Buikstra, A., Neth, H., Schooler, L., ten Teije, A., & van Harmelen, F. (2011). Ranking query results from Linked Open Data using a simple cognitive heuristic. In Workshop on Discovering Meaning on the Go in Large Heterogeneous Data 2011 (LHD-11). Barcelona, Spain: 22nd International Joint Conference on Artificial Intelligence (IJCAI-11).

Ranking query results from Linked Open Data using a simple cognitive heuristic

Arjon Buikstra*, Hansjörg Neth†, Lael Schooler†, Annette ten Teije*, Frank van Harmelen*
*Dept. of Computer Science, VU University Amsterdam
†Max Planck Institute for Human Development, Berlin

Abstract

We address the problem how to select the correct answers to a query from among the partially incorrect answer sets that result from querying the Web of Data.

Our hypothesis is that cognitively inspired similarity measures can be exploited to filter the correct answers from the full set of answers. These measure are extremely simple and efficient when compared to those proposed in the literature, while still producing good results.

We validate this hypothesis by comparing the performance of our heuristic to human-level performance on a benchmark of queries to Linked Open Data resources. In our experiment, the cognitively inspired similarity heuristic scored within 10% of human performance. This is surprising given the fact that our heuristic is extremely simple and efficient when compared to those proposed in the literature.

A secondary contribution of this work is a freely available benchmark of 47 queries (in both natural language and SPARQL) plus gold standard human answers for each of these and 1896 SPARQL answers that are human-ranked for their quality.

1 Introduction

The Web of Data has grown to tens of billions of statements. Just like the traditional Web, the Web of Data will always be a messy place, containing much correct, but also much incorrect data. Altough there has been surprisingly little structured research on this topic, anecdotal evidence shows that even the highest rated and most central datasets on the Web of Data such as DBPedia and Freebase contain factually incorrect and even nonsensical assertions. Consider the following results from some of the benchmark queries that we will discuss later, when executed against a combination of DBPedia, Geonames and Freebase:

> "AmeriCredit" is not an American car manufacturer (instead, it's a financial company owned by General Motors to help customers finance their cars)
>
> "Richard Bass" is not one of the highest summits on the seven contintents (instead, he was the first mountaineer that climbed all of them)
>
> "Cosima" is not a Nobel Prize for Literature Laureate (instead, it is a novel written by Grazia Deledda, who received the 1926 Nobel Prize for Literature)
>
> "Stig Anderson" was not one of the members of ABBA (instead, he was their manager)

These examples (which are just a few of many) illustrate *the central problem* that we tackle in this paper

> given a query to the Web of Data and the resulting answer set, how to separate the correct from the incorrect answers.

For well over a decade now, influential cognitive scientists have been proposing the notion of *fast and frugal heuristics*: heuristics that are surprisingly simple (sometimes even seemingly naive), but that on closer inspection perform very well on complex cognitive tasks. Their findings have shown convincingly that such simple heuristics are not only justified by gaining computational efficiency at the expense of output quality, but that such simple heuristics can even outperform complex decision rules [Gigerenzer *et al.*, 1999].

The main finding of this paper is that cognitively inspired heuristics can indeed be exploited to filter the correct answers from the noisy answersets obtained when querying the Web of Data. Perhaps the most surprising finding is that such heuristics are extremely simple when compared to those proposed in the literature, while still producing good results.

The overall benefit from this work is that it is now possible to efficiently select the most likely correct answers when querying the Web of Data. Our approach has as additional benefit that our selection heuristic can be tuned to favour either recall or precision.

An important secondary contribution of this work is the construction of a benchmark of general knowledge queries with their Gold Standard answers. Each of these has also been formulated as a SPARQL query, and the 1896 answers to these queries have been manually ranked on their quality. This collection is freely available for other researchers as an important tool in benchmarking their query strategies over the Web of Data.

