# Treo: Combining Entity-Search, Spreading Activation and Semantic Relatedness for Querying Linked Data

André Freitas[1], João Gabriel Oliveira[1,2], Edward Curry[1], Seán O'Riain[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.** This paper describes *Treo*, a natural language query mechanism for Linked Data which focuses on the provision of a precise and scalable semantic matching approach between natural language queries and distributed heterogeneous Linked Datasets. Treo's semantic matching approach combines three key elements: *entity search*, a *Wikipedia-based semantic relatedness measure* and *spreading activation search*. While entity search allows Treo to cope with queries over high volume and distributed data, the combination of entity search and spreading activation search using a Wikipedia-based semantic relatedness measure provides a flexible approach for handling the semantic match between natural language queries and Linked Data. Experimental results using the DBPedia QALD training query set showed that this combination represents a promising line of investigation, achieving a *mean reciprocal rank* of 0.489, *precision* of 0.395 and *recall* of 0.451.

**Keywords:** Natural Language Queries, Linked Data

## 1 Introduction

The problem of providing query and search capabilities for casual Linked Data consumers [1] poses new challenges for the areas of information retrieval and databases. Users querying Linked Data on the Web expect to query relationships represented in the data model behind datasets. Structured query languages such as SPARQL provide the capability of explicitly and unambiguously specifying these relationship constraints, at the cost of demanding from users an a priori understanding of the data representation. The scale and decentralized nature of the Web, however, represents a concrete barrier for this paradigm: it is not feasible for end users to become aware of all the vocabularies and possible representations of the data in order to query Linked Data. While this constraint can be manageable for developers building applications on the top of a limited number of datasets or vocabularies, it strongly limits the visibility and consequent utility of the data for casual users. At the heart of this problem lies the lack of

flexibility of query mechanisms to cope with lexical and structural differences between user queries and the data representation, which ultimately creates a semantic gap between users and datasets. In addition, querying Linked Data can usually imply coping with distributed, high-volume and dynamic data, bringing additional challenges for the construction of effective query mechanisms for Linked Data.

This work focuses on the description of *Treo*, a natural language query mechanism for Linked Data which focuses on the provision of a best-effort semantic matching approach, balancing precision and flexibility, while satisfying the ability to perform queries over distributed, large and dynamic data. With these objectives in mind a query mechanism based on a combination of *entity search*, *spreading activation* and a *Wikipedia-based semantic relatedness measure* is proposed. The final query processing approach provides an opportunity to revisit cognitive inspired spreading activation models over semantic networks [16] under contemporary lenses. The recent availability of Linked Data, large Web corpora, hardware resources and a better understanding of the fundamental principles behind information retrieval can provide the necessary resources to enable practical applications over cognitive inspired architectures.

This paper is structured as follows: section 2 outlines the motivation and the dynamics of the proposed approach. Section 3 describes the *entity recognition and entity search approach*, section 4 describes the *query parsing* strategy followed by the description of the *Wikipedia-based semantic relatedness measures* (section 5) and by the description of the *semantic relatedness spreading activation* approach. Each section details how each component part is used to build the final query approach. Section 7 provides a discussion and preliminary results using the QALD DBPedia training dataset, followed by related work on section 8. Finally, section 9 provides a conclusion and future work.

## 2 Outline of the Query Processing Approach

The query approach described in this work focuses on the construction of a query mechanism targeted towards bridging the semantic gap between user queries and Linked Data vocabularies and datasets, with the objective of enabling natural language queries over Linked Data for casual users.

The strategy employed in the construction of the *Treo* query mechanism is to provide a best-effort approach for natural language queries over Linked Data. Having this objective in focus, an answer format and an user interaction approach that could better match a best-effort natural language scenario was developed, where a set of triple paths (i.e. a set of ranked connected triples which are subgraphs in the Linked Data Web) supporting a query answer is returned as a result set. The exposure of a partial set of the data where users could cognitively assess the suitability of the results and search iteratively is a strategy widely used in the construction of document search engines and a variation of this strategy is used in this work in the context of natural language queries over Linked Data.

