Ad-hoc Object Retrieval in the Web of Data

Jeffrey Pound*
David R. Cheriton School of Computer Science
University of Waterloo
jpound@cs.uwaterloo.ca

Peter Mika
Yahoo Research
Barcelona, Spain
pmika@yahoo-inc.com

Hugo Zaragoza
Yahoo Research
Barcelona, Spain
hugoz@yahoo-inc.com

ABSTRACT

Semantic Search refers to a loose set of concepts, challenges and techniques having to do with harnessing the information of the growing Web of Data (WoD) for Web search. Here we propose a formal model of one specific semantic search task: *ad-hoc* object retrieval. We show that this task provides a solid framework to study some of the semantic search problems currently tackled by commercial Web search engines. We connect this task to the traditional *ad-hoc document retrieval* and discuss appropriate evaluation metrics. Finally, we carry out a realistic evaluation of this task in the context of a Web search application.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval—*Retrieval models*

General Terms

Experimentation, Measurement, Performance

Keywords

semantic search, evaluation, object retrieval

1. INTRODUCTION

In recent years the field of web search has diversified, bringing new challenges beyond the traditional text-based search problem studied by information retrieval researchers. Among these new paradigms is the field of semantic search, in which knowledge bases are used as the primary information source for search, or as a complement to text retrieval.

The last two years in particular has seen an increase both in the number of knowledge bases published as Linked Data[1] (such as Freebase[2], DBpedia[3], and others) and in the availability of metadata embedded inside web pages, thanks to the support for metadata standards by major commercial search engines (e.g. Yahoo's SearchMonkey[4] and Google's Rich Snippets[5] projects). Conservative estimates put the amount of structured data on the Web at over 100 billion triples today. The increase of size in the data that is handled first led to significant research on scalable indexing techniques with much input from the database community [1, 18, 26] and at the same time put a focus on the ranking of results. Ranking is also motivated by a growing number of end-user application scenarios where queries are given by ordinary users as keywords, and not in formal query languages such as SPARQL.

Despite the recent surge in semantic search research, there has been little work focusing on principled evaluation techniques for assessing the effectiveness of semantic search systems. In order to develop the specific components of semantic search technologies, as well as compare one system to another, a common evaluation methodology is needed. The lack of a common methodology specific to semantic search has also been identified by the participants of the Semantic Search workshop series [28] as one of the major stumbling blocks to future development of this research area.

Most current approaches to evaluating semantic search systems are adaptations of document evaluation techniques from the information retrieval community (e.g., [9]). In this setting, semantic search systems ultimately perform document retrieval, and the quality of documents returned is used as a metric of the quality of the entire system. These results make it difficult to interpret how well a semantic search system functions internally.

We make the following contributions.

- We define the *ad-hoc object retrieval task* for evaluating semantic search systems on the Web of Data. The task is based on answering arbitrary information needs related to particular aspects of objects, expressed in unconstrained natural language and resolved using a collection of structured data. We also explain the problems in mapping the current methodology for ad-hoc document retrieval to ad-hoc object retrieval.
- We analyse a real world web query log from a semantic search point of view, and propose a semantic search query classification based on this analysis.
- We propose a methodology for ad-hoc semantic search evaluation. Along the way, we discuss many of the difficult decisions that need to be made when designing a semantic search evaluation framework and the

*Work done while at Yahoo Research, Barcelona.
[1]http://www.linkeddata.org
[2]http://www.freebase.com
[3]http://dbpedia.org
[4]http://developer.search.yahoo.com/start
[5]http://googlewebmastercentral.blogspot.com/2009/05/introducing-rich-snippets.html

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

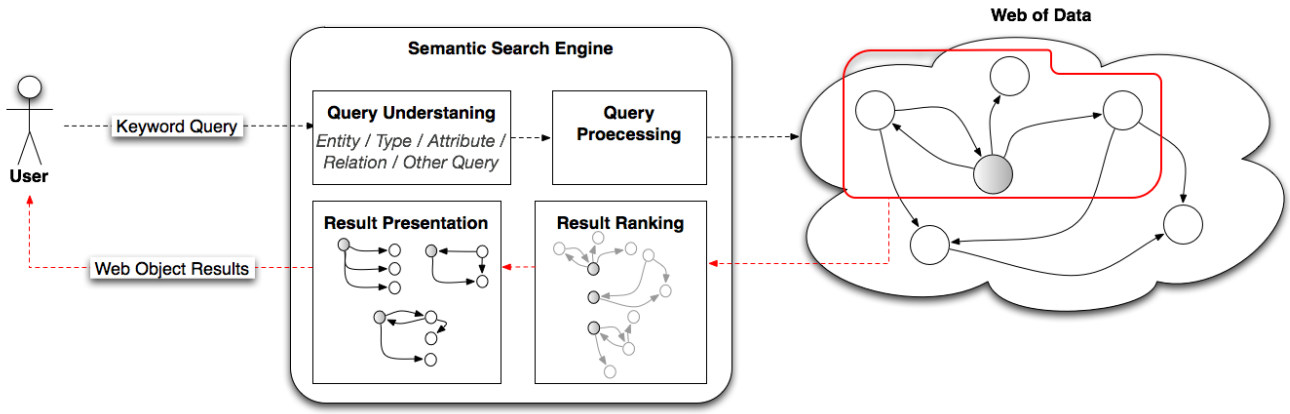

Figure 1: Overview of the semantic search process. A keyword query is used to identify relevant web objects. Relevant objects are ranked, and for each object a set of connected objects is selected as the result.

impact those decisions can have on the quality of the evaluation and the reusability of human judgments.

