Evaluating Ad-Hoc Object Retrieval

Harry Halpin[1], Daniel M. Herzig[2], Peter Mika[3], Roi Blanco[3], Jeffrey Pound[4], Henry S. Thompson[1], and Thanh Tran Duc[2]

[1] University of Edinburgh, UK
[2] Karlsruhe Institute of Technology, Germany
[3] Yahoo! Research, Spain
[4] University of Waterloo, Canada
H.Halpin@ed.ac.uk, herzig@kit.edu, pmika@yahoo-inc.com, roi@yahoo-inc.com, jpound@cs.uwaterloo.ca, ht@inf.ed.ac.uk, ducthanh.tran@kit.edu

Abstract. In contrast to traditional search, semantic search aims at the retrieval of information from factual assertions about real-world objects rather than searching over web-pages with textual descriptions. One of the key tasks to address in this context is ad-hoc object retrieval, i.e. the retrieval of objects in response to user formulated keyword queries. Despite the significant commercial interest, this kind of semantic search has not been evaluated in a thorough and systematic manner. In this work, we discuss the first evaluation campaign that specifically targets the task of ad-hoc object retrieval. We also discuss the submitted systems, the factors that contributed to positive results and the potential for future improvements in semantic search.

1 Introduction

Advances in information retrieval have long been driven by evaluation campaigns, and the use of official TREC evaluations and the associated queries and data-sets are ubiquitous in measuring improvements in the effectiveness of IR systems. We believe the rigor of evaluation in semantic search should be no different. Yet no such evaluation campaign exists for semantic search, and so usually very small and artificial data-sets are used to evaluate semantic search, using a diverse set of evaluation methods.

As a first step towards a common evaluation methodology, Pound et al. [10] defined the task of Ad-hoc Object Retrieval, where 'semantic search' is considered to be the retrieval of objects represented as Semantic Web data, using keyword queries for retrieval. While 'semantic search' is a broader notion that includes many other tasks, the one of object retrieval has always been considered as an integral part.

In this paper, we describe the way we created a standard data-set and queries for this task and the results of a public evaluation campaign we organized. Since the evaluated systems also run a gamut of the approaches used in semantic search, the evaluation results presented in this paper give an overview of the state-of-the-art in this growing field of interest.

Proceedings of the International Workshop on Evaluation of Semantic Technologies (IWEST 2010). Shanghai, China. November 8, 2010.

In the following, we will first give an overview of the often ambiguous term 'semantic search' (Section 2) before delving into the particular evaluation methodology used to evaluate search over Semantic Web data, including the creation of a standard data-set and queries (Section 3) and we briefly introduce our use of crowd-sourcing (Section 3.3). Then the submitted evaluated systems will be discussed along with the results of the evaluation (Section 4).

2 An Overview of Semantic Search

In detail the term 'semantic search' is highly contested, primarily because of the perpetual and endemic ambiguity around the term 'semantics.' While 'search' is understood to be some form of information retrieval, 'semantics' typically refers to the interpretation of some syntactic structure to another structure, the 'semantic' structure, that defines in more detail the meaning that is implicit in the surface syntax (or even the 'real-world' that the syntax describes). Semantics can be given to various parts of the information retrieval model, including the representations of the queries and the documents. This semantics can then be used to process queries against documents, as well as to support users during query construction and the presentation of results.

One main problem encountered by semantic search has been the general lack of a standard to capture the semantics. Knowledge representation formalisms vary widely, and up until the advent of the Semantic Web there have been no common standards for capturing the semantics of semantic search. The primary standard underlying the Semantic Web is RDF (Resource Description Framework), a flexible model that can capture graph-based semantics such as semantic networks, but also semi-structured data as used in databases. Semantic Web data represented in RDF is composed of subject-predicate-object *triples*, where the subject is an identifier for a resource (e.g. a real-world object), the predicate an identifier for a relationship, and the object is either an identifier of another resource or some information given as a concrete value (e.g. a string or data-typed value).

While more complex kinds of semantic search attempt to induce some sort of Semantic Web-compatible semantics from documents and queries directly, we focus instead on search methods that apply information retrieval techniques directly on Semantic Web data. We are motivated by the growing amount of data that is available directly in RDF format thanks to the worldwide Linked Data[5] movement that created a rapidly expanding data space of interlinked public data sets.

