Semantic Web 0 (0) 1
IOS Press

Recognizing the Truth: Automatically Ranking LOD Query Results with a Cluster Heuristic

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

Abstract

The primary contribution of this paper is the development and testing of a *cluster heuristic* that efficiently ranks the quality of answers obtained after querying Linked Open Data (LOD). The heuristic derives from Van Rijsbergen's [52] *cluster hypothesis*: Correct answers tend to be more similar to each other than incorrect ones. Using simple similarity metrics based on Tversky's [50] feature matching model, we show that the *cluster heuristic*'s answer rankings agree remarkably well with the rankings of a human rater.

An additional contribution of the paper is to shed some light on the quality of LOD. We found that on our benchmark set of questions, on average 70% of the answers retrieved from FactForge are correct, while on average 20% of the answers are clearly incorrect. However, we find great deviations from this average across our set of benchmark questions, with some questions scoring 100% correct answers, whereas others yield over 80% of incorrect answers.

The final contribution of this paper is the construction a publically available benchmark collection of 50 general knowledge questions formulated as SPARQL queries that are accompanied by gold standard answers and over 2000 answers obtained by posing the queries to FactForge.net, a large LOD repository. All these answers from FactForge have been manually ranked on their quality. This collection is freely available to other researchers as a benchmarking tool.

Keywords: Linked Open Data, Ranking, Cognitive Heuristic, Data 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. Although 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 FactForge.net, which includes DBPedia, Geonames and Freebase:

> "Stig Anderson" is a member of ABBA. (Actually, he was the band's manager.)
>
> "AmeriCredit" is a U.S.-American car manufacturer. (It is a financial company owned by General Motors to help customers finance their cars.)
>
> "Cosima" is a laureate of the Nobel prize for literature. (It is the title of a novel written by Grazia Deledda, who received this prize in 1926.)
>
> "Anthony H. Gioia" is a kind of pasta. (He is a past chairman of the National Pasta Association.)

* neth@mpib-berlin.mpg.de
** schooler@mpib-berlin.mpg.de
*** annette@cs.vu.nl
**** frank.van.harmelen@cs.vu.nl

> "Richard Bass" is one of the highest summits on the seven continents.
> (He was the first mountaineer that climbed all of them.)

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 can we recognize the truth, i.e., separate correct from incorrect answers?

Our solution to this problem is inspired by cognitive science and the psychology of human judgment and decision making. On a daily basis, people select among alternatives, need to classify food as edible or inedible, and to decide if another is friend or foe. In the 1950s, Herbert Simon noted that much of human problem solving and classification is based on rules of thumb, or heuristics, which simplify problems and cut their solutions down to a manageable size [40,41]. Due to limitations in time and computational capacity human cognition is *bounded* and *adapted* to its environment. Although the world contains an abundance of information, a satisfactory solution does not necessarily seek and integrate as much information as possible, but rather select a suitable amount of information to achieve the current goal.

The merits and potential pitfalls of heuristics have been the subject of extensive debates. In computer science and AI, heuristics are intelligent strategies that are used when optimization techniques are out of reach and provide "criteria, methods, or principles for deciding which among several alternative courses of action promises to be the most effective in order to achieve some goal" [32, p. 3]. In psychology, the research program on *heuristics and biases* [51,25] tends to highlight situations in which people make systematic errors in comparison to some normative standard, such as logic or probability theory. By contrast, the framework on *fast and frugal heuristics* [16,15] emphasizes the positive potential of efficient algorithms and tries to distinguish between the conditions that permit a heuristic to perform well from those conditions that thwart it.

What works well in nature can motivate and inspire the design of artificial systems [42]. In this paper, we borrow an insight from the information retrieval literature and use it to design a heuristic that separates correct from incorrect answers on the basis of semantic similarity. Van Rijsbergen's [52] *cluster hypothesis* "may be simply stated as follows: closely associated documents tend to be relevant to the same requests" (p. 30). To illustrate, Richard Bass, by any metric, should turn out to be less similar to Mount Everest and Mount Kilimanjaro than the two mountains are to each other; in this case, we would expect that a heuristic that identifies relevant answers based on similarity should do well. In contrast, it could be that Stig Anderson (ABBA's manager) and Benny Andersson (an ABBA member) are more similar to each other then they are to Anni-Frid Lyngstad (a female ABBA member); here similarity could lead us astray. As these examples illustrate, similarity will sometimes be a sound basis on which to classify answers obtained from Linked Open Data (LOD), and sometimes it will not. The suitability of semantic similarity as a proxy for the quality of an answer is an open, empirical question that we explore in this paper. Specifically, we test the *cluster heuristic*, an implementation of the *cluster hypothesis* that uses computationally simple measures of similarity.

The main finding of this paper is that cognitively inspired heuristics can indeed be exploited successfully to filter correct answers from the noisy set of answers obtained when querying the Web of Data. Such heuristics can be surprisingly simple when compared to those proposed in the literature, while still producing good results.

An additional contribution of the paper is to shed some light on the quality of LOD. We find that on our benchmark set of questions, on average 70% of all the answers is correct, while on average 20% of the answers are clearly incorrect. However, we find great deviations from this average across our set of benchmark questions, with some questions scoring 100% correct answers, whereas others yield over 80% of incorrect answers.

An *final 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 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[1].

The paper is structured as follows: In Section 2 we sketch some insights from cognitive science on semantic similarity and define the similarity metric imple-

[1] `http://dx.doi.org/10.6084/m9.figshare.882847`. To encourage proper data-citation practices, please cite this dataset as [30].

mented in the *cluster heuristic*. Section 3 describes the benchmark of general knowledge questions, their gold standard answers, and the hand-constructed ranking of the answers returned as a result of the SPARQL versions of these questions. Section 4 describes an experiment that we designed to investigate the performance of the *cluster heuristic* on the task of recognizing correct answers. Section 5 describes the results of these experiments and Section 6 provides an overview of related work. Section 7 discusses our findings, limitations and extensions, and concludes.

2. Defining Semantic Similarity

2.1. Similarity in Cognitive Science

Similarity plays a central role in cognitive science. Psychological notions of similarity typically refer to the proximity or relatedness between mental representations and serve as important explanatory constructs in theories of learning, categorization, memory, reasoning, and decision making (see for an overview [17]).

When an object X is similar to another object Y it may be possible to generalize from X to Y or predict aspects of Y by analogy to X. For instance, if an animal moves and looks like a shark it may be wise to assume that it behaves accordingly and avoid it, rather than to risk experiencing its actual behavior. But any appeal to similarity as an explanation requires specifying in *which way* some entity resembles another. For instance, a dolphin may look and behave like a shark in some ways, but its anatomical and reproductive features identify it as a mammal. Thus, a quantitative measure of similarity needs to be based on a set of principles that precisely define the meaning and dimensions of similarity.

Theoretical accounts of similarity combine precise definitions of the concept with instructions on how to measure similarity. We can distinguish between geometrical approaches, e.g., multidimensional scaling [38,39], featural approaches, e.g. the contrast model [50], structural approaches, e.g. [13,14], transformational approaches, [19,21], and statistical approaches, e.g. latent semantic analysis [12].

As our implementation of the *cluster heuristic* is based on the featural approach, we will introduce Tversky's (1977) notion of similarity in more detail. The basic assumptions of Tversky's (1977) *contrast model* are that objects are represented as collections of discrete features and that similar objects will share many relevant features. In its simplest form, the degree of feature overlap between two objects x and y can be expressed as a linear combination of their common and distinctive features [50, p. 322]:

$$S(X,Y) = \\ \theta f(X \cap Y) - \alpha f(X - Y) - \beta f(Y - X)) \quad (1)$$

Here, X and Y refer to sets of discrete features representing two objects x and y, and their intersection $(X \cap Y)$ represents their shared features. $(X - Y)$ represents the distinctive features of object x, $(Y - X)$ represents the distinctive features of object y. The non-negative parameters θ, α, and β specify the relative importance of the common and unique components and allow for asymmetric relationships (when $\alpha \neq \beta$). The function f is additive on disjoint sets (typically set cardinality) and specifies the contribution of any specific feature to the overall similarity. Thus, $f(X)$ expresses the salience or prominence of object x as a function of its features, which can depend on a variety of factors, including "intensity, frequency, familiarity, good form, and informational content" [50, p. 332f.].

Despite its simplicity, Tversky's contrast model is ubiquitous in discussions on semantic similarity and has been adopted in a variety of different measures, disciplines, and domains, e.g. [10,37]. In the next section, we define the specific version of the model implemented in the *cluster heuristic*.

2.2. *Defining the* Cluster Heuristic

Our explicit goal is to explore the potential of simple strategies to rank the results obtained by querying LOD. The *cluster hypothesis* assumes that better answers will be more similar to each other than worse answers [52]. Our *cluster heuristic* uses a simple measure of semantic similarity to implement such a strategy. Due to their simplicity and ubiquity we explore feature-based similarity metrics in the spirit of Tversky [50]. However, any notion of similarity based on the representation of objects as collections of features first needs to specify *how* the features of an object are defined and determined (see [10]). A substantial part of psychological research addresses the question *how* the relevant features of objects can be found [17]. In the following, we utilize the RDF data model to define the features used by our similarity metrics.

Predicate-object overlap Given the data model underlying RDF a "feature" of a resource or URI s can be defined as a triple $\langle s, p, o\rangle$. Consequently, two entities s_1 and s_2 share a feature if they contain the triples $\langle s_1, p, o\rangle$ and $\langle s_2, p, o\rangle$, respectively. For example, two entities share a feature if they both have a `skos:subject` property with object `dbp-cat:ABBA_members`. Formally,

Definition 1 (Similarity as Predicate-Object Overlap) *The similarity S_{po} based on* predicate-object *overlap $S_{po}(s_1, s_2)$ between two resources s_1 and s_2 in a graph G is defined as:*

$$S_{po}(s_1, s_2, G) = \\ ||\{(p, o)|\langle s_1, p, o\rangle \in G \text{ and } \langle s_2, p, o\rangle \in G\}|| \quad (2)$$

i.e., similarity is defined as the number of predicate-object pairs in G shared by two URIs s_1 and s_2.

This definition of similarity in terms of predicate-object overlap S_{po} looks even simpler as a schematic SPARQL query:

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

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

Quality Estimate and Cluster Hypothesis The *cluster hypothesis*, as it is known from Information Retrieval [52,49], states that documents relevant to a query (or in our case: correct LOD answers to a query) tend to be more similar to each other than to irrelevant (or incorrect) ones. In more formal terms, this means that a similarity measure S can be used as a quality estimate for query-answers, where S can be S_{po} or S_p from above:

Definition 2 (Quality Estimate of an Answer) *If Q is a query over a graph G, yielding a set of answers A, then a quality estimate $E(a, Q, G)$ for an answer $a \in A$ is defined as*

$$E(a, Q, G) = \Sigma_{a' \in A - \{a\}} S(a, a', G) \quad (3)$$

i.e., the estimated quality of an answer a is the aggregate similarity of a to every other answer a'.