- We deploy our proposed evaluation methodology on a real world data set and query workload. We show experimentally that our proposal is stable for some popular evaluation metrics, and that it can reliably distinguish among the effectiveness of systems with these metrics.

1.1 Background

It is common to call any information retrieval system a semantic search system if it performs the matching of queries and potential results at a conceptual level, i.e. by processing information beyond the surface forms of symbols. Unfortunately, this definition is hard to operationalize when it comes to semantic search evaluation, because it leaves open the type of queries processed by the system, the format of the content and the internal knowledge representation paradigm applied within the search system.

To arrive at a comparable subset of search systems we will have to restrict the above definition to systems that retrieve data from a knowledge base containing RDF data [17]. RDF is the core part of the Semantic Web stack and defines the abstract data model for the Semantic Web in the form of triples that express the connection between web resources and provide property values describing resources. (Since many knowledge formalisms can be encoded as sets of triples, this defines a large class of systems which we will focus on in the remainder of the paper.) Broadly speaking, the goal of retrieval in this context is to retrieve parts of the knowledge base (sets of triples or components of triples) that best match the query entered by the user.

Note that "resources" in RDF describe "objects" in a Web of Data. Since the Semantic Web is built upon such data models, it is also equivalent to think of these objects as Semantic Web "entities". We will use each of these notations in our discussion depending on the context.

Given the structured nature of the data, semantic search engines also bring new query capabilities to search. In principle, each new query type afforded by the search engine represents specific functionality that could be evaluated separately. In practice, not all features of any structured query language would be equally necessary to perform common tasks. Further, we expect that users will be limited by the capabilities of the input interface; in particular, we expect keyword queries to remain the prevalent form of input. To establish a realistic evaluation that covers the most commonly occurring query types, we will perform a study of keyword queries in a web search query log from the perspective of the semantic query functionality required to resolve the query. This will lead us to a simplified taxonomy of web queries as well as a measure of the commonality of each type of query, allowing semantic search engines to be evaluated on realistic work loads.

Figure 1 illustrates a generic architecture for semantic search systems that perform object retrieval for keyword queries. In this setting, a keyword query is used to identify relevant web objects occurring in the WoD. Relevant objects are ranked based on their content and relationships to other objects. Finally, each object is *explained* by expanding a set of connected objects to provide the final result.

1.2 Organization

The remainder of the paper is organized as follows. In Section 2 we outline the ad-hoc object retrieval task and discuss the challenges and choices made in adapting the ad-hoc document retrieval task. Section 3 reports on the results of an analysis of a web query log which is then used to conduct an evaluation of a base-line object ranking algorithm using our proposed evaluation task. In Section 4 we contrast our approach with related work. We conclude in Section 5 and discuss future work.

2. AD-HOC OBJECT RETRIEVAL TASK

Because Semantic Search is a relatively new problem, there have been no formalizations of standard tasks or evaluation methodologies in order to assess the quality of object retrieval on the Web. In this work we try to map an object ranking problem to the well known problem of *ad-hoc* document retrieval (ADR). Formalization of the ADR task and

of its evaluation opened up the way for research in ranking models in Information Retrieval. By providing a clear problem setting, it allowed the creation of standard metrics and collections, allowing researchers in IR to communicate and collaborate. This in turn yielded many fundamental developments in ranking models, evaluation methods, efficiency and user models.

An overview of the ADR task can be found in Chapter 1 of [16], we summarized the task as follows:

- INPUT: a user query (a keyword query, without structure) q, a collection of documents C.
- OUTPUT: a ranked list of document identifiers $o = (r_1, r_2, ..., r_k)$, where $d_{r_i} \in C$.
- EVALUATION: each document d_{r_i} is labeled with a score (independently of the rest) by a judge with respect to the information need originating the query q.

Although ADR was initially developed for research in information science and library search, it anticipated web search. Indeed ADR models well the typical situation for a user searching the web: the user types a keyword query q, and expects a ranked list of snippets o which represent web pages (documents). For this reason, web search engines from their infancy built on all the advances in ADR research, and adopted ADR metrics.

However, the Web is no longer simply a collection of web pages. As the WoD grows, we see objects of different types surfacing in web search results. For example, for queries with a strong "local" interpretation (such as "Pizza in New York"), the first results of commercial web search engines are typically constructed from structured objects and no longer from web pages. For such applications, web search engines need to rank all the candidate objects.

For the analysis of such tasks, we propose to modify the traditional ADR task. Intuitively we would like to replace documents by objects, leaving everything else the same. However, this brings up several non-trivial issues that need to be dealt with. These issues in turn will restrict our definition of the object ranking task.

The ad-hoc object retrieval task (AOR) is defined as follows.

- INPUT: a user query (a keyword query, without structure) q which has query type t and query intent z, and a data graph G.
- OUTPUT: a ranked list of resource identifiers $o = (o_1, o_2, ..., o_k)$ such that each o_i occurs in G.
- EVALUATION: each object (or resource) o_i is labeled with a score (independently of the rest) by a judge with access to all the information contained in or linked to by o_i, with respect to the query q, query type t, and the query intent z.

In the following sections, we highlight two of the key differences in evaluating AOR and ADR: the dependence between the type of the query and the type of result returned, and the larger role played by result presentation. We then discuss metrics applicable to AOR.

