Querying Linked Data Using Semantic Relatedness: A Vocabulary Independent Approach

André Freitas[1], João Gabriel Oliveira[1,2], Seán O'Riain[1], Edward Curry[1], and João Carlos Pereira da Silva[2]

[1] Digital Enterprise Research Institute (DERI)
National University of Ireland, Galway
[2] Computer Science Department
Universidade Federal do Rio de Janeiro

Abstract. Linked Data brings the promise of incorporating a new dimension to the Web where the availability of Web-scale data can determine a paradigmatic transformation of the Web and its applications. However, together with its opportunities, Linked Data brings inherent challenges in the way users and applications consume the available data. Users consuming Linked Data on the Web, or on corporate intranets, should be able to search and query data spread over potentially a large number of heterogeneous, complex and distributed datasets. Ideally, a query mechanism for Linked Data should abstract users from the representation of data. This work focuses on the investigation of a vocabulary independent natural language query mechanism for Linked Data, using an approach based on the combination of entity search, a Wikipedia-based semantic relatedness measure and spreading activation. The combination of these three elements in a query mechanism for Linked Data is a new contribution in the space. Wikipedia-based relatedness measures address existing limitations of existing works which are based on similarity measures/term expansion based on WordNet. Experimental results using the query mechanism to answer 50 natural language queries over DBPedia achieved a mean reciprocal rank of 61.4%, an average precision of 48.7% and average recall of 57.2%, answering 70% of the queries.

Keywords: Natural Language Queries, Linked Data.

1 Introduction

The last few years have seen Linked Data [1] emerge as a de-facto standard for publishing data on the Web, bringing the potential of a paradigmatic change in the scale which users and applications reuse, consume and repurpose data. However, together with its opportunities, Linked Data brings inherent challenges in the way users and applications consume existing Linked Data. Users accessing Linked Data should be able to search and query data spread over a potentially large number of different datasets. The freedom, simplicity and intuitiveness

R. Muñoz et al. (Eds.): NLDB 2011, LNCS 6716, pp. 40–51, 2011.

provided by search engines in the Web of Documents were fundamental in the process of maximizing the value of the information available on the Web, approaching the Web to the casual user.

However the approaches used for searching the Web of Documents cannot be directly applied for searching/querying data. From the perspective of structured/semi-structured data consumption, users are familiar with precise and expressive queries. Linked Data relies on the use of ontologies (also called vocabularies) to represent the semantics and the structure of datasets. In order to query existing data today, users need be aware of the structure and terms used in the data representation. In the Web scenario, where data is spread across multiple and highly heterogeneous datasets, the semantic gap between users and datasets (i.e. the difference between user queries terms and the representation of data) becomes one of the most important issues for Linked Data consumers. At Web scale it is not feasible to become aware of all the vocabularies that the data can be represented in order to formulate a query. From the users' perspective, they should be abstracted away from the data representation. In addition, from the perspective of user interaction, a query mechanism for Linked Data should be simple and intuitive for casual users. The suitability of natural language for search and query tasks was previously investigated in the literature (Kauffman [2]).

This work focuses on the investigation of a fundamental type of query mechanism for Linked Data: the provision of *vocabulary independent and expressive natural language queries for Linked Data*. This type of query fills an important gap in the spectrum of search/query services for the Linked Data Web, allowing users to expressively query the contents of distributed linked datasets without the need for a prior knowledge of the vocabularies behind the datasets.

This paper is structured as follows. Section 2 describes the proposed approach, detailing how the three elements (*entity search*, *spreading activation* and *semantic relatedness*) are used to build the query mechanism. Section 3 covers the evaluation of the approach, followed by section 4 which describes related work in the area. Finally, section 5 provides a conclusion and future work.

2 Query Approach

2.1 Introduction

The central motivation behind this work is to propose a Linked Data query mechanism for casual users providing flexibility (vocabulary independence), expressivity (ability to query complex relations in the data), usability (provided by natural language queries) and ability to query distributed data. In order to address these requirements, this paper proposes the construction of a query mechanism based on the combination of *entity search*, *spreading activation* and a *semantic relatedness measure*. Our contention is that the combination of these elements provides the support for the construction of a natural language and data model independent query mechanism for Linked Data. The approach represents a new contribution in the space of natural language queries over Linked Data. The remainder of this section describes the proposed approach and its main components.