There are already a number of semantic search systems that crawl and index Semantic Web data such as [2][5][9], and there is active research into algorithms for ranking in this setting. Despite the growing interest, it has been concluded in plenary discussions at the Semantic Search 2009 workshop[6] that the lack of standardized evaluation has become a serious bottleneck to further progress in

[5] `http://linkeddata.org`
[6] `http://km.aifb.kit.edu/ws/semsearch09/`

this field. In response to this conclusion, we organized the public evaluation campaign that we describe in the following.

3 Evaluation Methodology

Arriving at a common evaluation methodology requires the definition of a shared task that is accepted by the community as the one that is most relevant to potential applications of the field. The definition of the task is also a precondition for establishing a set of procedures and metrics for assessing performance on the task, with the eventual purpose of ranking systems [1]. For the field of text retrieval, this task is the retrieval of a ranked list of (text) documents from a fixed corpus in response to free-form keyword queries, or what is known as the ad-hoc document retrieval task.

In ad-hoc object retrieval the goal is to retrieve a ranked list of objects from a collection of RDF documents in response to free-form keyword queries. The unit of retrieval is thus individual objects (resources in RDF parlance) and not RDF documents[7]. Although some search engines do retrieve RDF documents (thus provide coarser granularity), object retrieval is the task with the highest potential impact for applications. Pound et al. [10] also proposed an evaluation protocol and tested a number of metrics for their stability and discriminating power. In our current work, we instantiate this methodology in the sense of creating a standard set of queries and data on which we execute the methodology.

3.1 Data Collection

Current semantic search engines have vastly different indices, with some specializing on only single data-sources with thousands of triples and others ranging over billions of triples crawled from the Web. Therefore, in order to have a generalized evaluation of the ranking of results, it is essential to normalize the index in order.

We required a data-set that would not bias the results towards any particular semantic search engine. The data-set that we wanted to use in the evaluation campaign needed to contain real data, sizable enough to contain relevant information for the queries, yet not so large that its indexing would require computational resources outside the scope of most research groups. We have chosen the 'Billion Triples Challenge' 2009 data set, a data-set created for the Semantic Web Challenge[8] in 2009 and which is well-known in the community. The raw size of the data is 247GB uncompressed and it contains 1.4B triples describing 114 million objects. This data-set was composed by combining crawls of multiple semantic search engines. Therefore, it does not necessarily match the coverage of any particular search engine. Also, it is only a fragment of the data that can be found on the Semantic Web today that is representative and still manageable by

[7] An RDF graph connects a number of resources through typed relations.
[8] `http://challenge.semanticweb.org`

individual research groups. We refer the readers to `http://vmlion25.deri.ie/` for more information on the dataset.

The only modification we have done is to replace local, document-bound resource identifiers ('blank nodes' in RDF parlance) with auto-generated URIs, i.e., globally unique resource identifiers. This operation does not change the semantics of data but it is necessary because resources are the unit of retrieval. With the announcement of the evaluation campaign, this modified 'Billion Triples Data-set' was released for download and indexing by participants[9].

3.2 Real-world Web Queries

As the kinds of queries used by semantic search engines vary dramatically (ranging from structured SPARQL queries to searching directly for URI-based identifiers), it was decided to focus first on keyword-based search. Keyword-based search is the most commonly used query paradigm, and supported by most semantic search engines.

Clearly, the type of result expected, and thus the way to assess relevance depend on the type of the query. For example, a query such as *plumbers in mason ohio* is looking for instances of a class of objects, while a query like *parcel 104 santa clara* is looking for information for one particular object, in this case a certain restaurant. Pound et al. [10] proposed a classification of queries by expected result type, and for our first evaluation we have decided to focus on object-queries, i.e. queries demonstrated by the latter example, where the user is seeking information on a particular object. Note that for this type of queries there might be other objects mentioned in the query other than the main object, such as *santa clara* in the above case. However, it is clear that the focus of the query is the restaurant named *parcel 104*, and not the city of Santa Clara as a whole.