2.1 Query Types

In ADR, a human judge must evaluate the relevance of a document to a query. Could a human judge evaluate the relevance of an object to a query? It would seem so intuitively. For example, if one queries for the name of a person, then any hCard (a micro-format encoding information similar to a business card) exactly matching the query would be judged relevant, and other hCard matching the name partially would be judged related but not as relevant. However, as we push our investigation of object relevance further, we quickly run into several problems. In particular, results may be connected to relevant objects, but not directly relevant on their own. Also, the expectation of what a result is may vary depending on the query.

Our first step towards studying this problem was to manually analyse a real web search query log, and try to understand how object retrieval in the Web of Data could improve upon these answers, if at all. For each query in the query log, we manually annotated any entities, entity types, or relations occurring among the query terms (see Figure 2). Furthermore we annotated which was the primary intent of the query.

In our web search log analysis we identified several query categories that would require different treatment in an AOR engine. We established five such categories (see examples in Figure 2):

- Entity query: the intention of the query is to find a particular entity. Correct results would be entities corresponding to some disambiguation of the query entity.
- Type query: the intention of the query is to find entities of a particular type or class. Correct results would be entities that are instances of the specified type, or an identifier of the type itself.
- Attribute query: the intention of the query is to find values of a particular attribute of and entity or type. Correct results would be the values of an attribute specified in the query.
- Relation query: the intention of the query is to find how two or more entities or types are related. Correct results would describe the relationship among the query entities or types.
- Other keyword query: the intention of the query is described by some keywords that do not fit into any of the above categories. Correct results would be resources providing relevant information.

The existence of these query types has great importance for our problem, since each type requires a different type of result, and would thus have to be evaluated differently by the human judge. In our opinion, any attempt to directly map the object retrieval task to the ADR task will fail because of the existence of these query types, and the differences required in relevance judgements. We believe this to be one of the most significant distinctions between AOR and ADR, and thus one of the most complicating factors in mapping the ADR task to object ranking. We further investigate this issue in a quantitative analysis in Section 3.

Query	Entities (Types)	Intent	Category	Dom
1978 cj5 jeep	cj5 jeep	cj5 jeep	Entity	Cars
applewood golf in windham nh	applewood golf, windham, nh	applewood golf	Entity	Other
north texas eye doctors eye surgery	north texas (eye doctors, eye surgery)	eye doctors	Type	Medical
akita dog	akita	akita	Entity	Other
best cold medication	(cold medication)	cold medication	Type	Medical
botanicals for hair	botanicals	botanicals	Entity	Other
CARS FOR SALE IN AUSTIN	austin (cars)	cars	Type	Prod.
cello players	(cello players)	cello players	Type	Ppl.
employment agencies w. 14th street nyc	w. 14th street, nyc (employment agencies)	employment agencies	Type	Other
zip code waterville Maine	waterville, Maine	zip code	Attr.	Other

Figure 2: A sample of annotated queries from our query log.

2.2 Result Presentation

There is another important difference of object search with respect to the ADR task: what constitutes a result? In ADR this is clear: a single document. Documents were invented by humans to be read, and therefore it is no challenge for a human judge to read a document result and decide its relevance. However, objects were not really intended for raw human consumption. Resources are complex, structured, and heavily interconnected. For example, many properties may be needed to "define" an object, and blank nodes are often used to hold information together. The correct unit of information for retrieval is unclear.

The ADR community has also encountered such problems when retrieving very large structured documents, for example books and XML documents. The main problem there is to define the right level of *granularity* of the results: sometimes relevance will be in a single sentence, sometimes in an entire section of XML subtree. Several approaches have been developed to tackle these issues, but they always lead to quite cumbersome task formalizations which demand a great amount of effort to the human evaluators (e.g., TREC sentence retrieval [21], and INEX entity search [14]).

We note that similar to the way a text retrieval engine provides an abstract of the search result, a typical semantic search engine will return more than the identifier of the resulting resource. For blank nodes, this is a strict requirement since the identifier of the blank node is not globally unique. However, even for results with URIs, the semantic search engine will provide more complete information to allow the user to inspect the result without retrieving its definition from the Web. The search engine may also highlight parts of the resulting graph, e.g. to show where the query terms have matched or which triples have been derived by inference.

However, treating the resulting (possibly decorated) RDF graphs as a result would break our commitment to component-wise evaluation, because the relevance of the results would be tied to particular strategies for explanation. This would be similar to judging ADR methods by the quality of the snippet generator. Take the following example as an illustration:

Example 1. Consider three semantic search systems (S1, S2, and S3) that all return the entity `Albert_Einstein` (identified by `URI-1`) as a search result for the keyword query *"einstein"*. The results are returned as follows:

S1: `URI-1`

S2: ⟨`URI-1`, *rdf:label*, "Albert Einstein"⟩

S3: ⟨`URI-1`, *rdf:label*, "Albert Einstein"⟩
⟨`URI-1`, *dbp:invented*, "Theory of Relativity"⟩
⟨`URI-1`, *namespace:some-relation*, `URI-2`⟩
⟨`URI-2`, *rdf:label*, "Chicken Soup with Broccoli"⟩
⟨`URI-2`, *food:ingredient*, "Chicken"⟩
⟨`URI-2`, *food:ingredient*, "Broccoli"⟩

It is evident from Example 1 that, despite all three search systems having essentially returned the same result, there is an inherent difference in the quality of how each item is returned. A human evaluator would likely have difficulty judging the relevance of the result returned by S1, while the verbosity of the result returned by S3 seems to make it of lesser quality than that of S2. It is this difficulty that leads us to the separation of retrieval from explanation previously discussed. This allows us to define an explicit evaluation method for the ranked retrieval of resources with a fixed explanation strategy (and presentation strategy for that matter), leaving explanation as a separate task from retrieval.

