Improving Semantic Search Using Query Log Analysis*

Khadija Elbedweihy, Stuart N. Wrigley, and Fabio Ciravegna

Department of Computer Science, University of Sheffield, UK
{k.elbedweihy, s.wrigley, f.ciravegna}@dcs.shef.ac.uk

Abstract. Despite the attention Semantic Search is continuously gaining, several challenges affecting tool performance and user experience remain unsolved. Among these are: matching user terms with the search-space, adopting view-based interfaces in the Open Web as well as supporting users while building their queries. This paper proposes an approach to move a step forward towards tackling these challenges by creating models of usage of Linked Data concepts and properties extracted from semantic query logs as a source of collaborative knowledge. We use two sets of query logs from the USEWOD workshops to create our models and show the potential of using them in the mentioned areas.

Keywords: semantic search, query logs, linked data, usability

1 Introduction and Problem Statement

The proliferation of structured data on the web is driving innovation in both 'conventional' search (information retrieval – IR) as well as in semantic search. For instance, the use of semantic markup (such as Microformats and RDFa) within existing HTML/XHTML webpages has helped mainstream web search engines such as Google and Bing to enhance their result pages by providing additional and related information to that which would normally have formed the query results.

In contrast to web search engines, Semantic Web search engines such as Swoogle [10], Watson [8] and Sindice [29] index data on the Semantic Web and act more as gateways to Semantic Web documents or data. The results of such systems are intended for Semantic Web professionals rather than end-users. In a more user-friendly approach, mashups like Sig.ma [28] and VisiNav [14] integrate data from different sources to provide rich descriptions about the searched-for

* This work was partially supported by the European Union 7th FWP ICT based e-Infrastructures Project SEALS (Semantic Evaluation at Large Scale, FP7-238975).

In: C. Unger, P. Cimiano, V. Lopez, E. Motta, P. Buitelaar, R. Cyganiak (eds.): Proceedings of Interacting with Linked Data (ILD 2012), Workshop co-located with the 9th Extended Semantic Web Conference, Heraklion, Greece, 28-05-2012, published at http://ceur-ws.org

concepts. At the heart of these technologies is the use of Linked Data providing the opportunity to exploit explicit and implicit knowledge.

Little work has been conducted on how to exploit Semantic Web search for end-users with potentially complex information needs and thus complex queries. One approach is to provide a natural language interface to such a search tool which allows a user to express their query in a (near) natural manner. Tools which have followed this approach include Querix [18], PowerAqua [19] and Freya [9]. However, such approaches suffer from a significant problem: a high degree of abstraction between the words which may be used by the user in formulating their query and the underlying semantically-corresponding terms in the ontology. Indeed, this problem is equally applicable to other approaches including a conventional keyword-based approach. Formal evaluations of semantic search technologies [17,30,32] have shown that it is very helpful (both in terms of user experience but also for response precision and recall) for users – especially who are unfamiliar with the underlying data – to explore the search space while building their queries.

In an attempt to provide context within the underlying data, a number of tools have adopted a visual approach to query construction (e.g., Semantic Crystal [4], K-Search [5], Corese [7]). However, such approaches tend to be focussed on relatively small data set sizes (especially when compared to the wider Linked Data context). The outstanding challenge is to adapt this approach to Linked Data in which there are multiple data sets, of widely varying size and spanning many domains. Recently, initial work in this vein demonstrated a visual query interface that helps users build a subgraph of interest from the underlying data through exploration and navigation [6]. However, even this can result in unwieldy lists of triples which are not conducive to a positive user experience.

The difficulties in Semantic Web search are not confined to abstraction, query construction or data visualisation. An additional problem focuses on the results of the query execution: what to return to the user and how to display it. We have shown previously [32] that semantic search tools should go a step further and augment the direct answer with associated information in order to provide a 'richer' experience for the user. Additionally, returning information related to entities and concepts found in a query might also be of interest to users [22,23].

In this paper we present a new approach which uses *collaborative knowledge* to address these problems. In a similar way in which traditional search engine logs record information about search histories including queries submitted by the users and their subsequent interactions (see [2,15,27,31] for early examples of analyses on such logs), the logs from interactions with Semantic Web search engines can be used to extract a rich picture of data use and user behaviour.

Whilst a number of studies have used log analysis to investigate the high level use and characterisation of Linked Data [13,23], [24] was the first to study it's use through analysing semantic query logs issued to SPARQL endpoints including DBpedia and SWDF. The analysis considered 'low-level' factors such as the type of requesting agent (human or software) and the structure of the query. Subsequent studies on the same query logs identified the most common query

types, SPARQL features and types of triple patterns [1]; proposed a method to derive more useful labels for LD entities based on the variable names used in the queries [12]; and showed how the analysis of semantic query logs can detect errors and weaknesses in LD ontologies and in turn support their maintenance [20].