We were looking for a set of object-queries that would be unbiased towards any existing semantic search engine. First, although the search engine logs of various semantic search engines were gathered, it was determined that the kinds of queries varied quite a lot, with many of the query logs of semantic search engines revealing idiosyncratic research tests by robots rather than real-world queries by actual users. Since one of the claims of semantic search is that it can help general purpose ad-hoc information retrieval on the Semantic Web, we have decided to use queries from actual users of hypertext Web search engines. As these queries would be from hypertext Web search engines, they would not be biased towards any semantic search engine. We had some initial concerns if within the scope of the data-set it would be possible to provide relevant results for each of the queries. However, this possible weakness also doubled as a strength, as the testing of a real query sample from actual users would determine whether or not a billion triples from the Semantic Web realistically could help answer the information needs of actual users, as opposed to purely researchers [4].

[9] `http://km.aifb.kit.edu/ws/semsearch10/#eva`

In order to support our evaluation, Yahoo! released a new query set as part of their WebScope program[10], called the *Yahoo! Search Query Log Tiny Sample v1.0*, which contains 4,500 queries sampled from the company's United States query log from January, 2009. One limitation of this data-set is that it contains only queries that have been posed by at least three different (not necessarily authenticated) users, which removes some of the heterogeneity of the log, for example in terms of spelling mistakes. While realistic, we considered this a hard query set to solve. Given the well-known differences between the top of the power-law distribution of queries and the long-tail, we used an additional log of queries from the Microsoft Live Search containing queries that were repeated by at least 10 different users.[11] We expected these queries to be easier to answer.

We have selected a sample of 42 entity-queries from the Yahoo! query log by classifying queries manually as described in Pound et al. [10]. We have selected a sample of 50 queries from the Microsoft log. In this case we have pre-filtered queries automatically with the Edinburgh MUC named entity recognizer [8], a gazetteer and rule-based named-entity recognizer that has shown to have very high precision in competitions. Both sets were combined into a single, alphabetically ordered list, so that participants were not aware which queries belonged to which set, or in fact that there were two sets of queries. We distributed the final set of 92 queries to the participants two weeks before the submission deadline.

3.3 Crowd-sourcing Relevance Judgments

We crowd-sourced the relevance judgments using Amazon's Mechanical Turk. A deep analysis on the reliability and repeatability of the evaluation campaign is left for future work. For the purpose of evaluation, we have created a simple rendering algorithm to present the results in a concise, yet human-readable manner without ontology-dependent customizations In order to achieve good throughput from the judges, each HIT consisted of 12 query-result pairs for relevance judgments. Of the 12 results, 10 were real results drawn from the participants' submissions, and 2 were gold-standard results randomly placed in the results. These gold-standard results were results from queries distinct from those used by the participants that were manually judged to be either definitely a 'relevant' or 'irrelevant' result. For each HIT, there was both a gold-standard relevant and gold-standard irrelevant result included.

65 Turkers in total participated in judging a total of 579 HITs over a three-point scale[12], covering 5786 submitted results and 1158 gold-standard checks. 2 minutes were allotted for completing each HIT. The average agreement and its standard deviation, computed with Fleiss's κ, for the two- and three-point scales

[10] `http://webscope.sandbox.yahoo.com/`

[11] This query log was used with permission from Microsoft Research and as the result of a Microsoft 'Beyond Search' award.

[12] *Excellent*- describes the query target specifically and exclusively
Not bad - mostly about the target
Poor - not about the target, or mentions it only in passing

are 0.44±0.22 and 0.36±0.18, respectively. There is thus no marked difference between a three-point scale and a binary scale, meaning that it was feasible to judge this task on a three-point scale. By comparing the number of times a score appeared and the number of times it was agreed on, 1s (irrelevant results) were not only the most numerous, but the easiest to agree on (69%), followed by 3s (perfect results, 52%) and tailed by 2s (10%). This was expected given the inherent fuzziness of the middle score. To see how this agreement compares to the more traditional setting of using expert judges, we have re-judged 30 HITs ourselves. We have again used three judges per HIT, but this time with all judges assessing all HITs. In this case, the average and standard deviation of Fleiss's κ for the two- and three-point scales are 0.57±0.18 and 0.56±0.16 and, respectively. The level of agreement is thus somewhat higher for expert judges, with comparable deviation. For expert judges, there is practically no difference between the two- and three-point scales, meaning that expert judges had much less trouble using the middle judgment. The entire competition was judged within 2 days, for a total cost of $347.16. We consider this both fast and cost-effective.

