Evaluating Question Answering over Linked Data[☆]

Vanessa Lopez[a,*], Christina Unger[b], Philipp Cimiano[b], Enrico Motta[c]

[a] IBM Research, Smarter Cities Technology Centre, Mulhuddart, Dublin, Ireland
[b] Semantic Computing Group, CITEC, Universität Bielefeld, Bielefeld, Germany
[c] Knowledge Media Institute, The Open University, Milton Keynes, UK

Abstract

The availability of large amounts of open, distributed and structured semantic data on the web has no precedent in the history of computer science. In recent years, there have been important advances in semantic search and question answering over RDF data. In particular, natural language interfaces to online semantic data have the advantage that they can exploit the expressive power of Semantic Web data models and query languages, while at the same time hiding their complexity from the user. However, despite the increasing interest in this area, there are no evaluations so far that systematically evaluate this kind of systems, in contrast to traditional question answering and search interfaces to document spaces. To address this gap, we have set up a series of evaluation challenges for question answering over linked data. The main goal of the challenge was to get insight into the strengths, capabilities and current shortcomings of question answering systems as interfaces to query linked data sources, as well as benchmarking how these interaction paradigms can deal with the fact that the amount of RDF data available on the web is very large and heterogeneous with respect to the vocabularies and schemas used. Here we report on the results from the first and second of such evaluation campaigns. We also discuss how the second evaluation addressed some of the issues and limitations which arose from the first one, as well as the open issues to be addressed in future competitions.

Keywords: Evaluation, Question Answering, Semantic Web, Linked Data, Natural Language

1. Introduction

With the rapid growth of semantic information published on the web, in particular through the linked data initiative [5], the question how typical web users can search and query these large amounts of heterogeneous and structured semantic data has become increasingly important. Promising research directed towards supporting end users to profit from the expressive power of these standards, while at the same time hiding the complexity behind an intuitive and easy-to-use interface, is offered by search and query paradigms based on natural language interfaces to semantic data [22, 28]. For example, question answering (QA) systems based on natural language allow users to express arbitrarily complex information needs in an intuitive fashion. The main challenge when developing such systems lies in translating the user's information need into a form that can be evaluated using standard Semantic Web query processing and inferencing techniques.

In recent years, there have been important advances in semantic search and QA over RDF data –a survey of existing systems and the challenges they face is presented by Lopez et al. [28]. In parallel to these developments in the Semantic Web community, there has been substantial progress in the areas of QA over textual data [34], natural language interfaces to databases (NLIDB) [2], as well as natural language search interfaces over structured knowledge. The latter are typically based

[☆] The authors wish to thanks Chris Welty for his invited talk "Inside the mind of Watson" at the QALD-1 workshop, and all participants in the open challenges QALD-1 and QALD-2 for valuable feedback and contributions.

[*] Corresponding author
Email addresses: vanlopez@ie.ibm.com (Vanessa Lopez), cunger@cit-ec.uni-bielefeld.de (Christina Unger), cimiano@cit-ec.uni-bielefeld.de (Philipp Cimiano), e.motta@open.ac.uk (Enrico Motta)

on data that is by and large manually coded and homogeneous (e.g., True Knowledge[1]).

Semantic search is also increasingly becoming interesting for commercial search engines. Google Knowledge Graph can be seen as a huge knowledge base that Google intends to exploit for enhancing search results, moving from a search engine to a knowledge engine. Wolfram Alpha[2] is a knowledge inference engine that computes answers to factual queries from a comprehensive structured knowledge base about the world, rather than providing a list of documents.

However, a great challenge for the Semantic Web and natural language processing (NLP) communities is scaling QA approaches to the large amount of distributed interlinked data that is available nowadays on the web, dealing with its heterogeneity and intrinsic noise [28]. Automatically finding answers to questions among the publicly available structured sources on the web has not been possible with NLIDB approaches. As databases are not interoperable and distributed over the web, NLIDB approaches focus on the exploitation of structured data in closed-domain scenarios. In contrast, ontology-based QA systems are able to handle a much more expressive and structured search space, where, as opposed to databases, the information is highly interconnected. For ontology-based approaches the knowledge and semantics encoded in an ontology, together with the use of domain-independent linguistic and lexical resources, are the primary sources for understanding user queries.