In our previous work [11], we introduced a new approach for analysing semantic query logs and found that a small set of concepts and relations in a data set often account for a large proportion of the queries and thus may be of more interest to Linked Data users. In the current paper, we extend this approach in order to demonstrate that careful log analysis can be viewed as a proxy for information need and be used to enhance the search process at a number of different stages from query construction, through search engine optimisation to result presentation. To achieve this, we use three different models, each of which captures information regarding different aspects of the patterns present in the multi-user query logs. We will show how combinations of these three models allow us to address the matching of user search terms to the underlying data vocabulary; the creation of data subgraphs for visualising the the underlying data and, finally, for enhancing the results returned to the user.

The remainder of the paper is structured as follows. Section 2 describes the analysis performed on the semantic query logs and the three models used to exploit this analysis. Subsequent sections show how these models can be used to address the three problems described above. Section 3 demonstrates using the models to improve the abstraction problem between user terms and the underlying data vocabulary. Section 4 shows how the results can be augmented in two different ways by exploiting the models. Section 5 illustrates how the models can be used to assist in visualising large data sets for query formulation. It should be emphasized that the details presented in Sections 3, 4 and 5 are a proof of concept for the usage of the models presented in Section 2 (i.e. the results shown come from a "pen and paper" exercise as opposed to having been produced by an implementation of the approach). Finally Section 6 discusses the strengths and weaknesses of the approach and Section 7 draws a number of conclusions and describes directions for future work.

2 Semantic Query Logs Analysis

In previous work [11] we introduced a new approach for analysing and representing information need using semantic query logs. Information need was defined as "the set of concepts and properties users refer to while using SPARQL queries". A SPARQL query can have one or more triple patterns, solution modifiers (such as LIMIT), pattern matching constructs (such as OPTIONAL) and mechanisms for restricting the solution space (such as FILTERs). A triple pattern consists of three components: a subject, a predicate and an object with each component being either bound (having a specific value) or unbound (as a variable).

Extending our previous analysis [11] we formulate the information inherent in semantic query logs into three models which capture:

- the concepts used together in a query: the query-concepts model

Table 1. Statistics summarising the query logs analysed.

	USEWOD2012	USEWOD2011
Number of queries	8866028	4951803
Number of unique triple patterns	4095011	2641098
Number of unique bound triple patterns	3619216	2571662

- the predicates used with a concept: the concept-predicates model
- the concepts used as types of one Linked Data entity: the instance-types model

We follow the same extraction steps but only extract triple patterns with bound subjects or objects to identify concepts (type of the subject/object) and predicates queried together which are used to build the proposed models.

2.1 Data Set

We use two sets of DBpedia query logs made available at the USEWOD[1] workshops (see Table 1). After extracting bound triple patterns [11], we identify the types associated with each distinct resource appearing as a subject or an object in the query by querying the Linked Data endpoint.

2.2 Models

In order to describe the proposed models, we use the following example query throughout the rest of this section:

```
SELECT DISTINCT ?genre, ?instrument WHERE
{  <http://dbpedia.org/resource/Ringo_Starr> ?rel <http://dbpedia.org/resource/The_Beatles>.
   <http://dbpedia.org/resource/Ringo_Starr> dbpedia:genre ?genre.
   <http://dbpedia.org/resource/Ringo_Starr> dbpedia:instrument ?instrument.
}
```

Query-Concepts Model This model captures the Linked Data concepts used in a whole query. All bound triple patterns (bound subject or object) in a single query are first identified and their types are retrieved from the Linked Data endpoint. The frequency of co-occurrence of each concept pair is accumulated. For the example query, the types retrieved for 'Ringo Starr' include `dbpedia:MusicalArtist` and `umbel:MusicalPerformer` while the 'The Beatles' has among its types `dbpedia:Band` and `schema:MusicGroup`. The frequency of co-occurrence of each concept in the first list with each concept in the second list is therefore incremented (e.g. `MusicalArtist` and `Band`).

Concept-Predicates Model This model captures the Linked Data concepts and predicates in a query. Again, bound triple patterns are identified; however, only types of instances used as subjects are retrieved. The frequency of co-occurrence of each of the types with the predicate used in the triple pattern – if available – is accumulated. To illustrate, the second triple pattern in the example query increments the co-occurrence of `dbpedia:MusicalArtist` with `dbpedia:genre` and `umbel:MusicalPerformer` with `dbpedia:genre`.