4 Evaluation Results

4.1 Overview of Evaluated Systems

For the evaluation campaign, each semantic search engine was allowed to produce up to three different submissions ('runs'), to allow the participants to try different parameters or features. A submission consisted of an ordered list of URIs for each query. In total, we received 14 different runs from six different semantic search engines. The six participants were DERI (Digital Enterprise Research Institute), University of Delaware (Delaware), Karlsruhe Institute of Technology (KIT), University of Massachusetts (UMass), L3S, and Yahoo! Research Barcelona (Yahoo! BCN).

All systems used inverted indexes for managing the data. The differences between the systems can be characterized by two major aspects: (1) the internal model used for representing objects and (2), the kind of retrieval model applied for matching and ranking. We will now first discuss these two aspects and then discuss the specific characteristics of the participated systems and their differences.

For object representation, RDF triples having the same URI as subject have been included and that URI is used as the object identifier. Only the **DERI** and the **L3S** deviate from this representation, as described below. More specifically, the object description comprises attribute and relation triples as well as provenance information. While attributes are associated with literal values, relation triples establish a connection between one object and one another. Both the attributes and the literal values associated with them are incorporated and stored on the index. The objects of relation triples are in fact identifiers. Unlike literal values, they are not directly used for matching but this additional information has been considered valuable for ranking. Provenance is a general notion that

can include different kinds of information. For the problem of object retrieval, participated systems used two different types of provenances. On the one hand, RDF triples in the provided data-set are associated with an additional context value. This value is in fact an identifier, which captures the origin of the triples, e.g. from where it was crawled. This provenance information is called here the 'context'. One the other hand, the URI of every RDF resource is a long string, from which the domain can be extracted. This kind of provenance information is called 'domain'. Clearly, the domain is different to the context because URIs with the same domain can be used in different contexts. Systems can be distinguished along this dimension, i.e., what specific aspects of the object they took into account.

The retrieval model, i.e. matching and ranking, is clearly related to the aspect of object representation. From the descriptions of the systems, we can derive three main types of approaches: (1) the purely 'text based' approach which relies on the 'bag-of-words' representation of objects and applies ranking that is based on TF/IDF, Okapi, or language models[6]. This type of approaches is centered around the use of terms and particularly, weights of terms derived from statistics computed for the text corpus. (2) Weighting properties separately is done by approaches that use models like BM25F to capture the structure of documents (and objects in this case) using a list of fields or alternatively, using mixture language models, which weight certain aspects of an object differently. Since this type of approaches does not consider objects as being flat as opposed to the text-based ones but actually decompose them according to their structure, we call them 'structure-based'. (3) While with this one, the structure information is used for ranking results for a specific query, there are also approaches that leverage the structure to derive query independent scores, e.g. using PageRank. We refer to them as 'query-independent structure-based' (Q-I-structured-based) approaches. To be more precise, the three types discussed here actually capture different aspects of a retrieval model. A concrete approach in fact uses a combination of of these aspects.

Based on the distinction introduced above, Table 1 gives an overview of the systems and their characteristics using the identifiers provided in the original run submissions. A brief description of each system is given below, and detailed descriptions are available at `http://km.aifb.kit.edu/ws/semsearch10/#eva`.

Delaware:*Object representation:* The system from Delaware took all triples having the same subject URI as the description of an object. However, the resulting structure of the object as well as the triple structure were then neglected. Terms extracted from the triples are simply put into one 'bag-of-words' and indexed as one document. *Retrieval model:* Three existing retrieval models were applied for the different runs, namely Okapi for **sub28-Okapi**, language models with Dirichlet priors smoothing **sub28-Dir**, and an axiomatic approach for **sub28-AX**.