The paper is structured as follows: In section 2, we first discuss the construction of this benchmark. In section 3, we report on how good a human subject is in recognising the Gold Standard correct answers for these benchmark questions. In section 4 we discuss some of the cognitive science literature that justifies the definition of our "fast and frugal" computational heuristic. In section 5 we then measure the perfor-

mance of this heuristic, and we show that its performance is comparable to that of the human subject. In section 6 we compare our approach to related work in the literature. In the final section 7 we compare our heuristic to those presented in the Semantic Web literature.

2 A benchmark for querying the Web of Data

Over the past decade, the Semantic Web community has built and adopted a set of synthetic benchmarks to test storage, inference and query functionality. Some of the most well known benchmarks are the Lehigh LUBM benchmark, [Guo *et al.*, 2005], the extended eLUBM benchmark [Ma *et al.*, 2006] and the Berlin SPARQL benchmark [Bizer and Schultz, 2009] are all examples of these[1]. However, all these are *synthetic* datasets. There is a shortage of *realistic* benchmarks that provide realistic queries plus validated ("Gold Standard") answers. The sample queries on the webpages of Linked Life Data [2] FactForge[3] are examples of such realistic queries, but they do not come with a validated set of Gold Standard answers.

Set of questions. For an experiment investigating how people search for information in their memory, [Neth *et al.*, 2009] designed a set of general knowledge questions. Each question identifies a natural category by a domain label (e.g.,'Geography') and a verbal description (e.g., 'African countries') and asks participants to enumerate as many exemplars as possible (e.g., 'Algeria', 'Angola', 'Benin', etc.). Questions were drawn from diverse areas of background knowledge (e.g., arts, brands, sciences, sports) and included "Name members of the pop band ABBA", "Name Nobel laureates in literature since 1945", etc.

Gold Standard answers. [Neth *et al.*, 2009] determined a set of correct answers for each question. The number of true exemplars varied widely between categories (from 4 to 64 items). Particular care was given to the completeness of the answer set by including alternative labels (e.g., 'Democratic Republic of the Congo', 'Zaire') and spelling variants ('Kongo')[4].

SPARQL queries. We have developed a set of 47 SPARQL queries, made to resemble the questions from [Neth *et al.*, 2009]. For this translation, we used a number of well-known namespaces, such as DBPedia, Freebase, Geonames, Umbel, etc. As an example, the question about ABBA members translates to the SPARQL query shown in figure 1.

SPARQL answers. To complete this benchmark collection, we executed all of our queries against FactForge[5]. FactForge [Bishop *et al.*, 2010a] is a collection of some of the most central datasources in the Linked Open Data cloud. It hosts 11 datasets, including DBPedia, Freebase, Geonames, UMBEL, WordNet, the CIA World Factbook, MusicBrainz and others. Several schemata used in the datasets are also loaded into FactForge, such as Dublin Core, SKOS and FOAF. FactForge uses the OWLIM reasoner [Bishop *et al.*, 2010b] to materialise all inferences that can be drawn from the datasets and their schemata. This results in some 10 billion retrievable statements, describing just over 400 million entities. Although FactForge is a subset of the entire Web of Data, it is currently one of the the largest available subsets that is both closed under inference and queryable.

```
SELECT DISTINCT ?member ?label
 WHERE {
  ?member skos:subject dbp-cat:ABBA_members
  ?member rdfs:label ?label
 FILTER(lang(?label) = "en")
}
```

dbpedia:Agnetha_Fältskog	Agnetha Fältskogen
dbpedia:Agnetha_Fältskog	Agneta øase Fältskogen
dbpedia:Anni-Frid_Lyngstad	Anni-Frid Lyngstaden
dbpedia:Anni-Frid_Lyngstad	Frida Lyngstaden
dbpedia:Benny_Andersson	Benny Anderssonen
dbpedia:Björn_Ulvaeus	Björn Ulvaeusen
dbpedia:Ola_Brunkert	Ola Brunkerten
dbpedia:Stig_Anderson	Stig Andersonen

Figure 1: Example query and answer-set

Running our 47 queries against FactForge[6] resulted in 1896 answers. An example answer-set is shown in figure 1.