2.2 Description of the Approach

The query mechanism proposed in this work receives as an input a natural language query and outputs a set of triple paths, which are the triples corresponding to answers merged into a connected path.

The query processing approach starts by determining the *key entities* present in the natural language query. Key entities are entities which can be potentially mapped to instances or classes in the Linked Data Web. After detected, key entities are sent to the entity search engine which determines the pivot entities in the Linked Data Web. A pivot entity is an URI which represents an entry point for the spreading activation search in the Linked Data Web (figure 1). The processes of key entity and pivot entity determination are covered in section 2.3.

After the key entities and pivots are determined, the user natural language query is analyzed in the query parsing module. The output of this module is a structure called *partial ordered dependency structure* (PODS), which is a reduced representation of the query targeted towards maximizing the matching probability between the structure of the terms present in the query and the *subject, predicate, object* structure of RDF. The partial ordered dependency structure is generated by applying Stanford dependency parsing [3] over the natural language query and by transforming the generated Stanford dependency structure into a PODS (section 2.4).

Taking as an input the list of URIs of the pivots and the partial ordered dependency structure, the algorithm follows a spreading activation search where nodes in the Linked Data Web are explored by using a measure of semantic relatedness to match the query terms present in the PODS to terms representing Linked Data entities (classes, properties and individuals). Starting from a pivot entity, the node exploration process in the spreading activation search is done by computing the relatedness measure between the query terms and terms corresponding to entities in the Linked Data Web. The semantic relatedness measure combined with a statistical threshold which represents the discrimination of the winning relatedness scores works as a spreading activation function which will determine the nodes which will be explored in the Linked Data Web.

Figure 1 depicts the core part of the spreading activation process for the example query *'From which university did the wife of Barack Obama graduate?'*. After parsing the natural language query into a partial ordered dependency structure (PODS) (light gray nodes), and after the pivot is determined (*dbpedia: Barack_Obama*) by the entity search, the algorithm follows computing the semantic relatedness between the next query term (*'wife'*) and all the properties, associated types and instance labels linked to the node dbpedia:Barack_Obama (*dbpedia-owl:spouse*, *dbpedia-owl:writer*, *dbpedia-owl:child*, ...). Nodes above a certain relatedness threshold are further explored (dereferenced). The matching process continues until all query terms are covered.

In the example, after the matching between *wife* and *dbpedia-owl: spouse* is defined (2), the object pointed by the matched property (*dbpedia: Michelle_Obama*) is dereferenced (3), and the RDF of the resource is retrieved. The next node in the PODS is *graduate*, which is mapped to both *dbpedia-owl:University* and *dbpedia-owl:Educational_Institution* (4) specified in the types. The algorithm then navigates to the last node of the PODS, *university*, dereferencing *dbpedia:Princeton_University* and *dbpedia:Harvard_Law_School* (5), matching for the second time with their type (6). Since the relatedness between the terms is high, the terms are matched and the algorithm stops, returning the subgraph containing the triples which maximize the relatedness between the query terms and the vocabulary terms. The proposed algorithm works as a best-effort query approach, where the semantic relatedness measure provides a semantic ranking of returned triples.

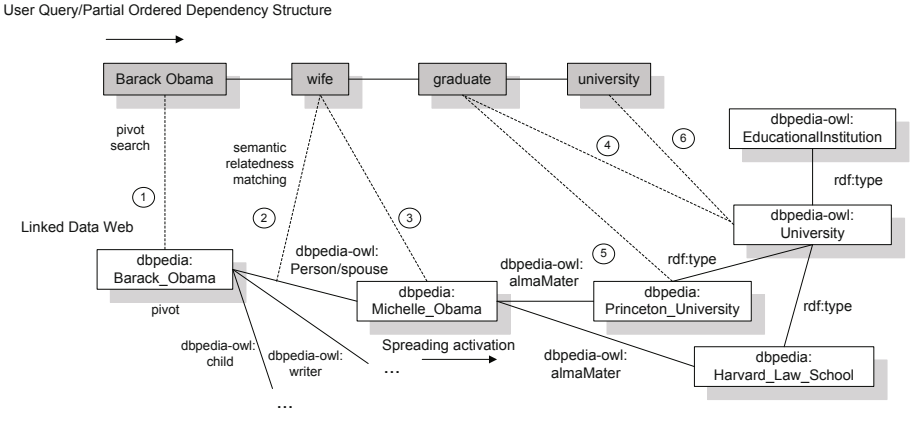

Fig. 1. The relatedness spreading activation for the question *'From which university did the wife of Barack Obama graduate?'*