DERI: *Object representation:* The Sindice system from DERI applied a different notion of object. All triples having the same subject and also the same context constitute one object description. Thus, the same subject that appears

Participant		Delaware			DERI			KIT	L3S	UMass			Yahoo!		BCN
	Run	sub28-Okapi	sub28-Dir	sub28-AX	sub27-dpr	sub27-dlc	sub27-gpr	sub32	sub29	sub31-run1	sub31-run2	sub31-run3	sub30-RES.1	sub30-RES.2	sub30-RES.3
Object representation	Attribute values	+	+	+	+	+	+	+	-	+	+	+	+	+	+
	Relations	-	-	-	+	+	+	-	-	-	-	-	-	-	-
	Context (+) / Domain (○)	-	-	-	+ ○	+ ○	+ ○	-	○	-	-	-	○	○	○
Retrieval model	Text based	+	+	+	+	+	+	-	+	+	+	-	-	-	-
	Structure-based	-	-	-	-	-	-	+	-	-	-	+	+	+	+
	Q-I-Structure-based	-	-	-	+	+	+	-	-	-	-	-	+	+	+

Table 1. Feature overview regarding system internal object representation and retrieval model

in two different contexts might be represented internally as two distinct objects. Further, the system considered relations to other objects, context information, and URI tokens for the representation of objects. *Retrieval model:* The context information, as well as the relations between objects are used to compute query independent PageRank-style scores. Different parameter configurations have been tested for each run, resulting in different scores. For processing specific queries, these scores were combined with query dependent TF/IDF-style scores for matches on predicates, objects and values.

KIT: *Object representation:* The system by KIT considered literal values of attributes and separately those of the $rdfs : label$ attribute as the entity description. All other triples that can be found in the RDF data for an object were ignored. *Retrieval model:* The results were ranked based on a mixture language model inspired score, which combines the ratio of all query terms to the number of term matches on one literal and discounts each term according to its global frequency.

L3S:*Object representation:* The system by L3S takes a different approach to object representation. Each unique URI, appearing as subject or object in the data set, is seen as an object. Only information captured by this URI is used for representing the object. Namely, based on the observation that some URIs contain useful strings, a URI was splitted into parts. These parts were taken as a 'bag-of-words' description of the object and indexed as one document. Thereby, some provenance information is taken into account, i.e., the domain extracted from the URI. *Retrieval model:* A TF/IDF-based ranking combined with using cosine similarity to compute the degree of matching between terms of the query and terms extracted from the object URI was used here.

UMass: *Object representation:* All triples having the same subject URI were taken as the description of an object. For the first two runs, **sub31-run1** and **sub31-run2**, the values of these triples are just seen as a 'bag-of-words' and no structure information was taken into account. For the third run, **sub31-run3**, the object representation was divided into four fields, one field containing all values of the attribute *title*, one for values of the attribute *name*, a more specific one for values of the attribute $dbpedia : title$ and one field containing the values for all the attributes. *Retrieval model:* Existing retrieval models were

applied, namely the query likelihood model for **sub31-run1** and the Markov random field model for **sub31-run2**. For **sub31-run3**, the fields were weighted separately with specific boosts applied to *dbpedia* : *title*, *name*, and *title*.

Yahoo! BCN: *Object representation:* Every URI appearing at the subject position of the triples is regarded as one object and is represented as one virtual document that might have up to 300 fields, one field per attribute. A subset of the attributes were manually classified into one of the three classes *important*, *neutral*, and *unimportant* and boosts applied respectively. The Yahoo! system took the provenance of the URIs into account. However, not the context but the domain of the URI was considered and similarly to the attributes, were classified into three classes. Relations and structure information that can be derived from them were not taken into account. *Retrieval model:* The system created by Yahoo! uses an approach for field-based scoring that is similar to BM25F. Matching terms were weighted using a local, per property, term frequency as well as a global term frequency. A boost was applied based on the number of query terms matched. In addition, a prior was calculated for each domain and multiplied to the final score. The three submitted runs represent different configurations of these parameters.

Only the top 10 results per query were evaluated, and after pooling the results of all the submissions, there was a total of 6,158 unique query-result pairs. Note this was out of a total of 12,880 potential query result pairs, showing that pooling was definitely required. Some systems submitted duplicate results for one query. We considered the first occurrence for the evaluation and took all following as not relevant. Further, some submissions contained ties, i.e. several results for one query had the same score. Although there exist tie-aware versions of our metrics [7], the *trec_eval* software[13] we used to compute the scores can not deal with ties in a correct way. Therefore we broke the ties by assigning scores to the involved result according to the order of occurrences in the submitted file.

