Repeatable and Reliable Semantic Search Evaluation

Roi Blanco[b], Harry Halpin[c], Daniel M. Herzig[a,*], Peter Mika[b], Jeffrey Pound[d], Henry S. Thompson[c], Thanh Tran[a]

[a]*Institute AIFB, Karlsruhe Institute of Technology, Karlsruhe, Germany*
[b]*Yahoo! Research, Barcelona, Spain*
[c]*University of Edinburgh, Edinburgh, UK*
[d]*David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada*

Abstract

An increasing amount of structured data on the Web has attracted industry attention and renewed research interest in what is collectively referred to as *semantic search*. These solutions exploit the explicit semantics captured in structured data such as RDF for enhancing document representation and retrieval, or for finding answers by directly searching over the data. These data have been used for different tasks and a wide range of corresponding semantic search solutions have been proposed in the past. However, it has been widely recognized that a standardized setting to evaluate and analyze the current state-of-the-art in semantic search is needed to monitor and stimulate further progress in the field. In this paper, we present an *evaluation framework for semantic search*, analyze the framework with regard to *repeatability* and *reliability*, and report on our experiences on applying it in the *Semantic Search Challenge 2010* and *2011*.

Keywords: Semantic search evaluation, semantic search, structured data, semantic data, Web data, RDF, Web search

1. Introduction

There exist a wide range of semantic search solutions targeting different tasks – from using semantics captured in structured data for enhancing document representation (and *document retrieval* [1–4]) to processing keyword search queries and natural language questions directly over structured data (*data retrieval* [5–7]).

In general, 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 more explicitly defines the meaning that is implicit in the surface syntax. Already in the early days of information retrieval (IR) research, *thesauri* capturing senses of words in the form of concepts and their relationships were used [8]. More recently, the large and increasing amount of structured data that are embedded in Web pages or available as publicly accessible datasets constitute another popular type of semantic structure. The advantage here is that these data are commonly represented in *RDF* (Resource Description Framework), a standard knowledge representation formalism recommended by the W3C. RDF is a flexible graph-structured model that can capture the semantics embodied in information networks, social networks as well as (semi-)structured data in databases. 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). As opposed to the wide range of proprietary models that have been used to capture semantics in the past, RDF provides a standardized vehicle for representation, exchange and usage, resulting in a large and increasing amount of publicly and Web-accessible data that can be used for search (e.g. Linked Data).

The explicit semantics captured by these structures have been used by semantic search systems for different tasks (e.g. document and data retrieval). More specifically, it can be used for enhancing the representation of the information needs (queries) and resources (doc-

*Corresponding author. Tel: +49 (721) 608 46108
Email addresses: roi@yahoo-inc.com (Roi Blanco), H.Halpin@ed.ac.uk (Harry Halpin), herzig@kit.edu (Daniel M. Herzig), pmika@yahoo-inc.com (Peter Mika), jpound@cs.uwaterloo.ca (Jeffrey Pound), ht@inf.ed.ac.uk (Henry S. Thompson), ducthanh.tran@kit.edu (Thanh Tran)

uments, objects). While this helps in dealing with the core task of search, i.e., *matching* information needs against resources, it has been shown that semantics can be beneficial throughout the broader search process [9], from the specification of the needs in terms of queries to matching queries against resources and ranking results, to refining the information needs and up to the presentation and analysis of results.

While there is active research in this field of semantic search, it has been concluded in plenary discussions at the Semantic Search 2009 workshop that the lack of *standardized evaluation* has become a serious bottleneck to further progress in this field. One of the principle reasons for the lack of a standardized evaluation campaign is the *cost* of creating a new and realistically sized "gold-standard" data-set and conducting annual evaluation campaign was considered too high by the community.

In response to this conclusion, we elaborate on an approach for *semantic search evaluation* that is based on crowdsourcing. In this work we show that crowdsourcing-based evaluation is not only *affordable* but in particular, it satisfies the criteria of *reliability* and *repeatability* that are essential for a standardized evaluation framework. We organized public evaluation campaigns in the last two years at the SemSearch workshops and tested the proposed evaluation framework. While the main ideas behind our crowdsourcing-based evaluation may be extended and generalized to the general case (i.e., other search tasks), the kind of semantic search we have focused on in the last two campaigns were keyword search over structured data in RDF. We were motivated by the increasing need to locate particular information quickly and effectively and in a way that is accessible to non-expert users. In particular, the semantic search task of interest is similar to the classic ad-hoc document retrieval (ADR) retrieval task, where the goal is to retrieve a ranked list of (text) documents from a fixed corpus in response to free-form keyword queries. In accordance to ADR, we define the semantic search task of *ad-hoc object retrieval* (AOR) [10], where the goal is to retrieve a ranked list of objects (also referred to as resources or entities) from a collection of RDF documents in response to free-form keyword queries. The unit of retrieval is thus individual entities and not RDF documents, and so the task differs from classic textual information retrieval insofar as the primary unit is structured data rather than unstructured textual data. In particular, we focus on the tasks of *entity search*, which is about one specific named entity, and *list search*, which is about a set of entities.

This paper provides a comprehensive overview of work on semantic search evaluation we did in the last three years and reports on recent progress on semantic search as observed in the evaluation campaigns in 2010 and 2011. It builds on the first work towards this direction on AOR [10], which provided an evaluation protocol and tested a number of metrics for their stability and discriminating power. We instantiated this methodology in the sense of creating a standard set of queries and data (Section 3) which we execute the methodology using a crowdsourcing approach (Section 4). A thorough study on the reliability and repeatability of the framework have been presented in [11]. Lastly, we discuss the application of this framework and its concrete instantiation in the *Semantic Search Challenge* held in 2010 and 2011 (Section 5). Details on these campaigns can be found in [12] and [13], respectively.

Outline This paper is organized as follows. In Section 2 we discuss different directions of related work. In Section 3, we present the evaluation framework, discuss its details and the underlying methodology. How the evaluation framework can be instantiated is detailed in Section 4, where we also examine its reliability and repeatability. In Section 5, we report on two evaluation campaigns, the Semantic Search Challenge, held in 2010 and 2011 and show the applicability of our evaluation framework in the real-world. Finally, we conclude in Section 6.

2. Related Work

We discuss related work from the perspectives of crowdsourcing-based evaluation, semantic search evaluation and search evaluation campaigns.

2.1. Crowdsourcing-based Evaluation

The main difference in using crowdsourcing to "gold standard" evaluation data-set creation in campaigns like TREC [14] is that human judges are no longer a relatively small group of professional expert judges who complete an equal-sized number of assessments, but large group of non-experts who may complete vastly differing numbers of assessments and may not actually have the required skill-set (such as command of English) to complete the task or be completing the task honestly. Earlier work in using crowdsourcing for information retrieval demonstrated quick turn-around times and the ability to have a much higher number of judges than previously thought possible [15]. This has led to a rapidly-expanding number of applications of crowdsourcing evaluation data sets to a wide range of information retrieval tasks such as XML-based retrieval [16].

Crowdsourcing has also been expanded successfully to related areas, such as machine translation [17].

In this vein, our primary contribution is in demonstrating the repeatability of crowdsourcing judgments in creating evaluation data sets, even when entirely different sets of judges are used on the same task over long periods of time, a necessary feature for running large-scale campaigns for novel information retrieval tasks on an annual basis. Previous work on crowdsourcing evaluation campaigns, such as work on replicating image labelling in ImageCLEF[18], has focused on determining the reliability of the judges over small subsets of the original campaign, but has not tested whether the evaluation campaign is repeatable over large time intervals (i.e., months or years), only inspecting differences over small amounts of time (4 days) and not comparing the judges performance over time to each other, but aggregating all judgments.

Previous work [15, 18] in general has focused on comparing crowdsourcing judgments to that of experts on existing campaigns with well-known "gold standards," not boot-strapping new evaluation campaigns for new search tasks where there are multiple competing but unevaluated search systems, such as in semantic search. Another goal of our work is to demonstrate the use of crowdsourcing for a large-scale evaluation campaign for a novel search task, which in our case is ad-hoc object retrieval over RDF. Many semantic search systems of this type, such as [5, 6, 19], have appeared in the past few years, but none have been evaluated against each other except on a very small scale. Semantic search systems are a subset of information retrieval systems, and thus it would be natural to apply existing IR benchmarks for their evaluation in a large-scale campaign.

It is of course possible to use crowd-sourcing for evaluation-type tasks without paying for participation, for example as part of a game [20] or as a side-effect of robot-blocking [21]. The short turn-around time required for our exercise, in the context of a competition, plus its scale, ruled this kind of approach out in this instance.

2.2. Semantic Search Evaluation

Especially through the series of SemSearch workshops, we observed a strong need for a standardized evaluation framework. To the best of our knowledge, we are the first to propose an evaluation framework and methodology as well as organizing the campaigns for participants to evaluate their semantic search systems. There are two difficulties in applying the ad-hoc document retrieval methodology directly to semantic search and the object retrieval problem in particular, as identified in [10]. The first and most apparent problem is that not all semantic search engines perform document retrieval, but rather retrieve knowledge that is already encoded in RDF, where factual answers may be found by aggregating or linking knowledge across RDF data, e.g. [22]. This is a clear difference to 'entity search' tracks such as the TREC Entity Track [14] or the INEX Entity Ranking Track [23]. With respect to addressing keyword retrieval on structured data, there is also existing work in the database literature (e.g., [24]), but this field of research has not produced a common evaluation methodology that we could have adapted. Second, in semantic search the unit of retrieval and thus the way to evaluate the results is dependent on the type of query. In turn, the types of queries supported may vary from search engine to search engine. By reducing the broad problem of semantic search to that of keyword-based ad-hoc object retrieval (i.e. retrieving objects given in RDF with relevant factual assertions connected as a property by a single link), we could invite multiple systems to our campaign, as most semantic search systems have this baseline feature. More complex query and result processing relies upon first retrieving a baseline of relevant objects, and so this baseline should be evaluated first.