The output of the algorithm is a list of ranked triple paths, triples following from the pivot entity to the final resource representing the answer, ranked by the average of the relatedness scores in the path. Answers are displayed to users using a list of triple paths and a graph which is built by merging the triple paths on a simple post-processing phase. This graph can be used by the user for further complementary exploration by navigation over the answer set. The query mechanism described above was implemented in a prototype named *Treo*, the word for *direction* in Irish, representing the direction that the algorithm takes in the node exploration process using semantic relatedness.

2.3 Entity Recognition and Entity Search

The query processing approach starts by determining the set of key entities (pivot candidates) that will be used in the generation of the partial ordered dependency

structure and in the determination of the final pivot entity. The process of generating a pivot candidate starts by detecting named entities in the query. The named entity recognition (NER) approach used is based on Conditional Random Fields sequence models [4] trained in the CoNLL 2003 English training dataset [5], covering people, organizations and locations. Named entities are likely to be mapped to the URIs of individuals in the Linked Data Web. After the named entities are identified, the query is tagged by a part-of-speech (POS) tagger, which assigns grammatical tags to the query terms. The POS tags are used to determine pivot candidates which are not named entities (typically classes or individuals representing categories). The POS tagger used is a log-linear POS tagger [6].

The terms corresponding to the pivot candidates are sent to an entity centric-search engine that will resolve the terms into the final pivot URIs in the Linked Data Web. Entity-centric search engines for Linked Data are search engines where the search is targeted towards the retrieval of individual instances, classes and properties in the Linked Data Web. This work uses the entity search approach proposed by Delbru et al. [7], implemented in SIREn, the Semantic Information Retrieval Engine. The approach used in SIREn uses a variation of the TF-IDF weighting scheme (term frequency - inverse subject frequency: TF-ISF) to evaluate individuals by aggregating the values of partial scores for predicates and objects. The TF-ISF scheme gives a low weight to predicates or objects which occur in a large number of entities. SIREn's keyword based search is used for resolving entities in the Linked Data Web, where the recognized entities (pivot candidates) are sent to the search engine and a list of URIs is returned. The query mechanism prioritizes named entities as pivots. In case the query has more than one named entity, both candidate terms are sent to the entity search engine and the number of properties connected to the URI is used to determine the final pivot entity.

2.4 Query Parsing

The relatedness spreading activation algorithm takes as one of its inputs a partial ordered dependency structure (PODS) which is a directed acyclic graph connecting a subset of the original terms present in the query. The idea behind PODSs is to provide a representation of the natural language input which could be easily mapped to the *(subject, predicate, object)* structure of an RDF representation. Partial ordered dependency structures are derived from Stanford typed dependencies [3] which represents a set of bilexical relations between each term of a sentence, providing grammatical relation labels over the dependencies. Additional details covering Stanford dependencies can be found in [3].

The query parsing module builds PODSs by taking as inputs both Stanford dependencies and the detected named entities/pivots and by applying a set of operations over the original Stanford dependencies. These operations produce a reduced and ordered version of the original elements of the query. The pivots and named entities combined with the original dependency structure determine the ordering of the elements in the structure.

Definition I: Let T(V, E) be a typed Stanford dependency structure over the question Q. The partial ordered dependency structure D(V, E) of Q is defined by applying the following operations over T:

1. *merge* adjacent nodes V_K and $V_{K+1} \in T$ where $E_{K,K+1} \in$ {nn, advmod, amod}.

2. *eliminate* the set of nodes V_K and edges $E_K \in T$ where $E_K \in$ {advcl, aux, auxpass, ccomp, complm, det}.

3. *replicate* the triples where $E_K \in$ {cc, conj, preconj}.

In the definition above the *merge* operation consists in collapsing adjacent nodes into a single node for the purpose of merging multi-word expressions, in complement with the NER output. The *eliminate* operation is defined by the pruning of a node-edge pair and eliminates concepts which are not semantically relevant or covered in the representation of data in RDF. The *replicate* operation consists in copying the remaining elements in the PODS for each coordination or conjunctive construction.

