Identifying Information Needs by Modelling Collective Query Patterns

Khadija Elbedweihy, Suvodeep Mazumdar, Amparo E. Cano, Stuart N. Wrigley and Fabio Ciravegna

OAK Group, Dept. of Computer Science,
University of Sheffield, UK
{k.elbedweihy, s.mazumdar, a.cano, s.wrigley,
f.ciravegna}@dcs.shef.ac.uk

Abstract. With individuals, organisations and Governments releasing large amounts of linked data, users have now access to an immense repository of highly structured data, ready to be queried and reasoned upon. However, it is important at this stage to ask questions like What do Linked Data users look for and how do they search for information? Understanding the information needs of users accessing such data could be invaluable to researchers, developers and linked data providers and consumers. In this paper, we present an approach to formalise query log analysis and how we consume such analysis. We present SEMLEX, a visualisation interface that facilitates exploration of user's information needs by analysing queries issued to a public dataset.

Keywords: linked data, information visualisation, semantic query log analysis, information needs

1 Introduction

Over the last two decades traditional search engines have improved accuracy by adapting their processing to address the information needs of web users. Part of this progress has been possible thanks to the analysis and interpretation of query logs [12,6]. These studies addressed statistics such as query length, term analysis and topic classification [5,13], as well as the identification of changes in users' search behaviour over time [7]. However, the structure and information captured in traditional query logs limits the analysis to a set of timestamped keywords and URIs, which lacks structure and semantic context.

The movement from the 'web of documents' towards structured and linked data has made significant progress in recent years. Semantic Web gateways (e.g., Sindice [14]) expose SPARQL endpoints, which allow users or software agents to perform more complex querying and reasoning over the 'web of data'. Although the use of these gateways has built up a rich semantic trail of users' information needs in the form of semantic query logs, little research has been done on the interpretation of query logs as clues for analysing and predicting information needs at the semantic level.

Previous studies have focused on metadata statistics derived from Semantic Web search engines (e.g., [9]). In this work, we investigate the size of the semantic gap between supply and demand within the Semantic Web by analysing the semantic content

of query logs. For our analysis, we define information needs as the set of concepts and properties users refer to while using SPARQL queries. Consider:

```
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?manufacturer
WHERE {
     <http://dbpedia.org/resource/Acura_ZDX> dbo:manufacturer ?manufacturer.
}
```

This query shows a user looking for the manufacturer of a particular car. The user's information needs would be represented as `http://dbpedia.org.../Automobile` and `dbo:manufacturer`. The concept `Automobile` would be inferred by querying the linked data endpoint.

The contributions of this paper are as follows:

1. We provide a new approach for analysing semantic query logs.
2. We describe a set of methods for extracting patterns in semantic query logs.
3. We implemented these methods in an interactive tool which enables the exploration of information needs revealed by the semantic query logs analysis.

We use a DBpedia query log dataset as a case study for testing our methodology. In this study, we explore aspects such as what information individuals or software agents commonly look for and the manner in which they perform the query. Such analyses can give an insight into the coverage and distribution of queries over the data and whether users and agents are making use of the whole or just a small portion of a dataset. Our visualisation tool supports the identification of interesting trends or hidden patterns.

This paper is structured as follows: Section 2 presents a review of the current state of the art in analysing query logs. Section 3 discusses our approach in analysing query logs by modelling log entries and describes the subsequent analysis results. Section 4 describes the dataset we have used for our analysis. Section 5 presents our approach in consuming our analysis results and some observations. Section 6 concludes the paper and discusses the next stages of our research.

2 Related Work

With the size of the Semantic Web currently approaching 40 billion triples, there has been a growing interest in studying different aspects related to its use and characteristics. Two recent studies [9,4] investigated whether casual web users can satisfy their information needs using the Semantic Web. The first study focused on extracting the main objects and attributes users were interested in from query logs which were then compared with Wikipedia templates to examine whether the schema of structured data on the web matched the users' needs as a key indicator of the success of semantic search. On the other hand, Halpin [4] used a named entity recogniser to identify names of people and places together with WordNet [3] to identify abstract concepts found in the users' queries. To investigate whether the Semantic Web provided answers to these queries, Falcon-S [2] was used as the Semantic Web search engine and the results of executing the queries were analysed. On average, 1,339 URIs were returned for entity queries, while 26,294 URIs were returned for concept queries. The authors explained this finding that semantic search engines similar to FalconS contain interesting information for ordinary users.