4.2 Evaluation results

Participant	Run	P@10	MAP	**NDCG**
Yahoo! BCN	**sub30-RES.3**	0.4924	0.1919	0.3137
UMass	**sub31-run3**	0.4826	0.1769	0.3073
Yahoo! BCN	**sub30-RES.2**	0.4185	0.1524	0.2697
UMass	**sub31-run2**	0.4239	0.1507	0.2695
Yahoo! BCN	**sub30-RES.1**	0.4163	0.1529	0.2689
Delaware	**sub28-Okapi**	0.4228	0.1412	0.2591
Delaware	**sub28-AX**	0.4359	0.1458	0.2549

Participant	Run	P@10	MAP	**NDCG**
UMass	**sub31-run1**	0.3717	0.1228	0.2272
DERI	**sub27-dpr**	0.3891	0.1088	0.2172
DERI	**sub27-dlc**	0.3891	0.1088	0.2171
Delaware	**sub28-Dir**	0.3652	0.1109	0.2140
DERI	**sub27-gpr**	0.3793	0.1040	0.2106
L3S	**sub29**	0.2848	0.0854	0.1861
KIT	**sub32**	0.2641	0.0631	0.1305

Table 2. Results of submitted Semantic Search engines.

The systems were ranked using three standard information retrieval evaluation measures, namely mean average precision (MAP), precision at 10 (P@10)

[13] `http://trec.nist.gov/trec_eval/`

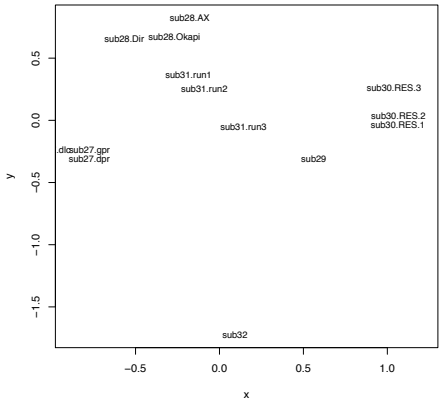

Fig. 1. Visualizing the distances between systems using MDS.

and normalized discounted cumulative gain (NDGC). Refer to [6] for a detailed explanation on these metrics. Table 2 shows the evaluation results for the submitted runs. The third run submitted by Yahoo!, together with the third run of the UMass system, gave the best results. The ordering of the systems changes only slightly if we consider MAP instead of NDCG. Precision at 10 is much less stable as it has been observed in previous evaluations.

It was interesting to observe that the top two runs achieve similar levels of performance with retrieving very different sets of results. The overlap between these two runs as measured by Kendall's τ is only 0.11. By looking at the results in detail, we see that **sub31-run3** has a strong prior on returning results from a single domain, dbpedia.org, with 93.8% of all results from this domain. DBpedia, which is an extraction of the structured data contained in Wikipedia, is a broad-coverage dataset with high quality results and thus the authors have decided to bias the ranking toward results from this domain. The competing run **sub30-RES3** returns only 40.6% of results from this domain, which explains the low overlap. Considering all pairs of systems, the values of τ range from 0.018 to 0.995. Figure 1 visualizes the resulting matrix of dissimilarities (1-τ) using non-metric multi-dimensional scaling (MDS). The visualization manages to arrange similar runs within close proximity. Comparing Figure 1 with Table 2, we can observe that at least in this low dimensional projection there is no obvious correlation between any of the dimensions and the systems' performance, except for the clear outlier **sub32**, which is distant from all other systems in the second (y) dimension and performs poorly. In general, this image also suggest that similar performance can be achieved by quite dissimilar runs.

Figure 3 shows the per-query performance for queries from the Microsoft and Yahoo! data-sets, respectively. Both Figures show the boundary of the first and third quartiles using error bars. It is noticeable that the Yahoo! set is indeed more difficult for the search engines to process, with larger variations of NDCG across both queries and across systems. The performance on queries from the Microsoft log, which are more frequent queries, shows less variation among queries and

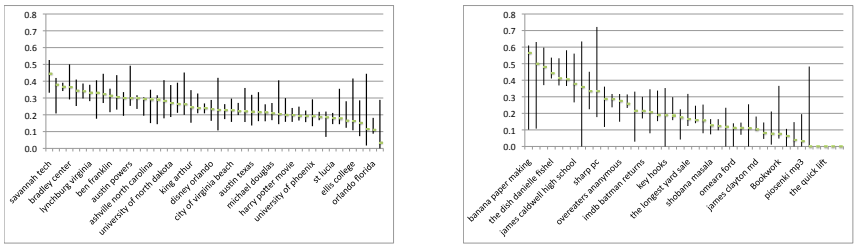

Table 3. Average NDCG for queries from the Microsoft data-set (left) and Yahoo! data-set (right)

between systems processing the same queries. This confirms that popular queries are not only easier, but more alike in difficulty.