The transversal sequence should maximize the likelihood of the isomorphism between the partial ordered dependency structure and the subgraph of the Linked Data Web. The transversal sequence is defined by taking the pivot entity as the root of the partial ordered dependency structure and following the dependencies until the end of the structure is reached. In case there is a cycle involving one pivot entity and a secondary named entity, the path with the largest length is considered.

2.5 Semantic Relatedness

The problem of measuring the semantic relatedness and similarity of two concepts can be stated as follows: given two concepts A and B, determine a measure f(A,B) which expresses the semantic proximity between concepts A and B. The notion of semantic *similarity* is associated with taxonomic (is-a) relations between concepts, while semantic *relatedness* represents more general classes of relations. Since the problem of matching natural language terms to concepts present in Linked Data vocabularies can cross taxonomic boundaries, the generic concept of semantic relatedness is more suitable to the task of semantic matching for queries over the Linked Data Web. In the example query, the relation between *'graduate'* and *'University'* is non-taxonomic and a purely similarity analysis would not detect appropriately the semantic proximity between these two terms. In the context of semantic query by spreading activation, it is necessary to use a relatedness measure that: (i) can cope with terms crossing part-of-speech boundaries (e.g. verbs and nouns); (ii) measure relatedness among multi-word expressions; (iii) are based on a comprehensive knowledge base.

Distributional relatedness measures [11] meet the above requirements but demand the processing of large Web corpora. New approaches propose a better balance between the cost associated in the construction of the relatedness measure and the accuracy provided, by using the link structure of Wikipedia. Wikipedia Link-based Measure (WLM), proposed by Milne & Witten [12], achieved high correlation measurements with human assessments. This work uses WLM as the

relatedness measure for the spreading activation process. The reader is directed to [12] for further details on the construction of the relatedness measure.

2.6 The Semantic Relatedness Spreading Activation Algorithm

The semantic relatedness spreading activation algorithm takes as an input a partial ordered dependency structure D(V, E) and searches for paths in the Linked Data graph W(V, E) which maximizes the semantic relatedness between D and W taking into account the ordering of both structures. The first element in the partial ordered dependency structure is the pivot entity which defines the first node to be dereferenced in the graph W. After the pivot element is dereferenced the algorithm computes the semantic relatedness measure between the next term in the PODS and the *properties*, *type terms* and *instance terms* in the Linked Data Web. Type terms represent the types associated to an instance through the *rdfs:type* relation. While properties and ranges are defined in the terminological level, type terms require an instance dereferenciation to collect the associated types. The relatedness computation process between the next query term k and a neighboring node n takes the maximum of the relatedness score between properties p, types c and instance terms i:

$$r_{k,n} = max(r(k,p), r(k,i), \underset{\forall c \in C}{max}(r(k,c))) \tag{1}$$

Nodes above a relatedness score threshold determine the node URIs which will be activated (dereferenced). The activation function is given by an adaptive discriminative relatedness threshold which is defined based on the set of relatedness scores. The adaptive threshold has the objective of selecting the relatedness scores with higher discrimination and it is defined as a function of the standard deviation σ of the relatedness scores. The activation threshold of a node I is defined as:

$$a(I) = \mu(r) + \alpha \times \sigma(r) \tag{2}$$

where I is the node instance, $\mu(r)$ is the mean of the relatedness values for each node instance, $\sigma(r)$ is the standard deviation of the relatedness values and α is a empirically determined constant. The value of α was determined by calculating the difference in terms of $\sigma(r)$ of the relatedness value of the correct node instances and the average relatedness value for a random 50% sample of the nodes instances involved in the spreading activation process in the query dataset. The empirical value found for α is 2.185. In case no node is activated for the first value of α, the original value decays by an exponential factor of 0.9 until it finds a candidate node above *a(I)*.

In case the algorithm finds a node with high relatedness which has a literal value as an object (non dereferenceable), the value of the node can be re-submitted to entity search engine. In the case an URI is mapped, the search continues from the same point in the partial ordered dependency structure in a different pivot in the Linked Data Web (working as an entity reconciliation step).

From a practical perspective the use of type verification in the node exploration process can bring high latencies in the node exploration process. In order to be effective, the algorithm should rely on mechanisms to reduce the number unnecessary HTTP requests associated with the dereferenciation process, unecessary URI parsing or label checking and unnecessary relatedness computation. The *Treo* prototype has three local caches implemented: one for RDF, the second for relatedness pairs and the third for URI/label-term mapping. Another important practical aspect which constitutes one of the strengths of the approach is the fact that it is both highly and easily parallelizable in the process of semantic relatedness computation and on the de-referenciation of URIs.