For this reason, in this work we use resources as the unit of retrieval. We show to the evaluator all of the resource's contents and structure including the connections to other resources. This allows us to reuse human judgments of resources to evaluate different ranking strategies independent of how systems present results. Section 3 will discuss a particular evaluation tool and the guidelines used for the human evaluators, and will present results obtained from the evaluation of a baseline object ranking system.

Judging a single resource, while making use of any information linked to that resource, seemed to us the best possible compromise for evaluation. It has the advantage of decoupling the evaluation of objects from its presentation. Furthermore, it makes no limiting assumptions on the uses an application may make of the object; on the contrary, it assumes that the application will know how to utilize a resource identifier. It is this decoupling of relevance from presentation that allows the reuse of human judgments. We consider this to be of paramount importance for an AOR task.

A last consideration is how object data should be rendered, as objects can include complex structure and there are a number of different ways in which they could be displayed. In our evaluation we use a simple text-based rendering of object results (see Figure 3), a user study of object display interfaces is outside the scope of this work.

2.3 Evaluation and Performance Measures

One of the advantages of mapping the object ranking problem to the ADR task is that we can reuse performance

measures designed for ADR. Defining new performance measures is very problematic, because it requires studying many aspects of the measure, such as its generalization ability, its stability, appropriate statistical significance measures, etc. In fact, being able to use well established performance measures for the object ranking problem was one of our main motivations in trying to map the problem to ADR.

Under the AOR task definition, for a given query q and query type t, a retrieval system will return a list of objects $\{o_1, o_2, \cdots, o_n\}$, which are evaluated and given relevance values $R(o_i, q, t)$. We consider three popular metrics from the IR community, normalized discounted cumulative gain (NDCG), mean average precision (MAP), and precision at k (P@k) for a k value of 5. Details of these metrics can be found in Chapter 8 of [16].

3. EMPIRICAL STUDY

In order to evaluate a retrieval task such as ADR or AOR, one needs to set up a real search task including a data collection and human evaluators. As we saw in the previous section, in the case of object ranking, we also need to fix the resource presentation application and the query category. In this section we set up a realistic evaluation of an AOR task on the Web. We use real query logs from a commercial search engine and a real large-scale collection of resources from a crawl of the Web of Data.

For resources, we focused on metadata publicly available on the Web. We used a subset of 240 million web pages which have been crawled by Yahoo! and contain some form of metadata (RDFa and various types of microformats). This produced approximately 8 billion triples when normalized as RDF, or about a 1.1 terabytes data graph (uncompressed). This graph contains schematic as well as instance data constructs, allowing a semantic search engine to exploit explicit assertions in the data as well as semantic inferences over RDFS or OWL.

In this evaluation we used the annotated query set described in Section 3.2. As argued before, the advantage of this approach is the diversity of the information needs that query logs capture, i.e. a sample of the aggregate information needs of Web users in the US geography. As opposed to a manually created or synthetic benchmark these queries also provide real information needs. Lastly, since our data set has been gathered from the Web it is natural to rely on Web queries for our evaluation. The disadvantage of our method is that the original intent of the query is not directly available. We deal with this situation simply by allowing evaluators to skip the queries that they do not understand. We do not instruct evaluators to skip the queries where multiple interpretations are possible but rather to consider all possible interpretations as valid. In practice, the majority of queries can be easily interpreted by the evaluators.

As an object retrieval engine we used the baseline system described in Section 3.1. The retrieval implementation uses an inverted index over the text fields occurring in the underlying triples. This index was used to retrieve a subset of resources, from which a subset of the RDF graph could be constructed to produce and rank results. We use MG4J [29] for our index.

With a fixed retrieval, explanation and presentation strategy, we conducted an evaluation of ranking semantic resources as follows. We computed the top-5 results of a baseline ranking strategy. We then did a human evaluation of relevance of the results, and also established the ideal ranking based on the ideal re-ordering using the human evaluations.

3.1 Baseline Ranking Approach

The baseline ranking strategy used is an adaptation of TF-IDF to RDF graphs. In this setting, we compute an IDF score for each term in the vocabulary with respect to each RDF property occurring in the graph. This means, for example, that the term "John" can have different IDF values for the properties "vcard:name" and "vcard:address", a desirable property to distinguish among common terms in particular properties. In the example, the term "John" may be very distinguishing as a street name, while very common as a person name. We also blacklist a set of RDF properties that contain a lot of diverse text causing noise in the results. These properties were chosen manually based on an inspection of a sample of query results. The blacklisting includes properties such as review text and blog summaries, which tend to match a diverse set of terms while providing little insight into the meaning of a resource. While this is certainly not an ideal ranking approach in general, it provides sufficient quality as a baseline metric to evaluate our methodology. A thorough investigation of RDF ranking methods is outside the scope of this paper.

3.2 Query Analysis

Our query log analysis was conducted as follows. First, the queries themselves were annotated by human judges. Judges segmented the query into entities, types, relations, attributes and remaining relevant keywords. Furthermore, judges annotated the intent of the query, as discussed in 2.1. The intent of the query is defined as the component of the query for which the query is primarily seeking information. Besides the primary focus, a query may be further tagged with additional information such as secondary entities that appear in the query to disambiguate the primary intent or to provide additional information. For example, the keyword query *"doctors in barcelona"* is primarily seeking information about *"doctors"*, which is an entity type. The entity *"barcelona"* which also appears in the query would be referred to as a *context entity*.