The query processing starts with the determination of 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 detection, key entities are sent to the *entity search engine* which resolves *pivot entities* in the Linked Data Web. A pivot entity is a URI which represents an entry point for the spreading activation search in the Linked Data Web. After the entities present in the user natural language query are determined, the query is analyzed in the *query parsing module*. The output of this module is a *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.

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 using a *semantic relatedness measure* as an activation function to match the query terms present in the PODS (user query representation) to dataset terms (classes, properties and instances). Starting from the pivot node, the algorithm navigates through neighboring nodes in the Linked Data Web computing the semantic relatedness between query terms and vocabulary terms in the node exploration process. The query answer is built through the node navigation process. The algorithm returns a set of ranked triple paths determined by the navigation from the pivot entity to the final resource representing the answer, ranked by the average of the relatedness scores over each triple path. Answers are displayed to users using a list of triple paths merged in a graph after a simple post-processing phase. Figure 1 depicts the spreading activation process for the example query *'From which university did the wife of Barack Obama graduate?'* over DBPedia.

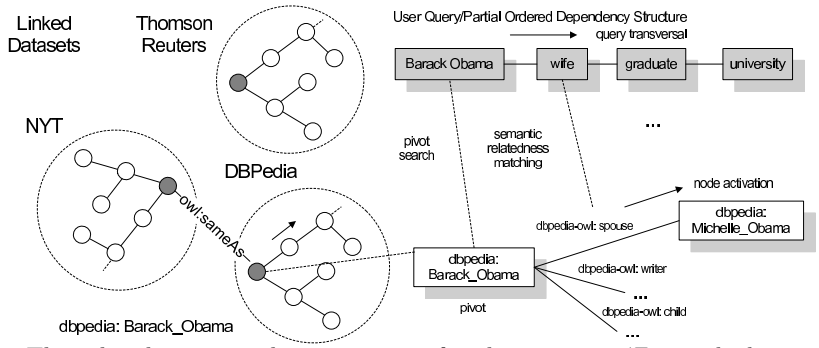


Fig. 1: The relatedness spreading activation for the question *'From which university did the wife of Barack Obama graduate?'*.

The squared gray nodes depict the PODS, a (graph pattern-like) representation of the natural language query while the squared white nodes represent nodes in the Linked Data Web. In the example, the query entity Barack Obama

is mapped to a pivot entity in the Linked Data Web, which is the starting point of the spreading activation process. The following sections detail each step in the query processing approach. The query above is used as a motivational scenario to introduce each step.

## 3 Entity Recognition and Entity Search

The query processing starts with the determination of the set of key entities 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 [3] trained in the CoNLL 2003 English training dataset [17], covering people, organizations and locations. After the named entities are identified, the query is tagged by a part-of-speech (POS) tagger, which assigns grammatical tags to query terms. Rules based on the POS tags are used to determine pivot candidates which are not named entities (classes or individuals representing categories). The POS tagger used is a log-linear POS tagger based on [4]. In the case of the example query, the named entity *Barack Obama* is recognized as the main entity.

The terms corresponding to the pivot candidates are sent to an entity search engine which 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 instances, classes and properties in the Linked Data Web, instead of approaching the problem of treating an entire RDF document as a single resource or addressing complex structured queries over Linked Data. This work uses the entity search approach proposed by Delbru et al. [5], implemented in Siren, the Semantic Information Retrieval Engine. The approach proposed by Delbru, combines query-dependent and query-independent ranking techniques to compute the final entity scores. The query-dependent score function uses a variation of the TF-IDF weighting scheme (term frequency - inverse subject frequency: TF-ISF) to evaluate entities 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. This work only uses the query dependent approach.

The detected entities are sent as keywords to the entity search engine. In this process, if named entities are available they are prioritized as pivots candidates. In case the query has more than one named entity, both candidate terms are sent to the entity search engine and the entities with a larger number of properties are prioritized in the determination of the final pivot entity. In the example query, the named entity *Barack Obama* is mapped to a list of URIs representing the entity Barack Obama in different datasets (e.g. *http://dbpedia.org/resource/Barack_Obama*).

