Recommending Tripleset Interlinking through a Social Network Approach

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

[1] Department of Informatics, Pontifical Catholic University of Rio de Janeiro,
Rio de Janeiro/RJ – Brazil, CEP 22451-900
{grlopes,bernardo,casanova}@inf.puc-rio.br
[2] Computer Science Institute, Fluminense Federal University,
Niterói/RJ – Brazil, CEP 24210-240
lapaesleme@ic.uff.br
[3] L3S Research Center, Leibniz University Hannover, Appelstr. 9a, 30167 Hannover,
Germany
dietze@l3s.de

Abstract. Tripleset interlinking is one of the main principles of Linked Data. However, the discovery of existing triplesets relevant to be linked with a new tripleset is a non-trivial task in the publishing process. Without prior knowledge about the entire Web of Data, a data publisher must perform an exploratory search, which demands substantial effort and may become impracticable, with the growth and dissemination of Linked Data. Aiming at alleviating this problem, this paper proposes a recommendation approach for this scenario, using a Social Network perspective. The experimental results show that the proposed approach obtains high levels of recall and reduces in up to 90% the number of triplesets to be further inspected for establishing appropriate links.

Keywords: Linked Data, Recommender Systems, Social Networks.

1 Introduction

One of the design principles of Linked Data is to include URIs linkages [1], or simply *links*, which allow the "navigation" among triplesets and the discovery of related resources and additional data [2]. Therefore, an important task in the publishing process of a tripleset t involves the selection of triplesets for which one may define links with t.

However, this is a non-trivial task. Indeed, a fully manual process requires considerable effort from the data publisher and will become impractical as the number of triplesets grows. According to Nikolov et al. [3], the selection of a tripleset u for which one may define links with t can be influenced by three factors: (i) *degree of overlap* - the number of resources of u related to resources of t; (ii) *additional information provided by the tripleset* - the amount of additional information u can provide for the resources of t; and (iii) *popularity of the tripleset* - how easy it will be for t to be discovered because it has links to popular triplesets.

We refer to the problem of the discovery and selection of triplesets for which one may define links with a given tripleset as the *tripleset recommendation problem.*

In this paper, we propose to address the tripleset recommendation problem using strategies borrowed from Social Networks. We introduce a procedure that receives as input a tripleset t and a set of triplesets S, and returns a ranked list of triplesets $u \in S$ such that links from t to u are more likely to be defined for the triplesets in the beginning of the list. Therefore, the effort of creating links from t to triplesets in S would be reduce, since one would have to analyze just the first few triplesets in the ranking. The procedure we propose could be used as an initial filtering phase to other more costly recommendation techniques based, for example, on schema and ontology matching, which might be applied only to the better ranked triplesets.

To generate the ranked list, the procedure uses a recommendation function adapted from link prediction measures used in Social Networks. Informally, we say that a tripleset t is *connected* to another tripleset u iff there are at least one link between resources from t to resources in u. Basically, to adapt the link prediction measures, we interpreted the connections between triplesets as relational ties and the triplesets as the actors. In the paper, we evaluate the performance of two link prediction measures, using data obtained from the Data Hub catalogue.

In general, recommendation systems [4] alleviate problems associated with information overload [5]. Recommendation systems aim at suggesting items to users based on their interests, i.e., from the analysis of their profiles. Currently, many e-commerce Web sites use this type of system to rank suggestions of their products to potential buyers [6]. It is noteworthy that such systems not only gained prominence in e-commerce [7], but also in several application areas. Indeed, such systems have been applied to different domains such as recommendation of books [8], restaurants [9], movies [10], news [11] and social networks [12]. In particular, in the context of Social Networks, measures based on analysis of the relational ties between actors have been used to recommend links between actors [13–15].

The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 details our recommendation approach. Section 4 shows an experimental evaluation and discusses the results obtained. Section 5 presents the conclusions and suggests further work.

