datasets were extracted from existing metadata (property *relationships*) in the Data Hub catalogue.

In order to define a performance measure, recall that the technique aims at reducing the search space for defining links by ranking existing datasets. Without an appropriate ranking of datasets, the discovery of new connections to a dataset S requires the search for links possibly in all known datasets, which is unfeasible. With the appropriate ranking, datasets more likely to contain connections from S will be better positioned in the ranking and the search could be concentrated on those datasets at the top of the ranking. It is clear, however, that the reduction in effort will only be good if one can search only a small portion of the ranking. As we are going to show later, the results indicate that, on the average, only 15% of the ranking was needed to find all connections of S.

[9] `http://ckan.org`
[10] `http://okfn.org`
[11] `http://datahub.io/api/rest/dataset/[datasetid]`

From the above, we defined a performance measure based on the ranking positions of the discovered connections of datasets in the testing partitions. Intuitively, for example, if the less relevant discovered connection of a dataset was in the tenth position in a rank of one hundred datasets, it would mean that the search space for links could be reduced to 10% of the complete set of datasets, since no more connections would be found further down the ranking.

More formally, we define the performance measure as follows. Let,

- S be a dataset in a test partition
- C be the set of connections of S in the test partition
- $\{F, R\}$ be a partition of C
- F be the set of connection chosen as features of S
- R be the set of connections to be found

Table 2.a, extracted from the Data Hub catalogue at `http://datahub.io/api/rest/dataset/rkb-explorer-acm`, and two different choices of feature sets F_1 and F_2, shown in Tables 2.b and 2.c. For each set of features, one wants to find the remaining connections of the ACM dataset.

Table 2. Existing connections of the Association for Computing Machinery (ACM) in the Data Hub catalogue (a), two sets of features of ACM (b and c).

C			
budapest	citeseer	cordis	courseware
curriculum	dblp	dbpedia	deepblue
dog-food	dotac	ecs-eprints	eprints
epsrc	eurecom	freebase	ft
ibm	ieee	irit	kisti
laas	newcastle	nsf	oai
pisa	rae2001	resex	risks
roma	southampton	ulm	wiki

(a)

F_1
deepblue
eurecom
nsf
resex

(b)

F_2
dblp
ecs-eprints
laas
rae2001

(c)

Start by computing the ranking $score(T_i, S)$ of all datasets T_i in the training partition for S represented by F_j, $(j = 1, 2)$ and let P_j be the position furthest down in the ranking among all the positions of the datasets in $R_j = C - F_j$. The rankings for both feature sets are shown in Table 3.a and 3.b respectively.

Let P be the number of datasets that must be inspected, if one wants to find all connections of S following the ranking. Let M be the total number of distinct datasets in the training partition, then $P' = P/M$ is the proportion of datasets necessary to find all connections of S. The smaller the proportion is, the better the ranking will be. If one repeats the above process for each S in a testing partition and for each different partition $\{F, R\}$ of S, one can calculate the arithmetic mean of P' in a test partition p, denoted by $\overline{P'_p}$. If we repeat the

process for all test partitions one can take the arithmetic mean of $\overline{P'_p}$, denoted by $\overline{\overline{P'_p}}$, as the overall performance.

In our running example, we have that $M = 768$. The result for the set F_1 shows that the worst dataset is in thirty-sixth place and, therefore, the performance in this case is calculated as $36/768 = 4,69\%$. On the other hand, if we take the feature set F_2, the performance is $128/768 = 16,67\%$.

Table 3. Fragment of the recommendation ranking given that ACM was represented by the set of features F_1 (a) and fragment of a second ranking given the set of features F_2 (b).

Fragment of ranking 1	
position	dataset
4	freebase
5	ecs-eprints
6	kisti
7	southampton
9	roma
10	wiki
11	dblp
14	budapest
20	oai
22	citeseer
27	ibm
29	ieee
32	risks
33	epsrc
36	dbpedia

(a)

Fragment of ranking 2	
position	dataset
6	wiki
7	eprints
8	oai
9	dotac
10	citeseer
12	southampton
15	ieee
17	budapest
25	curriculum
29	ibm
58	eurecom
60	dbpedia
67	risks
127	deepblue
128	freebase

(b)

5 Performance analysis

Recall from the previous section that

- S is a dataset in a test partition
- F is the set of connection chosen as features of S
- R is the set of connections to be found
- P is the number of datasets that must be inspected, if one wants to find all connections of S following the ranking
- P' is the proportion of datasets necessary to find all connections of S
- $\overline{P'_p}$ is the arithmetic mean of P' in a test partition p
- $\overline{\overline{P'_p}}$ is the the arithmetic mean of $\overline{P'_p}$ over all test partitions

- $\overline{P_p}$ is the arithmetic mean of P in a test partition p
- $\overline{\overline{P_p}}$ is the the arithmetic mean of $\overline{P_p}$ over all test partitions

Also recall that, given a dataset S, the purpose of the technique is to produce a ranking of datasets such that the closer a dataset S' is to the top, the higher the chances that S' contains resources that could be linked to resources of S.