2.3. Evaluation Campaigns

The Semantic Search Challenge differs from other evaluation campaigns on entity search. In comparison to the TREC 2010 Entity Track [25], the SemSearch Challenge searches over structured data in RDF rather than text in unstructured web-pages and features more complex queries. Likewise, in comparison to the INEX Entity-Ranking task [26], SemSearch focusses on RDF as opposed to XML as a data-format, and searches for relevance over entire RDF descriptions, not passages extracted from XML. Unlike the QALD-1 Question Answering over Linked Data [27] task, our queries were not composed of hand-crafted natural language questions built around particular limited data-sets such as DBPedia and MusicBrainz (i.e. RDF exports of Wikipedia and music-related information), but of both simple and complex real-world queries from actual query logs. The use of queries from actual Web search logs is also a major difference between our competition and all aforementioned competitions such as TREC and INEX. Keyword search over structured data gets also more attention in the database community [28] and an evaluation framework was recently proposed [29], but an standardized evaluation campaign is not yet available.

3. Evaluation Framework

In the Information Retrieval community the *Cranfield* methodology [30, 31] is the de-facto standard for the performance evaluation of IR-systems. The standardized setting for retrieval experiments following this methodology consists of a document collection, a set of topics and relevant assessments denoting which documents are (not) relevant for a given topic. We adapted this methodology to semantic search. In this section, we describe the data collection used in our evaluation framework and the query sets, which we developed for the Semantic Search Challenge in 2010 and 2011. How we obtained relevance assessments will be described in detail in Section 4.

3.1. Data Collection

A standard evaluation data collection should be not biased towards any particular system or towards a specific domain, as our goal is to evaluate general purpose entity search over RDF data. Therefore, we needed a collection of documents that would be a realistically large approximation to the amount of RDF data available 'live' on the Web and that contained relevant information for the queries, while simultaneously of a size that could be manageable by the resources of a research groups. We chose the 'Billion Triples Challenge' (BTC) 2009 data set, a data-set created for the Semantic Web Challenge [32] in 2009. The dataset was created by crawling data from the Web as well as combining the indexes from several semantic web search engines. The raw size of the data is 247GB uncompressed and it contains 1.4B RDF statements describing 114 million entities. The statements are composed of *quads*, where a quad is a four tuple comprising the four fields *subject*, *predicate*, *object*, as is standard in RDF, but also a URI for *context*, which basically extends a RDF triple with a new field giving a URI that the triples were retrieved from (i.e. hosted on). There was only a single a modification necessary for using this data-set for entity search evaluation which was to replace RDF blank nodes (an existential variable in RDF) with unique identifiers so that they can be indexed. Details of the dataset are given in Table 1.

Billion Triple Challenge 2009 Dataset	
RDF triples	1.4 billion
Size	247GB uncompressed
Download	`http://km.aifb.kit.edu/ws/dataset_semsearch2010`
Description	`http://vmlion25.deri.ie/`

Table 1: Statistics on the data collection

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. The type of result expected varies 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 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 [33].

3.2.1. Queryset 2010

In order to support our evaluation, Yahoo! released a new query set as part of their WebScope program[1],

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

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.[2] 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 [10]. We have selected a sample of 50 queries from the Microsoft log. In this case we have pre-filtered queries automatically, eliminating ones where no entities were found with the Edinburgh MUC named entity recognizer [31], 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. The 2010 query set is available at `http://km.aifb.kit.edu/ws/semsearch10/Files/finalqueries`. Ten random queries of the set are shown in Table 2.

james caldwell high school
44 magnum hunting
american embassy nairobi
city of virginia beach
laura bush
pierce county washington
university of north dakota
kaz vaporizer
david suchet
fitzgerald auto mall chambersburg pa
mst3000

Table 2: Examples queries from the 2010 Entity Query Set.

3.2.2. Querysets 2011

In 2011, the Semantic Search Challenge comprised two tracks. The Entity Search track is identical in nature to the 2010 challenge. However, we created a new set of queries for the entity search task based on the Yahoo! Search Query Tiny Sample v1.0 dataset. We selected 50 queries which name an entity explicitly and may also provide some additional context about it, as described in [10].

In the case of the List Search track, the second track of the 2011 challenge, we hand-picked 50 queries from the Yahoo query log as well as from TrueKnowledge 'recent' queries.[3] The queries describe a closed set of entities, have a relatively small number of possible answers (less than 12) which are unlikely to change.

Although many competitions use queries generated manually by the participants, it is unlikely that those queries are representative of the kinds of entity-based queries used on the Web. Therefore, we manually selected queries by randomly selecting from the query logs and then manually checked that at least one relevant answer existed on the current Web of linked data.

Table 3 shows examples from the query sets for both tracks. The entire query sets are available for download.[4]

08 toyota tundra
Hugh Downs
MADRID
New England Coffee
PINK PANTHER 2
concord steel
YMCA Tampa
ashley wagner
nokia e73
bounce city humble tx
University of York

gods who dwelt on Mount Olympus
Arab states of the Persian Gulf
astronauts who landed on the Moon
Axis powers of World War II
books of the Jewish canon
boroughs of New York City
Branches of the US military
continents in the world
standard axioms of set theory
manfred von richthofen parents
matt berry tv series

Table 3: Examples queries from the 2011 Entity Query Set (left) and 2011 List Query Set (right).

4. Reliability and Repeatability of the Evaluation Framework

Advances in information retrieval have long been driven by evaluation campaigns using standardized collections of data-sets, query workloads, and most importantly, result relevance judgments. TREC (Text REtrieval Conference) [34] is a forerunner in IR evaluations, but campaigns also take place in specialized forums like INEX (INitiative for the Evaluation of XML Retrieval) [23] and CLEF (Cross Language Evaluation Forum). The main premises of these campaigns is that a limited and controlled set of human *experts* decide the correctness of a given set of results, which will be used as a ground truth for evaluating the performance

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

[3] `http://www.trueknowledge.com/recent/`

[4] `http://semsearch.yahoo.com/datasets.php`

of different systems [34]. Early evaluation campaigns targeted relatively narrow domains and used small collections, where evaluations using a small number of queries provided robust results. Moving to the open domain of the Web resulted in significantly larger heterogeneity of data sources and an increase in the potential information needs (and so diverse tasks) that need to be evaluated. Current research in campaigns (like TREC) and information retrieval evaluation in general focus primarily on the following goals:

Repeatability - As observed by Harter [35], there can be substantial variation among different expert judges performing the same task. If evaluation is to drive the next generation of search technologies, it is important to validate that relevance assignment is a repeatable process. This fundamental requirement exacerbates the scalability problem, because the agreement between assessors needs to be tested not only for each new search task, but also for each set of judges that have been employed (agreement is a measure of the extent to which judges are interchangeable). However, outsiders who would like to validate an experiment will typically not have access to the original judges (or those judges may not be available or willing to repeat experiments at later times).

Reliability - The expert judges employed by campaigns such as TREC [36] are expected to be sufficiently reliable to produce a ground truth for evaluation. However, setting up new "tracks" for novel search tasks is often not feasible or expedient, due to the time and effort it takes to set up such tracks and the limited resources of the organizers. In such cases, researchers need to set up their own evaluation and seek replacements for experts, training others to be judges of their work, where training is often nothing more than providing a description of the task.

How can researchers create repeatable and reliable evaluation campaigns that scale over the number of new tasks brought about by the Web? An increasingly popular way of evaluating novel search tasks is the approach known as *crowdsourcing*. Crowdsourcing is a method of obtaining human input for a given task by distributing that task over a large population of unidentified human workers. In the case of building a search evaluation collection, crowdsourcing means distributing relevance judgments of pooled results over this crowd. The advantage of the crowd is that it is always available, it is accessible to most people at a relatively small cost, and the workforce scales elastically with increasing evaluation demands. Further, platforms such as Amazon Mechanical Turk[5] provide integrated frameworks for running crowdsourced tasks with minimal effort. We show how crowdsourcing can help execute an evaluation campaign for a search task that has not yet been sufficiently addressed to become part of a large evaluation effort such as TREC: ad-hoc Web object retrieval [10], for which we created a standard data set and queries for the task of object retrieval using real-world data, and the way we employed Mechanical Turk to elicit high quality judgments from the noise of unreliable workers in the crowd. The queries, index used, and results of the evaluation campaign are also publicly available for use in the evaluation of web-object retrieval systems.[6]

There are two research questions that must be answered for crowdsourcing to be used systematically in evaluation campaigns. First, are evaluation campaigns with crowdsourced workers **repeatable**, such that the resulting ranking of systems is the same for different pools of crowdsourced judges over a period of time? Second, are crowdsourced workers **reliable**, such that differences between experts and crowdsourced workers do not change the resulting ranking of the systems? As our primary contribution, we experimentally demonstrate the repeatability of our search system evaluation experiment using crowdsourcing. We also test the reliability of judges who are not task or topic-experts, which has been questioned in previous work [37], as crowdsourced workers do not have access to the original information need and may lack specialized training or background knowledge possessed by experts. The case of Mechanical Turk provides an extreme where the judges are not only likely to be untrained and non-expert, but they also sign up for payment and so have an incentive to "cheat" in order to gain monetary reward. Therefore, we repeat our evaluation and assess whether the results from the original campaign can be reproduced after six months with a new set of crowdsourced judges, and whether those results correspond to what we would have obtained using a more traditional methodology employing expert judges. We also explore the effect of different numbers of judges per result on the quality of judgments. Finally, we analyse the robustness of three popular information retrieval metrics under crowdsourced judgments. The metrics studied are discounted cumulative gain (NDCG), mean average precision (MAP), and precision at k (P@k). To the best of our knowledge, we are the first to analyze the repeatability of crowdsourcing in a real-world evaluation campaign.

