What SPARQL Query Logs Tell and do not Tell about Semantic Relatedness in LOD

Or: the Unsuccessful Attempt to Improve the Browsing Experience of DBpedia by Exploiting Query Logs

Jochen Huelss and Heiko Paulheim

University of Mannheim
Research Group Data and Web Science
jochen@huelss.de,heiko@dwslab.de

Abstract. Linked Open Data browsers nowadays usually list facts about entities, but they typically do not respect the relatedness of those facts. At the same time, query logs from LOD datasets hold information about which facts are typically queried in conjunction, and should thus provide a notion of intra-fact relatedness. In this paper, we examine the hypothesis how query logs can be used to improve the display of information from DBpedia, by grouping presumably related facts together. The basic assumption is that properties which frequently co-occur in SPARQL queries are highly semantically related, so that co-occurence in query logs can be used for visual grouping of statements in a Linked Data browser. A user study, however, shows that the grouped display is not significantly better than simple baselines, such as the alphabetical ordering used by the standard DBpedia Linked Data interface. A deeper analysis shows that the basic assumption can be proven wrong, i.e., co-occurrence in query logs is actually *not* a good proxy for semantic relatedness of statements.

Keywords: Semantic Relatedness, Linked Open Data, Linked Data Browsers, Query Log Mining, DBpedia

1 Motivation

Usefulness is considered as one of the key challenges to human users who attempt to benefit from the immense knowledge graph of semantic web [13]. This challenge implies that the user experience and visual presentation of linked data is currently not tangible for human users. Back in 2006, this lack of a tool which enables a curated, grouped, and sorted browsing experience of semantic data describing real-world entities was also mentioned by Sir Tim Bernes Lee in a talk on the *Future of the Web* at University of Oxford[1]. With the web of Linked Data having grown to more than 1,000 datasets [15], and the emergence of central

[1] http://webcast.oii.ox.ac.uk/?view=Webcast&ID=20060314

hubs in the semantic web such as DBpedia, the semantic web research community has put a lot of effort into the fields of browsing and interacting with linked data [4, 8] and summarizing important properties of a semantic entity. However, these problems persist since all major semantic web databases and browsers still present their linked data as an unordered or lexicographically ordered list to their users.

Traditionally, web usage logs are mined for behavioral patterns to cluster items of common interest and recommend them to users. Our approach applies web usage mining on SPARQL query logs and looks for patterns that relate equally interesting properties of semantic entities. Thus, our general hypothesis is that, in a data set of SPARQL queries, it should be possible to mine information about the semantic relatedness of statements. Such information again can be exploited to form coherent groups of properties which are beneficial for the human browsing experience of the semantic web.

2 Related Work

Although much work has been devoted to the creation of browsers for Linked Open Data, most of them essentially present facts about entities as lists, in which the facts have no relation among each other [4]. Examples for such classic browsers are *DISCO*[2] and *Tabulator* [2]. A semantic grouping of facts, as proposed in this paper, has been rarely proposed so far.

Some browsers, such as *Zitgist*[3], provide domain-specific templates that order information which uses popular ontologies, such as FOAF[4] or the Music Ontology[5]. While there is a trend towards reusing popular vocabularies for LOD, there is, at the same time, a trend towards using *multiple* vocabularies in parallel [15], which, in turn, creates new challenges for such template-based approaches.

A slightly different, yet related problem is the ranking and filtering of semantic web statements into more and less relevant ones. Here, some works have been proposed in the past, e.g., [3, 5, 6, 9, 17].

In [16], we have presented a first domain-independent attempt of creating a semantic grouping of facts. Here, we try mapping predicates to WordNet synsets, and measure the similarity among predicates in WordNet. Similar predicates are grouped together, with the labels for synsets of common ancestors being used as group headings. Like the work presented in this paper, no statistical significant improvements over baseline orderings of facts could be reported.