## 4 Query Parsing

After the key entities and pivots are determined, they are sent together with the natural language query to the query parsing module. The center of the query parsing strategy is to maximize the structural similarity between the query and the RDF-based representation of the data. The query parsing is based on the use of Stanford dependencies [2]. The fundamental notion behind dependency parsing is the idea that the syntactic structure of a sentence is determined by a set of directional bilexical relations and by the lack of phrasal nodes. Since the triple structure (subject, predicate, object) of the RDF representation minimizes the use of certain gramatical elements which are present in the free text query, the Stanford dependencies are reduced into a *partial ordered dependency structure* (PODS) by the application of a set of query transformation operations. The PODS also includes the introduction of an ordering which is defined by the relative position of the pivot and detected entities in the query. The final PODS is a directed acyclic graph connecting a subset of the original terms present in the natural language query.

After the Stanford dependencies are determined, four operations are applied to build the PODS: *merge*, *eliminate*, *join condition* and *ordering*. The *merge* operation consists in merging dependencies which are likely to form multi-words or compound expressions (e.g. Barack Obama). The *eliminate* operation remove dependencies which are less semantically significant, unlikely to be expressed in the RDF representation such as determiners, prepositions, etc. The *join condition* joins dependencies which represents conjunctive or disjunctive statements so that star-shaped graph patterns can be introduced in the final query, and finally, the *ordering* operation introduces the ordering based on the position of pivot and entities that is used in the final spreading activation algorithm. The final ordering 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.

For the example query the partial ordered dependency structure returned by the query parser is: Barack Obama → wife → graduate → university.

## 5 Semantic Relatedness

After the natural language query is parsed into the PODS and having the list of pivots determined, the spreading activation search process in the Linked Data Web starts. The center of the spreading activation search proposed in this work is the use of a semantic relatedness measure as the activation function, matching query terms to vocabulary terms. The proposed approach is highly dependent on the quality of the semantic relatedness measure. This section briefly describes the basic concepts behind semantic relatedness and the measure used in the algorithm.

Generally speaking the problem of measuring the semantic *relatedness* and *similarity* of two concepts is associated with the determination of a measure

f(A,B) which expresses the semantic proximity between these concepts. While the idea of semantic *similarity* is associated with taxonomic relations between concepts, 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 both taxonomic and part-of-speech 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 query-dataset semantic matching by spreading activation, it is necessary to use a relatedness measure that: (i) can cope with terms from different part-of-speech (e.g. verbs and nouns); (ii) measure relatedness among multi-word expressions; (iii) are based on comprehensive knowledge bases.

Existing approaches for querying Semantic Web/Linked Data knowledge bases are mostly based on WordNet similarity measures (see Related Work section). WordNet-based similarity measures [26] are highly dependent on the structure and scope of the WordNet model, not addressing the requirements above. Distributional relatedness measures [8][9][26] are able meet the previous requirements, providing approaches to build semantic relatedness measures based on large Web corpora. Recent 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 present in the corpora. One example of this class of measures is the Wikipedia Link-based Measure (WLM), proposed by Milne & Witten [10], which achieved high correlation measurements with human assessments. This work will use WLM as the relatedness measure for the spreading activation process.

The WLM measure is built based on the links between Wikipedia articles. The process of creation of the WLM relatedness measure starts by computing weights for each link, where the significance of each link receives a score. This procedure is equivalent to the computation of the TF-IDF weighting scheme for the links in the place of terms: links pointing to popular target articles (receiving links from many other articles) are considered less significant from the perspective of relatedness computation. The weighting expression is defined below:

$$w(s \rightarrow t) = log\left(\frac{|\ W\ |}{|\ T\ |}\right), \text{if s} \in T, 0 \text{ otherwise} \tag{1}$$

where $s$ and $t$ represent the source and target articles, $W$ is the total number of articles in Wikipedia and $T$ is the number of articles that link to $t$. The relatedness measure is defined by an adaptation over the Normalized Google Distance (NGD)[10][25].

$$r(a,b) = \frac{log(max(|\ A\ |, |\ B\ |)) - log(|\ A \cap B\ |)}{log(|\ W\ |) - log(min(|\ A\ |, |\ B\ |))} \tag{2}$$

where $a$ and $b$ are the two terms that the relatedness is being measured, $A$ and $B$ are the respective articles that are linked to $a$ and $b$ and $W$ is the set of all Wikipedia articles. The final relatedness measure uses a combination of the two measures. The reader is directed to [10] for additional details on the construction of the relatedness measure.