Fig. 1. An example of a combined log format entry [10]

Möller et al. [10] were the first to address the usage patterns of Linked Open Data (LOD). Unlike previous studies which had a primary focus on the content of the queries, this study had a broader view of web usage mining: it answered the questions of who is using LOD and how it is being used. The agents issuing the requests are classified into semantic and conventional based on their ability to process structured data. Additionally, the study investigated the relevance of a dataset according to how its usage statistics are affected by events of public interest such as conferences or political events. Similarly, Kirchberg et al. [8] used query logs provided by the USEWOD2011 data challenge[1] to analyze the relationship between traffic of queries to Linked Data resources and whether different time frames have an influence on this traffic.

The work done by Arias et al. [1] builds on [10] and performs further analysis on the nature of the SPARQL queries. The structure of the queries was examined to identify the most frequent pattern types, joins as well as SPARQL features such as `OPTIONAL` and `UNION`. This information is valuable in a number of ways including query optimisation.

3 Query Logs Analysis

3.1 Modelling Query Logs

In order to identify concepts and relations of interest from user queries, there is a need to formalise individual query logs to a structured and standardised representation. We propose the QLog (QueryLog) ontology[2] to represent the main concepts and relations that can be extracted from a query log entry and by its subsequent analysis stages. The ontology has been developed by identifying the concepts of a log entry that follows the Combined Log Format (CLF)[3]. Fig. 1 shows an example of a CLF log entry.

A query log entry is extracted to identify the different properties of the log entry including date and time, response size, response code, agent, query string (including SPARQL query) etc. In addition to the concepts that were identified from a CLF log entry, the QLog ontology also contains concepts to describe our analysis on the query log entry itself. The query string (identified as Request String in a CLF log entry) is further parsed and analysed to identify which concepts and relations have been queried for. The SPARQL query is also analysed to identify properties that can be derived like

[1] `http://data.semanticweb.org/usewod/2011/challenge.html`
[2] The QLog ontology and a video of SEMLEX are available at `http://galaxy.dcs.shef.ac.uk/QLogAnalysis/`
[3] `http://httpd.apache.org/docs/1.3/logs.html#combined`

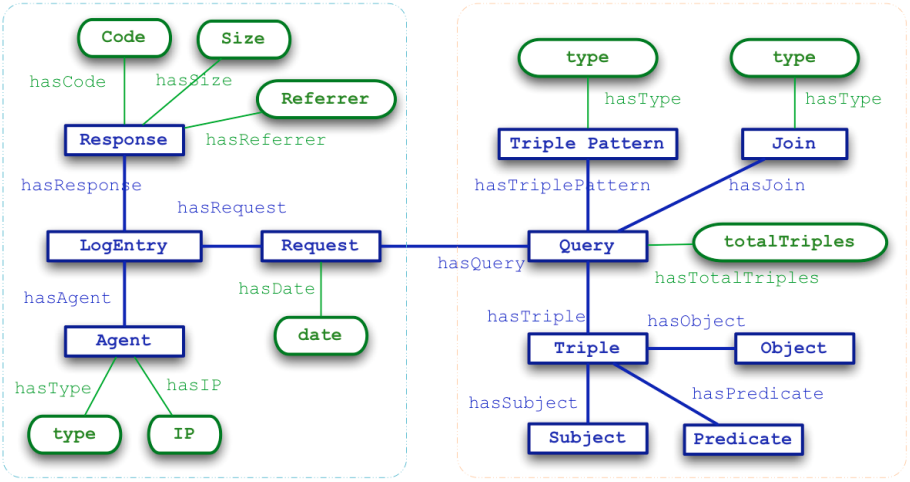

Fig. 2. The Query Log (QLog) Ontology. CLF concepts appear on the left and analysis concepts on the right.

types and number of triple patterns, joins, filters etc. Fig. 2 shows the proposed QLog ontology.