[2] http://wifo5-03.informatik.uni-mannheim.de/bizer/ng4j/disco/

[3] Meanwhile offline

[4] http://www.foaf-project.org/

[5] http://musicontology.com/

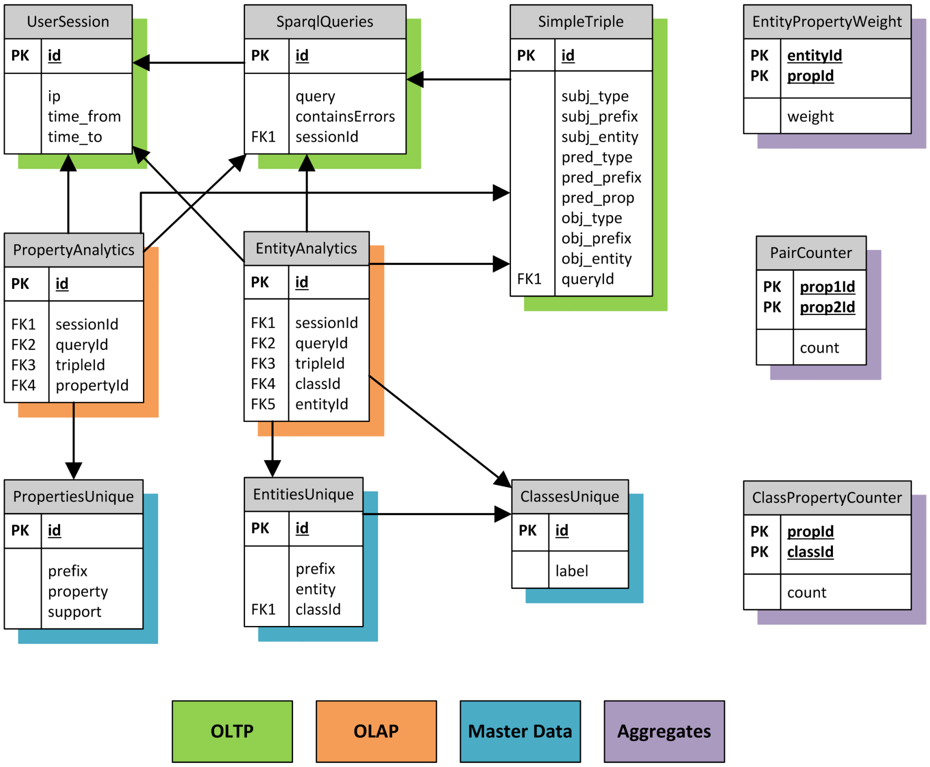


Fig. 1. DB schema for storing and analyzing SPARQL queries

3 Approach

In this paper, we present an approach for semantically grouping semantic web statements, based on SPARQL query logs. Those logs are read once and preprocessed into a database schema including basic statistics, as shown in Fig. 1.

Given that a user requests a URI, such as `http://dbpedia.org/resource/Mannheim`, the system first reads the corresponding set of triples, then uses the preprocessed database to create a grouping, with different possible algorithms (see below). The result of grouped statements is delivered to the user through the modular semantic web browser *MoB4LOD*[6].

3.1 Dataset and Preprocessing

The basis of our experiments is the UseWOD 2014 SPARQL dataset [1], which collects 300k SPARQL queries for the public DBpedia endpoint[7] over the period 06/12/2013 – 01/27/2014, out of which 249k are *valid* SPARQL queries,

[6] `http://www.ke.tu-darmstadt.de/resources/mob4lod`

[7] `http://dbpedia.org/sparql`

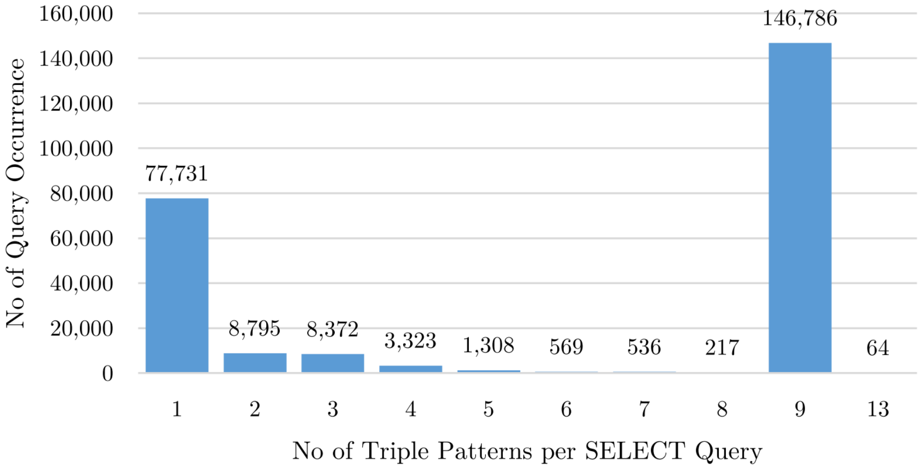


Fig. 2. Distribution of number of triple patterns per query

the vast majority (more than 98%) being `SELECT` queries.[8] The dataset is fully anonymized.