## 6 The Semantic Relatedness Spreading Activation Approach

Spreading activation is a search technique used in graphs based on the idea of using an activation function as a threshold for the node exploration process. Spreading activation has its origins associated with modeling the human semantic memory in cognitive psychology and have a history of applications in cognitive psychology, artificial intelligence and, more recently, on information retrieval [18]. The spreading activation theory of human semantic processing was first proposed by Quillian [20] and it was later extended by Collins & Loftus [19]. The spreading activation model introduced by Quillian [20] was proposed in the context of semantic networks, a graph of interlinked concepts which contained the basic elements formalized today under the scope RDF and RDFS. The recent emergence of the Linked Data Web is likely to motivate new investigations in spreading activation techniques.

The processing technique behind spreading activation is simple, consisting of one or more propagated pulses and a termination check. In addition, the model can implement propagation decays and constraints on the spreading activation process. The semantic relatedness spreading activation algorithm proposed in this work takes as an input a partial ordered dependency structure D($V$, $E$) and searches for paths in the Linked Data Web graph W($V$, $E$) maximizing the semantic relatedness between D and W taking into account the ordering of both structures, where both structures are defined by the set of vertices $V$ and edges $E$. In the approach used in this work, the propagation is defined by the computation of the relatedness measure between the terms present in the query (PODS) and the dataset terms, while the termination check is given by the PODS size.

The spreading activation works as follows. After the URI of the pivot element is dereferenced (the associated RDF descriptor for the URI is fetched), 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. Nodes above a relatedness score threshold (activation function) determine the node URIs which are explored in the search process. The activation function is given by an *adaptive discriminative relatedness threshold* which is defined based on the set of relatedness scores associated with a specific node (the adaptive threshold is defined for each explored node). The threshold selects

the relatedness scores with higher discrimination and it is defined as a linear function of the standard deviation $\sigma$ of the relatedness scores. The linear constant $\alpha$ associated with $\sigma$ is determined empirically and it represents the average discrimination in terms of $\sigma$ for the relatedness measure employed ($\alpha = 2.185$). The original value of $\alpha$ decays by an exponential factor of 0.9 until it finds a candidate node which is above the activation threshold. The final semantic relatedness spreading activation algorithm is defined below:

$D(V_G, E_G)$ : partial ordered dependency graph
$W(V_W, E_W)$ : LD graph
$A(V_A, E_A)$ : answer graph
$pivots$ : set of pivots URI's
**for all** $p$ in $pivots$ **do**
  $initialize(A, p)$
  **while** $hasUnvisitedNodes(V_A)$ **do**
    $v \leftarrow nextNode(V_A)$
    $dereference(v, W)$
    **for all** $nextT$ in $getNextDependencyNodes(D)$ **do**
      $best \leftarrow bestNodes(v, nextT, W)$
      $update(V_A, best)$
      $update(E_A, best)$
    **end for**
  **end while**
**end for**

For the example query *'From which university did the wife of Barack Obama graduate?'*, starting from the pivot node (*dbpedia: Barack_Obama*), 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 the adaptive relatedness threshold are further explored. After the matching between *wife* and *dbpedia-owl: spouse* is defined, the object pointed by the matched property (*dbpedia: Michelle_Obama*) is dereferenced, and the RDF of the resource is retrieved. The next node in the PODS is *graduate*, which is mapped to both *dbpedia-owl:University* an *dbpedia-owl:Educational_Institution* 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*, matching for the second time with their type. 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. The final algorithm returns a ranked list of triple paths (figure 2) which are (when possible) merged into a graph.