3.2 Analysing Query Logs

Fig. 3 shows the steps carried out in the analysis of the query logs. Since a web server log includes requests to particular web pages, RDF resources or to its SPARQL endpoint, the first step in the analysis is to filter the dataset and extract the requests issued to the SPARQL endpoint. The properties associated with a log entry, as shown in the QLog ontology are extracted first. These include the agent type and IP address, the request date as well as the response code, referrer and result size. The IP address can be used in different user-based studies that requires identifying requests coming from the same user. The request date was used in the study carried out in [8] to investigate the relationship between Linked Data resources and traffic of requests to these resources over different time windows. Agents requesting resources can be browsers (human usage), bots (machine usage), as well as tools (curl, wget, etc.) and data-services [10]. Identifying kinds of agents requesting resources and their distribution is useful for designers of Linked Data tools to understand what information is being accessed and how.

The next step was to verify the correctness of each SPARQL query before extracting its properties. Queries were parsed using Jena[4] and those producing parsing errors were excluded. For each successfully-parsed query, its type was first identified. The type can be either `SELECT`, `ASK`, `CONSTRUCT` or `DESCRIBE`. In this analysis, we only considered `SELECT` queries since it accounted for almost 97% of the query logs [1]. A SPARQL query can have one or more triple patterns, solution modifiers such as `LIMIT` and `DISTINCT`, pattern matching constructs such as `OPTIONAL` and `UNION` as well

[4] http://jena.sourceforge.net/index.html

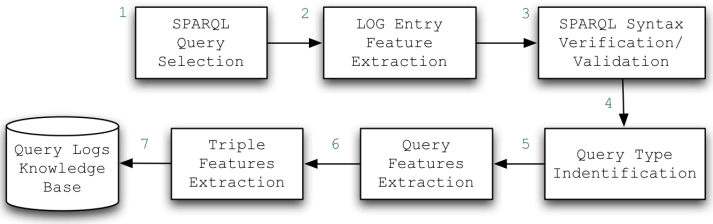

Fig. 3. Query Logs analysis process diagram

as `FILTER`s for restricting the solution space. These query parts are identified and triple patterns are analysed to extract the properties associated with the query and the triples.

A triple pattern consists of three components: a *subject*, a *predicate* and an *object* with each component in a triple pattern being either bound (having a specific value) or unbound (as a variable). There are 8 types of triple patterns according to the place of existence of variables and constants. The most general one is <?S, ?P, ?O> which is used to retrieve everything in the queried data. More specific ones include patterns having 1 variable such as <S, P, ?O> which retrieves the object values given a subject and a predicate, or 2 variables such as <S, ?P, ?O> retrieving all predicates and their values for a given subject. Finally the most specific triple pattern <S, P, O> does not ask for any data to be returned. After excluding the most general and specific triple patterns, the other types were identified when used in a query.

Two triple patterns used in a query can be joined by using the same unbound component in both of them. For instance `?x hasName ?y` and `?x hasAge ?z` are joined using the unbound subject `?x`. Using this approach, six different join types were identified according to the place of the common variable in both patterns. For instance, the *Subject-Subject* join is one in which the common variable is found in the Subject place in both triple patterns. The other types are *Subject-Predicate, Subject-Object, Predicate-Predicate, Predicate-Object* and *Object-Object*.

4 Dataset Analysis

DBpedia was the founding dataset of the LOD cloud and remains one of its largest; indeed, in 2009, it was shown that almost 83% of all Semantic Web queries related to DBpedia [4]. Its knowledge base currently describes more than 3.4 million things spanning multiple domains such as *People, Places* and *Species*.

The data used in this study is made available by the USEWOD2011 data challenge[5]. The query logs follow the combined log format. The challenge data however included two additional fields, namely *Country code* and *Hash of original IP* to support both location and user-based analyses. The logs contained around 5 million queries issued to DBpedia over a time period of almost 4 months. Table 1 shows the basic statistics of the query logs.