[5] `http://www.mturk.com`

[6] `http://semsearch.yahoo.com`

4.1. Crowdsourcing Judgments

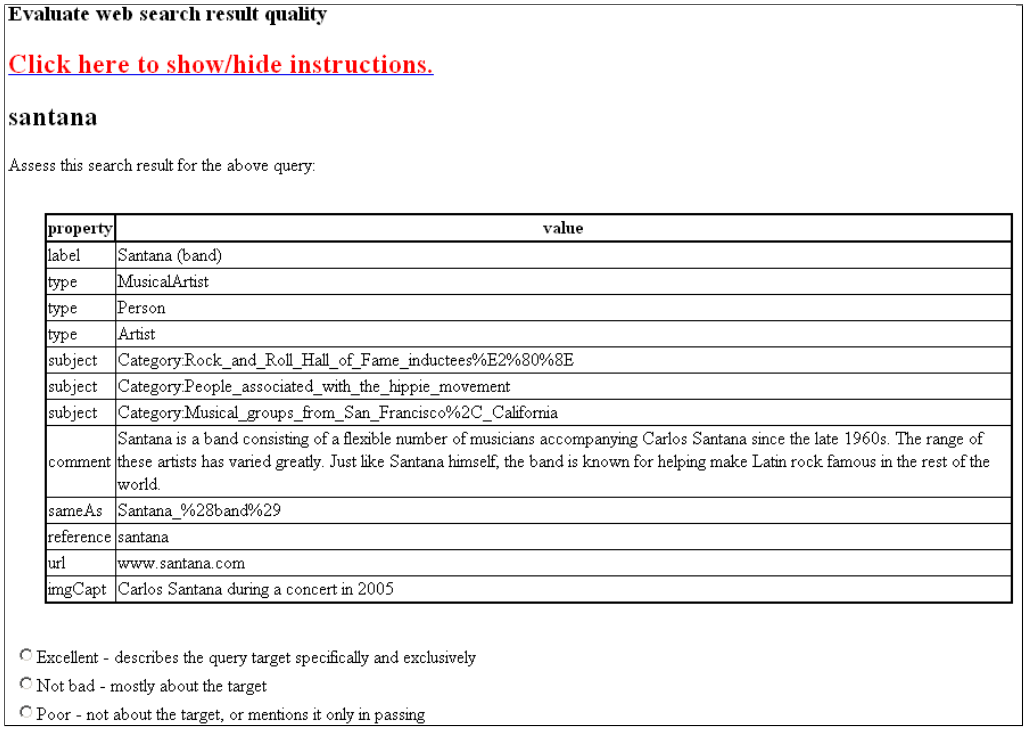
In this Section, we report how we used Amazon Mechanical Turk to assess the relevance of search results and describe the different sets of assessments we obtained for the evaluation. Using Mechanical Turk, tasks - called Human Intelligence Tasks (HITS) - are presented to a pool of human judges known as 'workers' who do the task in return for very small payments. Amazon provides a web-based interface for the workers that keeps track of their decisions and their payments. Because *anyone* can sign up to be a worker, we had to present each result for judgement in a way comprehensible to non-expert human judges. It was not an option to present the data in the native syntactic format of RDF such as RDF/XML or N-Triples, because they are too complex for average users, especially with the use of URIs as opposed to natural language terms for identifiers in RDF. In practice, semantic search systems use widely varying presentations of search results, sometimes tailored to particular domains. However, the rendering of results could possibly affect the valuation given by a judge. Allowing each participant to provide their own rendering would make it difficult to separate the measurement of ranking performance from effects of presentation, and would also eliminate the ability to pool results which reduces the total number of judgments needed.

For the purpose of evaluation, we have created a rendering algorithm to present the results in a concise, yet human-readable manner without domain-dependent customizations (see Figure 1). First, for each subject URI, all properties and objects were retrieved. Then the last rightmost hierarchical component of the property URI, often referred to as the local name, was used as the label of the property after tokenization. For example, the property `http://www.w3.org/1999/02/22-rdf-syntax-ns/type` was presented to the judge simply as `type`. A maximum of twelve object properties were displayed to the judge, based on previous experience that fitting the whole task on a single page improved participation rates. Preference was given to a few well-known property types defined in the RDF and RDF Schema namespaces, followed by custom-defined properties presented in the order retrieved from the dataset. In order to keep the amount of information given constant across judges and facilitate timely completion of the task, the URIs were not clickable and the judges were instructed to assess using only the information rendered, as to make the task of ad-hoc object retrieval directly comparable to tasks such as ad-hoc document retrieval. During the evaluation, we encountered the problem that some of the retrieved URIs only appear as objects, resulting in an empty display. Of the 6,158 URIs, a small minority of URIs (372) had triples only in the object position. For the current evaluation, we have ignored these results. Workers were given three options to judge each result: "Excellent - describes the query target specifically and exclusively", "Not bad - mostly about the target", and "Poor - not about the target, or mentions it only in passing." Note that we used the human-friendly labels "Excellent", "Not bad" and "Poor" for relevant, somewhat relevant and irrelevant results. We did not provide instructions to emphasize any particular properties (such as the "categories" in Figure 1), leaving the judgment to be based on general purpose judgment combining background knowledge about the entities and all of the displayed information. In the following, any grade higher than "Poor" will be considered as "Relevant" for metrics that compute performance values over binary relevance judgments (MAP and P@10).

4.2. Quality Assurance and Costs of Evaluation

In order to ensure quality in the presence of possible low-quality workers, 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 list of results. These gold-standard results were results from queries distinct from those used by the workers and have been manually judged earlier by an expert in RDF and information retrieval as being obviously 'relevant' or 'irrelevant'. For each HIT, there was both a gold-standard relevant and gold-standard irrelevant result included. These gold-standard results enabled the detection of workers who were not properly doing their task, as can be done by monitoring the average performance of judges on the gold-standard results hidden in their HITs. It is a common occurrence when using paid crowdsourcing systems for bogus workers to try to 'game' the system in order to gain money quickly without investing effort in the task, either by using automated bots or simply answering uniformly or randomly. Note that while we chose our gold-standards manually since we were evaluating a new task, one could in future campaigns use results with high inter-annotator agreement as new gold standards or apply machine learning techniques to predict spammers [38]. Amazon Mechanical Turk allows payment to be withheld at the discretion of the creator of the HIT if they believe the task has not been done properly.

Before publishing the final tasks, we had done small-scale experiments with varying rewards for the workers. Mason and Watts have already determined previously that increased financial incentives increase the quantity,



Evaluate web search result quality

Click here to show/hide instructions.

santana

Assess this search result for the above query:

property	value
label	Santana (band)
type	MusicalArtist
type	Person
type	Artist
subject	Category:Rock_and_Roll_Hall_of_Fame_inductees%E2%80%8E
subject	Category:People_associated_with_the_hippie_movement
subject	Category:Musical_groups_from_San_Francisco%2C_California
comment	Santana is a band consisting of a flexible number of musicians accompanying Carlos Santana since the late 1960s. The range of these artists has varied greatly. Just like Santana himself, the band is known for helping make Latin rock famous in the rest of the world.
sameAs	Santana_%28band%29
reference	santana
url	www.santana.com
imgCapt	Carlos Santana during a concert in 2005

○ 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

Figure 1: A sample HIT for semantic search evaluation.

but not the quality, of work performed by participants [39]. Thus our approach was to lower the payment to workers down to the price where the speed of picking up the published tasks was still acceptable. When our results were published via Amazon Mechanical Turk, workers were paid $0.20 per HIT. In the first experiment reported here 65 workers in total participated in judging a total of 579 HITs or 1737 assignments (3 assignments per HIT), covering 5786 submitted results and 1158 gold-standard checks. (Note that of these only a subset of 4209 results and 842 checks is relevant here, being those which were also evaluated in MT2 and EXP, see below). Three workers were detected to be answering uniformly or randomly, and their work (a total of 95 assignments) was rejected and their assignments returned to the pool for another worker to complete. Two minutes were allotted for completing each HIT. On average the HITs were completed in 1 minute, with only two complaints that the allotted time was too short. This means that workers could earn $6-$12 an hour by participating in the evaluation. The entire competition was judged within 2 days, for a total cost of $347.16. We consider this both fast and cost-effective. Given that this cost includes not only payments to judges, but also the provision of the entire testing infrastructure, this compares very favourably with the likely cost of recruiting, managing and paying graduate students for the same task.

To study repeatability of our evaluation campaign we have re-evaluated the relevance of the search results returned by our test systems using a second set of workers. This second experiment has been performed six months after the initial evaluation using the exact same procedure. In the following, we will refer to the original set of assessments as MT1 and the repeated set of assessments as MT2. For MT1 there were 64 judges in total. The top four judges did 131 HITs and did not differ from the experts on the gold-standard items, with the overall percentage of mistakes over the 2176 gold-standard items in those 1088 HITs was 3.2%. For MT2 there were 69 judges in total. The top five judges did 165 HITs and did not differ at all from experts on the gold-standard items, and the overall percentage of mistakes with regards the 1662 gold-standard items in those 831 HITs was 4.5%. For future campaigns items with a high inter-annotator reliability could be used to chose more gold-standard items.

To study the reliability of our crowdsourced judgments, we also created an "expert" set of relevance judgments over standard HITs that were not gold-standard items. Unlike repeatability, reliability concerns the ability of Mechanical Turk to reproduce a ground truth pro-

vided by experts. In our case, the authors of this paper have provided the ground truth by re-evaluating the same subset used in MT2. As this is a significant effort, we have used only one judge per HIT for re-evaluating the entire set of 4209 results, in 421 HITs of 10 results (leaving out the known-good and known-bad gold-standard check items). The resulting dataset is referred to as EXP herein.

For all of MT1, MT2, and EXP, we report here on the exact same set of queries and results. Some participants submitted more than one set of results (outputs from their system in differing configurations), of which we used the best submission of each of the competitor systems for testing repeatability. In total there were 6 competing systems with one submission each, which will be described in Section 5.1.1. Each result of every submission was judged by 3 crowdsourced workers, with systems results being judged to a depth of 10, given that it was a new unstudied task. We broke ties by taking the majority vote, except where the three judges each gave a different judgment, in which case we chose the middle, "Not Bad" assessment. In EXP, as mentioned above, each result was judged by a single expert, but a subset of 30 results were judged by three experts to determine intra-expert reliability.

Although the procedure for MT2 was the same as for MT1, the intervening six months appear to have seen a significant change in the worker pool: monitoring worker time-to-complete and performance on the known-good and known-bad gold-standard results revealed a total of 14 bogus workers for MT2, who completed a total of 1471 assignments between them before they were detected and blocked and their assignments returned to the pool. This change from 5% of assignments rejected in MT1 to 54% of assignments rejected in MT2 may indicate a significant increase in the number of bogus workers, and underlines the importance of including known-good and known-bad data in every HIT.