From those SPARQL queries, we extract *triple patterns* for further analysis. Fig. 2 depicts the distribution of the number of triple patterns over the dataset, showing that most of the datasets have only one triple pattern, while there is an anomaly at nine triple patterns, caused by a bot posing almost the same query repeatedly.

In particular, for our goal of semantically grouping statements, we are interested in *property pairs* and *class-property pairs*, i.e., pairs of two properties, or a property and a class, co-occurring in a query. From the 171k queries with more than one triple in the query pattern, we extracted a total of 12,078 unique property pairs and 1,141 unique class-property pairs. Here, we could use all triple patterns that do not have a variable in the predicate position, which holds for more than 80% of the triple patterns. During the pre-processing phase, we collect the frequency of each of those pairs, as well as of all class-property pairs, as shown in Fig. 1.

3.2 Approaches for Grouping Statements

For displaying results, we use two baseline approaches, as well as three approaches based on clustering statements together that have predicates often requested together.

Baseline 1: Lexicographic The first baseline follows the approach of traditional semantic web browsers, ordering facts about an entity lexicographically

[8] Note that the approach is not limited to DBpedia, but could be applied to any dataset for which such a logfile exists.

by their predicate. Groups are created by starting letters of the properties (A-F, G-K etc.).

Baseline 2: Counting The second baseline simply orders properties by their frequency in the SPARQL log for the class the retrieved resource belongs to. No grouping is made.

Approaches based on clustering To create groupings of statements for properties that co-occur frequently, we use three different clustering algorithms: DBSCAN [7], hierarchical clustering [18], and Partitioning Around Medoids (PAM), an implementation of k-medoids [10]. For the latter, we chose to use $k = 7$, so that seven groups of statements are produced, following the wide-spread paradigm that humans can perceive roughly seven items at once [12].

For all clustering algorithms, we use the implementation in *WEKA* [19] with the following distance function between two properties:

$$distance(p_1, p_2) = \frac{1}{count_{p_1,p_2} + \omega} \tag{1}$$

With this formula, the distance between two properties is the larger the more often they are queried together. ω is used as a smoothing factor which prevents a division by zero for pairs of properties that never occur together, and that influences the steepness of the curve between 0 and 1 co-occurences.

In our experiments, we have used $\omega = 5$. Furthermore, the following settings were used: (1) the top-7 properties showing the highest support count in the UseWOD queries were excluded from clustering since they are merely general purpose properties, such as `rdf:type` and `owl:sameAs`, and (2) properties not occurring at all in UseWOD data set were also excluded. The clusters shaped were ranked descendingly based on the median value for support of the properties assigned to a cluster. The employment of a median was anticipated to be better than an average function because it is not prone to very high or low outliers within the clusters.

4 Evaluation

We have evaluated the three different grouping options of our approach against the two baselines in an end-user study. In that study, users were presented an entity from DBpedia with the statements grouped according to one of the five approaches, and had to answer a question about the entity.

4.1 Setup

The study was conducted as an online experiment using the *MoB4LOD* semantic web browser introduced above. A sample screenshot for the property grouping for the DBpedia entity *Cambridge* is shown in Fig. 3. The hypotheses of this study are derived from studies conducted by [5], [14], and [16]:

Table 1. Average number and size of groups produced by the different approaches

Characteristic	Lexicographic	DBSCAN	Hierarchical Clustering	PAM
Avg. # Groups	3.0	4.33	6.00	7.00
Avg. Elem. / Group	8.2	5.68	9.20	7.88

H1 A participant finds a required fact significantly faster with a grouping based on our approach than with a baseline.

H2 The participant's subjective assessment of a grouping based on our approach is significantly better than the sorting of baseline.

H3 A participant is significantly more accurate in finding a required fact with a grouping based on our approach than with a baseline.

All of these hypotheses share the underlying assumption that the more coherent, i.e. semantically-related groups of statements are, the easier it becomes for humans to consume semantic web data and satisfy their information needs.