On the other hand, QA systems over free text are able to answer questions in open-domain environments. Such systems use information retrieval (IR) techniques to process large amounts of unstructured text and as such to locate the documents and paragraphs in which the answer might appear. IR methods scale well but often do not capture enough semantics. Documents containing the answer could be easily missed if the answer is expressed in a form that does not match the way the query is formulated, or if the answer is unlikely to be available in one document but must be assembled by aggregating answers from multiple documents [44]. Semantic QA systems over structured data can greatly benefit from exploiting ontological relationships in order to understand and disambiguate a query, inheriting relationships and linking word meanings across datasets.

Advances in information retrieval have long been driven by evaluation campaigns such as TREC[3] and CLEF [16]. For example, open question answering over unstructured documents or free text has been in the focus of the open-domain QA track introduced by TREC from 1999 to 2007. Recent evaluation campaigns for semantic search, such as SEALS [43, 31] and entity search evaluations [7, 20], work with structured RDF data rather than unstructured text. However, for natural language based question answering tools over linked data there are no systematic and standard evaluation benchmarks in place yet. Therefore, evaluations of such systems are typically small-scale and idiosyncratic in the sense that they are specific for certain settings or applications [38].

The lack of independent evaluation set-ups and frameworks undermines the value of direct comparisons across systems and the ability to evaluate the progress and assess the benefits of question answering technologies for the Semantic Web. In this paper we describe public evaluation challenges for question answering systems over linked data: QALD. The first instantiation of this challenge, QALD-1, was organised in the context of the ESWC workshop *Question Answering Over Linked Data* in 2011; the second instantiation, QALD-2, was run in the context of the ESWC workshop *Interacting With Linked Data* in May 2012. The second workshop, Interacting with Linked Data, had a broader scope than the first one and aimed at including other paradigms for interacting with linked data, to bring together research and expertise from different communities, including NLP, Semantic Web, human-computer interaction, and databases, and to encourage communication across interaction paradigms.

After giving an overview of existing evaluation approaches in Section 2, we will present the aims and methodology of the QALD challenge in Section 3 and the results from running QALD-1 and QALD-2 in Section 4. In Section 4 we also discuss what has been learned by developing a standard evaluation benchmark for these systems. In particular, as an improvement to the limitations that arose from QALD-1, in QALD-2 we aimed at further facilitating the comparison between different open QA approaches according to the challenges

[1] http://www.trueknowledge.com

[2] http://www.wolframalpha.com

[3] http://trec.nist.gov/

intrinsic to the different types of questions. In Section 5, we draw conclusions and highlight the main issues to be addressed in future competitions.

2. Existing evaluation methods and competitions

There is an increasing number of question answering systems over semantic data, evaluated through usability studies such as the one presented by Kaufmann & Bernstein [22]. But despite growing interest, there is a lack of standardized evaluation benchmarks to evaluate and compare the quality and performance of ontology-based question answering approaches at large scale [28].

To assess their current strengths and weaknesses, a range of such systems, for example GINGSENG [4], NLPReduce [23], Querix [21], FREyA [13] and PANTO [41], made use of the independent Mooney datasets[4] and corresponding queries. These are the only shared datasets that have been used to objectively[5] compare different ontology-based question answering systems for a given ontology or dataset. Standard precision and recall metrics have been adapted to evaluate these systems. Precision is consistently taken as the ratio of the number of correctly covered questions to the number of covered questions, i.e., questions for which the system produced some output, whether correct or not. Recall, on the other hand, is defined differently among systems. Damljanovic et al. [13] define recall as the ratio of the number of questions correctly answered by the system to the total number of all questions in the dataset, while for Wang et al. [41] recall is the ratio of the number of questions that deliver some output—independently of whether the output is valid or not—to the total number of all questions. Such differences, together with discrepancies in the number of queries evaluated, render a direct comparison difficult.