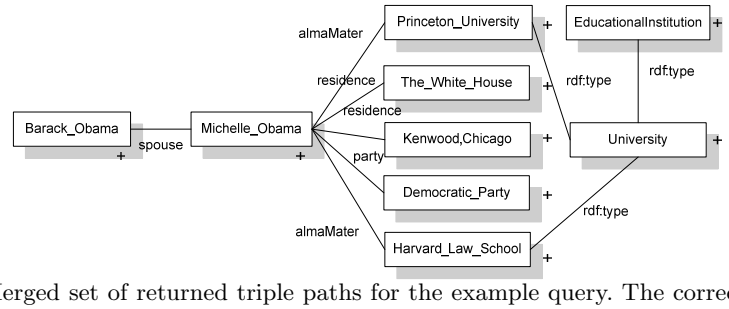


Fig. 2: Merged set of returned triple paths for the example query. The correct answer is given by the nodes Princeton University and Harvard Law School

## 7 Discussion & Preliminary Evaluation

This section has the objective of providing a brief analysis of the strenghts and weakenesses of the proposed approach and to describe the preliminary results achieved over the 1st QALD DBPedia training query set [7].

The proposed query approach consists of a two step process where the core entity in the query is resolved into entities in Linked datasets in the first step, followed by the structural query matching process using spreading activation combined with semantic relatedness. The process of first resolving the core entity plays an strategic role in the scalability and parallelism of the approach. The process of finding pivot entities associated with instances in the datasets, tends to be less ambiguous and it is related to the most informative part of the query (informativeness in this context is defined as the ability to constrain the search space in the Linked Data Web). Entity search demands the creation of an entity index for the datasets. As a strength, entity indexes, differently from structure indexes [22] can benefit from less challenging constraints in terms of index space, time and indexing frequency (related with dataset dynamics). In the proposed approach, entity search is used as the main mechanism for ranking pivots. One of the current limitations of using a pure entity search approach for pivot selection is the fact that it ignores the context provided by the remaining query terms. This ultimately can lead to an increase in query execution time or a decrease in precision. One possible solution for this problem is to extend the existing mechanism with dataset summaries [24], where additional query terms can be used in a summary-based search, composing the information provided in summaries with the entity search process.

The semantic relatedness spreading activation search algorithm traverses the Linked Data Web graph using the PODS query representation. Two important factors for the performance of the approach is the term ordering in the PODS and the overall performance of the semantic relatedness measure. The WLM semantic relatedness measure used in this work showed high discrimination in the node exploration process (average 2.81 $\sigma$ measured in a 50% sample of the query set) and robustness in relation to the computation of the relatedness among terms from different grammatical classes for the DBPedia query set. Another strength of the proposed approach is the fact that it is highly and

easily parallelizable both for spreading activation search and in the computation of semantic relatedness measure. The query mechanism currently uses sequences of URI dereferenciations as the primary way to navigate through Linked Data. From a practical perspective, the HTTP requests 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 computations. The prototype, *Treo*, have implemented three local caches: one for RDF, one for relatedness values and the third for URI/label-term mapping.

The quality of the approach was evaluated against the 50 queries of the QALD DBPedia training query set in terms of *precision*, *recall* and *mean reciprocal rank*. Online DBPedia (version 3.6) [6] was used as the dataset and a local version of Siren, indexing a local copy of DBPedia, was used as the entity search mechanism. In the scope of the system described on this paper, an answer is a set of ranked triple paths. Different from a SPARQL query result set or from typical QA systems, the proposed 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 set of subgraphs containing the answer triple paths. For the determination of *precision* we considered as correct answers triple paths containing the URI of the answer. For the example query used through this article, the graph in figure 2 containing the answer *Barack Obama* $\rightarrow$ *spouse* $\rightarrow$ *Michelle Obama* $\rightarrow$ *alma mater* $\rightarrow$ *Princeton University* and *Harvard Law School* is the answer provided by the algorithm, instead of just the names of the two universities. To determine both precision and recall, triple paths strongly supporting 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 starred in different movies (criteria which is used for both precision and recall). The QALD dataset contains queries with aggregate and conditional operators which were included in the evaluation. However, since Treo does not have a more sophisticated post-processing phase, triples supporting an answer for queries with operators were considered as correct answers. Table 1 contains the quality of results in terms of precision, recall and mrr. Three variations of query sets were taken into account. The first query set (Full DBPedia) evaluates the query mechanism against the 50 queries present in the query set. The second query set had queries with non-dereferenceable pivots (literals and classes) removed (total 42 queries evaluated), while the third category just considered queries which were answered by the approach (27 queries).