[1] `http://data.semanticweb.org/usewod/2011(2012)/challenge.html`

Instance-Types Model Ontologies consist of hierarchies of classes. In theory, these classes are linked together through subsumption or equivalence relationships. In practice, datasets in the Linked Data cloud are yet loosely coupled; lacking the required links [16,26]. A Linked Data entity can have several concepts as its types – from multiple datasets – which are not linked. The knowledge of one type is hence not sufficient for complete reasoning on the data. The *instance-types* model captures the concepts used as types for one instance. For the entity 'Ringo Starr', the frequency of co-occurrence of its types `dbpedia:MusicalArtist` and `umbel:MusicalPerformer` are accumulated in the model.

3 Matching User Terms to Linked Data Vocabularies

As explained above, non view-based semantic search approaches face the problem of matching user terms found in a query to the vocabulary of the search-space. [22] tried to tackle this problem on the traditional Web by mapping query terms to relevant concepts in DBpedia. However, scalability can be an issue for both since the matching process can be very expensive especially with the size of the Linked Data cloud. Additionally, although DBpedia is known as a central hub in the cloud, matching query terms to one specific dataset may lead to missing information associated with other semantically-equivalent concepts in different datasets due to the lack of links between them. This affects the ability to reason on the data and return complete results [16, 21, 26].

Based on their analysis of query logs issued to DBpedia, [11] and [20] drew two important conclusions. Firstly that a small number of classes appeared more frequently in the queries than others (e.g. Film, Place, MusicalArtist, Drug) and, secondly, that the dataset population is not an accurate indicator of the usage/interest in a specific concepts; some concepts had large number of instances and were only queried few times. Supported by the above observations, we believe that creating a model of usage of concepts and relations from query logs has a potential to improve the performance of a semantic search approach with respect to the matching task.

Initial stages followed by the proposed approach for processing the query involves standard NL parsing steps such as tokenization, stemming and lemmatization, and producing a parse tree. Entity extraction and classification is achieved by AlchemyAPI[2] and the NERD ontology[3] is used to map AlchemyAPI classes to the DBpedia classes found in the usage models. Rather than having to query all the underlying data, the models attempt to provide a small abstraction of that data which can act as a source of *collaborative knowledge* [25] capturing the most frequently queried Linked Data concepts and predicates. This could be used to provide matches and, in turn, answers for many commonly issued queries on Linked Data. Since the query terms associated with the entity can be either properties of that entity or concepts in a relation with it, both *query-concepts* and *concept-predicates* model are then queried for matches.

[2] http://www.alchemyapi.com
[3] http://nerd.eurecom.fr/ontology/

3.1 Illustrative Examples

In the rest of this section, we will use a set of examples and show the results returned by a selection of state-of-the-art systems in order to demonstrate the issues and limitations discussed above. The selected systems are of interest in the SW community, spanning different categories: SW gateways, QA systems and mashups. The examples are carefully selected so that they are not targeting a specific category, and will allow us to illustrate how the proposed models can be used in the matching task.

Example 1: What is the population of New York? The query can be given as a NL question to QA systems or as keywords – also entity query – to other systems; *population of New York*.

Sindice : The top 5 results returned by Sindice are as follows:

```
1. Armonk, New York           : http://www.mpii.de/yago/resource/Armonk,_New_York
2. About: New York City       : http://dbpedia.org/page/New_York_City
3. New York State Senate      : http://dbpedia.org/resource/New_York_State_Senate
4. Nova Iorque, New York      : http://linkeddata.uriburner.com/.../resource/New_York_City
5. Indian-American Population : http://www.scribd/.../New-York-Citys-IndianAmerican-Population
```

The second item in the results is showing the DBpedia page for New York city which contains the answer to the query. However, although being at the top of the list, 60% of the results are only syntactically related to the query (as opposed to *semantically* related); i.e., containing the terms 'New York', 'population' or both. This is due to using syntax-driven techniques in the matching task which in turn affects the precision of the results and the user experience.

PowerAqua : PowerAqua returns the following answer to the query:

```
Richard Lewontin  :  < PopulationGeneticists , birthPlace,  New_York >
```

Although PowerAqua could provide correct answers for other queries, this one shows the effects of the matching problem on the tool's performance. Attempting to find mappings for the query terms in the whole dataset – DBpedia in this case – resulted in 17 different ones for 'population' and three for 'New York'. They included non-semantically equivalent ones like `yago:RussianPopulationGroups` and `res:General_Population`, which – although they could be excluded before returning the final answer to the user – affected the tool's performance causing it to require approximately 13 seconds to return the found triples.