2 Related Work

Recommendation of triplesets to be interlinked in the Linked Data domain is a research area in expansion. However, there are still few approaches developed specifically for this purpose. In this section, we briefly review the research more closely related to ours.

Nikolov et al. [3, 16] propose an approach to identify relevant triplesets for data linking. Their approach establishes two main steps: (i) searching for potentially relevant resources in other triplesets using as keywords a subset of labels in the new published tripleset; and (ii) filtering out irrelevant triplesets by measuring semantic similarities applying ontology matching techniques. In the filtering step, they consider only the triplesets with higher degrees of semantic similarity, discarding the others.

The following two references [17, 18] aim at recommending triplesets relevant to answer queries expressing the user requirements. Lóscio et al. [17] propose the recommendation of relevant triplesets that contribute for answering queries posed to an application. The authors argue that a tripleset may contribute to answer queries of an application, but the returned response may not meet the user requirements. Thus, they propose to discover triplesets relevant for applications in a specific domain using information quality (IQ) as multidimensional criteria. Their recommendation function estimates a degree of relevance of a given tripleset based on the following IQ criteria: correctness, schema completeness and data completeness.

Oliveira et al. [18] use application queries and user feedback to discover relevant triplesets in Linked Data. The application queries help filter triplesets that are potentially strong candidates to be relevant and the user feedback helps analyze the relevance of such candidates. They argue that the consideration of both queries and user feedback helps recommending triplesets related to the user requirements.

To summarize, all previous works perform an analysis at the instance or schema levels, using techniques such as filtering by keyword-based searches, schema and ontology matching, user feedback and information quality.

Our proposed approach differs from these since it considers the links among triplesets as a "high" level information and it does not require an analysis at the instance or schema levels.

Our recommendation function aims at recommending candidate triplesets $u \in S$ to a tripleset t, such that t could possibly be interlinked with u. The inputs of our approach are the previous links among the candidate triplesets and some known triplesets that t can be interlinked with. For the generation of the recommendation ranking, we propose to apply link prediction measures adopted in Social Networks to the Linked Data context. To the best of our knowledge there is no previous work that takes this approach.

3 A Recommendation Approach

3.1 Recommendation Procedure

Briefly, recall that an *RDF triple* is a triple of the form (s, p, o), where s is the *subject* of the triple, which is an RDF URI reference or a blank node, p is the *predicate* or *property* of the triple, which is an RDF URI reference, and o is the *object*, which is an RDF URI reference, a literal or a blank node.

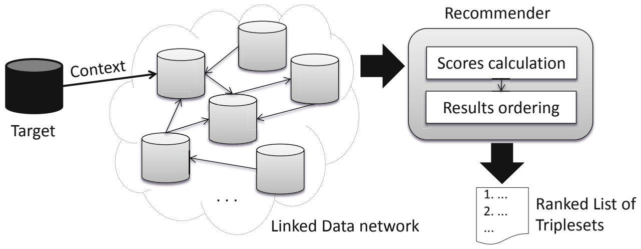


Fig. 1. Schematic description of the recommendation procedure

A *tripleset* t is a set of RDF triples. A resource identified by an RDF URI reference s is *defined* in t iff s occurs as the subject of a triple in t.

Let t and u be two triplesets. A *link* from t to u is a triple of the form (s, p, o), where s is an RDF URI reference identifying a resource defined in t and o is an RDF URI reference identifying a resource defined in u; we say that (s, p, o) *interlinks* s and o. We also say that t can be *interlinked* with u iff it is possible to define links from t to u.

A *Linked Data network* is a graph $G = (S, C)$ such that S is a set of triplesets and C contains edges (t, u), called *connection* from t to u, iff there is at least one link from t to u; we also say that t *points at* or *references* u. Note that there can be only one edge from t to u, even when there are multiple distinct links from t to u.

Let $G = (S, C)$ be a Linked Data network. The *context* of a tripleset $u \in S$, denoted C_u, is the set of all $v \in S$ such that $(u, v) \in C$; and the *inverse context* of $u \in S$, denoted C'_u, is the set of all $v \in S$ such that $(v, u) \in C$.