In order to count the number of unique triple patterns used in the queries, the variables found in the patterns were first normalised. In that sense, the two triple patterns

[5] `http://data.semanticweb.org/usewod/2011/challenge.html`

Table 1. Statistics summarising the query logs

Number of analysed queries	4951803
Number of unique triple patterns	2641098
Number of unique subjects	1168945
Number of unique predicates	2003
Number of unique objects	196221
Number of unique vocabularies	323

'*...dbpedia...resource/X hasPage ?page*' and '*...dbpedia...resource/X ?hasPage ?homepage*' were considered to be similar since the same information is being requested. The large difference between the number of unique subjects and objects supports the findings of [1], as they showed that the most frequent triple pattern is <S P ?O>. This means that most of the queries request the value of the object, given a specific subject and predicate; the object is given as a variable and thus not counted.

In a similar way to analysing complexity of keyword queries on the Web of Documents in terms of query length, the first metric for Linked Data queries is the number of triple patterns used in a query. Almost 65% of the queries contained only 1 triple pattern, 18% contained 2 triple patterns while 15% contained 3 triple patterns. This shows that queries follow a power-law distribution in which most of the queries are simple and lie at the head of the distribution, while more complicated queries with triple pattern counts ranging from 4 to 20 lie at the tail of the distribution. After excluding the most general and specific types of triple patterns (?S,?P,?O and S,P,O), the distribution of the other types is shown in Table 2.

Table 2. Distribution of triple pattern and join types in the queries.

TP Type	Queries Percentage	No. Queries	No. Joins	Queries Percentage	No. Queries
S P ?O	49.55%	3760649	0	85.8%	4242899
S ?P ?O	25.94%	1968511	1	9.8%	485307
?S P ?O	12.84%	974882	2	2.6%	132128
?S P O	9.51%	722091	3	0.8%	37646
S ?P O	1.17%	88679	5	0.6%	30539
?S ?P O	0.97%	73888	6	0.07%	3560

As shown in [1], the analysis shows that the most frequent triple pattern is <S P ?O>. This means that for almost 50% of all queries, the information need is very specific: the value of a specific predicate for a given resource is required. Indeed, since the second most frequent pattern is <S ?P ?O>, over 75% of queries are about a specific resource. Some Linked Data querying approaches build indexes to identify the relevant sources for answering a query or even use them to obtain the answer itself. In this sense, the identification of the most frequent triple patterns is valuable to optimise the indices which in turn would improve the search performance.

Additionally, Table 2 shows that around 86% of the queries are simple with no joins. The number of joins then increase from 1 to 20 with an inverse relation with the percent of queries. An interesting finding of the analysis is that more than half of the joins (54%) were of type *Subject-Subject* and almost 32% were of type *Subject-Object*.

Knowing this information is valuable for query planning and optimisation during the query execution process.

In addition to the basic graph patterns, there are three other constructs that can be used: `OPTIONAL`, `UNION` and `FILTER`. Only `FILTER` occurred in more than half of the queries (55%). It is used to restrict the results according to a given criteria. The most frequently observed use was with `LANG` which restricts the results to the specified language. The `OPTIONAL` feature increases flexibility: it allows information to be returned if found but does not reject the solution when part of the query does not have matches in the data. However, it is arguably the most expensive operator in query evaluation [11]. It is interesting to find that it occurred in only 15% of the observed queries. Although this low rate will be beneficial to search engines, it raises the question of why it is not used more frequently in Linked Data queries. One explanation could be the knowledge and experience of the query language required to use such a construct effectively.

Finally, the `UNION` construct combines graph patterns in the same way as `OR` is used in SQL and occured in only 9.5% of queries.

The number of variables found in the `SELECT` part of a SPARQL query shows how many data items the user needs in the results. These can be instances, concepts or relations between them. This was found to range between 1 and 13 with a variable count of 2 being the most frequent followed by 1 and 3. Using `SELECT *` indicates either a lack of knowledge regarding the structure of the data or a broad and non-specific information need (e.g., data exploration). Interestingly, this accounted for only 9.5% of the queries; thus, more than 90% of queries had a specific information need and knowledge of the data structure.