FalconS : The top 5 results returned by the object retrieval search option provided by FalconS are as follows:

```
1. New York City          : http://dbpedia.org/resource/New_York_City
2. York                   : http://dbpedia.org/resource/York
3. New York State Assembly: http://dbpedia.org/resource/New_York_State_Assembly
4. York County            : http://www.rdfabout.com/rdf/usgov/geo/us/pa/counties/york_county
5. New York State         : http://www.rdfabout.com/rdf/usgov/geo/us/ny
```

Although it returns the resource 'New York City' in the first rank, the other results retrieved are again only syntactically related to the query terms.

Our Approach : For this example, 'New York' is extracted and classified as `alchemyapi:City` which is then mapped to `dbpedia:City`. While no matches were found for 'population' associated with the concept `dbpedia:City` in the *query-concepts* model, the *concept-predicates* model returned several matches including `populationTotal` and `dbprop:populationDensityKm`. Since the user query did not provide a specific intent for 'population', all the mappings are considered and the answers from equivalent ones (`dbprop:populationTotal` and `populationTotal`) are merged.

To illustrate the use of the *query-concepts* model, consider the query 'Which islands belong to Portugal'. With syntactical matching, both 'island' and 'Portugal' could be matched with several instances, predicates or concepts (e.g., `island` and `res:Administrative_divisions_of_Portugal`). However, our approach attempts to find mappings only associated with the concept `dbpedia:Country` – classified type for Portugal – in the *concept-predicates* and the *query-concepts* models resulting in `dbpedia:Island` as the only match returned. The search is therefore done for a relation linking these two concepts. Querying the DBpedia endpoint, these concepts are found to be linked with the property `dbpedia:country` and a list of islands is returned, including `dbpedia-res:Porto_Santo_Island` and `dbpedia-res:Santa_Maria_Island` among others.

Example 2: Give me all the soccer clubs in Spain This query is from QALD workshop open challenge[4] where both Freya and PowerAqua – the participating tools – did not manage to return back all the results. We'll be using this example to explain the use of the *instance-types* model. We only compare it to PowerAqua since a demo for Freya is not available.

PowerAqua : PowerAqua matches Spain with six resources including `dbpedia-res:Spain` which is the main resource describing the country in the dataset. This however leads to missing results when the answer is given as cities or other places in Spain rather than the country itself. For instance, one such example of a missing result is `res:CD_Pozo_Estrecho` which is located in `res:Cartagena,_Spain`, a resource associated with several types including `dbpedia:Place`, `gml:_Feature`[5] and `schema:Place`. Tools following this approach thus favor precision over recall.

Tools favouring Recall : The other alternative that can be taken by tools in favour of recall over performance (time and scalability) is not to limit the results to a specific type. For the example query, this approach would search for soccer clubs that have a relation with any resource with a label containing the term 'Spain' (usually `rdfs:label` is used as a human identifier for the resource). The problem with this approach is that it affects the ability of a tool to scale over large datasets and to return answers for queries in real time [9, 19].

Our Approach : The two approaches explained above can be seen as the two ends of a 'precision and performance' versus 'recall' spectrum. Our approach attempts to balance the three. The same steps explained in Example 1 are followed

[4] http://www.sc.cit-ec.uni-bielefeld.de/qald-1

[5] `gml:_Feature` refers to `http://www.opengis.net/gml/_Feature`

resulting in extracting 'Spain' as an entity, classified as a Country and mapped to `dbpedia:Country`. The concepts associated with this class in the *instance-types* model are then extracted. An attempt to find mappings for the noun phrase 'soccer club' in the *query-concepts* and *concept-predicates* models results in `dbpedia:SoccerClub` as a match. Next, the approach tries to find instances of `dbpedia:SoccerClub` having a relation with instances of any of the following concepts (retrieved from the instance-types model for `dbpedia:Country`):

```
dbpedia:Country,dbpedia:Place,dbpedia:PopulatedPlace,schema:Place,schema:Country,
gml/_Feature,umbel:Country,umbel:PopulatedPlace
```

This query returns results including ones which would not be retrieved (e.g. `res:CD_Pozo_Estrecho`) if a specific type was specified (e.g., Country). On the other hand, it does not harm the performance since it limits the search space to a set of concepts rather than the whole dataset. One can argue that such information can be extracted for instances from the domain and range of certain properties. However, this not only requires knowledge of the exact properties that would return the results – in this example one of the properties is `dbprop:ground` – but also, as observed by [9], DBpedia properties included in `http://dbpedia.org/property/` do not provide domain and range classes.