Our recommendation procedure analyzes a Linked Data network much in the same way as a Social Network. The inputs of our recommendation procedure are (see Figure 1):

- a *Linked Data network* $G = (S, C)$
- a *target tripleset* t not in S (intuitively the user wishes to define links from t to the triplesets in S)
- a *target context* C_t for t consisting of one (or more) triplesets u in S (intuitively the user knows that t can be interlinked with u).

The procedure outputs a list L of triplesets in S, called a *ranking*. Intuitively, the triplesets in the initial positions of the ranking have a higher probability that resources in t can be interlinked with their resources.

The procedure adds a tripleset $u \in S$ to the ranking L iff $C_t \cap C_u$ is not empty, where, we recall, C_t is the context of t (given as input to the procedure) and C_u is the context of $u \in S$ (defined from the Linked Data graph).

To order the triplesets in L, the procedure estimates score values between t and the triplesets u in L: the higher the score of a tripleset u, the topmost u will be in the ranking. Intuitively, the score of a tripleset u is a predicted value of the relevance of u with respect to the probability of defining links from t to u. As stated before, to estimate the scores, the procedure applies measures used for link prediction in Social Networks, detailed in Section 3.2, to the Linked Data network $G = (S, C)$.

Finally, we remark that the recommendation procedure may be used iteratively, considering user feedback. The user indicates a first context for a target tripleset t. The procedure then outputs a ranking of triplesets such as t could possibly be interlinked with them. The user inspects the content of the topmost ranked triplesets and includes new links in t. Then, using the connections induced by the new links, the procedure outputs a new ranking, and so on.

3.2 Adapted Measures

Among the traditional measures originated from graph theory, we chose the Jaccard and the Adamic-Adar coefficients. We selected such measures because the results reported by Liben-Nowell and Kleinberg [13], which analyze co-authorship social networks, indicate that these two measures achieve good performance. Furthermore, they estimate non-zero score values only between nodes with two degrees of separation in the graph.

In what follows, let $G = (S, C)$, t and C_t respectively be the Linked Data network, the target tripleset and the target context given as input to the recommendation procedure. Let u be a tripleset in S. Recall that the context C_u of $u \in S$ is the set of all $v \in S$ such that $(u, v) \in C$, and that the inverse context C'_w of $w \in S$ is the set of all $v \in S$ such that $(v, w) \in C$.

Jaccard Coefficient. Intuitively, the larger the cardinality of the intersection of the contexts of t and u, the greater the likelihood that the two triplesets can be connected. This effect can be measured by the Jaccard coefficient, defined as follows.

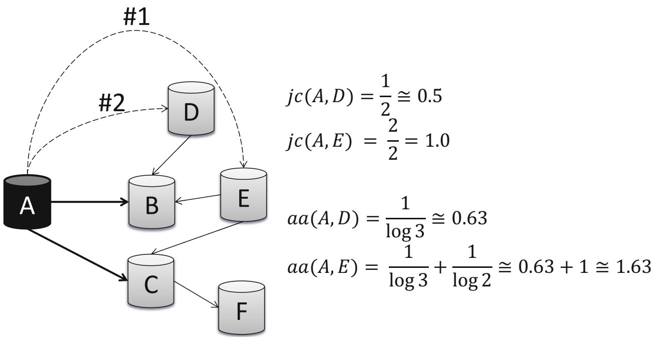
$$jc(t, u) = \frac{|C_t \cap C_u|}{|C_t \cup C_u|} \tag{1}$$

where:

- $|C_t \cap C_u|$ is the cardinality of the intersection of the contexts of t and u
- $|C_t \cup C_u|$ is the cardinality of the union of the contexts of t and u.