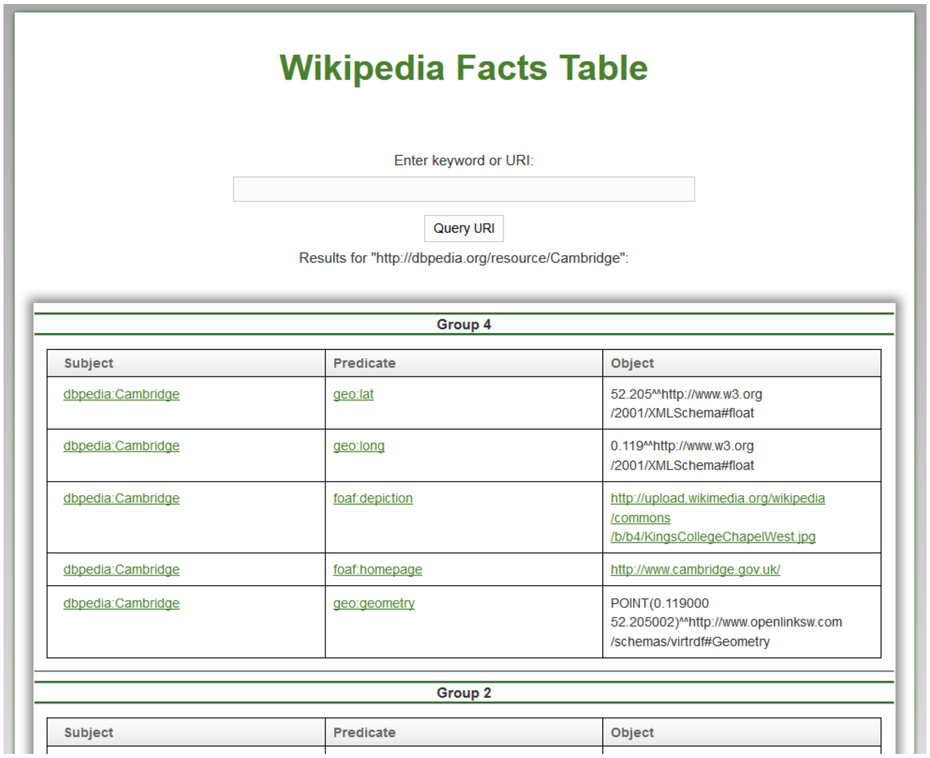
For investigating these hypotheses, the online experiment employed a 5x5 within-subject design with five questions and five groupings (i.e. five tasks) for each participant. For each data sample, we measured the completion time of a task (in seconds), the subjective assessment of a task (5-point Likert-type scale), and the accuracy of an answer of a task (true / false). These data items are the dependent variables for the study's independent variables which are the five sortings. The two baselines and the three groupings of our approach were exposed to a participant in a randomized manner that ensured that each tasks was answered equally often using one of the five sortings. Table 1 depicts the average number of groups and group sizes for each approach.

4.2 Tasks and Users

Each task of the online experiment consisted of a question and a grouped list of semantically related properties of a specified DBpedia entity. The five different sortings were the actual stimulus material for evaluating our approach. For the questions, entities from five DBpedia classes were employed (*Settlement*, *Film*, *Office Holder*, *Country*, *Musical Artist*). The chosen questions were intended to not be answerable based on the participants' general knowledge. A sample question of a task is: *What is the elevation of the city of Mannheim?*After each question, the participants were asked for their subjective assessment of a listing.[9]

80 participants from Germany completed the experiment. They were recruited by convenience sampling via social network sites, e-mailing, and other online channels. To remove obvious outliers, we removed all experiment data from participants who did not answer all questions, as well as those with a completion time outside of a 3σ confidence interval, i.e., extremely low or high

[9] The questionnaire is available at `http://dws.informatik.uni-mannheim.de/en/research/noise-2015-accompanying-material`



Wikipedia Facts Table

Results for "http://dbpedia.org/resource/Cambridge":

Group 4

Subject	Predicate	Object
dbpedia:Cambridge	geo:lat	52.205^^http://www.w3.org/2001/XMLSchema#float
dbpedia:Cambridge	geo:long	0.119^^http://www.w3.org/2001/XMLSchema#float
dbpedia:Cambridge	foaf:depiction	http://upload.wikimedia.org/wikipedia/commons/b/b4/KingsCollegeChapelWest.jpg
dbpedia:Cambridge	foaf:homepage	http://www.cambridge.gov.uk/
dbpedia:Cambridge	geo:geometry	POINT(0.119000 52.205002)^^http://www.openlinksw.com/schemas/virtrdf#Geometry

Group 2

Subject	Predicate	Object

Fig. 3. Screenshot of MoB4LOD browser with groups of RDF triples for DBpedia entity *Cambridge*

processing times. After data cleansing, the sample consisted of *65* participants, which means that each question was solved 13 times with each sorting, on average. Exactly 40% of the participants reported to be familiar with the concepts of semantic web.