Other systems, for example ORAKEL [10] and AquaLog [25], used their own datasets for evaluation purposes.

In contrast to the previously mentioned evaluation approaches, which are restricted to a specific domain ontology, the evaluation presented by Fernandez et al. [15] exploits the combination of information spaces provided by the Semantic Web and the (non-semantic) web. This evaluation was performed over an integrated system that includes the multi-ontology question answering system PowerAqua [27] as well as a semantic information retrieval system [15]. The ontology entities retrieved by PowerAqua are used to support query expansion in the information retrieval step. The output of the combined system consists of ontology elements that answer the user question together with a complementary ranked list of relevant documents. In order to judge the performance of this combined search tool, the TREC WT10G collection was used as an ontology-based evaluation benchmark. The main advantage is that the queries and relevance judgements are provided by external parties that do not participate in the competition, leading to the creation of objective gold standards that are not biased against any particular system, e.g., the TREC-9 and TREC 2001 collections.[6]

However, as argued by Fernandez et al. [15], there are several limitations of this benchmark when applied to data retrieval and ontology-based question answering systems. First, it does not directly evaluate the performance of these systems in returning answers, but rather how they perform within a query expansion task. Second, the queries selected for TREC-9 and 2001 were extracted from real web search engine logs, such that the queries are tailored to traditional keyword-based search engines and do not exploit the capabilities of ontology-based models in addressing more complex queries. Third, the approach is overshadowed by the sparseness of the knowledge available on the Semantic Web compared to the web. Indeed, at the time the study described by Fernandez et al. was conducted [15], only about 20 % of the query topics in the TREC dataset were covered to some extent by RDF data and ontologies available through the

[4] Raymond Mooney and his group from the University of Texas at Austin provide three datasets [35]: one on the domain of geography (9 classes, 28 properties and 697 instances), one on jobs (8 classes, 20 properties and 4141 instances) and one on restaurants (4 classes, 13 properties and 9749 instances). They were translated to OWL for the purpose of evaluation in [23].

[5] Objective in the sense that they have been created by other parties than those performing the evaluation.

[6] However, one flaw in the standard TREC methodology is that the evaluation is biased towards systems that contribute to the so called pooled assessment in which the top-ranked documents from many systems are evaluated by human assessors. Systems retrieving relevant documents that have not been part of the pooled assessment might thus be penalized for finding actually relevant documents that are judged as irrelevant (see [39]).

semantic search engines Watson[7] and Swoogle[8].

As already mentioned, prevailing differences in evaluation set-ups and techniques, including differences with respect to query samples and evaluation measures, undermine the value of direct comparisons, even over a common dataset. Therefore, efforts are being made towards the establishment of common datasets and methodologies to evaluate semantic technologies, most notably the SEALS evaluation methodology for semantic search [43]. SEALS implements a two-phase approach in which tools with different interfaces (in particular keyword-based, form-based and natural language interfaces) are evaluated and compared in a controlled scenario, both in a fully automated fashion as well as within a user study. SEALS uses a particular domain-specific ontology for the automated evaluation as well as the Mooney geography dataset to evaluate scalability and usability aspects. The most critical aspect of the SEALS evaluation, besides difficulties involved in benchmarking user experiences [43], is that systems are evaluated w.r.t. performance in a strictly controlled environment rather than their ability to solve open-ended, real-life problems. For example, the SEALS evaluation is based on a single well-defined, homogeneous ontology and is thus not particularly relevant in the context of applications that exploit the Semantic Web as a large-scale, distributed and loosely-coupled source of information.