Adamic-Adar Coefficient. Intuitively, if two triplesets t and u point to the same tripleset w and w is also pointed by many other triplesets, then w must be a generic tripleset and, therefore, it does not necessarily suggest any possible connection between t and u. On the other hand, if there is no tripleset other than t and u which points at w, then this might be a strong indication that w is a very particular tripleset for both t and u and, therefore, a connection between t and u could as well be defined. Thus, the strength of the belief in the existence of connections between t and u increases inversely proportional to the number of triplesets, other than t and u, which points at w, i.e., depends on the popularity of w.

The Adamic-Adar coefficient aa computes a measure of belief in the connection between t and u as a summation of the inverse of the logarithm of the


#1
#2
A
B
C
D
E
F
jc(A, D) = 1/2 ≅ 0.5
jc(A, E) = 2/2 = 1.0
aa(A, D) = 1/log 3 ≅ 0.63
aa(A, E) = 1/log 3 + 1/log 2 ≅ 0.63 + 1 ≅ 1.63

(a) (b)

Fig. 2. Example of (a) the inputs and the outputs of the recommendation procedure and (b) the values of the coefficients

popularity of the triplesets in the intersection of the contexts of t and u and is define as follows.

$$aa(t,u) = \sum_{w \in C_t \cap C_u} \frac{1}{log\,|C'_w|} \qquad (2)$$

where:

- $|C'_w|$ is the cardinality of the inverse context of w (the popularity of w).

3.3 Example

Figure 2 shows an schematic example of the computation of the coefficients, indicating: (a) the inputs and the outputs of the recommendation procedure; and (b) the values of the coefficients (using log_2). In the example depicted, the inputs are:

- The Linked Data network composed of the triplesets B, C, D, E and F and their connections, represented by solid lines
- The target tripleset A for which new connections must be recommended
- The context for A, pointed by thicker lines, consist of the triplesets B and C, which the user indicates that he can define connections from A to them.

The output of the procedure is a ranking of recommended triplesets, represented by dashed lines connecting the target to them (the number preceded by # indicates the ranking position of each recommended tripleset). Discarding the triplesets in the context of A, the recommendation technique has to rank the remainder triplesets D, E and F according to the chance of defining links from resources in A to resources in D, E and F. Adopting the Jaccard or the Adamic-Adar coefficient, the procedure will return E in the first position (#1) and D in the second (#2).

The tripleset F will not be recommended because there are no connections from F to triplesets pointed by A. Thus the score values of the Jaccard and Adamic-Adar coefficients between A and F are zero.

In this example, E points at two triplesets, B and C, which are pointed by A, whereas D points at just at one, B.

For the Jaccard coefficient, the number of triplesets in the intersection of the contexts of A and E with respect to the total number of triplesets in the union of their contexts is greater than that for the contexts of A and D.

For the Adamic-Adar coefficient between E and A, among the triplesets in the intersection of the contexts of A and E, the tripleset C is considered more important than B (just A and E points at C, while B is also pointed by D).

3.4 Interpretation of the Measures Application

The principle of our approach is that one can infer that t can be connected to u, i.e., t contains URIs that can be linked with URIs of u, iff the context of t overlaps the context of u. However, such analogy must be analyzed in order to better ground its validity.

If the two triplesets t and u share a connection to a tripleset w through the property *rdfs:sameAs*, then there would be triples of the form $(s_1, \mathit{rdfs{:}sameAs}, o_1)$, where $s_1 \in t$ and $o_1 \in w$, and $(s_2, \mathit{rdfs{:}sameAs}, o_2)$, where $s_2 \in u$ and $o_2 \in w$. Now, recall that *rdfs:sameAs* is reflexive and transitive. Thus, if $o_1 \equiv o_2$ holds then $(s_1, \mathit{rdfs{:}sameAs}, s_2)$ will also hold. That is, there will be a link from t to u.