5 Consuming Query Log Analysis

5.1 Visualisation of Query Logs

Analysis of query log entries can provide great insights into how individuals and software agents consume Linked Data. Making such analysis efforts available using a formalised representation is valuable since it facilitates a generic approach to consume such data. For example, experts can gain an understanding of the information needs that emerge from a dataset. Visualisation tools and interfaces can consume such data thereby providing a quick means to identify emerging trends and patterns from collective information needs. Fig. 4 shows how we make use of our analysis to provide visualisations to users.

In order to consume the query log analysis findings, we have developed software to visualise query log analysis data captured using the QLog ontology described above. It provides two different types of information:

1. Concept Graph: concepts according to query frequency (A, in Fig. 4)
2. Predicate Order Tree: query predicate order (B, in Fig. 4)

The query log analysis process described in Fig. 3 results in RDF triples that are stored in a local triplestore (‘KB’ in Fig. 4). In order to relate the information needs with concepts in the dataset, the Linked Data endpoint is initially queried to identify

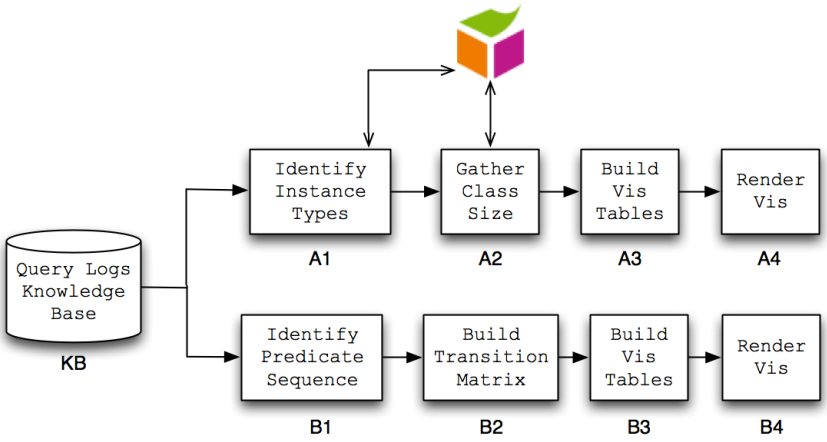

Fig. 4. Consumption of QueryLog analysis results

the types of the instances being queried for (A1). For example, querying the DBpedia endpoint for the type of the instance 'Acura ZDX' returns http://dbpedia.../Automobile. Once a type has been determined for a particular instance, the endpoint is queried again to understand how many instances in the data are associated with that type (A2). In this example, DBpedia will be queried again for how many instances of Automobiles exist. This process would continue until all the instances and classes have been analysed. The resulting information is then be assimilated into data tables (A3). A further interesting feature that can be identified by analysing SPARQL query logs is how users query for information especially when using multiple predicates to connect individual triple patterns. We refer to a predicate order as the order of the predicates that are observed in a query when the triple patterns use different predicates to identify different subsets of the data. Consider:

```
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?name ?place WHERE {
       ?person dbo:birthPlace ?place.
       ?person foaf:name ?name.
}
```

Here, the user's predicate of interest moves from `dbo:birthPlace` to `foaf:name`. In essence, the user is initially interested in looking at birthplaces of persons and then looking at their names. This can now be collectively studied after analysing all of the formalised query logs. Studying such patterns can provide insights into how the user's information need spread over different predicates and how these predicates are used together.

The process for visualising predicate order involves identifying the predicates that users have used. This can be retrieved by querying the KB for triple predicate instances, which provides the predicate order (B1). The triples are instantiated according to the order they appeared in the query. This ensures that the consistency for the predicate orders is maintained. The orders for all query log entries are then assimilated to construct a matrix (B2), which is then converted to data tables (B3). The data tables generated