In this section, we analyze the followings aspects of the recommendation technique:

Q1 Given a ranking of recommended datasets, how far down the ranking a dataset must be to contribute with new links? That is, what is the ranking efficiency?

Q2 What is the effect of the size of the feature set on the ranking efficiency. Would a bigger feature set lead to more precise rankings?

Q3 What is the effect of increasing the number of datasets to be found (or recommended)? If the number of irrelevant datasets increased relatively to what had to be found, then the method would be less efficient for large volumes of recommendations.

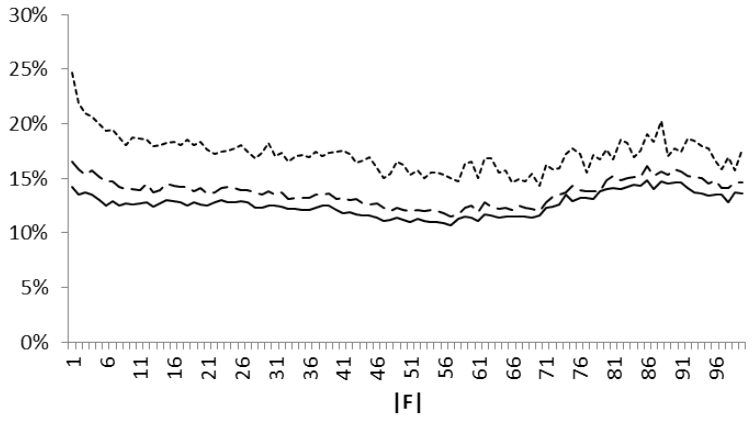

Fig. 1. Performance function $\overline{\overline{P_p}}(|F|)$, where $|F|$ is the size of the feature set: (a) *dotted line*: Performance computed with rank positions of all datasets in R; (b) *dashed line*: Performance computed by discarding the worst rank position of the datasets in R; (c) *solid line*: Performance computed by discarding the two worst rank positions of the datasets in R.

To answer the first two questions, we computed the average performance, $\overline{\overline{P'_p}}$, as a function of the size of the feature set $|F|$ (shown as *dotted line* in Fig. 1). One can see that the efficiency of the ranking is approximately 20%, no matter what is the size of the feature set $|F|$. Hence, the user may consider only the top 20% datasets in the ranking when searching for links.

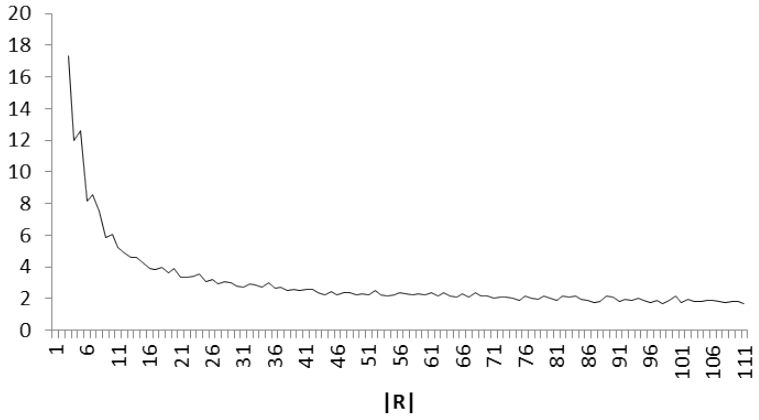

Fig. 2. Performance function $\overline{\overline{P_p}}(|R|)/|R|$, where $|R|$ is the size of the set of connections to be found. P is used instead of P' to compute the arithmetic means.

However, we realized that outliers caused by insufficient data distorted the average performance. Indeed, the size of training partitions was not always enough to compute the probabilities. Hence, for each set R, we computed a new performance measure that considers only the $(|R|-1)th$ best positions (shown in *dashed line* in Fig. 1)). To confirm that the first performance curve was really disparate, we computed a third performance measure that considered only the $(|R|-2)th$ best positions (shown as *solid line* in Fig. 1)). Note that the gap between the dotted curve and the dashed curve is greater than the gap between the dashed curve and the solid curve, which justifies the hypothesis. To summarize, Fig. 1 indicates that the performance measure is indeed better at about 15%, that is, the user may in fact consider only the top 15% datasets in the ranking when searching for links. This is the first relevant contribution of this paper.

To answer the third question, we analyzed the behavior of the ratio $\overline{\overline{P_p}}(|R|)/|R|$ (shown in Fig. 2). Note that here we used P instead of P' to compute the arithmetic means. This is because $|R|$ is also an absolute number of datasets. Therefore, $\overline{\overline{P_p}}(|R|)$ denotes the average worst position to find a total of $|R|$ datasets. It does not restrict the number of features of S. Actually, to compute $\overline{\overline{P_p}}(|R|)$ we considered datasets with any number of features. For instance, to compute $\overline{\overline{P_p}}(5)$ we selected all S in the testing partitions with $|C| > 5$ and for all partitions $\{F, R\}$ of each S where $|R| = 5$ we computed P. After that, we computed $\overline{\overline{P_p}}(5)$.