3 Evaluation

The focus of the evaluation was to determine the quality of the results provided by the query mechanism. With this objective in mind the query mechanism was evaluated by measuring *average precision*, *average recall* and *mean reciprocal rank* for natural language queries using DBPedia [8], a dataset in the Linked Data Web. DBpedia 3.6 (February 2011) contains 3.5 million entities, where 1.67 million are classified in a consistent ontology. The use of DBPedia as a dataset allows the evaluation of the system under a realistic scenario. The set of natural language queries annotated with answers were provided by the training dataset released for the Question Answering for Linked Data (QALD 2011) workshop [9] containing queries over DBPedia 3.6. From the original query set, 5 queries were highly dependent on comparative operations (e.g. *'What is the highest mountain?'*). Since the queries present in the QALD Dataset did not fully explore more challenging cases of query-vocabulary semantic gap matching, the removed queries were substituted with 5 additional queries. The reader can find additional details on the data used in the evaluation and the associated results in [10].

In the scope of this evaluation an answer is a set of ranked triple paths. Different from a SPARQL query, the algorithm is a best effort approach where the relatedness activation function works both as a ranking and a cut-off function and the final result is a merged and collapsed subgraph containing the triple paths. For the determination of *precision* we considered a correct answer a triple path containing the URI of the answer. For the example query used in this article, the triple path containing the answer *Barack Obama* → *spouse* → *Michelle Obama* → *alma mater* → *Princeton University* and *Harvard Law School* is the answer provided by the algorithm instead of just *Princeton University* and *Harvard Law School*. To determine both precision and recall, triple paths strongly supporting semantically answers are also considered. For the query *'Is Natalie Portman an actress?'*, the expected result is the set of nodes which highly supports the answer for this query, including the triples stating that she is an actress and that she acted on different movies (this is used for both precision and recall). The QALD dataset contains aggregate queries which were included in the evaluation. However, since the post-processing phase does not operate over aggregate operators we considered correct answer triples supporting the answer.

Table 1 shows the quality metrics collected for the evaluation for each query. The final approach achieved an *mean reciprocal rank*=**0.614**, *average precision*=**0.487**, *average recall*=**0.57** and % of *answered queries*= **70%**.

Table 1. Query dataset with the associated reciprocal rank, precision and recall

#	query	rr	precision	recall
1	From which university the wife of Barack Obama graduate?	0.25	0.333	0.5
2	Give me all actors starring in Batman Begins.	1	1	1
3	Give me all albums of Metallica.	1	0.611	0.611
4	Give me all European Capitals!	1	1	1
5	Give me all female German chancellors!	0	0	0
6	Give me all films produced by Hal Roach?	1	1	1
7	Give me all films with Tom Cruise.	1	0.865	1
8	Give me all soccer clubs in the Premier League.	1	0.956	1
9	How many 747 were built?	0.5	0.667	1
10	How many films did Leonardo DiCaprio star in?	0.5	0.733	0.956
11	In which films did Julia Roberts as well as Richard Gere play?	0	0	0
12	In which programming language is GIMP written?	0	0	0
13	Is Albert Einstein from Germany?	0.5	0.5	1
14	Is Christian Bale starring in Batman Begins?	0.125	0.071	1
15	Is Einstein a PHD?	1	1	1
16	Is Natalie Portman an actress?	1	0.818	0.273
17	Is there a video game called Battle Chess?	1	1	0.023
18	List all episodes of the first season of the HBO television series The Sopranos!	0.333	0.090	1
19	Name the presidents of Russia.	1	1	0.167
20	Since when is DBpedia online?	1	0.667	1
21	What is the band of Lennon and McCartney?	0	0	0
22	What is the capital of Turkey?	1	1	1
23	What is the official website of Tom Hanks?	1	0.333	1
24	What languages are spoken in Estonia?	1	1	0.875
25	Which actors were born in Germany?	1	0.033	0.017
26	Which American presidents were actors?	1	0.048	1
27	Which birds are there in the United States?	0	0	0
28	Which books did Barack Obama publish?	1	0.5	1
29	Which books were written by Danielle Steel?	1	1	1
30	Which capitals in Europe were host cities of the summer olympic games?	0	0	0
31	Which companies are located in California, USA?	0	0	0
32	Which companies work in the health area as well as in the insurances area?	0	0	0
33	Which country does the Airedale Terrier come from?	1	1	0.25
34	Which genre does the website DBpedia belong to?	1	0.333	1
35	Which music albums contain the song Last Christmas?	0	0	0
36	Which organizations were founded in 1950?	0	0	0
37	Which people have as their given name Jimmy?	0	0	0
38	Which people were born in Heraklion?	1	1	1
39	Which presidents were born in 1945?	0	0	0
40	Which software has been developed by organizations in California?	0	0	0
41	Who created English Wikipedia?	1	1	1
42	Who developed the video game World Warcraft?	1	0.8	1
43	Who has been the 5th president of the United states?	0	0	0
44	Who is called Dana?	0	0	0
45	Who is the wife of Barack Obama?	1	1	1
46	Who owns Aldi?	0.5	1	0.667
47	Who produced films starring Natalie Portman?	1	0.3	0.810
48	Who was the wife of President Lincoln?	1	1	1
49	Who was Tom Hanks married to ?	1	0.214	1
50	Who wrote the book The pillars of the Earth?	1	0.5	0.5