4 Results Selection

In an attempt to improve the user experience, Google, Yahoo! and Bing use structured data embedded in web pages to enhance their search results (for example, by providing supplementary information relevant to the query)[6]. Although Semantic Web search engines and question answering systems index much more structured data, a similar functionality (results enhancement) is not yet provided to their users. FalconS returns extra information together with each entity found as an answer to a query. It returns predicates associated with the entity in the underlying data (e.g. type, label, etc.); [32] showed that augmenting the answer with such extra information provides a richer user experience. This is, however, different from Linked Data mashups (e.g. Sig.ma) and browsers (e.g. Tabulator [3]) which attempt to create rich comprehensive views of entities and allow interactive exploration and navigation of Linked Data respectively. Furthermore, [23] and [22] suggested that returning information related to entities found in a query would be of interest to the user.

4.1 Illustrative Examples

In the rest of this section, we illustrate how the proposed models can be used to return more information with the results. We distinguish between providing more information about each result item and more information that is related to the query keywords including concepts and entities.

[6] For example, Google Rich Snippets: `http://googlewebmastercentral.blogspot.com/2009/05/introducing-rich-snippets.html`

Return additional result-related information To our knowledge, only VisiNav and FalconS return extra information about each entity in the result list. For the query given in 3.1 and for the entity 'New York City', FalconS lists the following 10 properties with their values:

```
populationAsOf,dbprop:populationTotal,populationTotal,PopulatedPlace:populationTotal,
populationDensity,PopulatedPlace:populationDensity,dbprop:populationDensitySqMi,
dbprop:populationBlank,dbprop:populationMetro,PopulatedPlace:populationUrban'
```

However, the strength of the proposed idea lies in utilising query logs as a source of collaborative knowledge able to capture perceptions of Linked Data entities and properties and use it to select which information to show the user rather than depending on a manually (or, indeed, randomly) predefined set. Additionally, [22, 23] observed that a class of entities is usually queried with similar relations and concepts.

In order to return more information about each result item, the type of instance returned is first identified then the most frequently queried predicates associated with it are extracted from the *query-predicates* model. The top ranked ones are shown to the user, limited by the space available without cluttering the view and affecting the user experience. The user is given the ability to add more results which would retrieve the next set in the ranked list of predicates. Examples of concepts with their associated predicates list are given below:[7]

```
MusicalArtist-> rdfs:label,rdf:type,thumbnail,.....,genre,associatedBand,occupation,instrument,
birthDate,birthPlace,hometown,prop:yearsActive,foaf:surname,prop:associatedActs, ...
Film-> rdfs:label,rdf:type,foaf:page,.....,prop:starring,prop:director,prop:name,releaseDate,
prop:gross,prop:budget,writer,producer,runtime,prop:language,prop:cinematography, ...
Country-> rdfs:label,rdf:type,thumbnail,...,capital,foaf:name,anthem,language,leaderName,
currency,largestCity,prop:areaKm,motto,..,geo:long,geo:lat,leaderTitle,prop:governmentType, ...
```

Return additional query-related information Returning related information with the results of a query is an attempt to place the queried entities and concepts within context in the surrounding data which indeed assist users in discovering more information and useful findings that otherwise would not be noticed. Following our approach, the query concepts (include concepts and types of entities used in the query) are first identified. The most frequently occurring concepts used with them are extracted from the *query-concepts* model. Again, only a limited set (the actual size of which is determined on an application requirements basis) from the top ranked ones is returned. A set of examples are listed below with their co-occurring concepts.

```
MusicalArtist-> Film,Work,Band,Album,.....,schema:Movie,MusicalWork,Place,Actor,Athlete,
TelevisionShow,WrittenWork,Model,City,GridironFootballPlayer,Writer,schema:Event, ...
City-> Book,Town,WorldHeritageSite,.....,Person,foaf:Person,Country,Organisation,SportsTeam,
SoccerClub,Scientist,Artist,MusicGroup,Film,RadioStation,University,River,Hospital,Park, ...
Company->RecordLabel,foaf:Person,Work,.....,LawFirm,Place,Software,schema:Place,Website,
Broadcaster,TelevisionStation,University,Country,GovernmentAgency,Magazine,Convention, ...
```

[7] `prop` is used as a prefix for `http://dbpedia.org/property/` while the default prefix () is for `http://dbpedia.org/ontology/`

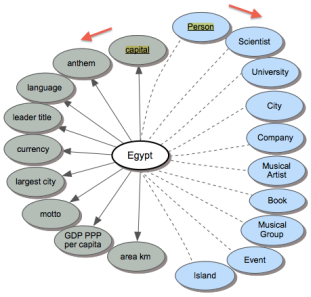

Fig. 1. Results returned by our approach for `Egypt`. Related concepts are on the right side and predicates on the left. For each side, elements are ranked with the top-most being most common and reducing in frequency in the direction of the arrows.