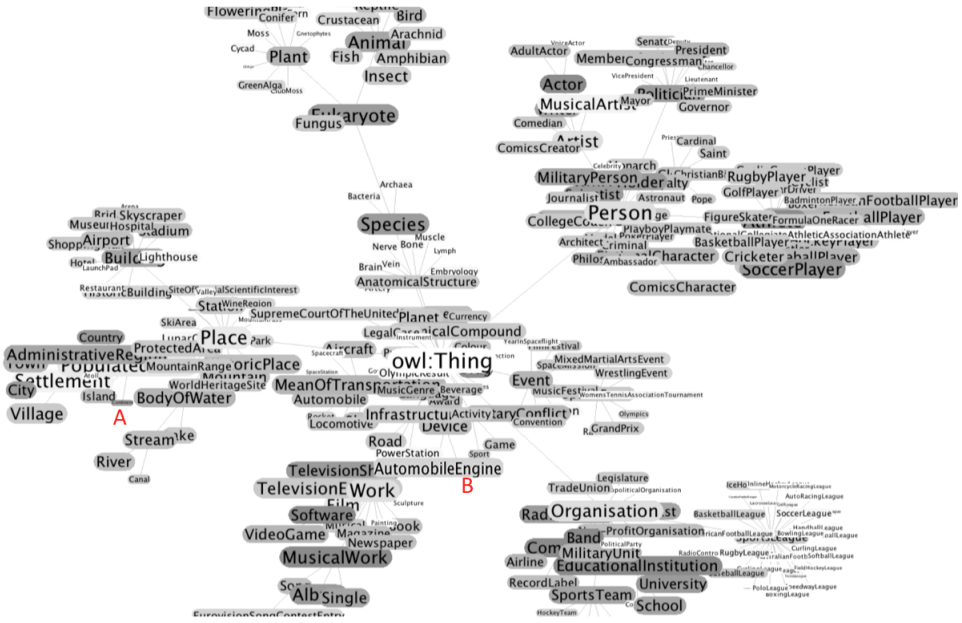

Fig. 5. Exploring information needs of DBpedia users (Concept Graph). Node size represents the amount of instances (larger nodes represent more instances), color represent the amount of user interest (darker nodes represent more interest)

in A3 and B3 are then rendered (A4 and B4) using the Prefuse Visualization Toolkit (`http://prefuse.org/`).

5.2 SEMLEX - Exploring Information Needs

SEMLEX (SEMantic Logs EXplorer) was designed to explore and present analysis on large query log datasets such as that described in Section 4 . The current implementation of SEMLEX provides the user with two visualisations: Concept Graph and Predicate Order Tree, though several other visualisations will be included in the future. Concept Graph essentially visualises the underlying ontologies, visually encoding nodes with information such as amount of data, query frequency.

Fig. 5 shows the relationship between query concept frequency and dataset concept frequency. In this example, the ontology classes have been visually encoded using two sets of information: size (to represent how many instances are types of the concept within the dataset) and colour (to represent how many times the concept has been queried). The larger the size of a class, the greater the number of instances. Similarly, the darker the colour for a class, the more frequently it has been observed in queries. For example, the concept ‘wrestler’ has been queried more times than ‘soccer player’ (Fig. 5, top right), even though the number of instances of wrestlers is fewer than for soccer players. While aggregating all the queries to identify which concepts

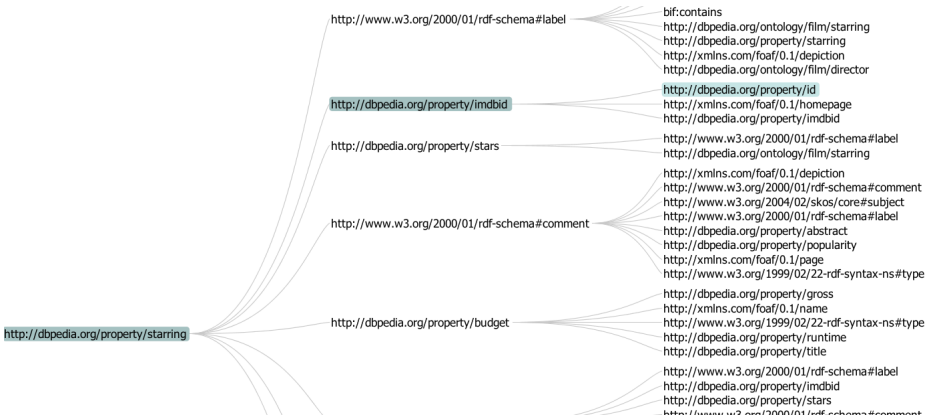

Fig. 6. Exploring information needs of DBpedia users (Predicate Order Tree). This shows, for a particular property, which predicates are most likely to be used in a single query.