The entire resource (original questions, their SPARQL translations, the Gold Standard answers, as well as query-results against FactForge) are available online[7].

3 Human performance

In order to judge how good our heuristics will be at recognising correct answers, we measured how good a human was at this task. A human subject (educated at university level, using the Web as a reference source, and asked to do this at reasonable speed) ranked all 1896 answers on a 5 point scale, with 5 indicating the answers on which the subject was most confident that they are correct, and 1 indicating answers on which the subject was most confident they were incorrect [8], [9]. We are now interested in the question whether the human subject can recognise correct answers with sufficiently high confidence. For this, we introduce the following notations:

Notation:
We use Q to indicate the query, with Q ranging from #1 to #47 in our benchmark collection.
The set of Gold Standard answers to query Q is written

[1] More benchmarks are described at http://www.w3.org/wiki/RdfStoreBenchmarking.

[2] http://linkedlifedata.com/sparql

[3] http://factforge.net/sparql

[4] The answers to some questions (e.g., the teams in particular leagues) are subject to periodic changes. This renders the current standard (which was constructed in 2009) partially out-dated.

[5] http://factforge.net/

[6] version of August 2010

[7] http://www.larkc.eu/resources/published-data-sources

[8] These human rankings are also available from the aforementioned URL.

[9] Because this was only a single human judge, we cannot measure how reliable the scoring is, since we have no measurement for inter-subject agreement. This would be a useful piece of future work to strengthen the value of our dataset

$G(Q)$.
The set of retrieved answers to query number Q are written $A(Q)$.
The set of answers to query Q that were scored with a confidence ranking of T or higher is written as $A_T(Q)$, $T = 1, .., 5$.

Obviously, $A_1(Q) = A(Q)$ (all answers are included at confidence threshold $T = 1$), and the size of $A_T(Q)$ decreases with increasing T.

In our experiment described below, the size of $G(Q)$ is typically a few dozen items (since this is how the cognitive scientists designed their queries). The size of $A(Q)$ varies greatly from a dozen to several hundreds, showing that some answer sets contain many wrong results, i.e. $A(Q) \not\subseteq G(Q)$, for some Q. We will see below that also $G(Q) \not\subseteq A(Q)$ for some Q, ie FactForge is not complete for all of our queries.

To judge the performance of our human subject in recognising correct answers, we will plot the recall and precision of $A_T(Q)$ as a function of his confidence threshold T, where the correctness of the answers in $A_T(Q)$ is determined against $G(Q)$. The comparison of the SPARQL results in $A_T(Q)$ against the (natural language) elements in $G(Q)$ is done using the `rdfs:label` of the elements in $A_T(Q)$.

Example query. As an illustration, Figure 2(a) shows the performance of our subject on query $Q = \#31$: "What are the highest mountains (peaks) of each continent". At threshold level $T = 5$ (i.e. when aiming to select only the answers about which he is most confident that they are correct), the subject scores a precision of 1.0 but recognises only $N = 4$ out of the 7 summits, ie the recall is only 0.57. When including answers at lower confidence levels, the recall increases, finally reaching 1.0 at $T = 1$. This shows that FactForge does indeed contain all correct set answers for this query, i.e. $G(\#31) \subset A(\#31)$. However, the increase in recall goes at the cost of also including some incorrect answers, with precision dropping to a final 0.5. The maximal performance (using the macro-averaged F-measure to combine precision and recall) is F=0.86, and is reached at confidence threshold $T = 4$.
Accumulated results. Figure 2(b) shows the recall and precision figures of our human subject accumulated over all 47 queries. It shows that even at $T = 1$ the recall is only just above 0.6. This tells us that FactForge is indeed incomplete for our set of queries, and it is simply impossible for any subject (human or machine) to do any better on this set of queries.