On the other hand, if the interlinking property was not *rdfs:sameAs* but, for instance, *hasAuthor* and *wasAttendedBy*, the probability that t and u share a connection would be lower, but still possible. Indeed, assume that there are triples of the form $(s_1, \mathit{hasAuthor}, o_1)$, where $s_1 \in t$ and $o_1 \in w$, and $(s_2, \mathit{wasAttendedBy}, o_2)$, where $s_2 \in u$ and $o_2 \in w$. Furthermore, assume that $o_1 \equiv o_2$. Then, we might understand s_1 as a paper presented in event s_2 and, therefore, a triple of the form $(s_1, \mathit{wasPresentedIn}, s_2)$ could be added to t to link t and u, provided that *wasPresentedIn* could be added to the vocabulary of t.

To sum up, in the second case one cannot say that the analogy holds in all situations in the context of Linked Data. However, as indicated in the literature [19], the prevalence of links of type *rdfs:sameAs* in the Web of Data justifies the use of the link prediction measures for the recommendation of triplesets based on their connections.

4 Experimental Evaluation

4.1 Description of the Data and the Experiment

We tested the recommendation procedure with data available in the Data Hub catalogue[1], a repository of metadata of open triplesets, in the style of Wikipedia.

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

It is openly editable and is running a data cataloguing software (CKAN)[2] maintained by the Open Knowledge Foundation[3].

The description of each tripleset includes a multivalued property, called *relationships*, exposed by the REST API[4] of the catalogue, whose range is the complete set of catalogued triplesets. This property permits asserting that a tripleset t points at a tripleset u by adding the assertions $t[relationships] = \_node$ and $\_node[object] = u$ to the catalogue data. We used the property *relationships* to extract the connections between triplesets in the Data Hub catalogue. Data was gathered at the end of the 2012, adding to 797 triplesets and 15,012 connections among them. This data therefore induced a Linked Data graph $G = (S, C)$.

To evaluate the technique, we adopted the 10-fold cross validation approach. We split the Linked Data graph $G = (S, C)$ into *recommendation partitions* and *testing partitions* in ten different ways, and defined *target contexts* as follows:

- A *recommendation partition* is a subgraph $G_i = (S_i, C_i)$ of $G = (S, C)$ such that S_i is a set of triplesets to be considered for recommendation and C_i is the set of connections among the triplesets in S_i induced by the *relationships* property
- A *testing partition* is a pair $Tp_i = (T_i, aC_i)$ such that T_i is the set of triplesets in S, but not in S_i, called *recommendation targets*, and aC_i is a set of sets such that, for each $t \in T_i$, aC_i contains the set aC_t of all triplesets u in S_i such that there is a connection from t to u in C
- For each recommendation target $t \in T_i$, a *target context* C_t consists of some chosen triplesets in aC_t.

Additionally, for each different recommendation partition $G_i = (S_i, C_i)$, testing partition $Tp_i = (T_i, aC_i)$, recommendation target $t \in T_i$, with target context $C_t \in aC_i$, we define:

- the *gold standard* for t and is defined as the set $Gs_t = aC_t - C_t$ and represents the triplesets that must be recommended
- a *relevant tripleset* to be recommended for t is a tripleset in Gs_t
- a *candidate tripleset* to be recommended for t is a tripleset in $S_i - C_t$.

Unlike the traditional cross-validation approach, where partitions are used as training sets, the recommendation partitions were used as recommendation subgraphs only, since the proposed technique does not require a training step. The overall performance is taken as the average of the performances in the testing partitions.

In the experiments, the results were evaluated using traditional Information Retrieval measures [20, 21], Recall and Mean Average Precision (MAP). The *overall Recall* is the mean of the recall of each testing partition. The recall of a

[2] `http://ckan.org`

[3] `http://okfn.org`

[4] `http://datahub.io/api/rest/tripleset/[triplesetid]`

testing partition Tp_i is defined as the average of the recall values of each tripleset $t_j \in T_i$:

$$Recall(Tp_i) = \frac{\sum_{j=1}^{|T_i|} Recall(t_j)}{|T_i|} \tag{3}$$

where:

- $Recall(t_j)$ is defined as the ratio between the number of relevant triplesets that are recommended for t_j and the total number of triplesets that must be recommended $|Gs_{t_j}|$.