Definition 3 (Van Rijsbergen's *Cluster Hypothesis*) *If a_1 and a_2 are two answers to a query Q over a graph G, then the* cluster hypothesis *states that*

$$a_1 \text{ is a better answer than } a_2 \text{ iff} \\ E(a_1, Q, G) > E(a_2, Q, G) \quad (4)$$

Besides the above definition of S_{po} there is a large variety of other similarity metrics that could be used as the basis for defining E, and hence for formalizing the *cluster hypothesis*. Our current implementation of the *cluster heuristic* uses the metric defined by Equation 3, but we discuss some even simpler, normalized, and asymmetric alternatives in Section 7.2.

Example Assume that a query "*Name the members of the pop band ABBA*" returns six answers: Agnetha Fältskog, Anni-Frid Lyngstad, Benny Andersson, Björn Ulvaeus, Ola Brunkert, and Stig Anderson. Table 1 shows that *Agnetha Fältskog* shares 76 property-value pairs with *Ola Brunkert*, but as many as 318 property-value pairs with *Benny Anderson*, etc. Thus,

S_{po}(Agnetha Fältskog, Ola Brunkert) = 76,
S_{po}(Agnetha Fältskog, Benny Anderson) = 318.

Due to the symmetry of S_{po} (i.e., $S_{po}(a_1, a_2) = s_{po}(a_2, a_1)$ for any answers a_1 and a_2), the matrix of Table 1 is also symmetric. Applying Definition 2, we find that the quality estimate for the answers *Agnetha Fältskog* and *Ola Brunkert* are:

E(Agnetha Fältskog) = 1119,
E(Ola Brunkert) = 356.

Thus, when asking for members of ABBA, the quality of answer *Agnetha Fältskog* exceeds the quality of answer *Ola Brunkert*. This is encouraging, as *Agnetha Fältskog* is indeed one of the four band members, whereas *Ola Brunkert* is a drummer who appeared on all of their albums.

3. 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 [18], the extended eLUBM benchmark [29], and the Berlin SPARQL benchmark [8].[2] However, all these benchmarks refer to *synthetic* datasets. There is a shortage of *realistic* benchmarks that provide both real-world queries and validated ("gold standard") answers. The sample queries on the webpages of Linked Life Data (`http://`

[2] Additional benchmarks are described at `http://www.w3.org/wiki/RdfStoreBenchmarking`.

	Ola Brunkert	Agnetha Fältskog*	Anni-Frid Lyngstad*	Björn Ulvaeus*	Benny Andersson*	Stig Anderson
Ola Brunkert		76	70	67	80	63
Agnetha Fältskog*	76		356	276	318	93
Anni-Frid Lyngstad*	70	356		271	287	91
Björn Ulvaeus*	67	276	271		431	102
Benny Andersson*	80	318	287	431		102
Stig Anderson	63	93	91	102	102	
Quality estimate	356	1119	1075	1147	1218	451

Table 1

Similarity matrix for Query 41 ("Name the members of the pop band ABBA"). Names with an asterix denote the four actual members.

`linkedlifedata.com/sparql`), FactForge (`http://factforge.net/sparql`) are examples of such realistic queries, but they do not come with a validated set of gold standard answers.

3.1. Set of Questions

For an experiment investigating how people search for information in their memory, [31] designed a set of 50 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 the members of The Beatles", "Name the Nobel laureates in literature since 1945", etc.

3.2. Gold Standard Answers

[31] determined a set of correct answers for each question. The number of true exemplars varied widely between questions, 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').

3.3. SPARQL Queries

We have developed a set of 50 SPARQL queries, made to resemble the questions from [31]. For this translation, we used a number of well-known namespaces, such as DBPedia, Freebase, Geonames, UMBEL, etc. (See Section 4.2 for details.)

3.4. SPARQL Answers

To complete this benchmark collection, we executed all of our queries against FactForge (`http://factforge.net`). FactForge [7] is a collection of some of the most central datasources in the LOD cloud and 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 [26] to materialize all inferences that can be drawn from the datasets and their schemata. This results in some 10 billion retrievable statements, describing just over 500 million entities. Although FactForge is only a subset of the entire Web of Data, it is currently one of the the largest available queryable subsets. We used the version of FactForge that is closed under inference, since this reflects the semantics of the Semantic Web languages used on the LOD cloud.

Our fifty queries produced 2197 distinct answers (i.e., 2197 distinct URIs). These URIs came with 4836 distinct natural-language labels (specified through an `rdfs:label` property), in a variety of languages.

3.5. Human Performance

In order to measure computer performance on ranking answers by an implementation of the *cluster heuristic*, we first obtained the ranking by a human judge as our baseline. To this end, all 4836 candidate `rdfs:labels` were scored by a human judge on a 5-point Likert scale, indicating their perceived correctness. (See Section 4.4 for details.)

The entire set of resources (original questions, their SPARQL translations, the gold standard answers, query-results against FactForge, as well as all human rankings of these query results) are available online [30].

4. Experimental Design

In this section we describe the experiment we designed in order to evaluate the quality of our simple similarity measures and the success of the *cluster heuristic* in ranking answers over LOD sources. In brief, our experiment consists of the following steps:

1. Construct a set of natural language general knowledge questions;
2. Translate these questions into computer representations;
3. Run these computer queries against a large LOD repository to obtain candidate answers;
4. Rate the correctness of all candidate answers by a human judge;
5. Rate the correctness of all candidate answers by the *cluster heuristic*, using a measure of semantic similarity;
6. Compare the human ratings (Step 4) with the *cluster heuristic*'s ratings (Step 5).

We now provide additional details to each of these steps.

4.1. Step 1: Construct a Set of Questions

We are using the set of general knowledge questions described in Section 3.1 for this purpose. These questions are all *enumeration* questions, which means that their answers consist of a set of objects. Typical examples for this type of questions from our queries include: "Name the highest summit on each of the seven continents" or "Name the members of the pop band ABBA".

Such enumeration questions are an important class of questions, not only whenever general knowledge questions are concerned, but also in many areas of scientific enquiry. As an example, we refer to [53], where a large pharmaceutical research consortium defined a set of 20 top-ranked research questions that should be answerable by a Linked Data question-answering system. Examples of such questions are "Give all oxidoreductase inhibitors with an activity $<$100nM in both human and mouse", or "For a given compound, which targets have been patented in the context of Alzheimer's disease?". In fact, the vast majority of these top-ranked research questions are of the enumeration type.

4.2. Step 2: Translate the Questions into Computer Representations

Each of the 50 questions were manually translated into SPARQL queries. As an example, the question about ABBA's members translates to the SPARQL query shown in Figure 1.

Any such translation is difficult, error-prone, and raises the question how faithful the SPARQL queries are to the original natural-language questions. To analyse this, we arranged matters as follows: First, the original 50 questions were designed by one of the authors. A second author then translated them into SPARQL queries. In a third step, the creator of the original questions verified the veracity of the SPARQL queries. Thirty-eight (76%) of the SPARQL queries were deemed "identical" to the original question, and the others were described as "close". Consequently, we trust that the correspondence between the questions and corresponding SPARQL queries is high and sufficient for our purposes, especially as the outcome of our experiment is only minimally affected by these choices. Whatever the SPARQL query is, both human judge and computer are asked to rank the same result sets from the SPARQL queries, and are hence both subject to the same "noise" that might have inadvertently been introduced by any choices in the formulation of the SPARQL queries.

4.3. Step 3: Run the Queries on FactForge

Running our 50 queries against FactForge (in its version of November 2011) resulted in 2197 distinct URIs as candidate answers, with a total of 4836 candidate `rdfs:labels`. The average size of an answer set is just under 100 URIs per query, but the exact number varies from 6 to 360 (median = 45.5). All answer URIs contain multiple `rdfs:labels`, with an average of just over two per URI, but sometimes as many as 10. Names of people and geographic places in particular tend to have many different labels.

Just by looking at the magnitude and variance of the answer sets it is obvious that FactForge is a noisy dataset in which queries return many incorrect answers: The answer set for "Name the winners of the Nobel peace prize" contains no less than 322 elements.

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

URI	rdfs:label
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. An example query and candidate answer-set.

An example answer-set is shown in Figure 1. Again, we see the messiness of LOD sources: A query that only has four correct answers yields six URIs, synonyms in the answer set, as well as two incorrect answers.

4.4. Step 4: Rate All Candiate Answers by a Human

One of our authors ranked the perceived correctness of all candidate `rdfs:labels` on a 5-point Likert scale, with 5 indicating the answers on which he was most confident that they were correct, and 1 indicating answers on which he was most confident that they were incorrect. During this extensive scoring task (of 4836 items), the human judge used background knowledge sources (online pages and encyclopediae) and was asked to proceed at reasonable speed.

Since the semantics of OWL and RDF assumes that URIs are simply meaningless identifiers (for example `opencyc:Mx4rwUIMiJwpEbGdrcN5Y29ycA`, denoting the oil company BP), the rankings of the human judge were based on the natural language `rdfs:labels` for each of the answers. All answer URIs contained multiple `rdfs:labels`, and we calculated the ranking of a URI as the maximum of the ranks assigned to its associated `rdfs:labels`. This is reasonable because unfamiliar labels received low scores from the human judge even if there was another label for the same URI that was recognized as a correct answer. As we will discuss in Section 5, this step provides us with unique insights into the correctness of FactForge and, by extension, of LOD sources in general.

4.5. Step 5: Rate All Candiate Answers by the Cluster Heuristic

Next, we applied the *cluster heuristic* to the same dataset. Specifically, we calculated $E(a, Q, G)$ for every candidate answer a (2197 distinct URIs in total) to all our questions Q (50 in total) using FactForge.net as our graph G. This amounts to calculating a similarity matrix as shown in Table 1 for each of the questions Q, summing up the row (or column) for every candidate answer a, and sorting all answers to each question on the basis of this value. These resulting similarity matrices (as obtained in November 2011) have been made available in our dataset [30].

4.6. Step 6: Compare Human Ratings with the Cluster Heuristic

The two rankings completed in Steps 4 and 5 yielded two sets of partially ordered lists, i.e., all answers to each question in our corpus ranked by both the human judge and by using the similarity-based *cluster heuristic*. These lists are partially ordered because multiple answers can share the same rank. To determine the correspondence between the human-ranked and the machine-ranked lists we use two different measures: A first measure will compare the relative ordering among list elements, and a second measure will compare the absolute scores on the 1-to-5 scale.

Comparing Relative Orderings Our problem is similar to ranking the results of search engines: We can view the ranked answers of the *cluster heuristic* as the

results of a search engine, and the scores of our human judge as the target answers for the same query.