5 Data Visualisation

Query logs have been used in IR to provide query recommendations and suggest similar queries to users [2, 33]. [23] analyzed a set of Yahoo! query logs to learn prefixes and postfixes that can be suggested for a specific type of entities. Similarly, Google and Bing return *related searches* for a query. These approaches are however limited to suggesting parts or complete queries provided by other users in a related context rather than guiding users in formulating their queries.

On the Semantic Web, supporting query formulation is provided by *view-based/visual-query interfaces* (e.g., [4,6]) which allow users to explore the underlying data. This can be very helpful for users, especially those unfamiliar with the search domain. A problem facing these tools is the technical limitations such as the number of items that can be included in a graph without cluttering the view and affecting user experience. This increases in heterogenous spaces like the open web since it is a challenge to decide what should be shown to users.

In an attempt to tackle this challenge and to identify a specific area of interest, Smeagol [6] introduces a "specific-to-general" interface where it starts from an entity or a term entered by the user and builds a related subgraph extracted from the underlying data. After the user disambiguates the query term from a list of candidates, the tool returns a list of triples containing that term for the user to select from and add to his specific subgraph of data. In a dataset such as DBpedia – currently used by the tool's demo –, this list will often contain thousands of triples for the user to examine in order to select the required ones.

Our proposed approach uses the *concepts-predicate* and *query-concepts* models to move a step forward towards a more specific subgraph that allows users to explore the data around the entities they start with. It exploits the collaborative knowledge collected from different users and applications to derive the selection of concepts and predicates added to the subgraph of interest.

Using `Egypt` as a starting entity, Fig. 1 shows a set of concepts and predicates associated with this entity's type in the models. Selecting a related concept retrieves a similar subgraph for the new one and shows the predicates connecting the two concepts.

Table 2. PowerAqua matches for the given queries. The second column gives the number of matches found in the dataset and the third one shows triples generated from the matches to return the answers. Timings in the fourth column are approximate.

Query #	# Found Matches	Relevant Facts	Time (sec)
1	official language: 7 philippines: 4	Language, officialLanguage, Philippines ?, prop:officialLanguages, Philippines	28
2	official websites: 6 Charmed: 4, actors: 5 television show: 3 websites: 5	Award, geminiAward, Actor Award, laurenceOlivierAward, Actor Charmed, IS_A, TelevisionShow Actor, starring, Charmed	22
3	1950: 11 organisations: 7 founded: 9	Organisation, title, 1950's Organisation, foundation, 1950's Organisation, established, 1950's Organisation, founded, 1950's ... Organisation, artist, Project_1950 Organisation, recordLabel, Project_1950	29

6 Discussion

The previous sections illustrate how our models could be adopted in the two main issues discussed: matching user terms to Linked Data vocabulary and returning more information with the results. In this section, we use the following queries to facilitate the discussion of the main strengths and weaknesses of the proposed approach. The queries are from the 'Interacting with Linked Data' workshop[8].

1. What are the official languages of the philippines?
2. Give me the official websites of actors of the television show Charmed.
3. Which organisations were founded in 1950?

In order to show the effect of the matching problem on the tool performance, the mappings for the given queries and the time required – as given by PowerAqua – to find them are given in Table 2.

Query# 2 makes use of the *instance-types* model since the match `prop:website` is found among the predicates queried with `Person` rather than with `Actor`. Additionally, Query# 3 shows that the proposed approach is not limited to queries containing entities. In this as well as similar queries, the *query-concepts* and the *concept-predicates* models are used to find matches for the query terms (e.g. organisations). The matches are then ranked according to their syntactical similarity (exact or partial match) as well as the frequency of usage in the models, resulting in selecting the concept `dbpedia:Organisation` for this query.

As shown in Section 4.1, both *query-concepts* and *concept-predicates* models can contain semantically-equivalent concepts and predicates associated with one concept, from one or more datasets. Although they are used in the matching process in order not to miss candidate results (e.g., `prop:founded, foundingDate` in Query# 3), showing several concepts or properties to the user which have the same meaning but different names affects the readability of the results and the user experience. Therefore, the approach should include a schema-matching step before returning information to the user. This is not yet achieved by our approach; it is a challenging task and is part of our future work.