4.3. Analysis of Results

We seek to answer the following in our experiments:

- **Repeatability** Are judges really interchangeable?
  - Can we expect anonymous crowdsourced workers to agree on judgments?
  - Can we expect repeated experiments to produce the same results in terms of relevance metrics and the rank-order of the evaluated systems?

This requires also confirming previous results [16]:

- **Reliability** Can crowdsourced workers reliably reproduce the results we would have obtained if we were using expert judges?
  - Are the same items scored similarly by workers and experts?
  - Can worker evaluations produce the same results in terms of our relevance metrics and the rank-order of the evaluated systems?

We will use as parameters both the evaluation metric, the number of assessors per item and the relevance scale used. In particular, we would like to find out the following:

- Which of our three evaluation metrics (MAP, NDCG, P@10) are more robust to changing the pool of workers, and when replacing experts with workers?
- Do we obtain better results with increasing number of assessments per item?
- Do our results hold for both binary and ternary scale assessment?

4.3.1. Repeatability

As previously discussed, in IR evaluation the notion of repeatability is tied to measuring the extent to which judges are interchangeable. The argument being that if we show judges from a particular pool of assessors are interchangeable, the experiment can be repeated with any subset of judges from the pool: the judges will agree on the relevancy of items to be judged, which will be reflected in the metrics to be computed, and the eventual ranking of the competing systems.

The most common measures of inter-annotator agreement in IR evaluations are Cohen's κ for the case of two judges, and Fleiss's κ for the case of multiple judges, which has a free-marginal version [18]. While we report inter-annotator agreement, we note that the applicability of standard metrics to the case of crowdsourced workers can be questioned. The reason is that although we have a fixed number of workers for each HIT, in the crowdsourcing scenario the workers select the tasks, and thus they are not necessarily the *same* workers who assess each item. Figure 2 shows the number of items judged by each worker in our first experiment with Mechanical Turk. In the case of traditional expert-based evaluation, this distribution would be flat as each expert would assess the same items. In our case, each worker may assess a different number of the total set of HITs. Some workers assess a large number of HITs, with the most diligent worker going through 273 HITs, while a long

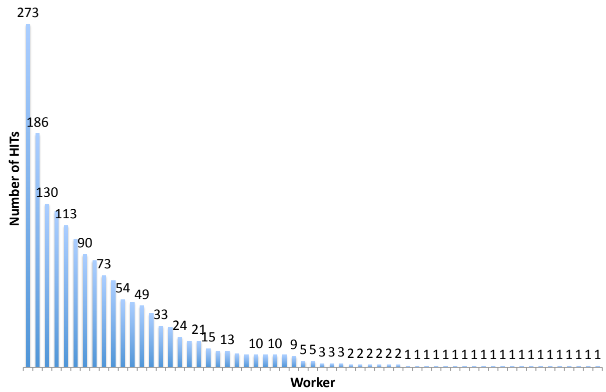


Figure 2: Workers ordered by decreasing number of items assessed.

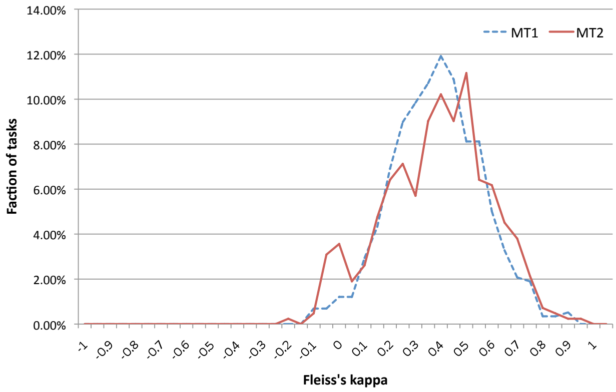


Figure 3: Agreement between workers.

tail of workers worked on a single task only. This long tail is especially problematic since there is much less data about these workers on which to base reliability tests.

Based on our knowledge of the related work, it seems that there is not yet consensus as to how to account for this deficiency [40] and the question of reliability is sometimes ignored altogether [41]. We believe the most prudent way to proceed is to report the distribution of Fleiss' κ values considering all HITs as individual assessments of a small number of 12 items. In Figure 3 we show this distribution for our first and second experiment. As the Figure shows, the level of agreement is very similar. The average and standard deviation are 0.36±0.18 for the first experiment (MT1) versus 0.36±0.21 for MT2. In fact, the difference between the average agreement appears at the fourth digit, strongly supporting the idea of a homogeneous pool of workers. We achieve slightly higher levels of agreement for binary relevance (with somewhat relevant and relevant judgments counted both as relevant), 0.44±0.22 and 0.47±0.25. 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.

Agreement numbers are not easy to interpret even in the context of related work, and agreement is only a proxy for a repeatable evaluation: what we are ultimately after is whether different pools of workers used in different experiments lead to the same results in terms of evaluation metrics, and ultimately the same ordering of the evaluated systems. Figure 4 shows Mean Average Precision (MAP) scores for the different systems using the two different evaluation sets obtained via Mechanical Turk (MT1 and MT2). The results are also included

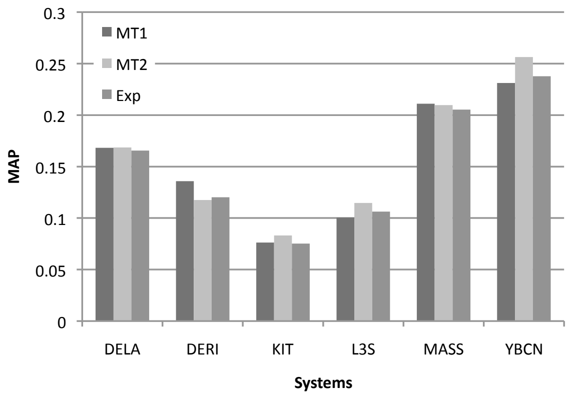


Figure 4: Mean average precision (MAP) for the systems using different test sets.

in Table 5. We can see that the scores are close in value, and in fact there is no change to the rank-order of the systems. The result holds for both binary and ternary scale, and for both MAP, P@10 and NDCG. Broadly, this confirms our hypothesis that crowdsourced ad-hoc evaluation is repeatable. The relative change in scores across the two sets, for all systems in average, is 7.85% for MAP, 4.24% for NDCG and 6.87% for P@10. This gives us a first indication that two systems would need to be very close in performance in order to change places in the ranking produced by repeated experiments.

In fact, Mechanical Turk gives surprisingly robust results with just a single assessment per item. We have tested this by subsampling, i.e. selecting randomly a single assessment for each item from the six assessments we have collected in total. We have repeated this 100 times and computed the min, max, mean and standard deviation of our metrics. Figure 5 shows the min,

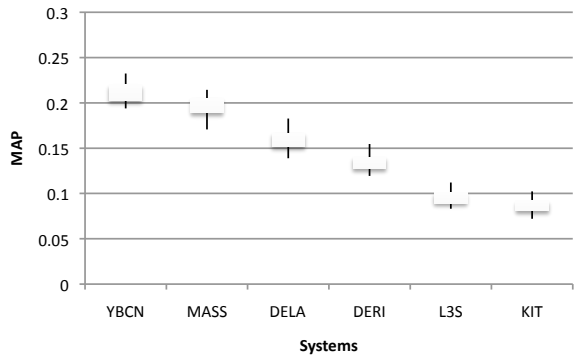


Figure 5: Mean average precision (MAP) for the systems using different test sets and a single worker.

max, and the range of one standard deviation from the mean for each system, using MAP as the metric. This figure furthermore shows that even one standard deviation intervals provide different ranges for the different systems and effectively separate them. Though the score of a system in a particular sample may surpass the score of an overall inferior system, such cases would be rare. Note that there is a particular robustness to Mechanical Turk. Though conventional wisdom would certainly be against running an evaluation with a possibly unreliable single judge, in the case of crowdsourcing the assessments will come from not a single expert judge for all the results, but multiple workers. These workers may be individually unreliable, but each will judge a small number of items. When considering three judges, see Figure 6, the intervals around the mean get even tighter.

The decrease of standard deviation around the mean is also shown in Figure 7. This Figure shows the standard deviation on the y-axis, for different numbers of workers (x-axis), and using different metrics. We see that P@10 benefits the most from increasing the number of workers and that adding more workers decreases the standard deviation between workers.

4.3.2. Reliability

Repeatable evaluations require that each evaluation be reliable, and while work such as Alonso et al. [16] has shown that crowdsourced judges can be reliable in information retrieval tasks, we should show that this reliability holds over repeated experiments. We measured the agreement between expert judges on a subset of the items (30 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, respectively. The level of agreement is thus higher for expert judges, with comparable deviation. For expert judges, there is practically no difference between the two- and three-point scales,

Set	Total items	Irrelevant	Somewhat R.	Relevant
MT1	4209	2593	970	646
MT2	4209	2497	975	737
EXP	4209	2847	640	722

Table 4: Scoring patterns in different evaluation sets.

meaning that expert judges had much less trouble using the middle judgment.

Moving on to comparing expert reliability with crowdsourced judgements from MT1 and MT2, Table 4 shows that again different sets of workers behave very similarly, though different from the experts on the whole. Fleiss's κ is similar with 0.412 between MT1 and experts, and 0.417 between MT2 and experts. In particular, experts are more pessimistic in their scoring, marking irrelevant many of the items that the workers would consider somewhat relevant.

This effect is also visible in Figure 8, which shows the assessments of the two worker sets compared to the assessments of the experts for the three assessment options. Whereas the two worker sets display similar behaviour compared to each to other, the difference towards more positive assessments compared to the experts can be observed. This may suggest that crowdsourced judgments cannot replace expert evaluations. Based on comments and the data, the source of this effect is likely the fact that experts understood "describes the query target specifically and exclusively" to be much of a more sharp distinction about objects than workers. An expert would note that the IMDB article about a movie featuring actor David Suchet would not be considered 'relevant', while workers would often judge that result as relevant if the query asked for David Suchet.