The *overall MAP* is defined as the mean of the MAP of each testing partition. The MAP of a testing partition Tp_i is in turn defined as the mean of the average precision scores of each tripleset $t_j \in T_i$:

$$MAP(Tp_i) = \frac{\sum_{j=1}^{|T_i|} AveP(t_j)}{|T_i|} \tag{4}$$

where:

- $AveP(t_j)$ is the average precision in the ranking of the tripleset t_j. It is computed as a average of the precision values obtained for each relevant tripleset. For this calculation, the position k in which a relevant tripleset was ranked is considered. Each precision value in position k, only the triplesets whose positions are lower or equal to k are considered, i.e., precision will be the ratio between the number of relevant triplesets recommended until the position k and this position number k. For instance, if in the tenth position was ranked the fifth relevant tripleset from a complete set of twenty relevant then the precision in p_{10} would be $p = 5/10$. For each relevant not recommend, the precision value used to calculate the $AveP$ is zero.

4.2 Evaluation and Results

To better understand the available data, Figure 3 presents the total number of triplesets calculated in function of a minimum number of connections (number of triplesets pointed by them). Figure 3 shows that most of the triplesets in the Data Hub catalogue has very few connections. The average of the number of connections per tripleset in the Data Hub catalogue was approximately 18.83.

In the experiments, we evaluated the ranking recommendations generated using the measures presented in Section 3.2. As the measures depend on both t and a tripleset s that points to at least one tripleset u which is also pointed by t, they estimate a score different from zero for the same triplesets in the *recommendation partition*. The overall recall was calculated as a function of the

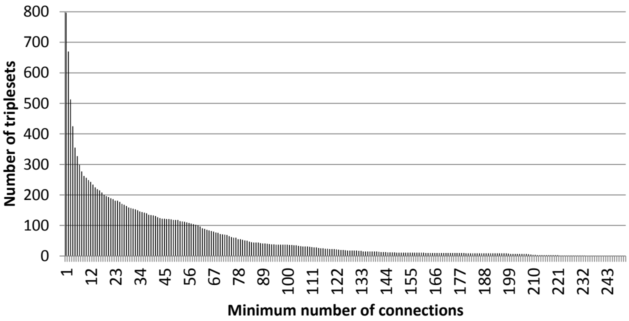


Fig. 3. Number of triplesets *vs.* minimum number of connections

cardinality of the target context (remember that the target context consists of some chosen triplesets pointed by t). Recall is used to analyze the coverage of the recommendation procedure. We considered all triplesets for which the score values are greater than zero. The results obtained showed that:

- For small target contexts, the overall recall is relatively high, being on the average greater than 75%
- For context sizes greater than 4 triplesets, the overall recall is higher than 90%.

These results show evidences that it is possible to recommend many relevant triplesets, even knowing just a few connections from the target tripleset. This is very important to validate the practical applicability of these Social Networks measures (that consider only the direct neighbors) to recommend triplesets in a Linked Data environment.

After these analyses, we evaluated the ordering of the recommendations in the rankings. For this purpose, we used the overall Mean Average Precision (MAP) to verify the accuracy of the generated ranking. Remember that the overall MAP estimation considers the gold standard induced by the choice of the context C_t, i.e., it is not defined by users. The results are presented in Figure 4 and show that the overall MAP values for Adamic-Adar are higher than those for the Jaccard coefficient, for context sizes smaller than 36, which means that, on average, for the same recall, the Adamic-Adar is more precise than Jaccard coefficient. This probably happens because the Adamic-Adar coefficient better differentiates the importance of the common triplesets pointed by the target and the candidate triplesets which tends to require less knowledge, or known triplesets in the context of the target.