The *average query execution time* was 728s with no caching and 353s with active caches using an Intel Centrino 2 machine with 4 GB RAM.

The experiments revealed two main directions for improvements. The first improvement direction is related to addresssing limitations in the pivot determination process related to the detection of complex-type classes (e.g. pivot classes with more than 2 terms typically from YAGO) and coping with non-

| Query Set Type | MRR | Avg. Precision | Avg. Recall |
|---|---|---|---|
| Full DBPedia Training | 0.489 | 0.395 | 0.451 |
| DBPedia Training (no non-deref. pivots) | 0.661 | 0.534 | 0.609 |
| DBPedia Training (answered queries) | 0.906 | 0.706 | 0.805 |

Table 1: Quality of results for the Treo query mechanism

dereferenceable pivots (i.e. queries having object literals as pivots). The second improvement direction aims towards coping with multiple candidate PODS representations, taking into account metrics of term informativeness (e.g. TF-IDF). 98% of the final set of PODSs contained the correct ordering in relation to the RDF structure. However, the occurrency of less informative terms before the target terms and the associated process of term-by-term PODS traversal created a semantic discontinuity that the relatedness measure was not able to handle. The use of PODS terms informativeness metrics in the generation of alternative query traversal sequences is a future direction for investigation. One important dimension which was not evaluated in the approach was the generality of the Wikipedia-based relatedness measure beyond the scope of generic datasets such as DBPedia. For these cases the construction of domain-specific distributional relatedness measures which are less dependent on the link structure and organization of Wikipedia (e.g. LSA) are likely be more general solutions across different domains.

## 8 Related Work

There is an extensive literature in natural language query systems over unstructured and structured data. The analysis of the related work focuses on natural language query approaches for Semantic Web/Linked Data datasets. PowerAqua [11] is a question answering system focused on natural language questions over Semantic Web/Linked Data datasets using PowerMap, a *hybrid matching algorithm comprising terminological and structural schema matching techniques with the assistance of large scale ontological or lexical resources*. PowerMap [15] uses WordNet based similarity approaches as a semantic approximation strategy. NLP-Reduce [13] approaches the query-data matching 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 [14], is also based on the synonyms expansion 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. In case ambiguities are detected, a disambiguation dialog is returned for user feedback. Ginseng [12] 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. ORAKEL [21] is a natural language interface focusing on the portability problem across different domains. For this purpose, ORAKEL implements a lexicon engineering functionality, which allows the creation of explicit frame mappings. Instead of allowing automatic approximations, ORAKEL focuses on a precise manually engineered model. Comparatively, Treo provides a query mechanism exploring Wikipedia-based semantic relatedness measures as a semantic approximation technique, where the use of semantic relatedness combined with spreading activation can cope with the heterogeneity of the query-vocabulary problem on Linked datasets on the Web. In addition, Treo's design supports the query of dynamic and distributed Linked Data by relying on entity search and sequential dereferenciations. Treo also differentiates itself from existing approaches in the query strategy: (i) instead of building a SPARQL query, Treo navigates the Linked Data nodes from a pivot node and (ii) Treo focuses on a best-effort ranked approach.

## 9 Conclusion & Future Work

This work describes *Treo*, a natural language query mechanism for Linked Data. The cognitively inspired architecture of Treo, combining *entity search*, *spreading activation* and *semantic relatedness* is designed to cope with critical features for natural language queries over Linked Data, including the ability to query high volume, heterogeneous, distributed and dynamic data. The approach was evaluated using the QALD query datasets containing 50 natural language queries over DBPedia, achieving an overall *mean reciprocal rank* of 0.489, *precision* of 0.395 and *recall* of 0.451, answering 56% of the queries. The proposed query mechanism approaches natural language queries over Linked Data using a best-effort approach where results are displayed in the form of triple paths, ranked paths containing the desired entities in the Linked Data Web. Future work will concentrate in two main directions: the improvement of the query execution time of the approach, exploring optimizations through parallelization, indexing and pipelining [23] strategies and the introduction of answer post-processing techniques for the generation of answers from triple paths.