In the field of Information Retrieval, a measure of normalized discounted cumulative gain ($nDCG$) is used frequently for judging the results of search engines, e.g. [23]. The $nDCG$ measure is based on the non-normalized discounted cumulative gain (DCG). If l is the length of a ranked list of answers, and Q_i is the human-judged quality of the element at position i in the list, then DCG is defined as:[3]

$$DCG_l = \sum_{i=1}^{l} \frac{Q_i}{log_2(i+1)} \quad (5)$$

This says that the DCG of a ranked list answers of length l is the summation over all items in the set of the quality of each item (as determined by the human judge) divided by the (log of) the position of the item in the list, increased uniformly by 1 (to avoid division by $log_2 1$). The intuition behind this definition is that the overall value of a list l is increased for each correct item in the list, that this gain should be proportional to an item's quality Q_i, but that this gain should be lower ("discounted") for later list items (i.e., division by the log_2). In short, the total "gain" is the score received for putting the correct items in the answer set, and the "discount" is the reduction of this gain by putting items at the wrong position of a list.

We have chosen to use $nDCG$ as our measure of relative ordering over the use of other well-known measures, such as Kendall's Tau distance or Spearman's Rho. Both of these treat errors low in the list equal to errors high in the list, while $nDCG$ penalizes errors high in the list more heavily, which is more appropriate for our task of recognizing correct answers.

Example We will illustrate our definitions using Query 41 ("Name the members of the pop band ABBA"). Table 2 shows the ranking of the answers as determined by the human judge, and repeats the bottom row of Table 1 for the quality measure as determined by the *cluster heuristic*. The DCG value is now computed as:[4]

Answer	Human Rank	Cluster Heuristic
Benny Andersson*	5	1218
Björn Ulvaeus*	5	1147
Agnetha Fältskog*	5	1119
Anni-Frid Lyngstad*	5	1075
Stig Anderson	1	451
Ola Brunkert	2	356

Table 2
Human ranking and cluster similarity score for Query 41 ("Name the members of the pop band ABBA").
Terms with an asterix denote gold standard answers.

$$\begin{aligned} &DCG \\ &= \tfrac{5}{log_2 2} + \tfrac{5}{log_2 3} + \tfrac{5}{log_2 4} + \tfrac{5}{log_2 5} + \tfrac{1}{log_2 6} + \tfrac{2}{log_2 7} \\ &= 13.91 \end{aligned}$$

The ideal ordering would have had the answers *Ola Brunkert* and *Stig Anderson* swapped, so that the DCG of the ideal ordering (typically written as $iDCG$) is

$$\begin{aligned} &iDCG \\ &= \tfrac{5}{log_2 2} + \tfrac{5}{log_2 3} + \tfrac{5}{log_2 4} + \tfrac{5}{log_2 5} + \tfrac{2}{log_2 6} + \tfrac{1}{log_2 7} \\ &= 13.94 \end{aligned}$$

The values of DCG grow of course with the size of the answer set l, making it unsuitable for use across answer sets of different length (as in our case, since the different questions in our dataset result in widely different sizes of answer sets). This motivates the use of the *normalized* Discounted Cumulative Gain $nDCG$. In the literature, this is simply defined as $DCG/iDCG$, which ranges from 0 to 1, independent of the length of the answer set. However, in our setting it would be too easy to gain a high $nDCG$ score under this definition. Typically, DCG is used in the literature to score the results of search engines. In that setting the gain-scores (the human-ranked column in Table 2 are 0 for the vast majority of the answers (after all: for any particular query, the vast majority of web-pages are entirely irrelevant). In such a setting, the worst DCG score is 0 (when the computer picks only irrelevant webpages). However, in our setting, all the answer are already given, and the computer only

[3] Slight variants of this measure exist (e.g., by dividing by $log_2 i$), but these do not affect the substance of the measure

[4] This example also illustrates that the DCG measure is robust against buckets of equal score: any ordering of the answers with rank 5 will give the same DCG value

has to rank them in the correct order. As a result, in our case, the minimal DCG score is far from 0. In the example from Table 2, the lowest possible DCG score (for the worst possible gain sequence: [1,2,5,5,5,5]) is in fact as high as 10.63. And even this minimal DCG score is too low of a lower-bound on the performance of our algorithm. A simple algorithm that just guesses a random sequence would often obtain a higher DCG than the minimal value. In our example from Table 2, a simple numerical simulation tells us that a random algorithm scores on average a DCG value as high as 12.66.

Thus, in order to (a) normalize the DCG value irrespective of the length of the answer set, and to (b) set an informative baseline for the performance of our algorithm, we normalize the DCG value with respect to the DCG value scored by a random guessing algorithm (denoted as $rDCG$) as follows:

Definition 4 (Measure for Relative Ordering)

$$nDCG = \frac{DCG - rDCG}{iDCG - rDCG} \tag{6}$$

This $nDCG$ value measures the improvement of our algorithm over a random baseline, with a value that's independent of the size of the answer set. Notice that this measure has the appropriate properties:

1. $nDCG > 0$ when our algorithm's ranking is better then random; $nDCG = 0$ when our algorithm ranks answers no better than random; $nDCG < 0$ in the unfortunate case that our algorithm's ranking of answers would be worse then random,
2. $nDCG = 1$ when our algorithm returns the perfect ranking (since then $DCG = iDCG$).
3. $nDCG \leq DCG/iDCG$ (since $DCG \leq iDCG$, by definition of $iDCG$), reflecting our intention that our version of $nDCG$ is indeed a tougher measure than the one found in the literature. In our example, $DCG = 13.91, iDCG = 13.94$ and $rDCG = 12.66$, making $nDCG = 0.977$ while $DCG/iDCG = 0.998$.
4. In the traditional setting from the literature, the vast majority of answers has a gain value of 0. In that case $rDCG \approx 0$ and our definition of $nDCG$ converges to the standard definition $DCG/iDCG$.

On lists of with only equally ranked answers (e.g., only correct answers, all ranked 5 by the human expert), we would have $DCG = iDCG = rDCG$, and we define the $nDCG = 1$, because in that case the heuristic has returned the perfect ranking.

Summarizing: In order to provide an informative baseline for our ranking algorithm, we begin with the human ranking for all answers to each of our 50 queries, then determine the $rDCG$ value for each of these human rankings by simple computational simulation, and then calculate the $nDCG$ score of the ranking provided by our *cluster heuristic*, which thus reflects the improvement of our algorithm over a random baseline.

Comparing Absolute Rank The previous measure only scored whether the relative rankings by human expert and the *cluster heuristic* agree. However, this still allows for the possibility that although the relative rankings agree, the absolute estimates of the correctness differ widely between human expert and computational algorithm. Consider the human expert producing a ranking like [5,1,1], clearly distinguishing a single correct answer from two incorrect ones. If the computer heuristic produces something like [300,299,298], the relative rankings fully agree (and the corresponding quality measure will tell us $DCG = 1$, even though the computer clearly failed to distinguish the single correct answer from the two incorrect ones. We will therefore apply a second quality measure to our heuristic, in order to compare not just relative, but rather absolute ranks.

This, it is necessary that we first scale back the scores obtained by our *cluster heuristic* (as in Table 1) to the 5-point Likert-scale used by our human expert. We do this by linearly scaling the interval between the highest and the lowest *cluster heuristic* value to a 5-point scale for every question.

After this linear scaling into a 5-point scale, we now have two vectors of equal length with elements from 1-to-5. A standard way to compare the distance between two of such vectors is to simply take the Manhattan distance between these vectors, see Spearman's footrule distance [43]. Again, this measure is widely used for comparing ranked data in diverse areas such as search engines, bioinformatics, genomics, and information science [36].

If H is the human expert rating of all the answers on a [1–5] scale, and C is the *cluster heuristic* rating of the same answers, scaled back to a [1–5] scale, then the Manhattan distance MD between H and C is simply

$$\sum_{i=1}^{n} |H_i - C_i|$$

where H_i is the i-th element of the vector H, and n is the length of the vectors. We adjust this usual definition somewhat in order to make the MD-value independent of the length size n of the answer set, and to give this measure the same direction as $nDCG$ above, with 1 being the ideal value. We then have

$$MD = 1 - \frac{\sum_{i=1}^{n} |H_i - C_i|}{4n}$$

i.e., rescaling by $4n$, the maximal distance between H and C, and inverting the direction of the measure by subtracting it from 1.

Again, as in the previous section, we will use the average distance between H and a randomly guessed sequence (written rMD) as our baseline. The normalized Manhattan Distance that we use in our experiment is then

Definition 5 (Measure for Absolute Ranks)

$$nMD = \frac{MD - rMD}{1 - rMD}$$

As with $nNDCG$, this measure has the same expected properties:

1. $nMD > 0$ when our algorithm's ranking of answers is better then random; $nMD = 0$ when our algorithm ranks answers no better than random; $nMD < 0$ in the unfortunate case that our algorithm's ranking of answers would be worse then random,
2. $nMD = 1$ when our algorithm returns the perfect ranking (since then $MD = 1$).

As before, the case where all answers receive equal rank leads to $MD = rMD = 1$, and we define $nMD = 1$ because the heuristic found the perfect ranking.

In our running example (Query 41,"Name the members of the pop band ABBA"), the the human ranked vector was [5,5,5,5,1,2] (see Table 2). A randomly guessed sequence of six Likert-values has average a distance of $nMD = 0.48$ to that vector. By scaling the scores for the *cluster heuristic* to the 1–5 interval, we obtain the vector [5,5,5,5,2,2]. The Manhattan distance between that vector and the human ranked vector is $MD = 0.95$. This renders nMD for the *cluster heuristic* on this query $nMD = (0.95 - 0.48)/(1 - 0.48) = 0.904$.

5. Results

In this section, we present the results of the experimental setup described in the previous section and discuss these results.

5.1. Quality of FactForge and the LOD Cloud

As explained before, our human judge rated all 4836 answer labels for correctness on a 5-point Likert scale, and these ratings were aggregated to score the 2197 distinct URIs that corresponded to the 4836 labels. Under the reasonable assumption that this resulted in a reliable score of the correctness of the answers, this provides us with a unique insight into the correctness of FactForge. And since FactForge contains some of the most prominent and central elements of the LOD cloud, the results for FactForge can be taken as indicative for the quality of the entire LOD cloud.

Figure 2 shows the distribution of scores from the human ranking effort. Looking at the average percentage of answers for each rank, we see that 58% of answers were ranked as correct, and a total of 70% (ranks 4 and 5) were ranked as correct or probably correct. For two out of fifty questions FactForge even yielded only correct answers (the 100% in the bottom right cell). These were the questions on the names of U.S.-American states and of all U.S. post-war presidents. On the one hand, this is an encouraging result for the quality of the LOD cloud. On the other hand, Figure 2 also shows ample room for improvement. For instance, 30% of all answers were deemed less than "almost correct" (ranks 1–3), there exist questions for which as much as 82% of the answers are incorrect (rank-1), and vice versa, there are questions for which only 4% of the answers are correct (rank-5). Summarizing, this suggests a very mixed view of the quality of the LOD cloud, with many answers being correct, but also with some result sets that are almost entirely incorrect.

This mixed view justifies the general motivation of this work, namely that LOD sources are indeed noisy, that they often return incorrect answers, and that methods are needed to (preferably automatically) rank answers from correct to incorrect.