Looking at agreement rate in other settings, such a κ of 0.55 at TREC 2005 on sentence relevance at TREC 2004 Novelty Track [42], our experts are clearly reliable, with agreement ratings of 0.57 (binary scale) and 0.56 (ternary scale). The reliability of non-expert crowdsourced judges of 0.36 in our experiment then appears to be less than ideal. However, does it change the ranking of the systems? This would be the ideal test of how far reliability has to degrade in order to impact an evaluation campaign.

Even if the level of agreement is higher amongst expert judges, if the ranking of the systems does not change when non-experts are employed, then a crowdsourcing approach is still reliable enough for the task (even if their reliability is strictly speaking relatively lower than expert judges). The relative change in scores

	MAP			NDCG			P@10		
System	MT1	MT2	EXP	MT1	MT2	EXP	MT1	MT2	EXP
YBCN	0.23	0.26	0.24	0.35	0.38	0.33	0.48	0.54	0.45
MASS	0.21	0.21	0.21	0.34	0.34	0.33	0.48	0.51	0.40
DELA	0.17	0.17	0.17	0.29	0.27	0.28	0.41	0.43	0.35
DERI	0.14	0.12	0.12	0.24	0.24	0.22	0.39	0.36	0.30
L3S	0.10	0.11	0.11	0.20	0.21	0.20	0.28	0.30	0.24
KIT	0.08	0.08	0.08	0.15	0.14	0.15	0.26	0.28	0.23

Table 5: Evaluation results using different evaluation sets and metrics.

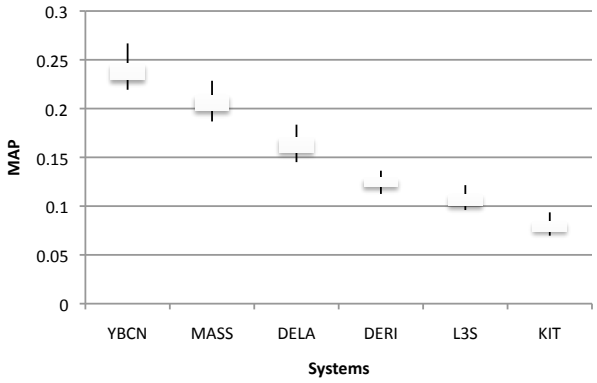


Figure 6: Mean average precision (MAP) for the systems using different test sets and three workers.

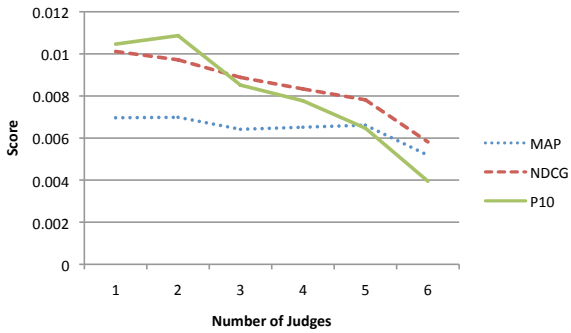


Figure 7: Average standard deviation around the mean for different numbers of workers and using different metrics.

when going from experts to workers (moving from EXP to three-samples of MT1 and MT2), for all systems on average, and using three judgments, is 1.8% for MAP, 3.5% for NDCG and 12.8% for P@10 (see also Table 5). These are comparable changes to what we have seen when moving from one worker set to another, but the changes are mostly positive, with notable increases in P@10 when changing from experts to workers. In particular, the increase in somewhat relevant scores explains the increase of the binary P@10 measure. Somewhat relevant results (counted as relevant for the binary measures) that are coming in at lower ranks boost P@10 more than MAP and NDCG, which are less sensitive to changes in the lower ranks. While the reliability of non-expert judges is lower than expert judges, the reliability of non-expert judges is still sufficient for ranking systems in the evaluation.

Figure 4 illustrates the performance values for MAP for the different systems using the two MT evaluation sets and the expert judgments. The values are not only close, but in fact again the obtained values for the experts produce the same rank-order of the systems as with any of the MT evaluation sets.

As in the case of repeatability, we might ask whether crowdsourced assessments become more reliable when adding more judges. We have already shown in Figure 7 that increasing the number of workers decreases their standard deviation and increases the reliability of workers, and this trend seems to continue beyond 6 workers. Figure 9 shows the deviation resulting from using the workers' assessments instead of the expert assessments, in particular the average relative change in our metrics for subsamples, for different numbers of workers. We can see a clear benefit to using three workers instead of 1 or 2 workers, but there is comparatively less benefit from employing more than three judges. Figure 10 shows the same for MAP and NDCG using the average values of Kendall's τ between the subsamples of worker judgments and the expert assessments. This value of τ is already very close to one for three judges independent of the metric. While intra-worker reliability increases as the number of workers increase, adding more than three workers will lead to a higher number of disagreements with expert judges.

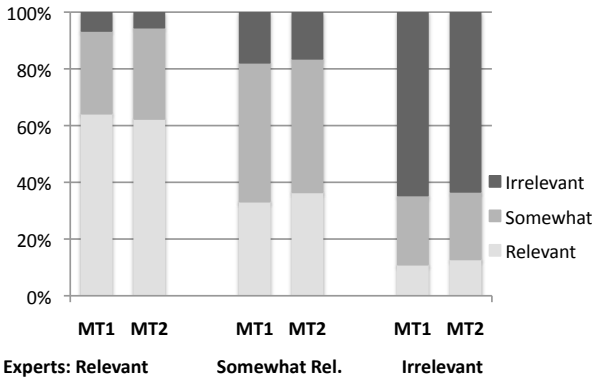


Figure 8: Assessments of the two workers' sets compared to the experts' assessments for the three assessment options.

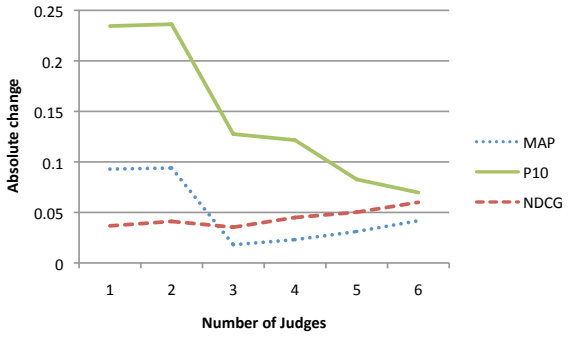


Figure 9: Average deviation of sample means from the expert assessments.

4.4. Conclusions on Reliability and Repeatability

With the advent of crowdsourcing platforms like Amazon Mechanical Turk, creating a "gold standard" evaluation data set of relevance judgments for new kinds of search tasks is now cheap, scalable, and easy to deploy. We have shown how to quickly boot-strap a repeatable evaluation campaign for a search task that has not previously been systematically evaluated, such as the object information retrieval task in semantic search, using Mechanical Turk. However, are such crowdsourced evaluation campaigns trustworthy? Are the relevance judgments of crowdsourced judges both reliable compared to experts and can such judgments be repeated with entirely different crowdsourced judges over time?

Regarding the **repeatability** of such crowdsourced judgments, we have shown that the level of agreement is the same for two pools of crowdsourced judges even when the evaluation is repeated after six months. Repeating an evaluation using crowdsourcing after six months led to the same result in evaluation metrics and

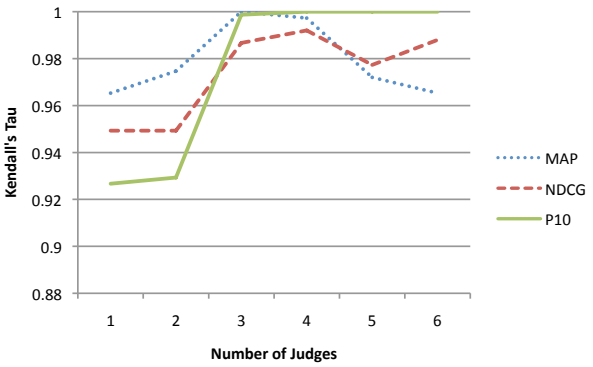


Figure 10: Kendall's Tau between workers and experts for different number of assessments per item.

the rank-order of the systems being unchanged. Concerning the *reliability* of crowdsourced judgments, we have observed that experts in general rate more results negative than crowdsourced judges. This is likely due to the object retrieval task and the time pressure on workers, as experts were more adept at discriminating between queries exclusively about an object to ones simply mentioning an object given time limits. However, the rank ordering of systems does not change when moving from experts to crowdsourced workers. Three judges seems to be a sufficient number and, surprisingly, increasing the number of crowdsourced judges has little effect unless the systems are particularly close. As regards evaluation metrics, *P@10* is more brittle than measures such as *MAP* and *nDCG* and so benefits most from collecting additional judgments.

We have successfully shown how a number of real-world and research semantic search systems can be evaluated in a repeatable and reliable manner via creating a new evaluation campaign using crowdsourcing. While the study here has focused on agreement between judges and workers over time and holding the items (queries and results) constant, future research needs to study the agreement between judges and workers on a per-item basis. For example, how does the ambiguity of entity queries affect reliability and repeatability? Future work should also take into account if these results hold over different kinds of entity queries or different kinds of tasks that vary in the levels of ambiguity. So far, the Semantic Search evaluation campaign focused on the case of entity search. It will be broadened to deal with new kinds of semantic search tasks such as relational keyword search and complex question answering featuring more expressive and complex queries beyond keyword-based entity search queries. The methodology demonstrated in this work should be repeated for these new tasks because the differences in ambiguity may have im-

pact on the reliability and repeatability of the results.

5. Semantic Search Challenge

We applied the evaluation framework in the Semantic Search Challenge 2010 and 2011, which were held as part of the Semantic Search Workshop at WWW2010 and WWW2011. The main difference between the challenges is that that 2011 challenge comprised also a List Search Track in addition to the Entity Search Track.

5.1. Semantic Search Challenge 2010

In the following, we describe the participating systems and discuss the results of the Semantic Search Challenge 2010 as reported in [12].

Entity Search Track. The Entity Search Track aimed to evaluate a typical search task on the web, keyword search where the keyword(s) is generally the name of the entity. Entities are ranked according to the degree to which they are relevant to the keyword query. This task was part of the Semantic Search Challenge 2010 and 2011.