Table 1 shows the results of this study. We found a large bias towards entity centric queries, or queries that do not fit easily into a semantic classification ("Other keyword queries", for example, the query "nightlife in Barcelona" has the primary intention of finding information about "nightlife" within the context of the entity "Barcelona"). We found that more than half of the queries focus on an entity or entity type. Interestingly, among all of the "Other keyword queries" (those which do not have any semantic resource as the primary intention), 14.3% of them also contain a context entity or type. This means that over 70% of all queries contain a semantic resource (entity, type, relation, or attribute), with almost 60% having a semantic resource as the primary intent of the query.

We also asked judges to label queries with a number of topical domains. These domains were inspired by various vertical search domains as well as domains manually identified during an initial analysis of a sample of the queries. Table 2 shows the distribution over the top domains for all queries in our sample. Almost half of the queries do not fit any domains at all, illustrating that a semantic interpre-

Query Type	Percent
Entity query	40.60%
Type query	12.13%
Attribute query	4.63%
Relation query	0.68%
Other keyword query	36.10%
Uninterpretable	5.89%

Table 1: Distribution of query types for a sample of web queries.

Query Domain	Percent
Other	49.05%
Navigational	14.44%
Product	14.31%
People	9.81%
Medical	3.13%
All others	< 3.00%

Table 2: Distribution of query domain for all queries.

Query Domain	Percent
Other	39.44%
Navigational	14.29%
Product	19.61%
People	14.50%
Medical	3.20%
All others	< 3.00%

Table 3: Distribution of query domain for all queries with any semantic resource intent.

Query Domain	Percent
Other	26.51%
Navigational	22.15%
Product	20.47%
People	18.79%
Medical	3.36%
All others	< 3.00%

Table 4: Distribution of query domain for queries with entity intent.

tation of user queries yields a largely domain independent workload. Table 3 considers only queries with the primary intention being a semantic resource (ie., remove all of the "Other keyword queries"), we see that this result is still present, though less pronounced. However, navigational, product, and people searches account for most of the remaining queries, showing there is still room for optimization based on vertical domains. In Table 4 we see that among the queries with an entity as the primary intention, the percentage of queries that do not fit a particular domain category is almost half in comparison to the full query set. The navigational, product, and people queries constitute a much larger percentage of the query domains when only considering entity queries. This implies that entity queries, though still primarily (26% of the time) domain independent, do have significant sized classes of queries which may benefit from domain dependent vertical search. We focus on domain independent evaluation to cover the more general case, though our methodology could still apply to vertical search evaluation.

Note that in this classification, a navigational query (e.g., "amazon.com") may also be considered an entity query, and a semantic search engine may choose to exploit this by returning information about the entity (e.g., information about the company amazon.com) rather than the entities home page as in traditional handling of navigational queries.

This analysis shows that type and entity queries constitute the vast majority of structured queries and thus we can restrict ourselves to these types of queries. For systems where these query types have a particular significance, separate evaluations can be devised in the future.

3.3 Object Relevance Analysis

For each of the queries in the analysis set, we run the baseline object retrieval engine (discussed in Section 3.1) and asked human judges to evaluate the results. For this, we developed an evaluation tool which allowed judges to see all the query information, and the resources being evaluated.

The explanation strategy we used was a form of *concise-bounded description*[6], where all properties directly connected to a qualifying resource are included in the explanation, and blank nodes are recursively expanded to provide explanations of how blank nodes connect to any qualifying resource. We limit the expansion to a maximum of ten properties to avoid overwhelming the user with high degree nodes (which are often large collections of tags, generally unrelated to the resource and can thus be considered as spam). We do not enforce a limit on how many blank nodes can be expanded, meaning the size of the RDF subgraph shown to the user is unbounded and can vary considerably.

The tool displays a query along with a single entry of a semantic search result (the URI along with its explanation). A user can then rate the relevance of the result to the query on a four-point zero to three scale. We measure two dimensions of relevance, the relevance of the result to the main intention of the query; and relevance of the result to the full query text. For example, given the query *"john smith barcelona"* with entity set {*"john smith", "barcelona"*} and *intent = "john smith"*, we would evaluate the relevance of each query result to both the entity *"john smith"* as well as the full query. In this setting, any "John Smith" is relevant in the first case, while only the "John Smiths" who are associated with "Barcelona" are relevant in the second case. We design our experiment in this manner because we are interested in both ranking for entity based retrieval and for full query answering.

Human judges were instructed to rate the relevance of the main entity, and only use the query explanation as aid in understanding what the entity is. Thus over-explanation or any irrelevant information in the explanation does not affect the relevance rating, ensuring we only measure result quality and not explanation quality. The judges were given the following two questions and a four-point scale for each.

[6] `http://www.w3.org/Submission/CBD/`

Figure 3: A screen capture of our evaluation tool showing a query for the entity "Aerocalifornia" with context entity "La Paz".

- Is the *resource* in the result relevant to ⟨`the query intent resource`⟩?
- Is the *resource* in the result relevant to ⟨`the full text query`⟩?

In each case, the words "resource" and the resource or query were highlighted and colour coded to coincide with the rendering of the query, query intent, and result resource. This ensured that the judge could easily distinguish what was being evaluated for each question. The description of the scale given to the judges was as follows.

- 0: Not relevant - the resource in the result is not at all relevant to the query resource.
- 1: Somewhat relevant - the resource in the result is moderately relevant to the query resource. For example, the result has the entity contained in a text property (review text, summary, etc...) or tag.
- 2: Relevant - the resource in the result is related to the resource in the query.
- 3: Perfect match - the resource in the result exactly describes the resource in the query. For example, the result is a VCard (address book entry) for a person entity in the query. In the cases of a type query, a result which is an instance of the type is also a perfect match.