5.2. Performance of the Cluster Heuristic

This section presents this paper's main results. Figure 3 shows the performance of our *cluster heuristic* on our corpus of fifty benchmark queries, as mea-

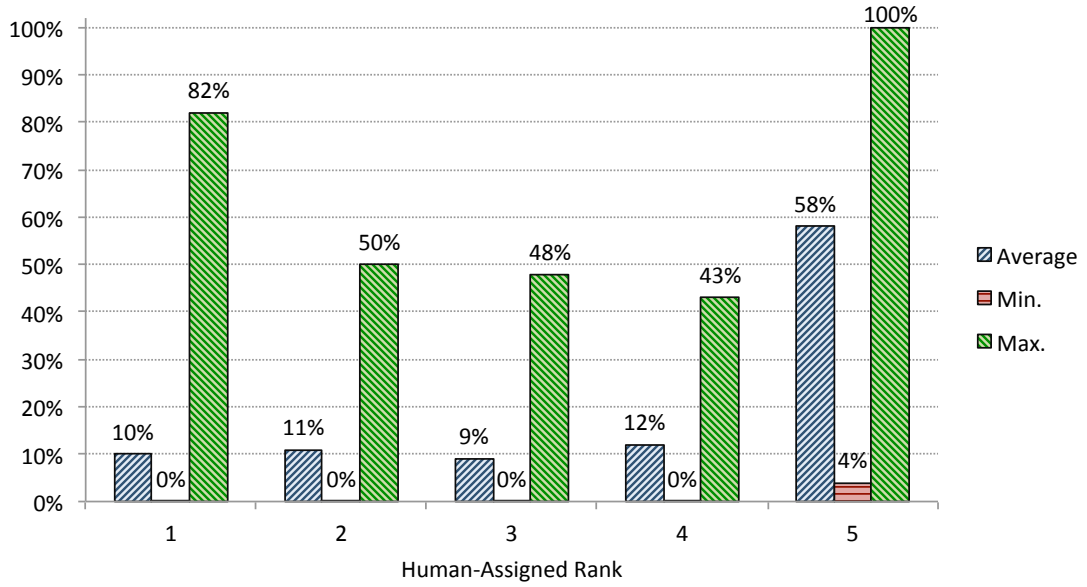


Figure 2. Distribution of human rankings of FactForge answers. (A maximal rank of 5 indicated correct answers.)

sured by our two metrics $nDCG$ (correspondence of relative ranking with human expert) and nMD (correspondence of absolute ranking with human expert).

The fact that the *cluster heuristic* yields positive values for the vast majority of queries demonstrates that it performs very well as a method for distinguishing correct from incorrect answers. It is outperforming a random guess on both of the metrics in 43 out of 50 cases, deviating significantly from a random baseline ($p < .001$). This is all the more surprising because the *cluster heuristic* does not use domain knowledge of any kind. Thus: *without any domain knowledge, and only counting overlap in feature/value pairs, the* cluster heuristic *highly reliably distinguishes correct answers from incorrect answers on a corpus of fifty general knowledge questions*.

Figure 3 also illustrates that correspondence on the absolute rank is harder to achieve than correspondence on the relative rank (i.e., grey bars are generally shorter than black bars). In the following, we will discuss the behavior of the *cluster heuristic* in more detail — both its successful cases and the few cases for which it fails.

5.3. *Example Results of the* Cluster Heuristic

Figure 4 shows the results for our running example (Query 41, "Name the members of the pop band ABBA"). As we saw in Section 4.6, for this query we obtained $nDCG = 0.977$ and $nMD = 0.904$. As a value of 1 would indicate a perfect score, such high values demonstrate that the *cluster heuristic* does a really good job on this query to mimic the selection of the correct answers by a human judge.

Figure 4 is a *graphical* representation of the same effect. Every node in this graph is one of the six answers to this query. Line thickness and node attraction is proportional to the similarity measure S_{po} from Definition 1. The layout of the graph has been calculated using a standard force-based algorithm. The human ranking is indicated in brackets in the label of each node. The central hypothesis of this paper from Definition 3 can now can be restated in graphical terms: Nodes with rank 5 (i.e., the answers that are judged as correct by the human rater) should cluster together in the graph (because the value of their quality estimate E should be high) while nodes with lower numbers should be further removed from the clustered nodes. And indeed, Figure 4 shows this desired property: the answers with a rank of 5 cluster together, while the two erroneous answers (with ranks of 1 and 2) are outliers in the cluster diagram. In essence, Figure 4 provides a visualization of the values of the *cluster heuristic* reported in Table 2, which are themselves accumulated from the similarity matrix in Table 1.

A second, and slightly more elaborate illustration of how well the *cluster heuristic* works is provided by Figure 5, which illustrates the results to Query 4 ("Name the planets of our solar system"). For this query, there also is a close correspondence between the human ranking and the *cluster heuristic*: $nDCG =$

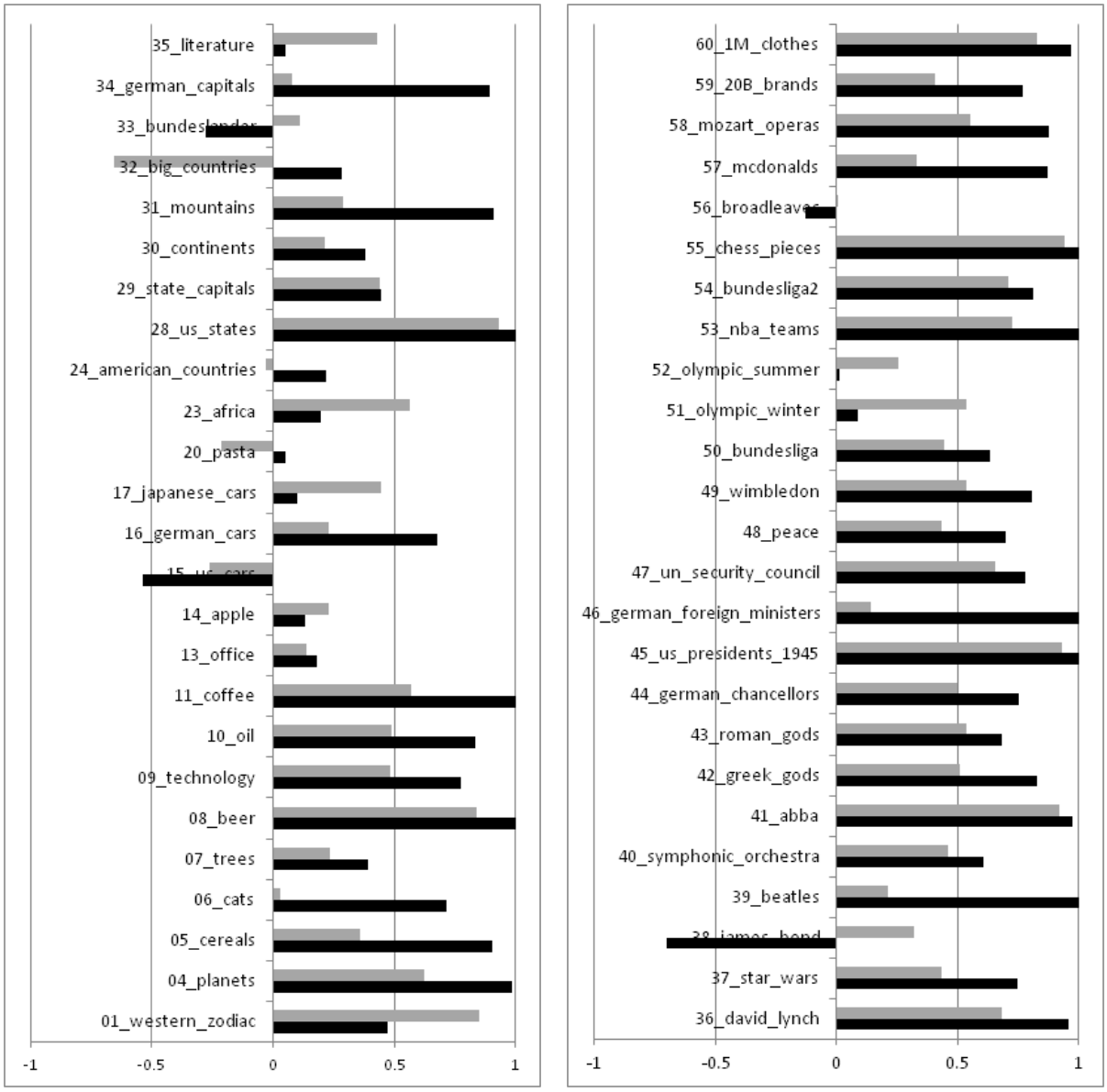
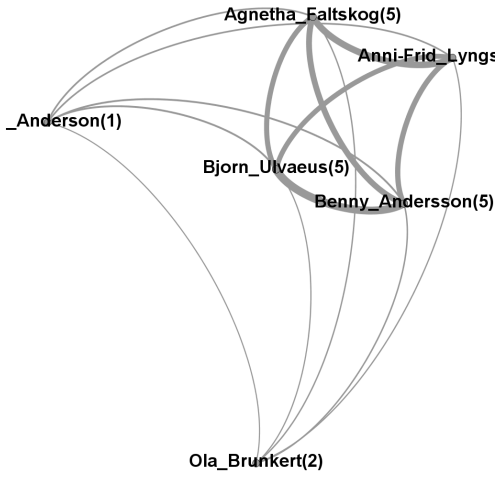


Figure 3. $nDCG$ scores (black) and nMD scores (grey) for all fifty benchmark queries. A value of 0 corresponds to the performance of a random ranking algorithm and a value of 1 represents perfect behavior (see Section 4.6).

0.985 and $nMD = 0.625$. And indeed all the rank 5 nodes are clustered together in the graph, with the rank 1 and rank 2 nodes (i.e., incorrect answers) showing up as outliers in the similarity network. Interestingly, Pluto was originally included as a correct answer in the gold standard, but has officially been recategorized as a dwarf planet of the Kuiper belt in 2008. Both the human judge was uncertain about its status (assigning a rank of 3) and the *cluster heuristic* places it in between the high-ranked cluster and the low-ranked outliers. This re-iterates and graphically visualises that our similarity index and the *cluster heuristic* yield good proxies for judgments about an item's correctness.