To analyze the results, queries with errors were classified according to 5 different categories, based on the components of the query approach. The first category, *PODS error*, contains errors which were determined by a difference between the structure of the PODS and the data representation which led the algorithm to an incorrect search path (Q35). In this case, the flexibility provided by semantic relatedness and spreading activation was unable to cope with this difference. The second error category, *Pivot Error*, includes errors in the determination of the correct pivot. This category includes queries with non-dereferenceable pivots (i.e. pivots which are based on literal resources) or errors in the pivot determination process (Q5, Q27, Q30, Q44). Some of the

difficulty in the pivot determination process were related to overloading classes with complex types (e.g. for the query Q30 the associated pivot is a class *yago:HostCitiesOfTheSummerOlympicGames*). *Relatedness Error* include queries which were not addressed due to errors in the relatedness computation process, leading to an incorrect matching and the elimination of the correct answer (Q11, Q12). The fourth category, *Excessive Dereferenciation Timeout Error* covers queries which demanded a large number of dereferenciations to be answered (Q31, Q40). In the query Q40, the algorithm uses the entity *California* as a pivot and follows each associated *Organization* to find its associated type. This is the most challenging category to address, putting in evidence a limitation of the approach. The last categories cover small errors outside previous categories or combined errors in one query (Q32, Q39, Q43).

Table 2. Error types and distribution

Error Type	% of Queries
PODS Error	2%
Pivot Error	10%
Relatedness Error	4%
Excessive Dereferenciation Timeout Error	6%
Combined Error	8%

The approach was able to answer 70% of the queries. The relatedness measure was able to cope with non-taxonomic variations between query and vocabulary terms, showing high average discrimination in the node selection process (average difference between the relatedness value of answer nodes and the relatedness mean is 2.81 $\sigma(r)$). The removal of the queries with errors that are considered addressable in the short term (PODS Error, Pivot Error, Relatedness Error) leads to precision=0.64, recall=0.75 and mrr=0.81.

From the perspective of *query execution time* an experiment was run using an Intel Centrino 2 computer with 4 GB RAM. No parallelization or indexing mechanism outside the pivot determination process was implemented in the query mechanism. The average query execution time for the set of queries which were answered was 635s with no caching and 203s with active caches.

4 Related Work

Different natural language query approaches for Semantic Web/Linked Data datasets have been proposed in the literature. Most of the existing query approaches for semantic approximations are based on WordNet. PowerAqua [13] is a question answering system focused on natural language questions over Semantic Web/Linked Data datasets. PowerAqua uses PowerMap to match query terms to vocabulary terms. According to Lopez et al. [17], *PowerMap is a hybrid matching algorithm comprising terminological and structural schema matching*

techniques with the assistance of large scale ontological or lexical resources. PowerMap uses WordNet based similarity approaches as a semantic approximation strategy. NLP-Reduce [15] approaches the problem from the perspective of a lightweight natural language approach, where the natural language input query is not analyzed at the syntax level. The matching process between the query terms and the ontology terms present in NLP-Reduce is based on a WordNet expansion of synonymic terms in the ontology and on matching at the morphological level. The matching process of another approach, Querix [16], is also based on the expansion of synonyms based on WordNet. Querix, however, uses syntax level analysis over the input natural language query, using this additional structure information to build the corresponding query skeleton of the query. Ginseng [14] follows a controlled vocabulary approach: the terms and the structure of the ontologies generate the lexicon and the grammar for the allowed queries in the system. Ginseng ontologies can be manually enriched with synonyms.