4.3 Results

For all hypotheses, the independent variable was the set of the five sortings and the hypotheses are individually analyzed on task level. An overall determination of the best sorting is impossible because the equality of all tasks' level of difficulty cannot be assumed. Table 2 exposes the descriptive statistics (i.e. means of the dependent variables) of our experiment to the readers.

For the recorded completion times T1-5, the analysis of the means does not lead to a conclusive picture. For three out of five tasks, the best mean completion is even taken by one of the baselines. A one-way ANOVA investigates pair-wise significant differences between the three groupings and the two baselines in case of H1 and H2. For H1, only Task 1 depicted significant pair-wise differences. A

Table 2. Descriptive statistics for H1-3, mean time in seconds (shorter is better), mean assesment as intervall [1,5] (higher is better) and mean accuracy as percentage of correctly given answers (in %)

Dep. Variable	Lexic. Baseline	Count. Baseline	DBSCAN	Hier. Clustering	PAM
Time T1	30.9	42.4	24.6	**21.8**	39.2
Time T2	28.3	**26.2**	38.5	44.2	38.5
Time T3	24.7	26.5	35.9	**23.7**	23.9
Time T4	**33.0**	38.9	38.4	58.1	39.4
Time T5	40.2	**22.7**	30.5	33.2	38.6
Assessment	3.66	3.57	3.66	3.57	**3.69**
Accuracy	94.0	**98.0**	94.7	92.7	96.7

Bonferroni posthoc test indicated that the hierarchical clustering grouping had a significant difference with the simple count baseline ($p < .05$). Therefore, H1 cannot be confirmed consistently across the tasks.

Regarding the subjective assessment of the groupings, Table 2 shows the average assessment for each grouping. The best assessment is given to the PAM grouping which is contradictive to the completion time findings. The executed one-way ANOVA does not reveal any significant pair-wise differences between any of the cluster groupings and either one of the baselines. Therefore, also H2 cannot be confirmed for all tasks.

The results of the experiment also show that the percentage of correctly answered tasks exceeds 92% for all sortings (see Table 2). H3 cannot be validated with an ANOVA since it is measured as a nominal variable. It can be accepted or refused by using frequency scales partitioned by the different sortings. However, these frequency scales revealed only non-significant differences between the groupings and the baselines. Thus, H3 cannot be confirmed either.

To support the assumption of the previous section that an overall evaluation of the hypotheses is impossible, Table 3 shows the mean working time and mean assessment of all tasks. Time has got a range of 15.52 seconds. This indicates that the different levels of difficulty led to varying answering times. Moreover, the table shows that the mean time and the mean assessment of the individual questions correlate negatively using Pearson's correlation ($\rho = -0.88$). The longer the time, the more negative the assessment. This finding is significant for Tasks 3-5. Table 3 also shows that, for the chosen ontology classes, the number of property pairs found in our database is small compared to the total amount of triples retrieved for the DBpedia entities.

5 Discussion

The experiments presented in the previous section have shown that the hypotheses formulated for this research work could not be confirmed, at least not for

Table 3. Effect of time and assessment of all sortings on task level (mean), a correlation of longer time and more negative assessment is revealed, $n = 65$, $**p < .01$

	T1	**T2**	**T3**	**T4**	**T5**
Ontology Class	Settlement	Film	OfficeHolder	Country	MusicalArtist
Pairs found	98	29	123	92	133
Total Triples	2,242	225	257	4,248	337
Time	31.76 (16.94)	35.09 (17.59)	26.42 (12.79)	41.94 (21.77)	32.47 (18.24)
Assessment	3.65 (1.268)	3.52 (1.200)	3.98 (1.192)	3.46 (1.187)	3.57 (1.274)
Correlation	-.162	-.170	-.321**	-.369**	-.276**

DBpedia. In particular, the assumption that the visual grouping of properties co-occurring in SPARQL logs leads to an improved human consumption of semantic web data is proven wrong. Since three different clustering algorithms were tried in the experiments, the cause is most likely not a shortcoming of the clustering method, but the approach itself.