In addition, we also calculated the average position of the last relevant tripleset in the ranking. These results were divided by the total number of triplesets in the corresponding recommendation partition. This analysis estimates the average percentage of the top of the ranking that needs to be verified to discover all the relevant triplesets that were recommended. Figure 5 presents these results. To better understand the results, we also calculated what would be, on the

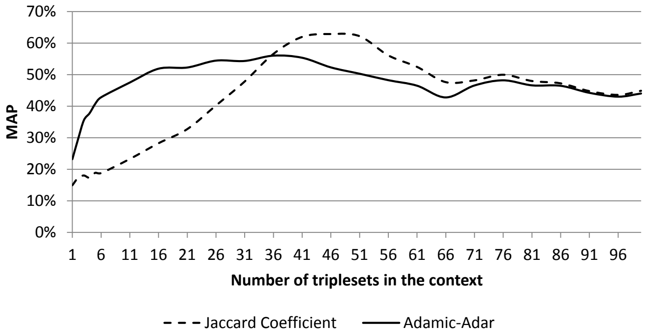


Fig. 4. Overall Mean Average Precision *vs.* context size

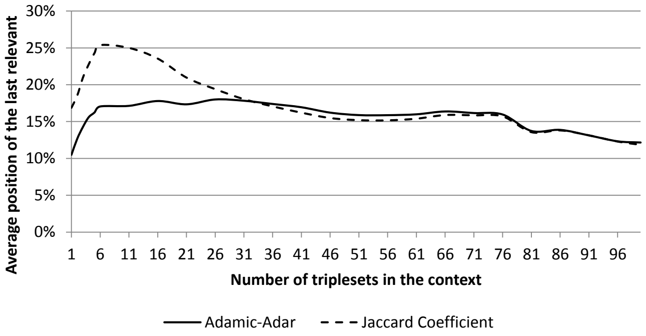


Fig. 5. Average relative position of the last relevant recommended tripleset in the ranking *vs.* context size

average, the maximum reduction possible (finding all relevant triplesets at the top positions of the ranking, for all triplesets) using a context size equal to 1. We obtained a maximum reduction value of 3.85%.

The worst performance of the Jaccard and Adamic-Adar coefficients indicates that one needs to examine, on the average, respectively, 25% and 18% of the top ranking triplesets to find all the relevant recommended triplesets. The best result, obtained using the Adamic-Adar coefficient considering only one connection in the context, indicates that one needs, on the average, to examine only 10% of the ranking to find all the relevant recommended triplesets. It shows evidences that the Adamic-Adar coefficient is more appropriate to rank the results in this scenario than the Jaccard coefficient. This also shows that is not necessary to know many triplesets in the context of target (what would otherwise invalidate the practical application of the procedure) to obtain suitable rankings.

5 Final Remarks

In this paper we proposed the use of link prediction measures to address what we called the *tripleset recommendation problem* in the Linked Data domain.

Our approach generates a ranking of triplesets to be linked with a tripleset t to be published. The ranking can be used to reduce the candidates that t can be interlinked with, thereby reducing the set of triplesets to be further inspected by other more costly techniques, if necessary. The experiments tested two different link prediction measures. The results show that such measures obtain good results, even when few triplesets in the context of t are available. Specifically, the results show that the approach can reduce up to 90% of the search space for the interlinking candidates.

We have defined the tripleset network as an unweighted graph $G = (S, C)$, thus disregarding the number of links between triplesets. This assumption favors triplesets from related information domains and penalizes generic ones. For instance, DBpedia is frequently referenced by many triplesets because it is a generic repository. Therefore, most likely, the weight of the connections to DBpedia would be very high, which would end up influencing the ranking in favour of DBpedia. However, the tripleset to be published would neither get more visibility nor unveil more hidden information from other more specific triplesets, because it is connected to DBpedia.

As further work, we plan to test other score measures for ranking generation and to perform experiments using other catalogues of triplesets. We will also consider using domain information to improve the results.