are most queried can provide an insight to data providers on which sections of an ontology are more 'interesting' to all users, it may be useful to explore how users are querying the dataset. We found that the most commonly queried concepts in DBpedia were as follows: '*Person, Work, Organisation, Artist, Film, Place, PopulatedPlace, MusicalArtist, Settlement, Drug, Company, Software, Band, Actor, Athlete, MusicalWork, EducationalInstitution, Album, OfficeHolder, RadioStation, Country, Species, Politician, City, SoccerPlayer*'.

SEMLEX also enables users to see how predicates have been used along with other predicates in individual queries. The tool accumulates all the predicates to build a matrix, which records which predicate has been used with the next and in which order. This matrix is rendered as a tree. Fig. 6 shows an example in which a user explores the most commonly used predicates associated with http://dbpedia.org/property/starring. The subtrees of the node are arranged according to their usages: label being used most often while budget being used less frequently. In our example, we focus on how users have queried for individuals who have starred in movies and then focus their search on IMDB entries. However, it seems that more users have looked for individuals who have starred in movies and then queried for the movies they have starred in or the movie directors. Observations such as this can be interesting to other applications such as automatic query suggestions, recommender systems, search tools, etc.

Fig. 7 shows the relationship between the information available in the dataset (instances) and the queries requesting this information. As per our expectation, we observed a direct relationship between the number of instances of a concept and the number of times they were queried. We explain this as users more often query for concepts for which there is a larger amount of information in the dataset.

However, the graph also shows some interesting anomalies. For instance, point A (shown in the figure) refers to the concept '*Continent* which has only 10 instances but appeared in almost 10,000 queries The same concept appears in Fig. 5 only as a small

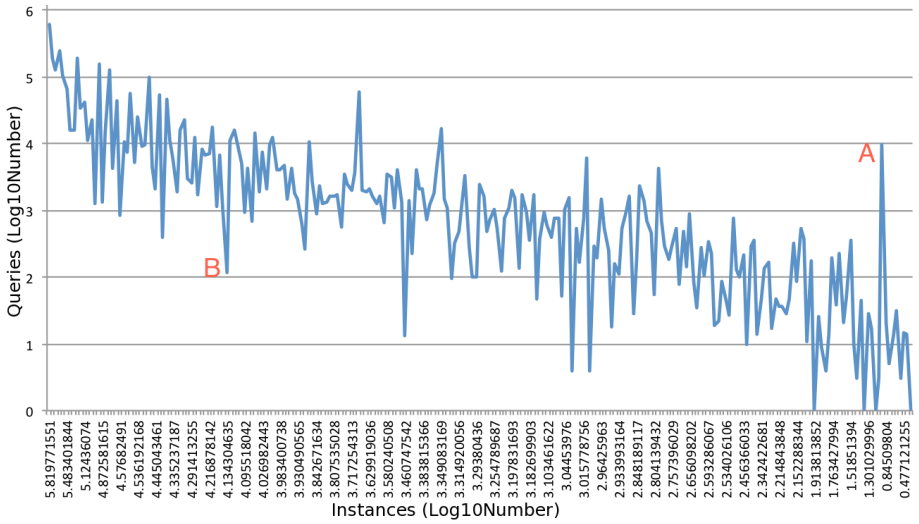

Fig. 7. Distribution of number of queries referring to a concept versus number of instances of that concept in the dataset.

node. In contrast, point B relates to the concept *AutomobileEngine* which exhibited lower than expected interest given the amount of available information.

Being aware of such a distribution (and dataset-specific points of interests) is valuable for both producers of Linked Data in terms of improving the structure of their data to better suit their users' information needs, as well as consumers such as designers of semantic search and visualisation tools who can better support their users when they know more about their needs in advance.