5.1.1. Participating Systems 2010

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 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 and 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 rankings [43], 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 [44], BM25 [45], or language models [46]. This type of approach 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 [47] 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 approach 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) For the last approach, the structured 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 6

gives an overview of the systems and their characteristics. 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 objects. All triples having the same subject and also the same context constitute one object description. Thus, the same subject that appears 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 split 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, it was 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! [48] 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.

5.1.2. 2010 Entity Track Evaluation Results

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

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 6: Feature overview regarding system internal object representation and retrieval model

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
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 7: Results of submitted Semantic Search engines.

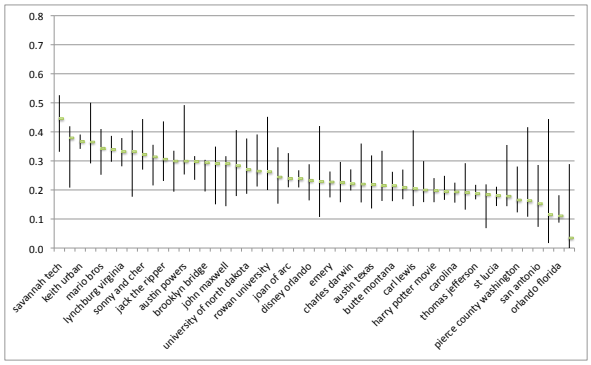


Figure 11: Average NDCG for queries from the Microsoft data-set.

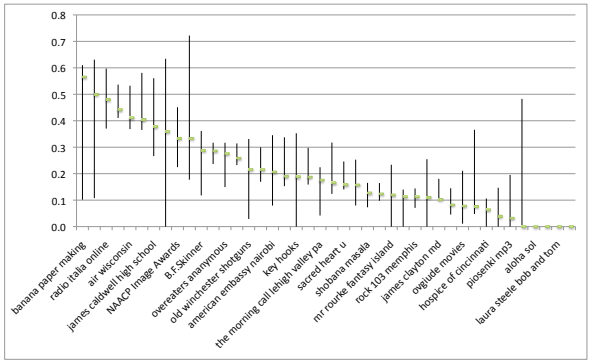


Figure 12: Average NDCG for queries from the Yahoo! data-set.

of our metrics [49], the *trec_eval* software[7] 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.

Table 7 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.

It was interesting to observe that the top two runs achieved 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. The performance difference is also visible in Figure 13, which shows the NDCG per query for both runs. Also we can observe that **sub30-RES3** exceeds **sub31-run3** for 40 of 92 queries.

Figure 11 shows the per-query performance for queries from the Microsoft and Figure 12 for the queries from the Yahoo! log. 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

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

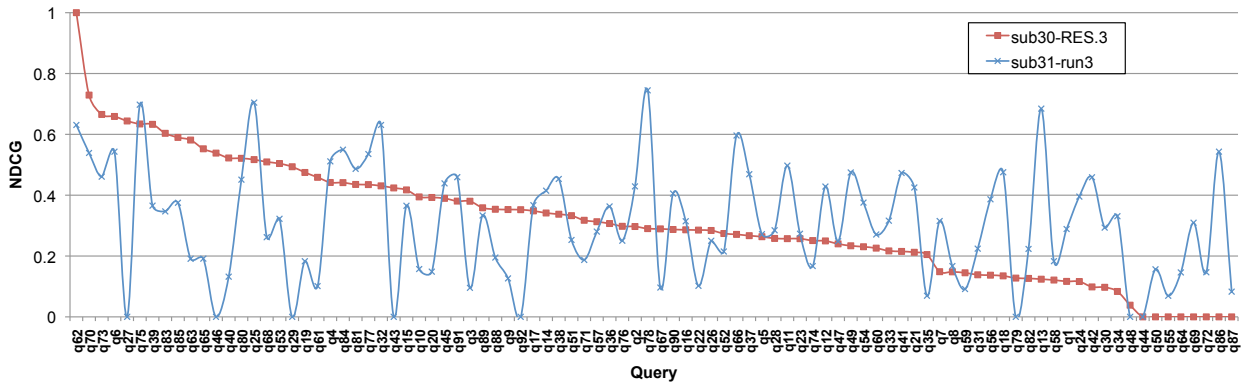


Figure 13: Comparison between runs **sub30-RES3** and **sub31-run3** in terms of NDCG per query for the Entity Track 2010.

log, which are more frequent queries, shows less variation among queries and between systems processing the same queries. This confirms that popular queries are not only easier, but more alike in difficulty.

5.1.3. Discussion of the 2010 Challenge

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.

5.2. Semantic Search Challenge 2011

As described in Section 5.1 the evaluation in 2010 was centered around the task of entity search. This choice was driven by the observation that over 40% of queries in real query logs fall into this category [10], largely because users have grown accustomed to reducing their query (at least initially) to the name of an entity. However, the major feedback and criticism of the 2010 SemSearch Challenge was that by limiting the evaluation to keyword search for named entities the evaluation excluded more complex searches that would hypothetically be enabled by semantic search over RDF. Therefore, the 2011 SemSearch competition introduced a second track, the "List Search" track, that focused on queries where one or more entities could fulfill the criteria given to a search engine.

The Semantic Search Challenge 2011 comprised two different tracks, the Entity Search Track, just like in 2010, and the List Search Track as reported before in [13].

5.2.1. Participating Systems in the Entity Track 2011

Four teams participated in both tracks. These teams were University of Delaware (**UDel**), Digital Enterprise Research Institute (**DERI**), International Institute of Information Technology Hyderabad (**IIIT Hyd**), and Norwegian University of Science and Technology (**NTNU**). Dhirubhai Ambani Institute of Information and Communication Technology (**DA-IICT**) participated additionally in the List Search Track.

Each team was allowed to enter up to three different submissions per track, in order to experiment with different system configurations.

In total, 10 runs were submitted for the Entity Search Track and 11 runs for the List Search Track.

In the following sections, we briefly describe and characterize the systems for each track and report on their performance. Detailed system descriptions are available at the challenge website[8].

In order to categorize the systems and illustrate their different approaches to the entity search task, two major aspects can be distinguished: (1) the internal model for *entity representation*, and (2) the *retrieval model* applied for matching, retrieval, and ranking. Before, we characterize the systems, we discuss these two major aspects.

Entity representation. teams used a *quad* having the same subject URI as the representation of an entity. Only DERI deviated from this representation and took all quads having the same subject and their contexts as the representation as the representation of an entity. The applied representations of an entity can be characterized by four aspects, which describe how the specifics of the data are taken into account. The RDF data model makes a distinction between object and datatype properties. Datatype properties can be seen as *attribute-value* pairs, where the value is a literal value, usually a text string. In contrast, object properties are typed *relations* in the form of attribute-object pairs, where the object is the URI identifier of another entity rather than a literal value. Since URIs are used as identifiers, each URI has a *domain name*, which can be seen as one kind of provenance. Another provenance aspect is the *context*, which describes the source of the triple in the BTC dataset. The domain is different from the context because URIs with the same domain can be used in different contexts. Whether these aspects are considered, is illustrated in Table 8 as follows:

- *attribute-value:* Are the attribute-values of the triples used in the entity representation (yes + / no −)?
- *relations:* Are the relations to other entities considered (yes + / no −)? The relations are potentially exploitable for ranking, because they form the data graph by linking to other entities. If this information is not taken into account, the relations usually treated as additional attribute-value pairs.
- *domain:* Is the domain information used (yes + / no −)? Entities of a certain domain are some times boosted, because certain domains are considered a-priori as relevant or of high quality. Often entities from *dbpedia.org* are considered for a-priori boosting.
- *context:* Is the context information included in the entity representation (yes + / no −)? This information can be used as well to favour certain sources.

Retrieval model. All participating systems used inverted indexes to manage their data. Still, the different approaches can be characterized by three main aspects introduced in Section 5.1.1 Table 8 gives an overview of the systems based on the characteristics introduced above.

5.2.2. Overview of Evaluated Systems Entity Track 2011

		UDel		DERI			NTNU		
	Run	VO	Prox	1	2	3	Olav	Harald	Godfrid
Entity representation	attribute-value	+	+	+	+	+	+	+	+
	relations	-	-	-	-	+	-	-	+
	domain	+	-	-	+	+	+	+	+
	context	-	-	+	+	+	-	-	-
Retrieval model	Text-based	+	+	+	+	+	+	+	+
	Structure-based	-	-	+	+	+	-	+	+
	Q-I-structure	-	-	-	-	+	-	-	-

Table 8: Feature overview regarding system internal entity representation and retrieval model

UDel:

Entity representation: All quads having the same subject URI constituted one entity. Terms extracted from these quads are simply put into one 'bag-of-words' and indexed as one document.
Retrieval model: An axiomatic retrieval function was applied by University of Delaware [50]. For run **UDel-Prox**, query term proximity was added to the model, which favours documents having the query terms within a sliding window of 15 terms. The third run **UDel-VO** promotes entities whose URI has a direct match to a query term.

DERI:

Entity representation: In contrast to the other systems, the Sindice system from DERI took all quads having the same subject and the same context as the description of an entity. Only entity descriptions comprising more than 3 quads were considered. This entity description is internally

[8] `http://semsearch.yahoo.com`

represented as a labeled tree data model with an entity node as the root, and subsequent attribute and value nodes. In addition, run **DERI-3** used the entire graph structure, so exploiting the relationships of any given entity when ranking.
Retrieval model: BM25MF, an extension of BM25F, which allows fields to have multiple values was used by Sindice to rank entities for all runs. The second and winning run, **DERI-2**, applied additionally query specific weights, namely query coverage and value coverage. These weights indicate how well the query terms are covered by a root node, respectively value node, in the internal data model. The more query terms are covered by a node, the more weight is contributed to this node. In addition, query independent weights were assigned to attributes, whose URI contain certain keywords, e.g. *label*, *title*, *sameas*, and *name*. Run **DERI-3** used additionally the relations to compute query independent scores based on the graph structure.

IIIT Hyd:

Did not provide a system description.