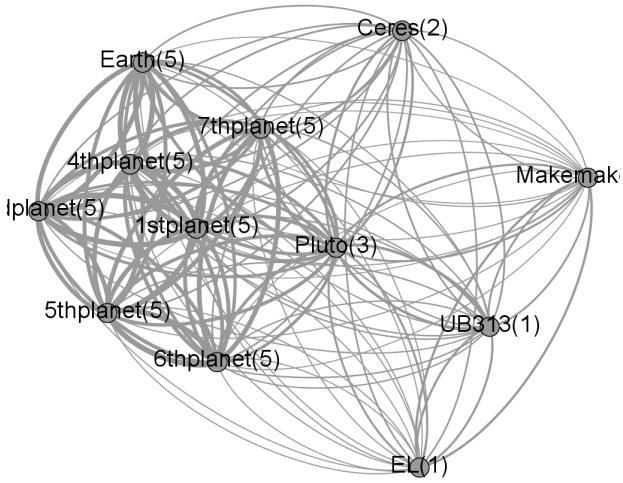
5.4. *Single Example of Failure of the* Cluster Heuristic

Among our fifty benchmark queries, there is only one result where the *cluster heuristic* clearly fails. Fig-



Answer	Human Rank	Cluster Heuristic
Benny Andersson*	5	1218
Björn Ulvaeus*	5	1147
Agnetha Fältskog*	5	1119
Anni-Frid Lyngstad*	5	1075
Stig Anderson	1	451
Ola Brunkert	2	356

Figure 4. The answers to Query 41 ("Name the members of the pop band ABBA"), visualized using the results of the *cluster heuristic*. Line thickness and node attraction is proportional to the similarity measure S_{po} from Definition 1. Terms with an asterix denote gold standard answers.



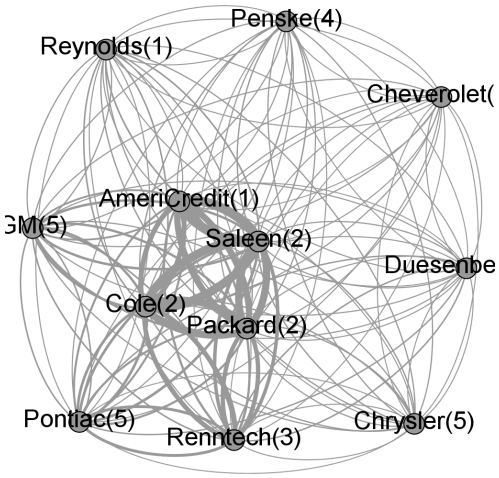
Answer	Human Rank	Cluster Heuristic
4th planet*	5	10892
1st planet*	5	10873
5th planet*	5	10781
2nd planet*	5	10298
Earth*	5	9988
6th planet*	5	9859
Pluto*	3	9435
7th planet*	5	9404
Ceres	2	5935
Eris	1	5604
Haumea	1	4979
Makemake	1	4634

Figure 5. The answers to Query 4 ("Name the planets of our solar system"), and the scores of both the human expert and the *cluster heuristic*. Terms with an asterix denote gold standard answers.

ure 3 shows that both of our metrics score worse than the random baseline for Query 15 ("Name all current U.S. car manufacturers"). The similarity graph for this query (as shown in Figure 6) explains why. Not only did this query yield quite a few incorrect answers (as shown by low human ranking values) but these incorrect answers shared more features between them than the correct answers (as shown by both the *cluster heuristic* scores and the clustering graph in Figure 6): The central cluster does not consist of nodes with a human-assigned rank of 5, but instead of nodes with low human-assigned ranks. Thus, the *cluster heuristic* here really fails — it is simply not the case that "correct answer look alike", but incorrect answers look alike instead. However, across our set of fifty benchmark queries, this is the only case where this phenomenon occurs. There are a few other cases in which the *cluster heuristic* scores poorly, but these are due to artifacts of our own metrics, which we discuss now.

5.5. The Effects of Scaling

The results in Figure 3 show that our two measures perform differently for Query 32 ("Name all countries with a population exceeding 80 million people"):



Answer	Human Rank	Cluster Heuristic
Saleen Incorporated	2	1960
Cole Motor Car	2	1785
Packard	2	1717
AmeriCredit	1	1562
Renntech	3	1207
Pontiac*	5	956
General Motors*	5	885
Reynolds Metals	1	663
Chrysler*	5	626
Duesenberg	2	622
Penske Automotive	4	620
Chevrolet*	5	577

Figure 6. The answers to Query 15 ("Name all current U.S. car manufacturers"), and the scores of both the human expert and the *cluster heuristic*. Terms with an asterix denote gold standard answers.

Although the $nDCG$ value outperforms the random baseline, the nMD value is significantly worse.

In this case, the distribution of E values is very uneven. The highest value is E(United States) = 172,758 while the second highest value is E(Indonesia) = 43,013, and the lowest value (27th place) is E(Russia) = 21,872. Due to the initial outlier, linearly scaling the interval [21,872–172,758] to the interval [1–5] assigns a minimal rank of 1 to all values below 52,049. This anomaly assignes a rank of 1 to many good answers, which received ratings of 4 or 5 by the human judge, resulting in a low score for the Manhattan metric. The $nDCG$ metric does not suffer from this problem: it only uses *relative* ranks, and those are not affected by the scaling procedure. We do not consider this to be a flaw in the *cluster heuristic*, but rather a limitation of our nMD metric.

5.6. The Effects of Meta-Elements

The other two remaining cases with poor scores (Query 33, "Name all German states (Bundesländer)" and Query 38, "Name all James Bond movies") both suffer from another effect, which we call *the presence of meta-elements*. Besides correct names of "James Bond movies", the answers to Query 38 also contain a number of lists of such movies, that correspond to pages in Wikipedia, and hence DBPedia [5]. Such meta-elements are not correct answers in themselves, but contain many links to correct answers (i.e., URIs for "James Bond movies"). Thus, such meta-elements are not filtered out by our *cluster heuristic*, and hence cause a reduced score.

[5] e.g. `http://en.wikipedia.org/wiki/List_of_James_Bond_films`

5.7. Runtime Costs

The cost of calculating the *cluster heuristic*'s similarity measure is quadratic in the size of the answer set. This sounds demanding, but the size of the *answer set* for typical queries is much smaller than the size of the *data set*. In our experiments, the size of the dataset (FactForge) is $O(10^8)$ while the size of the answer set never exceeds $O(10^2)$. Furthermore, the queries required to calculate the similarity measures are extremely simple, and are typically retrievable directly from the index structures of most triple stores. Notice also that only the *number* of shared feature-value pairs needs to be transferred from server to client, and not the actual set of feature-value pairs themselves, greatly reducing the network communication load.

6. Related Work

This section discusses related work in the Semantic Web literature. To the best of our knowledge, there is little or no work that is directly aimed at the central question of this paper: How to recognize the truth by separating correct from incorrect answers. In the absence of directly relevant preceding work, the closest related work would seem to be the literature on

ranking, which has been studied in the context of the Semantic Web. A number of semantic ranking approaches have been published. The ideas for ranking query results to Web of Data are based on methods, from different origins, some with influence from the ranking approaches devised for classical search systems in IR.

The recent literature on ranking for the Semantic Web is s very heterogeneous, with many different techniques, being used for very different purposes, and applied to very different datasets. In an an attempt to categorize the literature, an important distinction is to distinguish what is being ranked. Following [11], we discern three major categories of papers: (i) papers that discuss the ranking of predicates or relationships of RDF assertions, e.g. [5,4,3], whereas (ii) other papers, e.g. [6,22], discuss the ranking of Semantic Resource Instances, which can be either subjects or objects in the RDF assertion statement. Finally, (iii) some papers, e.g. [1,11,45,47], discuss ranking of entire ontologies. For comparison with our own work, obviously the second category (ranking instances) is the most relevant. Nevertheless, we will still consider the specific ranking methods studied in the other two categories.

A second important distinction is on which basis the ranking is being done. Here we distinguish between four categories: (i) comparison on the basis of semantic similarity among alternative answers; (ii) ranking on the basis of relevance to either the original question or a user-profile; (iii) ranking on the basis of various meta-properties such as authority, popularity, origin, etc., and finally (iv) ranking of complex objects (ontologies, services, etc) based on properties of these complex objects. We now discuss these four categories in more detail. (These categories are also listed as the third row in the tables of Appendix A (p. 20).

(i) Semantic Similarity Obviously our own approach discussed in this paper falls into this category. [27] also use a semantic similarity approach to rank the results of a query, but calculate semantic distance as the distance in a shared ontology. That paper proposes a merging algorithm that aggregates, combines and filters ontology-based search results and uses three different ranking algorithms that sort the final answers according to different criteria such as popularity, confidence and semantic interpretation of the results.

(ii) Ranking by Relevance Under this approach, answers are ranked on their relevance to the original query (often an unstructured query in natural language) or on their relevance to a user profile. [20] propose a ranking mechanism, xhRank, that is a summation of relevance, importance and query-length ranking. [44] propose an ontology-based ranking algorithm in which the relevance of a web resource to a users query is determined by utilizing the explicit semantics of relationships between ontological entities. [22] develop a querying system that performs an approximate matching of the users query to the data and ranks the answers in terms of how closely they match to the original query of user. The approximate matching framework incorporate standard notions of approximation such as edit distance as well as some RDFS inference rules, thereby capturing semantic as well as syntactic approximations.

[3] present an approach, SemRank, to rank the results of the query for semantic associations. Their method specifically focuses on adapting the ranking of relationships after determining the relative importance of relationships found with respect to a users context. This ensures that the same query made in different contexts and for different purposes does not yield the same ordering. This is achieved by measuring the likelihood that a user could have guessed the existence of the associations returned in results, called the predictability of the association. The ranking mechanism makes use of information theoretic techniques and the heuristics to determine the importance and relevance use semantic relationships.

(iii) Ranking Answers by Meta-Properties A third approach typically involves a variety of pageRank-like analyses of the structure of the Semantic Web, trying to locate which resources are more important, more authoritative, more trustworthy, etc. For instance, [6] introduce a ranking system that is dependent on a number of factors such as the number of triples relevant to the result, the importance of semantic web resources in triples, inverse property frequency of properties in triples and the effect of inference. [11] develop algorithms for ranking the importance of semantic web objects at three levels of granularity: documents, terms and RDF graphs. This algorithm is used for searching ontologies and ranking their importance.

(iv) Ranking of Complex Objects A fourth approach deals with ranking different kinds of objects, for example, ontologies, semantic web documents, services. This kind of ranking relies on a fairly sophisticated analysis of the object-to-be-ranked: the internal structure of the ontologies, semantic descriptions of the functionality of the services, etc. For example, [11] implemented an ontology search tool that uses AKTiveR-

ank to rank ontologies. The AKTiveRank [1] system aggregates a number of graph-analysis measures that use certain structural features of concepts, such as their hierarchical centrality, structural density and semantic similarity to other concepts. [47] propose an ontology QA system, OntoQA, that allows users to tune the ranking of ontologies towards certain features of ontologies to suit the needs of their applications. The ontologies are evaluated on two dimensions: Schema and instances. The first dimension evaluates the ontology design and its potential for rich knowledge representation. The second dimension evaluates the placement of the instance data within the ontology. Finally a cumulative score is calculated to rank the ontologies.

This brief attempt to categorize the very heterogeneous literature on ranking for the Semantic Web is summarized in Appendix A (p. 20f.). Our survey is very much in accordance with the recently appeared [24].