6 Conclusions and Future Work

This paper has presented an approach which can advance our understanding of the information needs of Linked Data consumers and help Linked Data providers match these needs. We described the analysis of semantic query logs and how the subsequent results can be represented in a formalised model. We presented a visualisation tool – *SEMLEX* – that demonstrates how these analyses can be consumed to explore trends and patterns in the queries.

Using DBpedia as a case study, we followed our proposed approach to analyse a sample of its query logs (around 5 million queries) from the USEWOD2011 data challenge. However, our proposed approach, model and visualisation tool are independent of any dataset and can, therefore, be used for any similar analysis of SPARQL query logs. Nevertheless, this study provides a useful insight into the information needs of Linked Data users by highlighting patterns and trends inherent in their queries. This reveals great potential for different applications consuming Linked Data. For instance, a semantic search tool could benefit from having an advance knowledge of the most queried categories and the associated search patterns followed by users.

In future work, we intend to apply our approach to examine other datasets with different features such as SWDogFood as a domain-specific dataset targeting Semantic Web researchers. We further intend to study query logs that span multiple datasets such as the ones in the Linked Open Data Cloud Cache[6]. This could present a more representative view of Linked Data queries in terms of size and domain coverage. Additionally, it would show how the query exchange between different datasets in the cloud occur and whether the Linked Data principle of connecting datasets is being used in real-world queries. Further on, we also intend to understand how the user queries evolve time.

Acknowledgements Elbedweihy and Wrigley are funded by the EU FP7 Project SEALS (Semantic Evaluation at Large Scale, FP7-238975); Cano is funded by CONACyT, grant 175203; Mazumdar is funded by SAMULET, a project supported by Rolls Royce Plc and the UK Government Technology Strategy Board.

References

1. M. Arias, J. D. Fernández, M. A. Martnez-Prieto, and P. de la Fuente. An empirical study of real-world sparql queries. *CoRR*, abs/1103.5043, 2011. informal publication.
2. G. Cheng, W. Ge, and Y. Qu. Falcons: searching and browsing entities on the semantic web. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 1101–1102, New York, NY, USA, 2008. ACM.
3. C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.
4. H. Halpin. A query-driven characterization of linked data. 2009.
5. C. Hölscher and G. Strube. Web Search Behavior of Internet Experts and Newbies. *Computer Networks*, 33(1-6):337–346, 2000.
6. B. J. Jansen and A. Spink. An analysis of web searching by european alltheweb.com users. *Information Processing and Management: an International Journal*, 41(2):361–381, 2005.
7. B. J. Jansen, A. Spink, and J. Pedersen. A temporal comparison of altavista web searching: Research articles. *J. Am. Soc. Inf. Sci. Technol.*, 56:559–570, April 2005.
8. M. Kirchberg, R. K. L. Ko, and B. S. Lee. From linked data to relevant data – time is the essence. *CoRR*, abs/1103.5046, 2011.
9. P. Mika, E. Meij, and H. Zaragoza. Investigating the demand side of semantic search through query log analysis. In *SemSearch*, 2009.
10. K. Möller, M. Hausenblas, R. Cyganiak, and G. A. Grimnes. Learning from linked open data usage: Patterns and metrics. In *Proceedings of the WebSci10: Extending the Frontiers of Society On-Line*, 2010.
11. J. Pérez, M. Arenas, and C. Gutierrez. Semantics and complexity of sparql. *ACM Trans. Database Syst.*, 34:16:1–16:45, September 2009.
12. C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.
13. A. Spink, S. Ozmutlu, H. C. Ozmutlu, and B. J. Jansen. U.s. versus european web searching trends. *SIGIR Forum*, 36:32–38, September 2002.
14. G. Tummarello, E. Oren, and R. Delbru. Sindice.com: Weaving the open linked data. In *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC2007), Busan, South Korea*, volume 4825 of *LNCS*, pages 547–560, Berlin, Heidelberg, November 2007. Springer Verlag.

[6] `http://lod.openlinksw.com/`