[8] `http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/index.php`

Table 3. Matches found by the proposed approach for the given queries. The second column shows the query entities and their types. The third one shows the matches found while the last 2 columns show the extra information extracted from the models

Query#	Entity:Type	Found Matches	More Info.	Related Info.
1	philippines Country	schema:Language, Language, language, prop:language prop:officialLanguages, officialLanguage, prop:languages	capital, anthem, language, leaderName, currency, largestCity, areaKm, motto	Person,Scientist, University,City, Company, Book MusicalArtist, Event, MusicalGroup
2	Charmed: Television-Show	Actor, prop:website	occupation, genre, birthPlace, birthDate, surName, hasPhotoCollection, nationality, gender	Film, Album, Organisation, MusicalArtist, SportsTeam, Broadcaster
3	N/A	Organisation, OrganisationMember Non_ProfitOrganisation, GeopoliticalOrganisation prop:founded, foundationPlace, prop:foundation, foundingDate	industry, city, country, website, divisions, subsid, president, established, staff, yearsActive, owner, founded foundationPlace	Company,Band Broadcaster, RadioStation, University, School, SoccerClub, MilitaryUnit, Airline

In order to reduce inconsistencies due to noise found in the query logs (incompatible concepts and predicates), the Linked Data endpoint is queried to check the validity of using a specific predicate with a given concept or the existence of a relation between two concepts before adding them to the models. However, there is no similar way to prevent errors in the *instance-types* model since they are caused by inconsistencies found in the dataset. For instance, the concept `Person` was found together with the concept `Country` as types for one Linked Data resource and thus was associated with it in the model. Fortunately, they have a very low frequency of co-occurrence and thus can be easily identified and removed. Another issue to consider is the existence of a few popular generic predicates (e.g. `label`) frequently occurring with most of the concepts and thus ranked at the top of the their associated predicates list. The ones we observed include `rdfs:label`, `rdf:type`, `thumbnail`, `foaf:page`, `rdfs:comment`, `foaf:depiction`, `abstract` and `foaf-homepage`. Although in deciding which predicates to show the user for a specific concept while building a query (Fig. 1), we chose to exclude these generic predicates similar to a stop list in IR, we believe this choice needs to be evaluated through a usability study which could reveal a different view.

7 Conclusions and Future Work

This paper has proposed an approach to support Semantic Search tools in challenges facing them such as matching user terms with Linked Data vocabulary, returning related information with the results and supporting users while building their queries. Following *wisdom of the crowds* and exploiting *collaborative knowledge* found in semantic query logs, the approach attempts to create models of usage of Linked Data concepts and properties. As a proof of concept, we analyzed around 13.5 million DBpedia queries. However, the proposed approach is independent from a specific dataset. Our preliminary results have shown the

potential of adopting the proposed models in an improved semantic search approach. We plan to further evaluate the approach with respect to its performance in matching user terms to Linked Data concepts as well as the quality and relevancy of the returned results as perceived by real users. We think it can additionally be used to create a new vocabulary relying on information needs of Linked Data users and applications and hence customised to best fit their queries. Although the current approach is promising, part of our future work is to investigate the potential benefits available of combining our current models with ones created from traditional query logs as opposed to semantic ones.

References

1. Arias, M., Fernández, J.D., Martínez-Prieto, M.A., de la Fuente, P.: An Empirical Study of Real-World SPARQL Queries. In: Proc. of USEWOD 2011
2. Baeza-Yates, R., Hurtado, C., Mendoza, M. In: Query Recommendation Using Query Logs in Search Engines. LNCS 3268 (2004) 588–596
3. 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: Proc. of SWUI 2006
4. Bernstein, A., Kaufmann, E., Göhring, A., Kiefer, C.: Querying Ontologies: A Controlled English Interface for End-users. In: Proc. of ISWC 2005
5. Bhagdev, R., Chapman, S., Ciravegna, F., Lanfranchi, V., Petrelli, D.: Hybrid Search: Effectively Combining Keywords and Ontology-based Searches. In: Proc. of ESWC 2008
6. Clemmer, A., Davies, S.: Smeagol: A specific-to-general semantic web query interface paradigm for novices. In: Proc. of DEXA 2011. LNCS 6860, Springer (2011) 288–302
7. Corby, O., Dieng-Kuntz, R., Faron-Zucker, C., Gandon, F.: Searching the semantic web: Approximate query processing based on ontologies. IEEE Intelligent Systems **21**(1) (2006) 20–27
8. D'Aquin, M., Baldassarre, C., Gridinoc, L., Angeletou, S., Sabou, M., Motta, E.: Characterizing Knowledge on the Semantic Web with Watson. In: EON (2007) 1–10
9. Damljanovic, D., Agatonovic, M., Cunningham, H.: Natural Language Interface to Ontologies: combining syntactic analysis and ontology-based lookup through the user interaction. In: Proc. of ESWC 2010
10. Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V.C., Sachs, J.: Swoogle: A Search and Metadata Engine for the Semantic Web. In: Proc. of CIKM 2004, ACM Press (2004)
11. Elbedweihy, K., Mazumdar, S., Cano, A.E., Wrigley, S.N., Ciravegna, F.: Identifying Information Needs by Modelling Collective Query Patterns. In: Proc. of COLD 2011
12. Ell, B., Vrandečić, D., Simperl, E.: Deriving human-readable labels from SPARQL queries. In: Proc. of I-Semantics 2011
13. Halpin, H.: A Query-Driven Characterization of Linked Data. In: Proc. of LDOW 2009
14. Harth, A.: VisiNav: A system for visual search and navigation on web data. J. Web Sem. **8**(4) (2010) 348–354