4.3 Discussion

The systems submitted to the evaluation represent an array of approaches to semantic search, as shown by the diversity of results. Most participants started with well-known baselines from Information Retrieval. When applied to object retrieval on RDF graphs these techniques yield workable results almost out-of-the-box, although a differential weighting of properties has been key to achieving top results (see the runs from **Yahoo! BCN** and **UMass**).

Besides assigning different weights to properties, the use of 'semantics' or the meaning of the data has been limited. All the participating systems focused on indexing only the subjects of the triples by creating virtual documents for each subject, which is understandable given the task. However, we would consider relations between objects as one of the strong characteristics of the RDF data model, and the usefulness of graph-based approaches to ranking will still need to be validated in the future. Note that in the context of RDF, graph-based ranking can be applied to both the graph of objects as well as the graph of information sources. Similarly, we found that keyword queries were taken as such, and despite our expectations they were not interpreted or enhanced with any kind of annotations or structures. The possibilities for query interpretation using background knowledge (such as ontologies and large knowledge bases) or the data itself is another characteristic of semantic search that will need to be explored in the future.

The lack of some of these advanced features is explained partly by the short time that was available, and partly by the fact that this was the first evaluation of this kind, and therefore no training data was available for the participants. For next year's evaluation, the participants will have access to assessments from this year's evaluation. This will make it significantly easier to test and tune new features by comparing to previous results. We will also make the evaluation software available, so that anyone can generate new pools of results, and thus evaluate systems that are very dissimilar to the current set of systems.

5 Conclusions

We have described the methodology and results of the first public evaluation campaign for ad-hoc object retrieval, one of the most basic tasks in semantic search. We have designed our evaluation with the goals of efficiency in mind, and have chosen a crowd-sourcing based approach. A natural next step will be to perform a detailed analysis of the mechanical-turk produced ground truth.

Our work could be also extended to new data-sets and new tasks. For example, structured RDF and RDF-compatible data can now be embedded into HTML pages in the form of RDFa and microformats, making it a natural next step for our evaluation campaign. This kind of data is used by both Facebook, Google, and Yahoo! for improving user experience. The selection of our queries could be biased toward queries where current search engines fail to satisfy the information need of the user due to its complexity, queries with a particular intents or an increased pay-off etc. We plan to extend the number of public query and data-sets in the future, and given funding we might open our system for continuous submission, while we continue to host yearly evaluation campaigns.

References

1. C. Cleverdon and M. Kean. Factors Determining the Performance of Indexing Systems, 1968.
2. 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*, pages 652–659, New York, NY, USA, 2004. ACM Press.
3. S. Elbassuoni, M. Ramanath, R. Schenkel, M. Sydow, and G. Weikum. Language-model-based ranking for queries on rdf-graphs. In *CIKM*, pages 977–986, New York, NY, USA, 2009. ACM.
4. H. Halpin. A query-driven characterization of linked data. In *WWW Workshop on Linked Data on the Web*, Madrid, Spain, 2009.
5. 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.
6. C. D. Manning, P. Raghavan and H. Schütze Introduction to Information Retrieval Cambridge University Press, 2008,
7. F. McSherry and M. Najork. Computing information retrieval performance measures efficiently in the presence of tied scores. In *ECIR*, Berlin, Heidelberg, April 2008. Springer-Verlag.
8. A. Mikheev, C. Grover, and M. Moens. Description of the LTG System Used for MUC-7. In *MUC-7*,1998.
9. 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.
10. J. Pound, P. Mika, and H. Zaragoza. Ad-hoc Object Ranking in the Web of Data. In *WWW*, pages 771–780, Raleigh, USA, 2010.