7. Discussion

The central problem addressed by this paper was: "Given a query to the Web of Data and the resulting answer set, how can we recognize the truth, i.e., separate correct from incorrect answers?" The primary contribution of our work was the design and test of the *cluster heuristic* — a simple, cognitively inspired heuristic that can separate correct from incorrect answers with a high degree of reliability. We first developed a set of benchmark questions with gold standard answers and translated them into corresponding SPARQL queries. By comparing the rating scores of a human judge with the results returned by our algorithm, we demonstrated that our heuristic yields highly promising results.

Secondary results of our work are (i) insights into the quality of the LOD cloud, and (ii) a publicly available benchmark of fifty general knowledge questions in natural language, rephrased in SPARQL, with with gold standard answers, and the 2197 answers returned when querying a significant subset of the LOD cloud, as well as a human ranking of the quality of these answers.

We now discuss some of the *cluster heuristic*'s current characteristics and limitations and conclude with open questions and possible extensions that point to promising avenues of future work.

7.1. Characteristics and Limitations

Some observations about the specific type of questions, queries, and data used in our experiment may help to elucidate the surprising success of our simple *cluster heuristic*.

Focus on Enumerative Questions All our questions (introduced in Section 3.1) asked for the enumeration of a set of objects of the same type (e.g., "highest summits", "U.S. presidents", "African countries", "members of ABBA", etc.). Although this constitutes an important class of questions (see [53] for applied examples) not all questions are of this kind. In particular, the *cluster heuristic* would not work for questions with only a single correct answer (e.g., "What is the age of President Obama?", or "Is Mont Blanc higher than the Matterhorn"). However, this is a limitation by design and we trust that other fast and frugal heuristics—like fluency, tallying, or take-the-best [15]—could be adopted to answer such questions.

Exploiting Natural Categories Due to the enumerative nature of our questions their correct answers are typically of a single type. This partly explains why the *cluster heuristic* works so well: Correct answers are similar to each other by belonging to the same *natural category* [34,35]. The success of our heuristic informally suggests that some structural aspect of human memory seems to be reflected in the Semantic Web. Natural categories seem to have played an important role in structuring the LOD sources that we were using (and which are in fact the most important LOD sources to date, such as DBPedia, Freebase, Geonames, etc.).

An interesting future research question would be to use an entirely different part of the Linked Data Cloud (e.g., the large amounts of knowledge available from the life-sciences) and test whether the *cluster heuristic* works equally well in a domain that is not as readily structured into natural types.

Exploiting Ontological Knowledge A hallmark of heuristics in general is that they can afford to be simple by exploiting some systematic structure in their environment [16]. The *cluster heuristic* is no exception in this respect. Although it knows nothing about ontologies, it still capitalizes on ontological knowledge in the data. All our queries were performed on FactForge under deductive closure, which means that all derivable properties have been turned into direct feature-value pairs that are used as input to the *cluster heuristic*. It is an interesting question for future research to query

FactForge without deductive closure, and to investigate how ontological derivations in the data have supported our heuristic's success.

Dynamic Developments in Facts, Queries, and Data Due to dynamic changes in the world and the rapid development of Semantic Web technologies our benchmark questions and the results and formulation of our queries are moving targets. Our experiments were performed in November 2011. Since then, the LOD cloud in general, as well FactForge itself, have undergone substantial changes. These changes involve not just a monotonic growth of the available information, but also renamed name spaces and changes in classes and their hierarchy. Consequently, the quantitative details of our results are subject to constant changes. Many of our queries now yield different results, and some will have to be rewritten to reflect changes in FactForge's vocabulary. Similarly, the correct answers to some of our questions (e.g., current sport teams in particular leagues) are subject to periodic changes. This renders our original standard [31] partially out-dated, but is an inevitable feature of realistic queries in Semantic Web contexts. In order to provide reproducible results as far as possible, we have made all our queries, answers, and similarity scores available online [30].

Despite these limitations, the results reported above are independent of the quality of the dataset, i.e., the completeness and correctness of FactForge. Crucially, both the human judge and the *cluster heuristic* ranked the answers that were returned after querying FactForge, and we were only comparing the correspondence between those rankings. Thus, if FactForge is incomplete (which it is) this only implies that some answers are missing from the list that is ranked by both human and computer. This does not affect our measurement of the correspondence between the *cluster heuristic* and the human ranking. An analogous argument holds for incorrect answers in FactForge.

Related results reported in [9] allow us to estimate the completeness of FactForge. When comparing the answers returned by FactForge against an expert-constructed gold standard set of answers for every query, FactForge contained only 60% of the answers deemed correct (without applying any ranking). In other words, 40% of the correct answers were simply not returned as a result of our queries to FactForge.

Similarly, as mentioned in Section 4.2, our results are also independent of the faithfulness of the natural language questions into SPARQL. Even though we have tested that this translation is indeed faithful (see Section 3.3), both human judge and the *cluster heuristic* rank the same results, so any incorrect answers that are possibly introduced by unfaithful translations into SPARQL should receive a low-rank by both of them.

7.2. Possible Extensions

In the spirit of the framework of fast and frugal heuristics [16] the *cluster heuristic* implements a very basic notion of semantic similarity. Whereas other approaches rely on complex constructs like the distance between classes in ontologies, e.g. [2,46,28], the *cluster heuristic* uses a feature-based measure of semantic similarity that dates back to [50] and has been explored extensively in cognitive science and other disciplines (see [17,10,37]).

It is encouraging that the simple metric based on predicate-object overlap that we defined in Section 2.2 yielded such promising results. By simply counting the number of shared feature-value pairs without assigning any weights or distances to features the *cluster heuristic* ignores most semantic information that is available in rich ontologies. The extreme simplicity of our heuristic is part of its attraction, and validates [16]'s claim that fast and frugal heuristics can perform well on sparse information. Our implementation and findings can also be counted as an empirical validation of Van Rijsbergen's [52] *cluster hypothesis*, which states that correct answers tend to cluster together. While originally intended for textual information retrieval, our results show that the hypothesis also holds when answering queries over LOD.

As our original choice of metric was one of many plausible instantiations of Tversky's [50] similarity measure it is possible that alternative metrics could fare even better. We now discuss some variants and extensions of our measure that seem promising at this point. An explicit test of the *cluster heuristic* against alternative and semantically more informed heuristics is an interesting piece of future work.

Alternative Simple Metrics of Semantic Similarity The *cluster heuristic* implemented a measure of semantic similarity based on predicate-object overlap (see Definition 1 in Section 2.2). Two alternative and less restrictive measures could define similarity in terms of mere predicate or object overlap, i.e., regard either the predicates p or objects o of a resource s as its features and entirely disregard the other component. The intuition behind such an approach would be that correct or relevant resources resemble each other

if they share the same predicates or objects. Formal versions of these notions and corresponding SPARQL queries can easily be defined in analogy to those in Section 2.2).

Rather than specifying a single similarity metric, Tversky's [50] feature-based notion of similarity defines a family of scales that are characterized by different values of the parameters (θ, α, and β in Equation 1) and frequently expressed as a parameterized ratio model:

$$S_{ratio}(X,Y) = \frac{\theta \cdot f(X \cap Y)}{\theta \cdot f(X \cap Y) + \alpha \cdot f(X-Y) + \beta \cdot f(Y-X)} \quad (7)$$

Normalizing this measure to $0 \leq S(X,Y) \leq 1$ and setting $\theta = \alpha = \beta = 1$ yields the so-called *Jaccard index* and setting $\alpha = \beta = 1/2$ yields *Dice's coefficient* of similarity.

A larger departure from these symmetric models are asymmetric models that put more weight on one of the objects than on the other one ($\alpha \neq \beta$). For instance, the moon may be considered to be more similar to the earth than vice versa. An extreme version of such asymmetry is defined by $\alpha = 1$ and $\beta = 0$. In this case, Equation 7 yields the *degree of inclusion* for X.

Interestingly, [52] advised against the use of any measure not normalized by the length of the document vectors, something that was experimentally verified (see [48, ref to Willet (1983), p. 620]). In our experiments, we omitted to normalize by the total number of features defined for a given object in the answer set. Apparently, this did not prevent the *cluster heuristic* from performing well, but it remains an interesting question to perform our experiments using some of the alternative measures mentioned in this section. Our initial explorations with normalized and asymmetric variants suggest that the results obtained by the *cluster heuristic* are robust under different simple metrics of semantic similarity.

Alternative Approaches to Semantic Similarity The success of the *cluster heuristic* demonstrates the effectiveness of a simple feature-based notion of semantic similarity. But beyond Tversky's [50] contrast model there are other candidate approaches towards semantic similarity to consider. Before capitalizing on the rich ontological knowledge contained in the class hierarchy of LOD it could be interesting to not only count the *number* of shared feature-value pairs, but to more closely inspect *which* features are being shared. Informal inspection of some of our results suggests that some properties are more predictive of object-similarity (and hence the ranking of correctness through clustering) than others. Again, this would represent a departure from the very simple *cluster heuristic* that we have employed here.

A different family of measures are based on geometric measures of semantic relatedness, e.g. latent semantic analysis [12]. To explore their potential, [33] measured the similarity between answers using a geometric semantic space based on Wikipedia articles. By using queries much like the ones we used (e.g., enumerate curries, teas, and owls) their tests showed that when similarity measures are grounded in semantic spaces, Van Rijsbergen's *cluster hypothesis* holds for answers to SPARQL queries. The answer to the "tea" query, for example, indeed showed that teas tended to be more similar to each other than many of the erroneously retrieved answers, such as the "Boston Tea Party".

More systematic explorations of such alternative approaches to semantic similarity will help us better understand the robustness and boundaries of the *cluster hypothesis*, and specific instantiations of the *cluster heuristic*. Given the current state of LOD, gaining further insights into the potential of similarity measures to rank and filter out incorrect query results in different domains and applications is an important challenge for future research.

7.3. Conclusion

Our work on the *cluster heuristic* demonstrates the potential for fast and frugal heuristics to solve important real-world problems posed by the Semantic Web. In all likelihood, except for the most meticulously curated repositories, LOD will continue to include incorrect and inconsistent information for the foreseeable future. To some extent, the shambolic state of LOD undoubtedly reveals the haphazard way some people enter data. But before we try to root out all these errors, we should consider that they might have positive, unintended consequences. These errors may help LOD repositories capture interesting ambiguities in the world. For instance, Stig Anderson, ABBA's manager, really could be a considered a member of the band. Similarly, several individuals — including the producer George Martin and the drummer Pete Best — were often called "the fifth Beatle". As LOD continues to grow, rather than using the *cluster heuristic* to identify unambiguously correct answers, we may end up using it to shape what we take to be the correct answers.

Acknowledgements This research was funded under FP7 project LarKC, nr. 215535 and by the Max Planck Society. The literature review section is based on work by our student Ravindra Harige.

References

[1] Harith Alani, Christopher Brewster, and Nigel Shadbolt. Ranking ontologies with AKTiveRank. In Isabel Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Mike Uschold, and LoraM. Aroyo, editors, *The Semantic Web - ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 1–15. Springer Berlin Heidelberg, 2006.