A similar description was given for the task of rating the relevance of the result resource to the full query.

The tool is shown in Figure 3. The query, *"aerocalifornia, la paz"* is rendered in black, with the main intent of the query, *"aerocalifornia"* below in parenthesis. The result being shown is a VCard for a person who works for Aerocalifornia in La Paz (the name is anonymized for privacy reasons). Because this person entity is related to the intent entity of the query, it would be given a score of two by human evaluators.

With this mechanism we judged 1162 results from 264 queries, corresponding to the (at most) top-5 results produced by our baseline ranking algorithm. Such a *shallow* evaluation scheme is typical of web search engine evaluation, where greater number of queries is preferred to large number of results evaluated per query.

3.3.1 Analysis of Human Judgments

Even with a given description of the evaluation task, determining the relevance of an RDF resource to a query is a difficult process that can be highly subjective at times. To validate our human assessments, we had the judges evaluate an overlapping set of 230 query results and compared the assessments. Figure 4 shows a table of evaluation decisions for the four-point relevance scale. Each cell in the (symmetric) table shows the frequency of a judgment for the value of its respective row and column. Thus the diagonal entries show the frequency of perfect agreement, which occurs on 64% of the result evaluations for query intent, and 71% of the result evaluations for full query. If we also consider the cells adjacent to the diagonals, we see that the "off-by-one" agreement of judges produces an agreement of 93% for query intent 94% for full query. This indicates that our judges give

Query Intent Resource

	0	1	2	3
0	46	16	6	0
1	-	61	37	19
2	-	-	13	9
3	-	-	-	23

Full Text Query

	0	1	2	3
0	148	29	6	0
1	-	15	7	5
2	-	-	3	9
3	-	-	-	8

Figure 4: Inter-judge agreement on relevance scores from 0 to 3 for both relevance to the query resource and relevance to the full text query.

generally similar valuations to results, despite the inherent subjectiveness of the task.

In Figure 5 a histogram for both relevance of query intent to the result and relevance of full query to the result is shown. These histograms illustrate the frequencies of each value on our four-point scale over the full result set. While our entity retrieval scores mostly in the moderate relevance range, the results are not very relevant to the full text queries. This is a result of our simple baseline metric used for retrieval. Our results, however, do contain enough diversity in relevance to conduct an evaluation for distinguishing ranking algorithms.

3.3.2 Stability

We evaluate the stability of our chosen metrics on the resource ranking task. In order to achieve a 95% confidence interval, we apply a standard bootstrap method [8]. We compute a sample of size n from our query workload of size n, sampling with replacement. From this sample, we compute the mean value of the metric being evaluated over all queries in the sample. We repeat this process one million times to get a set of means, and compute the mean and standard error over the resulting set of means to obtain a 95% confidence interval. This final mean can then be compared with the empirical mean obtained from the actual workload to test the stability of the metric.

Figure 6 shows the results of our stability test. Averaged across many samples, the bootstrapped mean of the metric becomes very close to the empirical mean and produces a tight confidence interval at 95% confidence (we find a difference after the sixth decimal place in all cases). This indicates that the metrics are stable across samples of semantic queries from the same distribution over RDF resource results. We note that this stability is also dependent on our data, the sampling method used to obtain our query workload from the web query log, and the human relevance judgments. We must also make the assumption that the bootstrap sampling draws scores from a normal distribution to apply this analysis.

3.3.3 Discriminant Power

We aim to show that applying our test metrics to RDF resources with a fixed retrieval, explanation, and presentation strategy has the ability to discriminate among ranking approaches in an intuitive way. We consider three ranking strategies, all based on re-ranking the same top-5 results returned by our fixed baseline retrieval strategy. The first approach is a random re-ranking, the second is our baseline ranking algorithm described in Section 3.1, and the third is the ideal re-ranking where results are ordered based on the valuations given by a human judge.

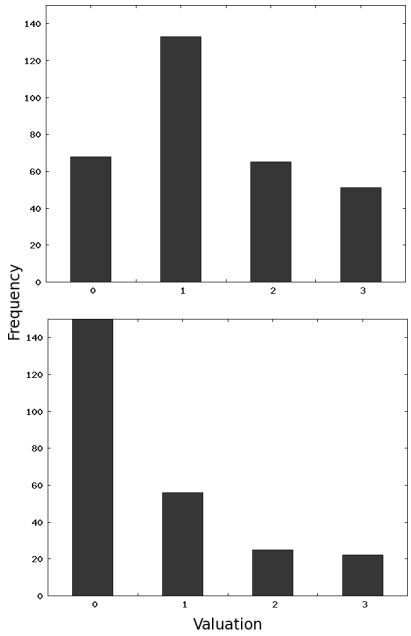

Figure 5: Histograms showing the frequencies of valuations from the four-point evaluation scale. Valuations were made for query intent (top) and full query (bottom).

Figure 7 shows the results of the three ranking approaches. We see that for both the query resource relevance and the full text query relevance the scores distinguish among the ranking approaches as expected. This is a positive result for discriminating power, given how easy it is to obtain a high score with a random re-ranking of only 5 results. Note that we omit P@5 since re-ranking the same results does not affect the score, and thus all systems would be ranked equally.

4. RELATED WORK

Due to the diverse nature of data models considered in semantic search systems (e.g., text, annotated data, ontologies) a wide spectrum of query languages are used for semantic search. These query languages range from expressive structured queries, like SPARQL [23], to simple keyword queries. A number of works also consider semi-structured variations (hybrid search) which integrate structured query parts with keyword context [27, 20, 25].