Figure 2(b) shows a near perfect performance by our human subject: when increasing his confidence levels T, the precision of $A_T(G)$ increases from 0.25 to 0.75, while paying almost no penalty in decreasing recall (dropping from 0.6 to 0.5). In other words: when stepping up the confidence level from T to $T + 1$, the sets $A_{T+1}(G)$ have lost some of the wrong answers that were still in $A_T(G)$ while maintaining most of the correct answers in $A_T(G)$. Or stated informally: our subject is actually rather good at recognising the correct answers from among $A_T(G)$. In terms of the graph in Figure 2(b), a perfect performer would result in a vertical plot (increasing precision at no loss of recall). The human subject comes close to that perfect plot. Consequently, the highest score (F=0.60) is obtained at confidence threshold $T = 5$.

4 Definition of selection heuristic

Fast and Frugal Heuristics. Biological cognitive agents (be they human or animal) have to perform a similar task on a daily basis: given a set of possible alternatives, which ones are "the best", ranging from distinguishing edible from inedible foods to deciding if another agent is friend or foe. Already in 1969, Herbert Simon [Simon, 1969] noted that this selectivity is based on rules of thumb, or heuristics, which cut problems down to manageable size. The idea is that the world contains an abundance of information and the solution is not necessarily to integrate as much information as possible, but rather to select some information and use that for reasoning.

In the late 20th century there was a debate between decision theorists on whether these rules of thumb had positive or negative consequences for the quality of decisions humans were making. In the heuristics and biases program, examples are shown where people make faulty decisions in situations containing uncertainty, as measured by what is expected of them from the optimal result in probability theory [Tversky and Kahneman, 1974]. The other side of the debate is the fast and frugal heuristics program, where situations are shown where simple rules requiring little information perform as well as (and, surprisingly, in special cases better than) algorithms attempting to reach the optimum [Gigerenzer *et al.*, 1999].

Such *fast and frugal heuristics* which in given situations outperform classical methods while using substantially less information and computational resources, are also relevant outside the area of cognitive science. Whether or not these algorithms acurately describe human (or animal) cognition, we can use these heuristics for complex decision making outside of the mind, in a computational setting.

The relevance to the Semantic Web lies in that these algorithms could improve yield without damaging quality excessively: producing the results we want at lower computational costs, but not guarantying optimal quality or completeness of results. In a term coined by Simon, we wish to satisfice [Simon, 1956].
Similarity as a heuristic. If one thinks about a query as defining a target region in some semantic space, one would expect the results of the query to be clustered in that target region. That is, the results that are most similar to each other are most likely to be close to the center of the targeted region. It seems reasonable to assume the the farther away a result is from the center of this estimated target region, the more likely it is to have been included in the results due to error.

Two classical approaches to formalizing similarity are featural approaches, epitomized in Tversky's contrast model [Tversky, 1977], and spatial models [Shepard, 1957]. In spatial models, similarity is defined as the distance in a defined metric space between two items. Spatial models have predefined spaces and each item is a separate point in the space, making symmetrical similarity natural. Tversky's contrast model focuses on features shared between items and features not shared between items. Similarity is then defined by the