Compared to existing approaches, *Treo* provides a query mechanism which explores a more robust semantic approximation technique which can cope with the variability of the query-vocabulary matching on the heterogeneous environment of Linked Data on the Web. Additionally, its design supports querying dynamic and distributed Linked Data. The proposed approach also follows a different query strategy, by following sequences of dereferenciations and avoiding the construction of a SPARQL query and by focusing on a best-effort ranking approach.

5 Conclusion and Future Work

This paper proposes a vocabulary independent natural language query mechanism for Linked Data focusing on addressing the trade-off between expressivity and usability for queries over Linked Data. To address this problem, a novel combination for querying Linked Data is proposed, based on entity search, spreading activation and Wikipedia-based semantic relatedness. The approach was implemented in the *Treo* prototype and evaluated with an extended version of the QALD query dataset containing 50 natural language queries over the DBPedia dataset, achieving an overall *mean reciprocal rank* of 0.614, *average precision* of 0.487 and *average recall* of 0.572, answering 70% of the queries. Additionally, a set of short-term addressable limitations of the approach were identified. The result shows the robustness of the proposed query mechanism to provide a vocabulary independent natural language query mechanism for Linked Data. The proposed approach is designed for querying live distributed Linked Data. Directions for future investigations include addressing the set of limitations identified during the experiments, the incorporation of a more sophisticated post-processing mechanism and the investigation of performance optimizations for the approach.

Acknowledgments. The work presented in this paper has been funded by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2).

References

1. Berners-Lee, T.: Linked Data Design Issues (2009), http://www.w3.org/DesignIssues/LinkedData.html
2. Kaufmann, E., Bernstein, A.: Evaluating the usability of natural language query languages and interfaces to Semantic Web knowledge bases. J. Web Semantics: Science, Services and Agents on the World Wide Web 8, 393–377 (2010)
3. Marneffe, M., MacCartney, B., Manning, C.D.: Generating Typed Dependency Parses from Phrase Structure Parses. In: LREC 2006 (2006)
4. Finkel, J.R., Grenager, T., Manning, C.D.: Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363–370 (2005)
5. Sang, F., Meulder, F.: Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL (2003)
6. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In: Proceedings of HLT-NAACL 2003, pp. 252–259 (2003)
7. Delbru, R., Toupikov, N., Catasta, M., Tummarello, G., Decker, S.: A Node Indexing Scheme for Web Entity Retrieval. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010. LNCS, vol. 6089, pp. 240–256. Springer, Heidelberg (2010)
8. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - A crystallization point for the Web of Data. J. Web Semantics: Science, Services and Agents on the World Wide Web (2009)
9. Hellmann, S.: 1st Workshop on Question Answering over Linked Data, QALD-1 (2011), http://www.sc.cit-ec.uni-bielefeld.de/qald-1
10. Evaluation Dataset (2011), http://treo.deri.ie/results/nldb2011.htm
11. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In: Proceedings of the International Joint Conference On Artificial Intelligence (2007)
12. Milne, D., Witten, I.H.: An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In: Proceedings of the First AAAI Workshop on Wikipedia and Artificial Intelligence (WIKIAI 2008), Chicago, I.L (2008)
13. Lopez, V., Motta, E., Uren, V.S.: PowerAqua: Fishing the semantic web. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 393–410. Springer, Heidelberg (2006)
14. Bernstein, A., Kaufmann, E., Kaiser, C., Kiefer, C.: Ginseng A Guided Input Natural Language Search Engine for Querying Ontologies. In: Jena User Conference (2006)
15. Kaufmann, E., Bernstein, A., Fischer, L.: NLP-Reduce: A naive but Domain-independent Natural Language Interface for Querying Ontologies. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 1–2. Springer, Heidelberg (2007)
16. Kaufmann, E., Bernstein, A., Zumstein, R.: Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 980–981. Springer, Heidelberg (2006)
17. Lopez, V., Sabou, M., Motta, E.: PowerMap: Mapping the real semantic web on the fly. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 414–427. Springer, Heidelberg (2006)