NTNU:

Entity representation: NTNU used the *DBPedia* dataset in addition to the BTC to represent entities. An entity is represented by three submodels, the first comprises all name variants of this entity in DBPedia, the second considers several attributes from DBPedia for this entity, and the third uses the data from BTC about this entity. On the syntactic level, all triples having the same subject URI were used for the models based on DBPedia. For run **NTNU-Olav**, the model based on the BTC used only literal objects and regarded them as one flat text representation. For the runs **NTNU-Harald** and **NTNU-Godfrid**, the model had two fields, the name field which contained values of attributes that mentioned the name of the entity, while all other attributes were put into the content field.
Retrieval model: Mixture language models were used to incorporate the different entity models in the retrieval function, while weights were applied for specific attributes of DBPedia. Run **NTNU-Godfrid** used *sameAs* (an equivalence link on the Semantic Web) relations to propagate scores, in order to rank directly related entities higher.

Participant	Run	P10	P5	**MAP**
DERI	2	0.260	0.332	**0.2346**
UDel	Prox	0.260	0.337	**0.2167**
NTNU	Harald	0.222	0.280	**0.2072**
NTNU	Godfrid	0.224	0.272	**0.2063**
NTNU	Olav	0.220	0.276	**0.2050**
UDel	VO	0.194	0.248	**0.1858**
DERI	1	0.218	0.292	**0.1835**
DERI	3	0.188	0.252	**0.1635**
IIIT Hyd	1	0.130	0.148	**0.0876**
IIIT Hyd	2	0.142	0.132	**0.0870**

Table 9: Results of the 2011 Entity Search Track.

5.2.3. 2011 Entity Track Results

Discussion of the 2011 Entity Search Track. The semantic search task of finding entities in an large RDF graph has been addressed by a spectrum of different approaches in this challenge as shown by the diversity of the results. The basis for most systems was well known Information Retrieval techniques, which yielded acceptable results. However, the winning system from DERI was a specialized system, which adapted IR methods and tailored them to RDF. The key feature for success, shared by the two top ranked systems in the 2011 challenge, was to take the proximity or coverage of query terms on individual attribute values into account. This was a consequent development step over the 2010 challenge, where weighting properties individually was the key feature for success. The general observation was that considering the particular pieces of the structured data yields higher performance over unstructured text-based retrieval methods.

Similar to 2010, one of the main and promising features of the RDF data model, namely the ability to express and type the relations between entities was only used by one run from DERI, which did not exceed the other runs. Whether relations are actually not helpful for entity search on large scale datasets or whether the usage of the relations is not yet understood remains to be investigated in the future. The List Search Track was designed with the intention in mind to get the systems to consider the relations as well. How the systems addressed this task is described in the next section.

5.2.4. 2011 List Search Track Evaluation

The List Search Track comprised queries that describe sets of entities, but where the relevant entities were not named explicitly in the query. This track was designed to encourage participating systems to exploit relations between entities and type information of entities, therefore raising the complexity of the queries. The information need was expressed by a number of keywords (minimum three) that describe criteria that need

to be matched by the returned results. The goal was to rank higher the entities that match the criteria than entities that do not match the criteria. Examples of the queries used in the two tracks are shown in Table 3 and described in the Section 3.2.2.

For the List Search track, the workers were presented additionally with a reference list of correct entities in addition to the criteria itself, which was obtained through manual searching by the organizers. This was done as the queries were of such difficulty that many assessors may not know the answers themselves.

In general the teams participated with the same systems in the List Search Track and adapted them only slightly to this new task, although the most high-performing system was specially designed for the List Track. The adaptions were mostly on query analysis and interpretation, because the queries were not just keywords but more complex descriptions in natural language, as described in Section 3.2.2. The modifications as well as the additional system were described in the next section followed by the results for this track.

5.2.5. Participating Systems in the List Search Track

Delaware:
The team from Delaware applied an NLP parser to process the queries for run **UDelRun1**, in order to find the target type of the entities. Only entities belonging to this type were considered as results. For the runs **UDelRun2** and **UDelRun3** the type information was manually expanded, because the automatic processing failed in some cases. Instead of the axiomatic retrieval function, model-based relevance feedback was applied for run **UDelRun3** [51].

DERI:
DERI participated with an identical system configuration in the List Search Track.

NTNU:
NTNU participated with a system especially designed for this track. The system used only the Wikipedia dataset and mapped the results to entities in the BTC collection. The queries were analyzed and potentially reformulated using the Wikipedia Miner software [52], in order to find the primary entity of the query. The query was run against an index of Wikipedia abstracts to get a candidate list of Wikipedia articles. The outgoing links from these articles were expanded and the resulting articles were also added to the candidate list. Scores are added if an article occurs multiple times and articles with a direct relation to the principal entity are boosted. In contrast to run **NTNU-1**, the runs **NTNU-2** and **NTNU-3** used an additional boosting for articles belonging to a Wikipedia set that had more than a certain fraction of its set of members in the candidate list. Run **NTNU-3** also applied an additional boost based on *sameAs* links.

DA-IICT:
The system by DA-IICT used a text-based approach build on Terrier [53] which favoured entities according to the number of query terms present in their textual description. Due to data loss, the queries were only run against a part of the BTC data collection.

5.2.6. List Search Track Results

The retrieval performance for the submitted runs are shown in Table 10. The metrics were computed the same ways as for the Entity Track. There are on average 13 relevant entities per query with a standard deviation of 12.8. The participating systems could not find relevant entities for 6 queries. These were the queries with numbers *q15*, *q23*, *q27*, *q28*, *q45* and *q48*, for example *q15: "henry ii's brothers and sisters"*.

Participant	Run	P10	P5	**MAP**
NTNU	3	0.354	0.356	0.2790
NTNU	2	0.348	0.372	0.2594
NTNU	1	0.204	0.200	0.1625
DERI	1	0.210	0.220	0.1591
DERI	3	0.186	0.216	0.1526
DERI	2	0.192	0.216	0.1505
UDel	1	0.170	0.200	0.1079
UDel	2	0.162	0.152	0.0999
IIIT Hyd	1	0.072	0.076	0.0328
IIIT Hyd	2	0.072	0.076	0.0328
DA-IICT	1	0.014	0.012	0.0050

Table 10: Results of the List Search Track.

Discussion of the 2011 List Search Track. The List Search Track proved to be a hard task and may require different techniques compared to the Entity Search Track. Since this track was new, most teams participated with their systems built for the Entity Search Track and adapted to the task mainly by analyzing and interpreting the query. Still, the performances showed that solutions can be delivered, although there was still room for improvement. The winning system by NTNU did not use the BTC data collection, but was built on the Wikipedia corpus and exploited the links between articles, demonstrating that the plain links between articles are a valuable resource for search. Ideally, such algorithms could eventually be adopted to more general-purpose RDF structured data outside that of Wikipedia.

5.2.7. Discussion of the 2011 Semantic Search Challenge

The Semantic Search Challenge started in 2010 with the task of (named) entity retrieval from RDF data crawled from the Web. Though this task was seemingly simple, because the query contains the name of the entity, it features many of the problems in semantic search, including the potential ambiguity of short-form queries, the varying degrees of relevance by which an entity can be related to the one named in the query and the general quality issues inherent to Web data. The List Search Track introduced in 2011 presented an even harder problem, i.e. queries that do not explicitly name an entity, but rather describe the set of matching entities.

The general direction of our work will continue toward exploring search tasks of increasing difficulty. In addition, there are a number of open questions that may impact the end-user benefits of semantic search engines and would still need to be investigated. For example, the retrieval engines above did not attempt to remove duplicates, and may return different, redundant descriptions of the same entity multiple times. A semantic search engine should remove such duplicates or merge them. Similarly, the user experience was largely impacted by the explanations given by the search engines. Similar to how current text search engines generate summaries and highlight keyword matches, a semantic search engine should attempt to summarize information from an RDF graph and highlight why a particular result is an answer to the user's query.

6. Conclusion

The topic of semantic search has attracted large interests both from industry and research, resulting in a variety of solutions that target different tasks. There is however no standardized evaluation framework that helps to monitor and stimulate the progress in this field. We define the two standard tasks of entity search and entity list search, which are commonly supported by semantic search systems. Starting with these tasks, we run evaluation campaigns organized in the context of the series of SemSearch workshops to assess the state-of-the-art in semantic search with respect two these basic tasks. Aiming at affordable, repeatable and reliable evaluation, we provide a crowdsourcing-based evaluation methodology alongside with a semantic search evaluation framework consisting of real-world queries and datasets. This work discusses the tasks, the framework, the performances achieved by the systems that participated in the campaigns, and the repeatability and reliability of the proposed evaluation methodology. Throughout these two years, we have observed that not only was the evaluation reliable and repeatable but also, experiments could be performed at an acceptable cost.

So far, the methodology has been tested only with respect to two tasks. We are planning to extend the Semantic Search Challenge to cover other retrieval scenarios such as search for consolidated objects (data integration and search), search for documents with embedded RDF (semantic document retrieval) and search for relations between objects (relational search). We consider the provided evaluation framework as a basis platform, which invites researchers to participate and extend towards these and other semantic search scenarios.

7. Acknowledgments

We acknowledge Yahoo! Research for making available under license the 'Yahoo! Search Query Log Tiny Sample, version 1.0' dataset as part of the WebScope program. We also thank Evelyne Viegas and Microsoft Research for allowing a portion of the Microsoft Live Query Log to be used in the 2010 campaign. In particular, we would like to thank Amazon and the NAACL workshop on using the Mechanical Turk for providing the initial funding for the 2010 evaluation, and would like to thank the European PASCAL project for the rest of the support. Of course, none of this work would have been possible without the groups that submitted search engines for evaluation at the Semantic Search Challenge 2010 and 2011. Harry Halpin was partially supported by a Microsoft Research grant on 'Beyond Search.' We acknowledge Yahoo! Labs for hosting the Semantic Search Challenge 2011 website and sponsoring the prizes. The costs of the 2011 evaluation have been sponsored by the European SEALS project, `http://www.seals-project.eu`.

References

[1] J. Chu-Carroll, J. M. Prager, K. Czuba, D. A. Ferrucci, P. A. Duboué, Semantic search via xml fragments: a high-precision approach to ir, in: SIGIR, 2006, pp. 445–452.

[2] J. Chu-Carroll, J. M. Prager, An experimental study of the impact of information extraction accuracy on semantic search performance, in: CIKM, 2007, pp. 505–514.