**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. Kaufmann, E., Bernstein, A.: Evaluating the usability of natural language query languages and interfaces to Semantic Web knowledge bases. Web Semantics: Science, Services and Agents on the World Wide Web 8 393-377 (2010).
2. Marneffe, M., MacCartney, B. and Manning, C. D., Generating Typed Dependency Parses from Phrase Structure Parses. In LREC 2006 (2006).

3. Finkel, J.R., Grenager, T. , and Manning, C. D., Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370 (2005).
4. Toutanova, K., Klein, D., Manning, C.D. and Singer, Y., Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252-259 (2003).
5. Delbru, R., Toupikov, N., Catasta, M., Tummarello, G., Decker, S.: A Node Indexing Scheme for Web Entity Retrieval. In Proceedings of the 7th Extended Semantic Web Conference (ESWC) (2010).
6. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S.r., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - A crystallization point for the Web of Data. Web Semantics: Science, Services and Agents on the World Wide Web 7 (2009).
7. 1st Workshop on Question Answering over Linked Data (QALD-1), `http://www.sc.cit-ec.uni-bielefeld.de/qald-1` (2011).
8. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using Wikipedia-based explicit semantic analysis. International Joint Conference On Artificial Intelligence (2007).
9. Deerwester, S., Dumais, S.T., Furnas, G.W.: Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science 41 (6) 391407 (1990).
10. Milne, D. and 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'08), Chicago, I.L. (2008).
11. Lopez, V., Motta, E., Uren, V.: PowerAqua: Fishing the Semantic Web. Proc 3rd European Semantic Web Conference ESWC, Vol. 4011. Springer 393-410 (2004).
12. Bernstein, A., Kaufmann, E., Kaiser, C., Kiefer, C.: Ginseng A Guided Input Natural Language Search Engine for Querying Ontologies. Jena User Conference 2006 (2006).
13. Kaufmann, E., Bernstein, A., Fischer, L.: NLP-Reduce: A naive but Domain-independent Natural Language Interface for Querying Ontologies. 4th European Semantic Web Conference ESWC 2007 1-2 (2007).
14. Kaufmann, E., Bernstein, A., Zumstein, R.: Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs. 5th International Semantic Web Conference (ISWC). Springer 980-981 (2006).
15. Lopez, V., Sabou, M., Motta, E.: PowerMap: Mapping the Real Semantic Web on the Fly. International Semantic Web Conference, Vol. 4273. Springer 5-9 (2006).
16. Cohen, P.: Information retrieval by constrained spreading activation in semantic networks, Information Processing & Management, Vol. 23, no. 4, pp. 255-268 (1987).
17. Conference on Computational Natural Language Learning (CoNLL-2003), http://www.clips.ua.ac.be/conll2003/ (2003).
18. Crestani, F. Application of Spreading Activation Techniques in Information Retrieval, Artificial Intelligence Review, vol. 11, no. 6, pp. 453-453 (1997).
19. Collins, A.M., Loftus, E.F.: A spreading activation theory of semantic processing, Psychological Review, vol. 82, no. 6, pp. 407-428 (1975).
20. Quillian, M.R.: Semantic Memory, Semantic Information Processing, MIT Press, pp. 227-270 (1968).
21. Cimiano, P., Haase, P., Heizmann, J., Mantel, M. and Studer,R.: Towards portable natural language interfaces to knowledge bases: The Case of the ORAKEL system, Data Knowledge Engineering (DKE), 65(2), pp. 325-354 (2008).
22. Dong, X., Halevy, A.: Indexing Dataspaces, In Proceedings of the ACM SIGMOD (2007).

23. Hartig, O., Bizer, C., and Freytag, J.C.: Executing SPARQL Queries over the Web of Linked Data, Proceedings of the 8th International Semantic Web Conference (2009).
24. Harth, A., Hose, K., Karnstedt, M., Polleres, A., Sattler, K.-U. and Umbrich, J.: Data summaries for on-demand queries over linked data, in Proceedings of the 19th international conference on World Wide Web, USA (2010).
25. Cilibrasi, R.L. and Vitanyi, P.M.B.: The Google Similarity Distance, IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 3, 370-383 (2007).
26. Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Pasca, M. and Soroa, A., A study on similarity and relatedness using distributional and WordNet-based approaches, In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 19-27 (2009).