15. Hölscher, C., Strube, G.: Web Search Behavior of Internet Experts and Newbies. Computer Networks **33**(1-6) (2000) 337–346
16. Jain, P., Hitzler, P., Sheth, A.P., Verma, K., Yeh, P.Z.: Ontology alignment for linked open data. In: Proc. of ISWC 2010
17. Kaufmann, E., Bernstein, A.: How useful are natural language interfaces to the semantic web for casual end-users? In: Proc. of ISWC/ASWC 2007
18. Kaufmann, E., Bernstein, A., Zumstein, R.: Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs. In: Proc. of ISWC 2006
19. Lopez, V., Motta, E., Uren, V.: PowerAqua: Fishing the Semantic Web. In: The Semantic Web: Research and Applications, Springer (2006) 393–410
20. Luczak-Rösch, M., Bischoff, M.: Statistical Analysis of Web of Data Usage In: Proc. of ISWC 2011
21. Mascardi, V., Locoro, A., Rosso, P.: Automatic ontology matching via upper ontologies: A systematic evaluation. IEEE Trans. on Knowl. and Data Eng. **22** (2010) 609–623
22. Meij, E., Bron, M., Hollink, L., Huurnink, B., de Rijke, M.: Mapping queries to the Linking Open Data cloud: A case study using DBpedia. Web Semantics: Science, Services and Agents on the World Wide Web **9**(4) (2011) 418 – 433
23. Meij, E., Mika, P., Zaragoza, H.: Investigating the Demand Side of Semantic Search through Query Log Analysis. In: Proc. of SemSearch 2009
24. Möller, K., Hausenblas, M., Cyganiak, R., Grimnes, G.A.: Learning from Linked Open Data Usage: Patterns and Metrics. In: Proc. of WebSci 2010
25. Murray, G.C., Teevan, J.: Query log analysis: social and technological challenges. SIGIR Forum **41** (2007) 112–120
26. Parundekar, R., Knoblock, C.A., Ambite, J.L.: Linking and building ontologies of linked data. In: Proc. of ISWC 2010
27. Silverstein, C., Marais, H., Henzinger, M., Moricz, M.: Analysis of a very large web search engine query log. SIGIR Forum **33**(1) (1999) 6–12
28. Tummarello, G., Cyganiak, R., Catasta, M., Danielczyk, S., Delbru, R., Decker, S.: Sig.ma: live views on the web of data. In: Proc. of WWW 2010
29. Tummarello, G., Oren, E., Delbru, R.: Sindice.com: Weaving the Open Linked Data. In: Proc. of ISWC/ASWC 2007
30. Uren, V., Lei, Y., Lopez, V., Liu, H., Motta, E., Giordanino, M.: The usability of semantic search tools: a review. The Knowledge Engineering Review **22**(4) (2007) 361–377
31. Wen, J.R., Nie, J.Y., Zhang, H.J.: Clustering user queries of a search engine. In: Proc. of WWW 2001, New York, NY, USA, ACM Press (2001) 162–168
32. Wrigley, S., Elbedweihy, K., Reinhard, D., Bernstein, A., Ciravegna, F.: Evaluating semantic search tools using the SEALS platform. In: Proc. of IWEST 2010
33. Zaïane, O.R., Strilets, A.: Finding Similar Queries to Satisfy Searches Based on Query Traces. In: Proc. of OOIS 2002