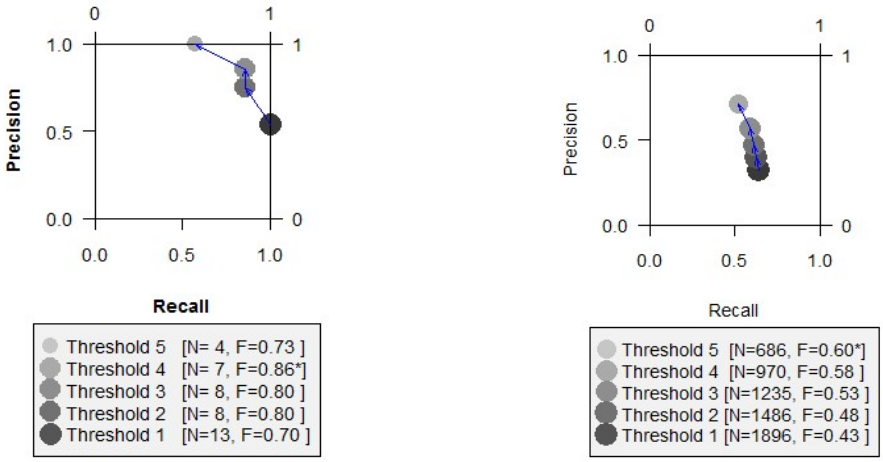

Figure 2: Performance of human subject: (a) on an example query and (b) accumulated results

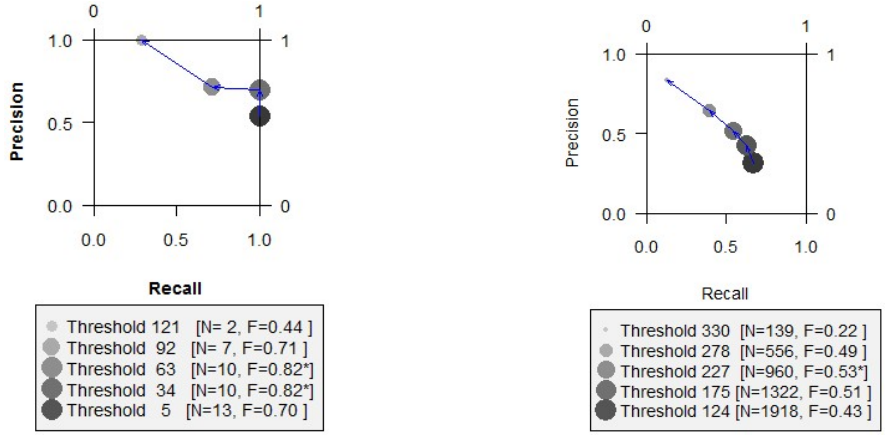

Figure 3: Performance of the similarity heuristic: (a) on an example query and (b) accumulated results

proportion of shared features in the total features of an item

Computational definition of similarity. Tversky's similarity model based on shared features fits very naturally with the datamodel underlying RDF: an "item" is a URI s_1, a "feature" is a triple $\langle s, p, o\rangle$, and two features are shared between two items s_1 and s_2 if they have the form $\langle s_1, p, o\rangle$ and $\langle s_2, p, o\rangle$. For example, two objects share a feature if they both have a `skos:subject` property with object `dbp-cat:ABBA_members`. Formally:

Definition 1 *the similarity $S(s_1, s_2)$ between two resources s_1 and s_2 in a graph G is defined as:*

$$S(s_1, s_2, G) = ||\{(p, o)|\langle s_1, p, o\rangle \in G \text{ and } \langle s_2, p, o\rangle \in G\}||$$

i.e. similarity is defined as the number of feature-value pairs in G that are shared between s_1 and s_2. This looks even simpler as a schematic SPARQL query:

```
SELECT COUNT(?p)
 WHERE {<s1> ?p ?q
        <s2> ?p ?q}
```

where `<s1>` and `<s2>` must be replaced by specific URIs.

This similarity measure can now be used to define a heuristic confidence estimate for query-answers:

Definition 2 *The confidence estimate $C(a, Q, G)$ for an answer $a \in A(Q)$ to a query Q over a graph G is defined as*

$$C(a, Q, G) = \Sigma_{a' \in A(Q)} S(a, a', G)$$

i.e. the confidence estimate of an answer a is simply the aggregate similarity of a to every other answer a'. This similarity heuristic is similar to the "clustering hypothesis" as it is known from Information Retrieval [Tombros and Van Rijsbergen, 2001], namely that relevant documents tend to be more similar to each other than to non-relevant ones, and therefore tend to appear in the same clusters.