[2] Harith Alani, Christopher Brewster, and Nigel Shadbolt. Ranking ontologies with AKTiveRank. In *The Semantic Web ISWC 2006*, pages 1–15. Springer, 2006.

[3] Kemafor Anyanwu, Angela Maduko, and Amit Sheth. SemRank: Ranking complex relationship search results on the Semantic Web. In *Proceedings of the 14th international conference on World Wide Web*, pages 117–127. ACM, 2005.

[4] Kemafor Anyanwu and Amit Sheth. The ρ operator: Discovering and ranking associations on the Semantic Web. *ACM SIGMOD Record*, 31(4):42–47, 2002.

[5] Kemafor Anyanwu and Amit Sheth. P-Queries: Enabling querying for semantic associations on the Semantic Web. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 690–699, New York, NY, USA, 2003. ACM.

[6] Bhuvan Bamba and Sougata Mukherjea. Utilizing resource importance for ranking Semantic Web query results. In Christoph Bussler, Val Tannen, and Irini Fundulaki, editors, *Semantic Web and Databases*, volume 3372 of *Lecture Notes in Computer Science*, pages 185–198. Springer Berlin Heidelberg, 2005.

[7] Barry Bishop, Atanas Kiryakov, Damyan Ognyanov, Ivan Peikov, Zdravko Tashev, and Ruslan Velkov. FactForge: A fast track to the web of data. *Semantic Web*, 2(2):157–166, 2011.

[8] Christian Bizer and Andreas Schultz. The Berlin SPARQL benchmark. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(2):1–24, 2009.

[9] Arjon Buikstra, Hansjörg Neth, Lael Schooler, Annette ten Teije, and Frank van Harmelen. 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)*. 22nd International Joint Conference on Artificial Intelligence (IJCAI-11), Barcelona, Spain, 2011.

[10] V. Cross. Tversky's parameterized similarity ratio model: A basis for semantic relatedness. In *Annual Meeting of the North American Fuzzy Information Processing Society, NAFIPS 2006.*, pages 541–546. IEEE, 2006.

[11] Li Ding, Rong Pan, Tim Finin, Anupam Joshi, Yun Peng, and Pranam Kolari. Finding and ranking knowledge on the semantic web. In Yolanda Gil, Enrico Motta, V.Richard Benjamins, and MarkA. Musen, editors, *The Semantic Web âĂŞ ISWC 2005*, volume 3729 of *Lecture Notes in Computer Science*, pages 156–170. Springer Berlin Heidelberg, 2005.

[12] S T Dumais and T K Landauer. A solution to Platos problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104:211–240, 1997.

[13] D. Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170, 1983.

[14] D. Gentner and A. B. Markman. Structure mapping in analogy and similarity. *American Psychologist*, 52(1):45–56, 1997.

[15] G. Gigerenzer, R. Hertwig, and T. Pachur, editors. *Heuristics: The foundations of adaptive behavior*. Oxford University Press, New York, NY, 2011.

[16] G. Gigerenzer, P. M. Todd, and the ABC research group. *Simple heuristics that make us smart*. Oxford University Press, New York, NY, 1999.

[17] R. L. Goldstone and J. Y. Son. Similarity. In *The Cambridge Handbook of Thinking and Reasoning*, pages 13–36. Cambridge University Press, New York, NY, US, 2005.

[18] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. LUBM: A benchmark for OWL knowledge base systems. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2):158–182, 2005.

[19] U. Hahn, N. Chater, and L. B. Richardson. Similarity as transformation. *Cognition*, 87(1):1–32, 2003.

[20] Xin He and Mark Baker. xhRank: Ranking entities on the Semantic Web. In *9th International Semantic Web Conference (ISWC2010)*, November 2010.

[21] C. J. Hodgetts, U. Hahn, and N. Chater. Transformation and alignment in similarity. *Cognition*, 113(1):62–79, 2009.

[22] Carlos A. Hurtado, Alexandra Poulovassilis, and PeterT. Wood. Ranking approximate answers to semantic web queries. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero HyvÃűnen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Simperl, editors, *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 263–277. Springer Berlin Heidelberg, 2009.

[23] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[24] V. Jindal, S. Bawa, and Sh. Batra. A review of ranking approaches for semantic search on web. *Information Processing and Management*, in press, 2013.

[25] D. Kahneman, P. Slovic, and A. Tversky. *Judgment under uncertainty: Heuristics and biases*. Cambridge University Press, Cambridge, UK, 1982.

[26] Atanas Kiryakov, Damyan Ognyanov, and Dimitar Manov. OWLIM: A pragmatic semantic repository for OWL. In *Web Information Systems Engineering (WISE 2005) Workshops*, pages 182–192. Springer, 2005.

[27] Vanessa Lopez, Andriy Nikolov, Miriam Fernandez, Marta Sabou, Victoria Uren, and Enrico Motta. Merging and ranking answers in the semantic web: The wisdom of crowds. In AsunciÃşn GÃşmez-PÃl'rez, Yong Yu, and Ying Ding, editors, *The Semantic Web*, volume 5926 of *Lecture Notes in Computer Science*, pages 135–152. Springer Berlin Heidelberg, 2009.

[28] Vanessa Lopez, Andriy Nikolov, Miriam Fernandez, Marta Sabou, Victoria Uren, and Enrico Motta. Merging and ranking answers in the semantic web: The wisdom of crowds. In *The semantic web*, pages 135–152. Springer, 2009.

[29] Li Ma, Yang Yang, Zhaoming Qiu, Guotong Xie, Yue Pan, and Shengping Liu. Towards a complete OWL ontology benchmark. In *Proceedings of the 3rd European conference on The Semantic Web: Research and applications (ESWC 2006)*, pages

125–139, Budva, Montenegro, 2006. Springer-Verlag.

[30] Hans-Jorg Neth, Arjon Buikstra, Lael Schooler, Annette ten Teije, and Frank van Harmelen. Linked open data q/a benchmark. FigShare, http://dx.doi.org/10.6084/m9.figshare.882847, 2013.

[31] Hansjörg Neth, Lael J Schooler, Jose Quesada, and Jörg Rieskamp. Analysis of human search strategies. LarKC project Deliverable 4.2.2. Technical report, The Large Knowledge Collider (LarKC), 2009.

[32] Judea Pearl. *Heuristics: Intelligent search strategies for computer problem solving*. Addison-Wesley Inc., Reading, MA, 1984.

[33] Jose Quesada, Stefan Otte, Ralph Brandao-Vidal, Lael J. Schooler, John Wong, and HansjÃűrg Neth. Subsetting by statistical semantics. Unpublished data., 2011.

[34] Eleanor H Rosch. Natural categories. *Cognitive Psychology*, 4(3):328–350, 1973.

[35] Eleanor H Rosch, Carolyn B Mervis, Wayne D Gray, David M Johnson, and Penny Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology*, 8(3):382–439, 1976.

[36] Pranab K. Sen, Ibrahim A. Salama, and Dana Quade. SpearmanâĂŹs footrule: Asymptotics in applications. *Chilean Journal of Statistics*, 2(1):3–20, 2011.

[37] B. Sheehan, A. Quigley, B. Gaudin, and S. Dobson. A relation based measure of semantic similarity for Gene Ontology annotations. *BMC Bioinformatics*, 9(1):468–493, 2008.

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

[39] R. N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. i. *Psychometrika*, 27(2):125–140, 1962.

[40] Herbert Alexander Simon. A behavioral model of rational choice. *The Quarterly Journal of Economics*, 69(1):99–118, 1955.

[41] Herbert Alexander Simon. Rational choice and the structure of the environment. *Psychological Review*, 63(2):129–138, 1956.

[42] Herbert Alexander Simon. *The Sciences of the Artificial*. The MIT Press, Cambridge, MA, 3rd edition, 1996.

[43] C. Spearman. Footrule for measuring correlation. *The British Journal of Psychiatry*, 2:89–108, 1906.

[44] Nenad Stojanovic, Rudi Studer, and Ljiljana Stojanovic. An approach for the ranking of query results in the semantic web. In Dieter Fensel, Katia Sycara, and John Mylopoulos, editors, *The Semantic Web - ISWC 2003*, volume 2870 of *Lecture Notes in Computer Science*, pages 500–516. Springer Berlin Heidelberg, 2003.

[45] Samir Tartir and I. Budak Arpinar. Ontology evaluation and ranking using OntoQA. In *Proceedings of the International Conference on Semantic Computing*, ICSC '07, pages 185–192, Washington, DC, USA, 2007. IEEE Computer Society.

[46] Samir Tartir and I Budak Arpinar. Ontology evaluation and ranking using OntoQA. In *International Conference on Semantic Computing, ICSC 2007*, pages 185–192. IEEE, 2007.

[47] Edward Thomas, Harith Alani, Derek Sleeman, and Christopher Brewster. Searching and ranking ontologies on the Semantic Web. In *Workshop on Ontology Management: Searching, Selection, Ranking, and Segmentation. 3rd K-CAP*, 2005.

[48] A. Tombros and C J Van Rijsbergen. Query-sensitive similarity measures for information retrieval. *Knowledge and Information Systems*, 6(5):617–642, 2004.

[49] Anastasios Tombros and Cornelis J van Rijsbergen. Query-sensitive similarity measures for the calculation of interdocument relationships. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, pages 17–24. ACM, 2001.

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

[51] A. Tversky and D. Kahneman. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–1131, 1974.

[52] C J Van Rijsbergen. *Information Retrieval*. Butterworths, London, UK, 2nd edition, 1979.

[53] Antony J Williams, Lee Harland, Paul Groth, Stephen Pettifer, Christine Chichester, Egon L Willighagen, Chris T Evelo, Niklas Blomberg, Gerhard Ecker, Carole Goble, et al. Open PHACTS: Semantic interoperability for drug discovery. *Drug Discovery Today*, 17(21):1188–1198, 2012.