Acknowledgments. This work was partly supported by CNPq, under grants 160326/2012-5, 301497/2006-0, 475717/2011-2 and 57128/2009-9, by FAPERJ, under grants E-26/170028/2008 and E-26/103.070/2011, and by CAPES under grant PROCAD/NF 1128/2010.

References

1. Berners-Lee, T.: Linked Data - Design Issues. W3C (June 2009), `http://www.w3.org/DesignIssues/LinkedData.html` (accessed on March 2013)
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems (IJSWIS) 5(3), 1–22 (2009)
3. Nikolov, A., d'Aquin, M.: Identifying Relevant Sources for Data Linking using a Semantic Web Index. In: WWW 2011 Workshop on Linked Data on the Web, LDOW. CEUR Workshop Proceedings, vol. 813. CEUR-WS.org (2011)
4. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender Systems: An Introduction. Cambridge University Press (2011)
5. Bergamaschi, S., Guerra, F., Leiba, B.: Guest editors' introduction: Information overload. IEEE Internet Computing 14(6), 10–13 (2010)
6. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Computing 7(1), 76–80 (2003)
7. Schafer, J.B., Konstan, J., Riedl, J.: Recommender systems in e-commerce. In: Proceedings of the 1st ACM Conference on Electronic Commerce, EC 1999, pp. 158–166. ACM, New York (1999)
8. Mooney, R.J., Roy, L.: Content-based book recommending using learning for text categorization. In: Fifth ACM Conference on Digital Libraries, DL 2000, pp. 195–204. ACM, New York (2000)

9. Burke, R.D.: Hybrid recommender systems: Survey and experiments. User Model. User-Adapt. Interact. 12(4), 331–370 (2002)
10. Golbeck, J., Hendler, J.: Filmtrust: movie recommendations using trust in web-based social networks. In: 3rd IEEE Consumer Communications and Networking Conference, CCNC 2006, vol. 1, pp. 282–286 (2006)
11. Montaner, M., López, B., de la Rosa, J.L.: A taxonomy of recommender agents on the internet. Artif. Intell. Rev. 19(4), 285–330 (2003)
12. Meo, P.D., Nocera, A., Rosaci, D., Ursino, D.: Recommendation of reliable users, social networks and high-quality resources in a social internetworking system. AI Commun. 24(1), 31–50 (2011)
13. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. J. Am. Soc. Inf. Sci. Technol. 58(7), 1019–1031 (2007)
14. Newman, M.E.J.: The structure and function of complex networks. SIAM Review 45(2), 167–256 (2003)
15. Quercia, D., Capra, L.: Friendsensing: recommending friends using mobile phones. In: Third ACM Conference on Recommender Systems, RecSys 2009, pp. 273–276. ACM, New York (2009)
16. Nikolov, A., d'Aquin, M., Motta, E.: What should I link to? Identifying relevant sources and classes for data linking. In: Pan, J.Z., Chen, H., Kim, H.-G., Li, J., Wu, Z., Horrocks, I., Mizoguchi, R., Wu, Z. (eds.) JIST 2011. LNCS, vol. 7185, pp. 284–299. Springer, Heidelberg (2012)
17. Lóscio, B.F., Batista, M.C.M., Souza, D., Salgado, A.C.: Using information quality for the identification of relevant web data sources: a proposal. In: 14th International Conference on Information Integration and Web-based Applications & Services, IIWAS 2012, pp. 36–44. ACM, New York (2012)
18. de Oliveira, H.R., Tavares, A.T., Lóscio, B.F.: Feedback-based data set recommendation for building linked data applications. In: 8th International Conference on Semantic Systems, I-SEMANTICS 2012, pp. 49–55. ACM, New York (2012)
19. Halpin, H., Hayes, P.J.: When owl: sameAs isn't the same: An analysis of identity links on the semantic web. In: Proceedings of the WWW 2012 Workshop: Linked Data on the Web (2010)
20. Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval - the concepts and technology behind search, 2nd edn. Pearson Education Ltd., Harlow (2011)
21. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (July 2008)