Alternatives. Of course a number of variations on this definition would be possible. Instead of counting the total number of shared features $\langle s_1, p, o\rangle$, we could calculate the *fraction* of shared features, as suggested in [Tversky, 1977]. Because of the fairly uniform arity of the nodes in RDF graphs, we would not expect this to make much difference.

We could also have used the weaker definition of only counting shared properties p without demanding that they

have the same values: $\langle s_1, p, \_ \rangle$ and $\langle s_2, p, \_ \rangle$. For example: two objects are similar if they both have a `dbp-prop:manufacturer` property, even if that property has different values. However, due to the weak nature of many of the features (e.g. `rdf:type`, `skos:subject`) we expect that this will generate too high similarity ratings.

More reasonable would be to include shared *inverse* features $\langle o, p, s_1 \rangle$ and $\langle o, p, s_2 \rangle$. This would account for inverse modelling in the RDF graph, for example using `is-manufacturer` instead of `manufactured-by`. Such inverse properties are rare in FactForge, but this would be worth further investigation.

5 Heuristic performance

We are now in position to measure how good the heuristic from Def. 2 is at selecting the correct answers for a query. In order to use the same evaluation procedure as for the human subject in section 3, we divide for every query Q the interval $[min_{a \in A(Q)} C(a, Q, G),\ max_{a \in A(Q)} C(a, Q, G)]$ uniformly in 5 equal steps.

Figure 3(a) shows the performance of our similarity based confidence estimate on the same "seven summits" query as in Figure 2. Trivially, the heuristic performance at the lowest confidence level equals that of the human performance at the lowest confidence level, at a reasonably high F-value of 0.43, achieved with trivially accepting all answers as correct. This is caused by the high quality of FactForge. Just as the human subject, the heuristic achieves a precision of 1.0 at the highest confidence level, but only manages to do so at a very low recall of 0.28 (2 out of 7), whereas the human subject managed to maintain a recall of 0.57 (4 out of 7).

Figure 3(b) shows the performance of the similarity based confidence estimate accumulated over all queries (as Figure 2(b) did for the human subject). The conclusion from this comparison is mixed: On the one hand, the human recall-precision curve lies everywhere above the heuristic curve, on the other hand the highest heuristic F-score ($F = 0.53$) is within 10% of the highest human F-score ($F = 0.60$). This is all the more surprising since our heuristic uses no background knowledge whatsoever, and only counts the number of shared feature-value pairs between the members of the answer set. This lends some support to the conclusion that well chosen very simple "fast and frugal" heuristics can achieve high performance levels.

6 Related work

The topic of "ranking" query results has been studied since the early days of the Semantic Web, and is itself based on even longer lines of research in fields such as Information Retrieval. In fact, our heuristic is closely related to the "clustering hypothesis" as it is known from Information Retrieval [Tombros and Van Rijsbergen, 2001]. Our space is insufficient here to provide an extensive literature survey. Instead, we will discuss a few salient differences between our approach and the literature:

Some of the literature on ranking is concerned with *relevance ranking*: determining which answers are relevant to an unstructured query in natural language, or relevant for a user based on their profile (see [He and Baker, 2010; Stojanovic *et al.*, 2003; Anyanwu *et al.*, 2005; Hurtado *et al.*, 2009] and others). Although interesting and important, this work is not pertinent to the current paper, since we start with SPARQL queries (hence query-relevance is not an isssue) , and we are not considering user-profiles, but we are trying to recognise objectively true answers.

Another part of the literature is concerned with ranking answers by *importance*. Typically, this is done by a variety of pagerank-style analysis of the structure of the Semantic Web, trying to locate which resources are more important, more authoratitive, more trustworthy, etc. [Bamba and Mukherjea, 2005; Ding *et al.*, 2005; Anyanwu *et al.*, 2005] . Our approach differs from all this work in an important way: we do not do any a priori analysis of the structure of the large RDF graph that we are querying (a graph with billions of edges and hundreds of millions of nodes). Instead, we only take the URIs that are returned as a result of a query, and we compute some very simple local properties of these URIs (namely the number of shared feature-value pairs). As we have shown in section 4 this is, surprisingly, already enough to rank the answers such that the best answers get a high ranking, performing within a 10% range of human performance.