From the point of view of evaluation, structured queries have an explicit semantics and thus a well defined query result. Such systems can be evaluated using standard approaches from the database community, and as such, we omit them from our discussion. Keyword queries and semi-structured queries however, have uncertain results and are thus much more challenging to evaluate.

A number of end-to-end semantic search systems have been developed up to date, e.g. [6, 7, 10, 11, 12, 19, 24]. Given that semantic search systems are a subset of information retrieval systems it would be natural to apply existing IR benchmarks. This is the choice made by Fernandez et al. who perform evaluation using the TREC benchmark

	Query Resource			Full Query		
Metric	Random	Baseline	Ideal	Random	Baseline	Ideal
DCG	5.004	5.072	5.402	2.878	2.923	3.0670
NDCG	0.942	0.952	1.0	0.968	0.972	1.0
MAP	0.644	0.719	0.820	0.603	0.668	0.790

Figure 7: Distinguishing among the baseline, random, and ideal rankings for various metrics.

Measurement	Query Resource	Full Query
Empirical NDCG	0.9529	0.9721
Bootstrap NDCG	0.9529 ± 1.02e-05	0.9721 ± 9.08e-06
Empirical MAP	0.7190	0.6684
Bootstrap MAP	0.7190 ± 7.04e-05	0.6684 ± 7.69e-05
Empirical P@5	0.4111	0.1533
Bootstrap P@5	0.4111 ± 5.25e-05	0.1533 ± 4.15e-05

Figure 6: Stability of metrics for resource ranking. Confidence intervals are computed at 95% confidence. Results are shown for both result resource to query resource, and result resource to full text query.

[9]. TREC has been designed for measuring the relevance of results returned by text retrieval engines and finds its origins in the Cranfield experiments [5]. However, there are two problems inherent in applying TREC measurements directly. First, many semantic search engines fall back to basic keyword search when the query can not be understood in terms of the available semantic models. Thus it is not known how much of the evaluation tests the part of the system that actually employs semantics, and how much covers traditional keyword search. In fact, in the work of Fernandez et al. only 20% of the TREC queries used are covered by ontology concepts, the rest are removed leaving 20 queries for evaluation. Simply dismissing the queries that can not be semantically interpreted is not easy as different search engines may have vastly different query understanding capabilities. Furthermore, reaching consensus on a query workload in this manner may be impossible as each approach to query interpretation can vary significantly. Thus, taking the intersection of queries that a set of semantic search engines can handle would likely produce an "easy" workload (if not empty), which can not be justified as representative of real users. Evaluating each search engine only on the queries it can interpret also has drawbacks since the results obtained by two different search engines may not be directly comparable.

The second and even more apparent problem is that not all semantic search engines perform document retrieval, but rather retrieve knowledge encoded in some semantic data model. In many contexts, the data in the system is not necessarily associated with any particular text document. This is the case for example with search engines that crawl and index Linked Data such as Sindice [19]. Even in cases where there is a document, an evaluation based on document rankings is not able to measure some of the key advantages of semantic search such as being able to give precise answers to factual questions or that answers can be computed by aggregating knowledge from different documents.

While TREC itself is thus not applicable to our scenario, the general concepts of the Cranfield experiments can be directly carried over. In particular, we take a system-oriented perspective with a reference collection, a fixed set of test queries and a set of relevance judgments, which form the benchmark. In this respect our work is analogous to the INEX series [14], which is an adaptation of the methodology to XML content retrieval. Among the various tracks of INEX, the entity-retrieval experiments the INEX Entity Ranking Track is most relevant in terms of the query type. However, this INEX track also focuses on textual corpora and that means that systems compete also on information extraction functionality. In contrast, semantic web search engines work with structured data.

In terms of the queries we consider, there are also commonalities to benchmarks that consider queries for particular types of entities such as the Web People Search Evaluation (WEPS) [2], or the expert finding task of the TREC Enterprise Track [22]. With respect to addressing keyword retrieval on structured data, there is also existing work in the database literature (e.g., [3, 13, 15]). This field of research has not produced a common evaluation methodology that we could adapt, and the development of evaluation methodology has been identified by this community as an important goal for future work in a recent SIGMOD tutorial [4].

5. CONCLUSION

We have outlined the details of a methodology for semantic search evaluation on the Web of Data: ad-hoc object retrieval. Our proposal builds on the well established ad-hoc document retrieval task, allowing us to reuse existing efforts. We empirically justified the applicability of ADR metrics to our object ranking problem showing that our adaptations to object retrieval are sound. We also have constructed a typology of Web queries from the object retrieval point of view and have defined the notion of a result and what relevance means in this context.

To a great extent, much of our work revolved around making the choices required to arrive at a benchmark that can be reliably executed, a property which we have demonstrated. We believe that the resulting evaluation still covers a large class of systems and it is able to discriminate among these systems based on their ranking functionality. One may argue that by opting to evaluate using Web queries we have excluded structured queries of higher complexity than what is typically expressed in Web search, e.g. queries involving variable matching or restrictions on attributes. We have chosen to evaluate on Web queries because there is no similar generic collection of real information needs in the semantic search domain. The queries we did find in the query logs are real, provide a diversity of topics, are highly relevant and fall within the common subset of query types supported by the majority of semantic search engines. Despite this,

our model could be applied in alternative scenarios where the relevance of an object to a query can be evaluated.

It has also become clear that in order to arrive to an executable benchmark, we needed to exclude significant parts of a semantic search system. We believe that additional evaluations can be designed for these components, e.g. for measuring the quality of result explanation. One may also decide to compare actual implementations of semantic search engines: while a benchmark may compare algorithms, the success of actual semantic search engines will depend on a number of variable factors including the comprehensiveness of the index and the freshness of the data.