[3] P. Castells, M. Fernández, D. Vallet, An adaptation of the vector-space model for ontology-based information retrieval, IEEE Trans. Knowl. Data Eng. 19 (2) (2007) 261–272.

[4] T. Tran, S. Bloehdorn, P. Cimiano, P. Haase, Expressive resource descriptions for ontology-based information retrieval, in: ICTIR, 2007, pp. 55–68.

[5] R. Guha, R. McCool, E. Miller, Semantic Search, in: WWW, ACM, 2003, pp. 700–709. doi:http://doi.acm.org/10.1145/775152.775250.

[6] T. Tran, H. Wang, P. Haase, Hermes: Data web search on a pay-as-you-go integration infrastructure, J. Web Sem. 7 (3) (2009) 189–203.

[7] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, G. Tummarello, Sindice.com: A document-oriented lookup index for open linked data, International Journal of Metadata, Semantics, and Ontologies 3 (1) (2008) 37–52.

[8] E. M. Voorhees, Query expansion using lexical-semantic relations, in: SIGIR, 1994, pp. 61–69.

[9] T. Tran, D. M. Herzig, G. Ladwig, Semsearchpro - using semantics throughout the search process, J. Web Sem. 9 (4) (2011) 349–364.

[10] J. Pound, P. Mika, H. Zaragoza, Ad-hoc Object Ranking in the Web of Data , in: Proceedings of the WWW, Raleigh, United States of America, 2010, pp. 771–780.

[11] R. Blanco, H. Halpin, D. M. Herzig, P. Mika, J. Pound, H. S. Thompson, D. T. Tran, Repeatable and reliable search system evaluation using crowdsourcing, in: SIGIR, 2011, pp. 923–932.

[12] H. Halpin, D. M. Herzig, P. Mika, R. Blanco, J. Pound, H. S. Thompson, D. T. Tran, Evaluating Ad-Hoc Object Retrieval, in: Int. Workshop on Evaluation of Semantic Technologies (IWEST 2010) at ISWC, 2010.
URL http://people.csail.mit.edu/pcm/tempISWC/workshops/IWEST2010/paper9.pdf

[13] R. Blanco, H. Halpin, D. M. Herzig, P. Mika, J. Pound, H. S. Thompson, D. T. Tran, Entity Search Evaluation over Structured Web Data, in: Proc. of the 1st Int. Workshop on Entity-Oriented Search (EOS 2011) at SIGIR, 2011.
URL http://research.microsoft.com/en-us/um/beijing/events/eos2011/20.pdf

[14] K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, T. Westerveld, Overview of the TREC 2009 Entity Track, in: NIST Special Publication: SP 500-278, 2009.

[15] O. Alonso, D. E. Rose, B. Stewart, Crowdsourcing for relevance evaluation, SIGIR Forum 42 (2) (2008) 9–15. doi:http://doi.acm.org/10.1145/1480506.1480508.

[16] O. Alonso, R. Schenkel, M. Theobald, Crowdsourcing assessments for XML ranked retrieval, in: ECIR, 2010, pp. 602–606.

[17] C. Callison-Burch, Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk, in: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2009, pp. 286–295.
URL http://www.aclweb.org/anthology/D/D09/D09-1030

[18] S. Nowak, S. M. Rüger, How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation, in: Multimedia Information Retrieval, 2010, pp. 557–566.

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

[20] J. Waitelonis, N. Ludwig, M. Knuth, H. Sack, Whoknows? evaluating linked data heuristics with a quiz that cleans up dbpedia, Interact. Techn. Smart Edu. 8 (4) (2011) 236–248.

[21] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, M. Blum, recaptcha: Human-based character recognition via web security measures, Science 321 (5895) (2008) 1465.

[22] D. M. Herzig, T. Tran, Heterogeneous web data search using relevance-based on the fly data integration, in: WWW, 2012, pp. 141–150.

[23] J. Kamps, S. Geva, A. Trotman, A. Woodley, 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 (2009) 1–28.

[24] Y. Luo, W. Wang, X. Lin, SPARK: A Keyword Search Engine on Relational Databases, in: ICDE, 2008, pp. 1552–1555.

[25] K. Balog, P. Serdyukov, A. de Vries, Overview of the trec 2010 entity track, in: TREC 2010 Working Notes, 2010.

[26] G. Demartini, T. Iofciu, A. P. De Vries, Overview of the inex 2009 entity ranking track, in: Proceedings of the Focused retrieval and evaluation, and 8th international conference on Initiative for the evaluation of XML retrieval, INEX'09, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 254–264.

[27] C. Unger, P. Cimiano, V. Lopez, E. Motta, QALD-1 Open Challenge, http://www.sc.cit-ec.uni-bielefeld.de/sites/www.sc.cit-ec.uni-bielefeld.de/files/sharedtask.pdf (2011).

[28] Proceedings of the Second International Workshop on Keyword Search on Structured Data, KEYS 2010, Indianapolis, Indiana, USA, June 6, 2010, ACM, 2010.

[29] J. Coffman, A. C. Weaver, A framework for evaluating database keyword search strategies, in: Proceedings of the 19th ACM international conference on Information and knowledge management, CIKM '10, ACM, New York, NY, USA, 2010, pp. 729–738.

[30] C. Cleverdon, M. Kean, Factors Determining the Performance of Indexing Systems (1966).
URL http://dspace.lib.cranfield.ac.uk/handle/1826/863

[31] C. W. Cleverdon, The Significance of the Cranfield Tests on Index Languages, in: SIGIR, 1991, pp. 3–12.

[32] C. Bizer, P. Mika, The semantic web challenge, 2009, Journal of Web Semantics 8 (4) (2010) 341.

[33] H. Halpin, A query-driven characterization of linked data, in: Proceedings of the WWW Workshop on Linked Data on the Web, Madrid, Spain, 2009.

[34] E. Voorhees, The philosophy of information retrieval evaluation, in: In Proceedings of the The Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems, Springer-Verlag, 2001, pp. 355–370.

[35] S. P. Harter, Variations in relevance assessments and the measurement of retrieval effectiveness, J. Am. Soc. Inf. Sci. 47 (1) (1996) 37–49. doi:http://dx.doi.org/10.1002/(SICI)1097-4571(199601)47:1¡37::AID-ASI4¿3.3.CO;2-I.

[36] E. M. Voorhees, D. K. Harman, TREC: Experiment and Evaluation in Information Retrieval, Digital Libraries and Electronic Publishing, MIT Press, 2005.
URL http://mitpress.mit.edu/catalog/item/default.asp?ttype=2\&\#38;tid=10667\&\#38;mode=toc

[37] P. Bailey, N. Craswell, I. Soboroff, P. Thomas, A. P. de Vries, E. Yilmaz, Relevance assessment: are judges exchangeable and does it matter, in: SIGIR, ACM, New York, NY, USA, 2008, pp. 667–674. doi:http://doi.acm.org/10.1145/1390334.1390447.

[38] H. Halpin, R. Blanco, Machine-learning for spammer detection in crowd-sourcing, in: Workshop on Human Computation at AAAI, Technical Report WS-12-08, 2012, pp. 85–86.
URL https://www.aaai.org/ocs/index.php/WS/AAAIW12/paper/viewPDFInterstitial/5321/5606

[39] W. Mason, D. J. Watts, Financial incentives and the "performance of crowds", in: Proc. of the ACM SIGKDD Workshop on Human Computation, HCOMP '09, ACM, New York, NY, USA, 2009, pp. 77–85. doi:10.1145/1600150.1600175.
URL http://doi.acm.org/10.1145/1600150.1600175

[40] B. Carpenter, Multilevel bayesian models of categorical data annotation. technical report, Tech. rep., Alias-I, http://lingpipe.files.wordpress.com/2008/11/carp-bayesian-multilevel-annotation.pdf (2008).

[41] S. Elbassuoni, M. Ramanath, R. Schenkel, M. Sydow,

G. Weikum, Language-model-based Ranking for Queries on RDF-graphs, in: CIKM, ACM, 2009, pp. 977–986. doi:http://doi.acm.org/10.1145/1645953.1646078.

[42] I. Soboroff, D. Harman, Novelty detection: the trec experience, in: HLT '05, ACL, USA, 2005. doi:http://dx.doi.org/10.3115/1220575.1220589.

[43] C. D. Manning, P. Raghavan, H. Schütze, Introduction to information retrieval, Cambridge University Press, 2008.

[44] G. Salton, A. Wong, C. S. Yang, A vector space model for automatic indexing, Commun. ACM 18 (1975) 613–620. doi:http://doi.acm.org/10.1145/361219.361220.
URL http://doi.acm.org/10.1145/361219.361220

[45] S. Robertson, S. Walker, Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval, Proceedings of the SIGIR conference on Research and development in information retrieval (1).

[46] C. Zhai, J. Lafferty, A study of smoothing methods for language models applied to information retrieval, ACM Transactions on Information Systems 22 (2) (2004) 179–214. doi:10.1145/984321.984322.

[47] S. Robertson, H. Zaragoza, The Probabilistic Relevance Framework: BM25 and Beyond, Foundations and Trends in Information Retrieval 3 (4) (2010) 333–389. doi:10.1561/1500000019.
URL http://www.nowpublishers.com/product.aspx?product=INR\&doi=1500000019

[48] R. Blanco, P. Mika, S. Vigna, Effective and efficient entity search in rdf data, in: International Semantic Web Conference (1), 2011, pp. 83–97.

[49] F. McSherry, M. Najork, Computing information retrieval performance measures efficiently in the presence of tied scores, in: Proceedings of the 30th ECIR, Springer-Verlag, Berlin, Heidelberg, 2008.

[50] H. Fang, C. Zhai, An exploration of axiomatic approaches to information retrieval, in: SIGIR, 2005, pp. 480–487.

[51] C. Zhai, J. D. Lafferty, Model-based feedback in the language modeling approach to information retrieval, in: CIKM, 2001, pp. 403–410.

[52] D. Milne, I. H. Witten, An open-source toolkit for mining wikipedia, in: Proc. New Zealand Computer Science Research Student Conf., Vol. 9, 2009.

[53] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, C. Lioma, Terrier: A High Performance and Scalable Information Retrieval Platform, in: Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006), 2006.