The main weak point about the assumption is that SPARQL queries and LOD interfaces serve the same information needs. First of all, a large fraction of SPARQL queries are posed by machine agents, while Linked Data interfaces are used by humans. Second, seasoned semantic web experts will be able to use SPARQL as well as Linked Data interfaces, and choose among them given the specific characteristics of their problem. These differences make the overall assumption problematic.

In the following, we will analyze this result in more detail. We exemplify potential problems both with the approach as well as the evaluation methodology.

5.1 Problems of the Approach

An a posteriori analysis revealed that one central problem of the approach presented in this paper is the coverage of the log file used for the experiments. According to DBpedia mapping statistics[10], there are currently 6,126 different DBpedia ontology properties. In the UseWOD data set we found 488 pairs consisting of DBpedia's ontology properties. Thus, the recall of class-property pairs is 7.96% (given all of those pairs that appear for at least one entity in DBpedia). For the property pairs generated from UseWOD, the recall is even lower at 1.9% (again given all such pairs that appear for at least one entity in DBpedia). This, in turn, means that the distance function for the majority of pairs is mostly uniform (i.e., $\frac{1}{\omega}$), with meaningful distances only assigned to a minority of pairs.

Another problem we observed was that redundant properties (such as `dbo:birthPlace`, `dbp:birthPlace`, and `dbp:placeOfBirth`) were rarely grouped

[10] `http://mappings.dbpedia.org/server/statistics/en/?show=100000`

into the same cluster by any of the clustering based approaches. At second glance, this is actually a built-in problem of the approach: when a user poses a query against DBpedia, he or she will likely pick one of the properties, and not use them in conjunction – for example, there are 2,687 using at least one of the three aforementioned properties, but only 41 (i.e., 1.53%) use at least two of those. This shows that redundant properties – which have the highest semantic relatedness! – are unlikely to frequently co-occur in queries, and hence, are likely not to end up in the same cluster.

In informal feedback, many users complained that the groupings of statements we created had only generic titles, which are essentially the cluster names (*group 1*, *group 2*, etc.). Furthermore, DBSCAN identifies "noise points" which do not belong to any cluster, that were displayed under the headline *noise*, which lead to additional confusion. This shows that the assignment of meaningful headlines to groups of statements is a desirable – if not required – property of an approach like the one presented in this paper. This claim can be formulated even more strongly, stating that grouping without assigning headlines is pointless, since a user will have to scan each group for the desired piece of information, and will thus not perceive any usability advantage. Assigning headlines to groups, however, is a hard research problem in itself and was out of scope of the research work.

5.2 Problems of the Methodology

Using lexicographic sorting as a baseline is a straightforward idea. Especially since many tools use that ordering, it is also necessary to show that a significant advancement can be made over that ordering in order to prove the utility of the approach.

However, in our case, the baseline is rather strong due to some particular characteristics of DBpedia. DBpedia has two major namespaces – i.e., `http://dbpedia .org/ontology/` holds all the higher quality properties mapped against the DBpedia ontology, while `http://dbpedia.org/property/` contains the lower-quality, raw extraction from the Wikipedia infoboxes [11]. The information conveyed by the former usually contains all the major information about an entity. In lexicographic ordering by property URI, the properties from the DBpedia ontology namespace are all listed before those from the raw extraction namespace, which leads to the major facts presented way up in the list.

Moreover, the properties that were required to answer the questions might have a different perceived importance (comparing, e.g., the elevation of a city to the governor of a state). Thus, users may implicitly search for properties they deem more important further up on the list. Since this importance is also partly reflected by the overall number of occurrences in DBpedia, this strategy may be successful on the counting baseline, which may explain why this is also a very strong baseline.

When analyzing the results in more detail, we found that there is a significant negative correlation between task completion time and assessment of the presentation. At the same time, the presented entities had significantly different sizes of the statement sets, which may furthermore influence both the completion time and the assessment. A more balanced selection of entities w.r.t. the number of statement displayed may have lead to more conclusive results.

6 Conclusion

In this paper, we have analyzed how SPARQL query logs can be used for creating meaningful groupings of statements for semantic web browsers. The analysis of the results show that this is not possible for various reasons, including the coverage of the SPARQL log files and blind spots of the approach, such as redundant properties.