We consider our results a first step towards a common methodology for semantic search evaluation. Our analysis provides a foundation that can be harnessed by the semantic search community at large to develop standard evaluation procedures for object retrieval on the Web. Based on our results, we plan to organize a public evaluation campaign of semantic search ranking as part of the yearly Semantic Search workshop series. To this end, we plan to release in the public domain both our annotated query set, our reference collection, and our evaluation tool.

6. REFERENCES

[1] D. J. Abadi, A. Marcus, S. Madden, and K. Hollenbach. SW-Store: a vertically partitioned DBMS for Semantic Web data management. *VLDB J.*, 18(2):385–406, 2009.

[2] J. Artiles, S. Sekine, and J. Gonzalo. Web people search: results of the first evaluation and the plan for the second. In *WWW*, pages 1071–1072, 2008.

[3] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword Searching and Browsing in Databases using BANKS. In *ICDE*, pages 431–440, 2002.

[4] Y. Chen, W. Wang, Z. Liu, and X. Lin. Keyword search on structured and semi-structured data. In *SIGMOD Conference*, pages 1005–1010, 2009.

[5] C. Cleverdon and M. Kean. Factors Determining the Performance of Indexing Systems, 1968.

[6] L. Ding, T. Finin, A. Joshi, R. Pan, S. R. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs. Swoogle: a search and metadata engine for the semantic web. In *CIKM '04: Proceedings of the thirteenth ACM conference on Information and knowledge management*, pages 652–659, New York, NY, USA, 2004. ACM Press.

[7] A. Duke, T. Glover, and J. Davies. Squirrel: An advanced semantic search and browse facility. In *ESWC*, pages 341–355, 2007.

[8] B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1):1–26, 1979.

[9] M. Fernandez, V. Lopez, M. Sabou, V. Uren, D. Vallet, E. Motta, and P. Castells. Semantic Search Meets the Web. In *ICSC '08: Proceedings of the 2008 IEEE International Conference on Semantic Computing*, pages 253–260, Washington, DC, USA, 2008. IEEE Computer Society.

[10] M. Fernandez, V. Lopez, M. Sabou, V. Uren, D. Vallet, E. Motta, and P. Castells. Semantic Search Meets the Web. In *IEEE Semantic Computing*, pages 253–260, 2008.

[11] R. Guha, R. McCool, and E. Miller. Semantic search. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 700–709, New York, NY, USA, 2003. ACM.

[12] A. Harth, J. Umbrich, A. Hogan, and S. Decker. YARS2: A Federated Repository for Querying Graph Structured Data from the Web. *The Semantic Web*, pages 211–224, 2008.

[13] V. Hristidis and Y. Papakonstantinou. DISCOVER: Keyword Search in Relational Databases. In *VLDB*, pages 670–681, 2002.

[14] J. Kamps, S. Geva, A. Trotman, A. Woodley, and M. Koolen. Overview of the inex 2008 ad hoc track. *Advances in Focused Retrieval: 7th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2008, Dagstuhl Castle, Germany, December 15-18, 2008. Revised and Selected Papers*, pages 1–28, 2009.

[15] Y. Luo, W. Wang, and X. Lin. SPARK: A Keyword Search Engine on Relational Databases. In *ICDE*, pages 1552–1555, 2008.

[16] C. Manning, P. Raghavan, and H. Schütze. *An Introduction to Information Retrieval*. Cambridge University Press, 2008.

[17] F. Manola and E. Miller. Rdf primer, February 2004.

[18] T. Neumann and G. Weikum. RDF-3X: a RISC-style engine for RDF. *PVLDB*, 1(1):647–659, 2008.

[19] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello. Sindice.com: a document-oriented lookup index for open linked data. *IJMSO*, 3(1):37–52, 2008.

[20] C. Rocha, D. Schwabe, and M. P. Aragao. A hybrid approach for searching in the semantic web. In *Proceedings of the 13th international conference on World Wide Web*, pages 374–383. ACM Press, 2004.

[21] I. Soboroff. Overview of the trec 2004 novelty track. In *The Thirteenth Text Retrieval Conference (TREC 2004)*, 2004.

[22] I. Soboroff, A. de Vries, and N. Craswell. Overview of the TREC 2006 Enterprise Track, page 32.

[23] SPARQL Query Language for RDF. http://www.w3.org/tr/rdf-sparql-query/, 2008.

[24] T. Tran, H. Wang, and P. Haase. SearchWebDB: Data Web Search on a Pay-As-You-Go Integration Infrastructure, 2008.

[25] H. Wang, T. Tran, and C. Liu. CE2: towards a large scale hybrid search engine with integrated ranking support. In *CIKM*, pages 1323–1324, 2008.

[26] C. Weiss, P. Karras, and A. Bernstein. Hexastore: sextuple indexing for semantic web data management. *PVLDB*, 1(1):1008–1019, 2008.

[27] L. Zhang, Q. Liu, J. Zhang, H. Wang, Y. Pan, and Y. Yu. Semplore: An ir approach to scalable hybrid query of semantic web data. In *In Proc. of the 7th Intl. Semantic Web Conference*, pages 652–665, 2008.

[28] Semantic Search Workshop. Homepage, 2009. `http://km.aifb.uni-karlsruhe.de/ws/semsearch09/`.

[29] MG4J: Managing gigabytes for java. Homepage. `http://mg4j.dsi.unimi.it/`.