Appendix

A. Literature Survey

Metrics & Paper Ref	Anyanwu (2003)	He (2010)	Anyanwu (2005)	Hurtado (2009)	Stojanovic (2003)	this paper
Paper Title Keywords	ρ-Queries	xhRank	SemRank	Approximate Answers	Ranking of Query Results	Cognitive heuristic
Type of Ranking	RELEVANCE	RELEVANCE	RELEVANCE, IMPORTANCE	RELEVANCE	ENTITIES	SEMANTIC SIMILARITY
Compared Objects	Predicates & relations		Predicates & relations	instances	Query results	Instances
Problem Area	How to finding complex yet meaningful and obscured relationships between entities? (Pre-SPARQL problem)	Classical IR searching & ranking approaches are not suitable to semantic web data and it does not cover all aspects of ranking entities in SW: importance, relevance and query length.	In existing relationship search systems, relationship search made in different contexts returns results in same ordering. How to adapt the ranking based on the context?	How to assist users querying semi structured data such as RDF without knowing its structure?	Classical IR searching and ranking approaches are not suitable to semantic web data. How to improve the searching and ranking the semantic web query result?	How to select the correct answers to a query from among the partially incorrect answer sets that result from querying the Web of Data
Approach	Formalizes the RDF data model with notion of property sequence – that capture paths in RDF. p-queries are performed on property sequences.	Proposes a ranking mechanism, which includes three categories of rankings: importance, relevance and query length tailored to SW data	Modulated Relevance model, uses Information theoretical techniques and heuristics to rank semantic association	Proposes a framework that performs an approximate matching of users query to the data and rank the answers in terms of how closely they match the original query. RDF data is modeled as graph structure and query language is restricted to that of conjunctive regular path queries .	Proposes a novel approach for determining relevance in ontology-based searching for information, which exploits the "full potential" of the semantics of such a semantically-based link structure	Proposes cognitively inspired similarity measures which can be exploited to filter the correct answers from the full set of answers
Dataset	NA: Evaluation not done	NA: Short paper	Synthetically generated data – schemas and resources instances of the schema	NA: Algorithms and evaluation techniques are proposed but not implemented.	Data from the semantic portal of the institute of authors.	FactForge (includes DBPedia, Freebase, Geonames, UMBEL,WordNet, the CIA World Factbook, MusicBrainz)
Method	A set of binary relations on the domain of resources are defined based on different types of Property Sequences. A notion of p-Query is defined as a set of operations that map from a pair of keys to the set of PS. Two strategies for processing p-queries are proposed: a query processing layer on top of RDF storage layer and b) using graph traversing algorithms for processing memory resident graph representation of RDF	To address 3 aspects of ranking resource in SW, xhRank employs different ranking algorithms and then compute the final ranking. Relevance ranking is achieved by a) phrase-level ranking b) graph-level ranking. Importance-based ranking is addressed by: resource (node) ranking and property (edge) ranking. Query length based ranking is used to evaluate a node within a graph against an user input.	Uses concept of information gain from information theory to build a model for measuring the information content of a semantic association. Defines a) measure of refraction, which is a deviation of semantic association path's representation at the schema layer b) semantic match measure to find associations that match the keywords supplied by user to augment the association search query. The cumulative score of these three measures are used to compute final SemRank value.	Proposed framework allows for approximate matching of queries on graphs using conjunct queries with weighted regular transducers to model the approximations. The model also incorporate standard notions of approximation such as edit distance as well as certain RDFS inference rules, thereby capturing semantic as well as syntactic approximations.	Ranking method, called as ontology-based-ranking, combines characteristics of the inferencing process and the content of the information repository used in searching.	Fast and Frugal Heuristics
Query Example	How is *Resource A* related to *Resource B*?For example, a security agency may want to find any relationship between a terrorist act and a terrorist organization or country known to support such activity.	Different queries to demonstrate each method	a) In an investigative context the focus of a search may be to uncover obscure relationships between entities b) In an conventional search, the purpose is to find commonly expected relationship for the purpose of validating/augmenting known information.	The RDF graph is assumed involving information about transport network, where nodes=(cities/name of city) & edge=mode of transport(train/nus/airplane).User Query: find the cities which we can travel to city X using only airplane as well as city Y using only trains or busses.	Find researcher who researchIn X. E.g.: Institute example. Researcher (worksIn|researchIn|teaches) (Project|topic|lecture). KB defined.Find researcher who researchIn X. there are multiple answers but some are more relevant than others. Thus ranking is required.	Enumerative type questions: "Name members of the pop band ABBA?

Metrics & Paper Ref	Bamba (2005)	Ding (2005)	Alani (2006)	Thomas (2005)	Tartir (2007)	Lopez (2009)
Paper Title Keywords	Utilizing Resource	Finding and Ranking	ATKive Ranking	Searching and Ranking	OntoQA	Merging and Ranking Answers
Type of Ranking	IMPORTANCE	IMPORTANCE, ENTITIES	ENTITIES	ENTITIES	ENTITIES	SEMANTIC SIMILARITY
Compared Objects	Instances	ontologies	ontologies	ontologies	ontologies	Query results
Problem Area	How traditional IR/www link analysis techniques can be improvised to calculate importance of semantic web resources?	How navigating Semantic Web documents on the Web can be made simpler?	How to rank the relevant ontologies for the user query for ontology?	How to search and rank the relevant ontologies for the user query for ontology?	How to rank the relevant ontologies for the user query for ontology?	How to rank the results of the query executed by QA systems which rely on freely available SW knowledge repositories? Considering issues of merging redundant information from different SW sources.
Approach	The approach to rank the results of semantic web query are based on parameters like the number of triples relevant to the result, the importance of the semantic web resource in the triples, the inverse property frequency of the properties in triples and the effect of inference.	Developed algorithms for ranking the importance of Semantic Web objects at three levels of granularity: documents, terms and RDF graphs	Proposes a prototype of an ontology ranking system which applies a number of analytic methods to rate each ontology based on how well it represents the given search terms.	Makes use of AKTiveRanking \cite{10} that ranks ontologies using certain features of concepts, such as their hierarchical centrality, structural denisity, and semantic similarity to other concepts of intetest	Rank ontologies based on their contents and relevance to a set of given keywords and preferences, uses ATKive Ranking\cite{10}	Proposes merging algorithm that aggregates, combines and filters ontology-based search results and three different ranking algorithms that sort the final answers according to different criteria such as popularity, confidence and semantic interpretation of results.
Dataset	Experiment evaluated on the proprietary dataset of biomedical patents	Index of Semantic Web documents published on the web (FOAF/RSS)	AKTiveRanking was used on the ontology retrieved from search query to Swoogle sytem.	Uses Google API and local repository of ontologies	Top 9 RDF and OWL ontologies ranked by Swoogle when searched for “Paper”.	High- level ontologies, e.g., ATO, TAP, SUMO, DOLCE and very large ontologies, e.g., SWETO or the DBPedia Infoboxes
Method	Use of graph structure to represent RDF data. The connectivity node (semantic web resource) with other nodes in graph data determines the importance of the semantic web resource. Kleinberg’s hub and authority scores are used to determine the connectivity of nodes and subjectivity/objectivity scores of SW resource.	Swoogle’s OntoRank ranking model is a variation of PageRank ranking system and it works at the document level. Thus the popularity of SW documents bias the ranking. For e.g., SW documents which are ontology (like OWL) are ranked higher than other RDF documents because they are referred to by many SW documents. Swoogle identifies different relationships /paths between SW documents, terms and ontology and then applies link analysis based ranking method to compute the final rank.	ATKiveRanking ontology ranking applies 4 measures to evaluate different representational aspects of the ontology and calculate ranking. Each ontology is evaluated separately. Once those measures are all calculated for all ontologies, the resulting values are merged to produce total rank for the ontology. The 4 measures are: Class Match measure, density measure, semantic similarity measure, Betweenness measure.	User query is executed on local repository of ontologies. If local information is not present the query is executed by Google API to return the ontological information from Semantic Web. The results are analyzed and added to local repository if not already present. Then ATKiveRanking is used to rank the ontologies. The ranking depends on the weights assigned to each metrics of ATKiveRanking by user.	The ontology evaluation is done on two dimensions: Schema and Instances	Merging algorithm considers different scenarios for the result received from each KB, e.g. valid but duplicate answer/part of composite answer/alternative answer derived from different ontological interpretations – and applies 3 types of operators over set of retireived results: union, intersection, and condition to merge the results. Then 3 types of ranking are employed to sort the final list of answer for the query. They are: ranking by semantic similarity, confidence and popularity.
Query Example	In BioMedical Patent Semantic Web data – find the inventor and assignee pairs who have a patent which has a term belonging to the UMLS class, say for e.g., Molecular_Funcation class.	Search for ‘person ontology’	Search result for ‘student university’ as returned by Swoogle search system.	For searching structures: N/Triple based query representation is used to describe sub graphs which might be present in ontology. For searching classes: keywords are used to match class information	The ontology evaluation is based on metrics defined on the two dimensions: Schema and Instances. Finally a score is computed based on the values on the metrics which is then used to rank the ontologies. Each metric has an associated weight, which can be tuned by the user to bias the ranking based on user preferences.	“Which languages are spoken in Islamic countries?”

B. General Knowledge Questions

1 Name the signs of the (Western) *zodiac*
4 Name the *planets* of our solar system
5 Name types of *grains* or *cereals* (genera)
6 Name species of *cats* (felidae)
7 Name genera of *pine trees* (Pinaceae)
8 Name *beer brands* with over $1 billion in sales (2008)
9 Name *technology brands* with a value exceeding $5 billion (2009)
10 Name *oil companies* with a value exceeding $700 million (2009)
11 Name *Coffee brands* with a value exceeding $600 million (2009)
13 Name products in the *Microsoft Office* Suite
14 Name *Apple* products (hard- or software) (2009)
15 Name *US-American car manufacturers* (current)
16 Name *German car manufacturers* (current)
17 Name *Japanese car manufacturers* (current)
20 Name *Italian pasta* varieties
23 Name *African countries* (nations)
24 Name *American countries* (nations)
28 Name *U.S.-American states* (current)
29 Name *capital cities* of U.S.-American states (current)
30 Name *continents* on earth
31 Name the *highest mountain* (peaks) of each continent
32 Name *countries* with a population exceeding 80 million (as of 2009)
33 Name the *German Bundesländer* (Federal States of the current Federal Republic of Germany)
34 Name the *capital cities* of German federal states (current)
35 Name *Nobel laureates in literature* (since 1945)
36 Name the feature *films by David Lynch*
37 Name the main characters of the first *Star Wars* movie (1977)
38 Name *James Bond movies* (titles)
39 Name the members of the pop band *The Beatles*
40 Name the musical *instruments of a symphonic orchestra*
41 Name the members of the pop band *ABBA*
42 Name the Olympian gods of the *Greek* mythology
43 Name the Olympian gods of the *Roman* mythology
44 Name the *German chancellors* (Federal Republic of Germany since 1949)
45 Name the *U.S.-American presidents* (since 1945)
46 Name the *foreign ministers of Germany* (Federal Republic since 1951)
47 Name the current member states of the UN Security Council (2009)
48 Name the laureates of the *Nobel Peace Prize* (since 1975)
49 Name *Wimbledon winners* for women's or men's singles (since 1980)
50 Name the teams of the *1st German football league Bundesliga* (2008/09)
51 Name the sites (host cities) of the modern *Olympic Winter games* (so far)
52 Name the sites (host cities) of the modern *Olympic Summer games* (so far)
53 Name the teams of the *NBA* (National Basketball Association in 2009)
54 Name the teams of the 2nd German football league *2. Bundesliga* (2008/09)
55 Name the types of *chess pieces*
56 Name the genera of *European broadleaf trees*
57 Name meat products offered by *McDonalds*
58 Name the *operas* by *Wolfgang Amadeus Mozart*
59 Name existing *brands* worth over $20 billion (in 2009)
60 Name existing *fashion brands* worth over $1 billion (in 2009)