In particular, it has been shown that co-occurence of properties in SPARQL queries is not a suitable proxy to determine semantic relatedness of those properties. This is best illustrated with the case of redundant properties, which are maximally semantically related, but extremely unlikely to co-occur in a query.

Many of the problems leading to the insignificant results – e.g., the problem of redundant properties or the strength of certain baselines – are specific to one dataset, in our case: DBpedia. For other datasets with different characteristics, those problem may or may not hold. Thus, evaluations on other datasets than DBpedia before eventually discarding the approach.

Still, we think that finding ways to create semantically coherent, visually appealing ways to present semantic web statements is a desirable property of Linked Open Data browsers. We hope that the results presented in this paper inspire future researchers to explore different ways of achieving that goal.

References

1. Berendt, B., Hollink, L., Hollink, V., Luczak-Rösch, M., Möller, K., Vallet, D.: Usage analysis and the web of data. In: ACM SIGIR Forum. vol. 45, pp. 63–69. ACM (2011)
2. Berners-Lee, T., Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., Lerer, A., Sheets, D.: Tabulator: Exploring and analyzing linked data on the semantic web. In: Proceedings of the 3rd International Semantic Web User Interaction Workshop (SWUI2006) at the 5th ISWC Conference. Athens, USA (2006)
3. Cheng, G., Tran, T., Qu, Y.: Relin: relatedness and informativeness-based centrality for entity summarization. In: Proceedings of the 10th International Semantic Web Conference (ISWC2011). pp. 114–129. Bonn, Germany (2011)
4. Dadzie, A.S., Rowe, M.: Approaches to visualising linked data: A survey. Semantic Web 2(2), 89–124 (2011)
5. Delbru, R., Toupikov, N., Catasta, M., Tummarello, G., Decker, S.: Hierarchical link analysis for ranking web data. In: The Semantic Web: Research and Applications, pp. 225–239. Springer (2010)
6. Ding, L., Pan, R., Finin, T., Joshi, A., Peng, Y., Kolari, P.: Finding and ranking knowledge on the semantic web. In: Proceedings of the 4th International Semantic Web Conference (ISWC2005). pp. 156–170. Galway, Ireland (2005)

7. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining. pp. 226–231. Portland, USA (1996)
8. García, R., Paulheim, H., Di Maio, P.: Special issue on semantic web interfaces. Semantic Web 6(8) (2015)
9. Kirchberg, M., Ko, R., Lee, B.S.: From linked data to relevant data - time is the essence. In: Proceedings of the 1st International Workshop on Usage Analysis and the Web of Data (USEWOD2011) at the 20th WWW Conference. Hyderabad, India (2011)
10. Van der Laan, M., Pollard, K., Bryan, J.: A new partitioning around medoids algorithm. Journal of Statistical Computation and Simulation 73(8), 575–584 (2003)
11. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., et al.: Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web Journal (2014)
12. Miller, G.A.: The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychological Review 63(2), 81 (1956)
13. Möller, K., Hausenblas, M., Cyganiak, R., Handschuh, S.: Learning from linked open data usage: Patterns & metrics. In: Proceedings of the 2nd Web Science Conference (WebSci10). Raleigh, USA (2010)
14. Paulheim, H.: Improving the usability of integrated applications by using interactive visualizations of linked data. In: Proceedings of the ACM International Conference on Web Intelligence, Mining and Semantics. Sogndal, Norway (2011)
15. Schmachtenberg, M., Bizer, C., Paulheim, H.: Adoption of the linked data best practices in different topical domains. In: The Semantic Web–ISWC 2014, pp. 245–260. Springer (2014)
16. Seeliger, A., Paulheim, H.: A semantic browser for linked open data. In: Proceedings of the Semantic Web Challenge at the 11th ISWC Conference. Boston, USA (2012)
17. Thalhammer, A., Toma, I., Roa-Valverde, A., Fensel, D.: Leveraging usage data for linked data movie entity summarization. In: Proceedings of the 2nd International Workshop on Usage Analysis and the Web of Data (USEWOD2012) at the 21st WWW Conference. Lyon, France (2012)
18. Ward Jr, J.H.: Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association 58(301), 236–244 (1963)
19. Witten, I., Frank, E., Hall, M.: Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, 3rd edn. (2011)