Note that $\overline{\overline{P_p}}(|R|)/|R|$ tends to be approximately equal to 2, which means that the number of datasets that should be inspected is twice the number of connections that have to be found. This result shows that the computational effort to find connections depends exclusively on the number of connections to

be found in a proportion of 2:1. We stress that it is not an intuitive conclusion. We expected that, as the number of datasets to discover grew, the proportion of irrelevant datasets amongst the relevant ones would increase faster. This would negatively impact the recommendation algorithm. Unlike our expectation the number of irrelevant datasets increased in the same proportion. This is the second and last relevant contribution of this paper.

6 Conclusions

Aligned with the Linked Data recommendations [1], and the W3C VoID Vocabulary [3] we proposed a ranking technique that can be used to recommend datasets and that can dramatically reduce the computational effort to find connections amongst datasets. The technique proved to reduce about 85% of the search space and to make the computational effort of finding datasets proportional to the number of datasets to be found.

As future work, we plan to explore how to improve the results by taking into account the information domain of the datasets. Given a dataset S, the other datasets could be clustered by information domain and valued proportionally to the information domains of S. To achieve this, one would have to add a preliminary classification step to find all possible information domains of S.

References

1. Berners-Lee, T.: Linked Data. In: Design Issues. W3C (July 2006)
2. Nikolov, A., d'Aquin, M., Motta, E.: What Should I Link to? Identifying Relevant Sources and Classes for Data Linking. In: Proceedings of the Joint International Semantic Technology Conference (JIST), Springer Berlin Heidelberg (2012) 284–299
3. Alexander, K., Cyganiak, R., Hausenblas, M., Zhao, J.: Describing Linked Datasets with the VoID Vocabulary. W3C (March 2011)
4. Schafer, J.B., Konstan, J., Riedi, J.: Recommender systems in e-commerce. In: Proceedings of the 1st ACM Conference on Electronic Commerce (EC). (1999) 158–166
5. Konstas, I., Stathopoulos, V., Jose, J.M.: On social networks and collaborative recommendation. In: Proceedings of the 32nd International ACM Conference on Research and Development in Information Retrieval (SIGIR). (2009) 195–202
6. Malinowski, J., Keim, T., Wendt, O., Weitzel, T.: Matching People and Jobs: A Bilateral Recommendation Approach. In: Proceedings of the 39th Annual Hawaii International Conference on Print (HICSS). (January 2006) 137c–137c
7. Ricci, F., Rokach, L., Shapira, B., Kantor, P.B.: Recommender Systems Handbook. Springer (2011)
8. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender systems: an introduction. New York : Cambridge University Press (2011)
9. Damljanovic, D., Stankovic, M., Laublet, P.: Linked Data-Based Concept Recommendation: Comparison of Different Methods in Open Innovation Scenario. In: Proceedings of the 9th Extended Semantic Web Conference (ESWC), Springer Berlin Heidelberg (2012) 24–38

10. Nikolov, A., d'Aquin, M.: Identifying Relevant Sources for Data Linking using a Semantic Web Index. In: Proceedings of the 4th Linked Data on the Web Workshop (LDOW). (2011)
11. Lóscio, B.F., Batista, M., Souza, D.: Using information quality for the identification of relevant web data sources. In: Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services. (2012) 36–44
12. de Oliveira, H.R., Tavares, A.T., Lóscio, B.F.: Feedback-based data set recommendation for building linked data applications. In: Proceedings of the 8th International Conference on Semantic Systems (I-SEMANTICS). (2012) 49–55
13. Kuznetsov, K.A.: Scientific data integration system in the linked open data space. Programming and Computer Software **39**(1) (January 2013) 43–48
14. Mühleisen, H., Jentzsch, A.: Augmenting the Web of Data using Referers. In: Proceedings of the 4th Linked Data on the Web Workshop (LDOW). (2011)
15. Leme, L.A.P.P., Casanova, M.A., Breitman, K.K., Furtado, A.L.: Instance-based OWL schema matching. In: Proceedings of Enterprise Information Systems: 11th International Conference, Springer (2009) 14–26
16. Leme, L.A.P., Brauner, D.F., Breitman, K.K., Casanova, M.A., Gazola, A.: Matching object catalogues. Innovations in Systems and Software Engineering **4**(4) (2008) 315–328
17. Nunes, B.P., Mera, A., Casanova, M.A., Breitman, K.K., Leme, L.A.P.P.: Complex Matching of RDF Datatype Properties. Technical Report MCC12/11 (December 2011)
18. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (January 2011)
19. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. The MIT Press (2002)
20. Cyganiak, R., Jentzsch, A.: Linking Open Data cloud diagram