Some of the literature on ranking deals with ranking different kinds of objects from what we consider: [E.Thomas *et al.*, 2005; Alani *et al.*, 2006; Tartir and Budak Arpinar, 2007] and others rank ontologies, Swoogle ranks Semantic Web documents [Ding *et al.*, 2005], [Vu *et al.*, 2005] and others rank services, etc. These works rely on fairly sophisticated analyses of the object-to-be-ranked: internal structure of the ontologies, semantic descriptions of the functionality of the services, etc. Instead, we rank only sets of atomic URIs (the members of $A(Q)$), and Although it might seem harder to rank such simple objects, since they come with very little structure to base the ranking on, we have shown in section 4 that a simple analysis of very little information is sufficient to obtain good ranking results, in line with the "fast and frugal heuristics" hypothesis proposed by [Gigerenzer *et al.*, 1999]. It would be interesting to investigate if such simple (or even: simplistic) analysis would also yield good results when applied to more complex objects such as ontologies or services, potentially replacing the more sophisticated ranking techniques found in the literature until now.

A work that is quite close in aim to ours is [Lopez *et al.*, 2009]. Their "semantic similarity" is similar in spirit to ours: it tries to spot wrong answers through their large semantic distance to many of the other answers. However, the semantic distance in [Lopez *et al.*, 2009] is calculated as the distance in a shared ontology. Ours is a much simpler method: we need no ontology at all, and distance is simply calculated as the number of shared feature-value pairs.

7 Conclusion

In this paper, we have shown that a simple cognitively inspired heuristics can be used to select the correct answers from among the query results obtained from querying Linked Open Data sets. Our heuristic is extremely simple and efficient when compared to those proposed in the literature,

while still producing good results, on a par with human performance.

Our work differs from previous work in the following important ways: Firstly, we do not require any expensive prior pagerank-style analysis of the structure of the entire dataspace we are querying. Instead, we do a simple count over information local to the URIs that are returned as query results. In the worst case, the cost of our analysis is limited to retrieving all properties of all elements in the answer set, a cost which is negligible when compared to the large scale network analysis needed for most ranking approaches. Also, any such a prirori large scale link-analysis is likely to be outdated when it is needed for querying. The information that our heuristic needs is so simple that it can be retrieved at query time itself, and is hence always up to date.

Secondly, we do not require any background knowledge or inferencing. No reference is made to any ontological background knowledge, it is not necessary to relate any answers to a shared ontology, and no inference of any kind is performed by our fast-and-frugal heuristic. All we require is to simply retrieve the properties of the elements in the answer set. These are simple atomic queries of the form $\langle s, ?, ? \rangle$, that are efficiently supported by the index-structures of any triple-store.

References

[Alani *et al.*, 2006] H. Alani, C. Brewster, and N. Shadbolt. Ranking ontologies with aktiverank. In *ISWC*, vol. 4273 of *LNCS*, pag. 1–15. Springer, 2006.

[Anyanwu *et al.*, 2005] K. Anyanwu, A. Maduko, A. Sheth. Semrank: ranking complex relationship search results on the semantic web. In *WWW '05*, pag. 117–127. ACM, 2005.

[Bamba and Mukherjea, 2005] B. Bamba and S. Mukherjea. Utilizing resource importance for ranking semantic web query results. In *Semantic Web and Databases*, vol. 3372 of *LNCS*, pag. 185–198. Springer, 2005.

[Bishop *et al.*, 2010a] B. Bishop, A. Kiryakov, D. Ognyanoff, I. Peikov, Z. Tashev, and R. Velkov. Factforge: A fast track to the web of data. *Semantic Web Journal*, 2010. under submission.

[Bishop *et al.*, 2010b] B. Bishop, A. Kiryakov, D. Ognyanoff, I. Peikov, Z. Tashev, and R. Velkov. Owlim: A family of scalable semantic repositories. *Semantic Web Journal*, 2010.

[Bizer and Schultz, 2009] C. Bizer and A. Schultz. The berlin sparql benchmark. *Int. J. On Semantic Web and Information Systems*, 5(1):1–24, 2009.

[Ding *et al.*, 2005] L. Ding, R. Pan, T. Finin, A. Joshi, Y. Peng, and P. Kolari. Finding and ranking knowledge on the semantic web. In *ISWC 2005*, vol. 3729 of *LNCS*, pag. 156–170. Springer, 2005.

[E.Thomas *et al.*, 2005] E.Thomas, H. Alani, D. Sleeman, and C. Brewster. Searching and ranking ontologies on the semantic web. In *K-CAP*, pag. 57–60, 2005.

[Gigerenzer *et al.*, 1999] G. Gigerenzer, P. M. Todd, and the ABC Research Group. *Simple heuristics that make us smart*. Oxford University Press, 1999.

[Guo *et al.*, 2005] Y. Guo, Z. Pan, and J. Heflin. Lubm: A benchmark for owl knowledge base systems. *Web Semantics*, 3(2-3):158 – 182, 2005.

[He and Baker, 2010] X. He and M. Baker. xhrank: Ranking entities on the semantic web. In *ISWC2010*, 2010.

[Hurtado *et al.*, 2009] C. Hurtado, A. Poulovassilis, and P. Wood. Ranking approximate answers to semantic web queries. In *ESWC*, vol. 5554 of *LNCS*, pag. 263–277. Springer, 2009.

[Lopez *et al.*, 2009] V. Lopez, A. Nikolov, M. Fernandez, M. Sabou, V. Uren, and E. Motta. Merging and ranking answers in the semantic web: The wisdom of crowds. In *ASWC*, vol. 5926 of *LNCS*, pag. 135–152. Springer, 2009.

[Ma *et al.*, 2006] L. Ma, Y. Yang, G. Qiu, Z.and Xie, Y. Pan, and S. Liu. Towards a complete owl ontology benchmark. In *ESWC*, vol. 4011 of *LNCS*, pag. 125–139. Springer, 2006.

[Neth *et al.*, 2009] H. Neth, L. Schooler, J. Quesada, and J.s Rieskamp. Analysis of human search strategies. LarKC project deliverable 4.2.2. Technical report, The Large Knowledge Collider (LarKC), 2009.

[Shepard, 1957] R N Shepard. Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space. *Psychometrika*, 22(4):325–345, 1957.

[Simon, 1956] H Simon. Rational choice and the structure of the environment. *Psychological Review*, 63(2):129–38, 1956.

[Simon, 1969] H A Simon. *The Sciences of the Artificial*. The MIT Press, 1969.

[Stojanovic *et al.*, 2003] N. Stojanovic, R. Studer, and L. Stojanovic. An approach for the ranking of query results in the semantic web. In *ISWC 2003*, vol. 2870 of *LNCS*, pag. 500–516. Springer, 2003.

[Tartir and Budak Arpinar, 2007] S. Tartir and I. Budak Arpinar. Ontology evaluation and ranking using ontoqa. In *Int. Conf. on Semantic Computing (ICSC)*, pag. 185 –192, 2007.

[Tombros and Van Rijsbergen, 2001] A. Tombros and C. Van Rijsbergen. Query-sensitive similarity measures for the calculation of interdocument relationships. In *Proc. of the 10th int. conf. on Information and knowledge management CIKM01*. ACM Press, 2001.

[Tversky and Kahneman, 1974] A Tversky and D. Kahneman. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–31, 1974.

[Tversky, 1977] A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.

[Vu *et al.*, 2005] L. Vu, M. Hauswirth, and K. Aberer. Qos-based service selection and ranking with trust and reputation management. In *OTM Conferences*, vol. 3760 of *LNCS*, pag. 466–483. Springer, 2005.