A slightly different kind of semantic evaluation challenge was provided by the entity search challenges, which were started in 2010 and were targeted at keyword-based entity search over semantic data [20, 7]. The goal was to retrieve a ranked list of RDF documents in response to a keyword query. The 2011 competition used a dataset that was based on the Billion Triple Challenge 2009 dataset, containing 1.4 billion triples describing 114 million objects. The competition provided queries obtained from Yahoo query logs that directly mention the entity in question in addition to hand-selected list queries that do not necessarily mention the entity to be retrieved. Assessments were crowd-sourced using Amazon Mechanical Turk[9]. In this scenario it turned out that approaches targeting list queries[10] either applied a combination of NLP and IR techniques in order to retrieve RDF data [24, 33], or relied on Wikipedia articles [3] and did not actually take advantage of the structure of semantic data. To Shah and Arora [33] this "seems like going in the wrong direction [. . .], as the whole point of RDF was to move away from unstructured documents towards semantic data". According to them, one of the reasons for applying traditional information retrieval techniques is that "traditional SPARQL queries on data of a large magnitude is not practical". This indicates that if a competition wants to attract and compare semantic approaches, it should be designed in a way that strongly encourages approaches to make use of the structure of the semantic data available nowadays and to deal with its size and heterogeneity.

In contrast to traditional IR approaches, which use the same type of input (keywords) and output (ranked documents), there is no standard model of ontology-based search. Thus, there has been no general adoption of evaluation regimes or methods for semantic search systems. Indeed, as we have seen, most of the state-of-the-art evaluations for question answering systems are generally conducted with respect to small scale and idiosyncratic tasks tailored to the evaluation of particular systems. The Semantic Web community has not yet adopted standard evaluation benchmarks for semantic question answering that focus on the ability to answer open-ended real life queries over real world datasets. With the goal of progressing on this issue, the evaluation challenge presented in this paper, QALD, aims to evaluate natural language based question answering interfaces to linked data sources, i.e., sources that are characterized by their large scale, openness, heterogeneity and varying levels of quality.

3. Evaluation methodology of QALD

The main goal of the QALD challenge is to evaluate and compare question answering systems that mediate between semantic data and users who express their information needs in natural language, especially with respect to their ability to cope with large amounts of heterogeneous and structured data. The main motivation behind QALD is to provide a common evaluation benchmark that allows for an in-depth analysis of the strengths and shortcomings of current semantic question answer-

[7] http://kmi-web05.open.ac.uk/WatsonWUI/

[8] http://swoogle.umbc.edu/

[9] https://www.mturk.com/

[10] Results of the list track can be found at http://semsearch.yahoo.com/results.php

ing systems and, potentially, their progress over time.

The task for participating systems is to return, for a given natural language question and a RDF data source, a list of entities that answer the question, where entities are either individuals identified by URIs or labels, or literals such as strings, numbers, dates and booleans.

In order to set up a common benchmark, the following basic ingredients are provided:

- **Datasets:** a collection of linked data sources in RDF format
- **Gold standard:** a set of natural language questions annotated with corresponding SPARQL queries and answers for the purpose of training and testing
- **Evaluation method:** a set of procedures and metrics for assessing the performance of a system on the task
- **Infrastructure:** a SPARQL endpoint and an online evaluation tool which is able to assess the correctness of the answers returned by the participating systems

In the following, we describe these ingredients in more detail.

3.1. Datasets

The selected datasets needed to contain real, large scale data, being challenging enough to assess the abilities and shortcomings of the systems. Two different datasets with complementary properties and requirements were selected: DBpedia and MusicBrainz.

- The **DBpedia**[11] project [6] is increasingly becoming the central interlinking hub for the emerging linked data cloud. The official DBpedia dataset for English describes more than 3.5 million entities extracted from Wikipedia, roughly half of them modelled in a consistent ontology with over 320 classes and 1650 properties. Version 3.6 (used for QALD-1) contains a total of about 280 million RDF triples, and version 3.7 (used for QALD-2) contains a total of about 370 million RDF triples. Both include links to YAGO[12] categories.
- **MusicBrainz**[13] is a collaborative open-content music database. A RDF export of the MusicBrainz dataset was provided, containing all of MusicBrainz' artists and albums as well as a subset of its tracks, leading to a total of roughly 15 million RDF triples. This data is modelled with respect to a small ontology with just a few classes and relations—the MusicBrainz ontology in case of QALD-1 and the more standard Music Ontology[14] in the case of QALD-2.

The reason for choosing closed datasets instead of using all the linked data available on the web is two-fold. First, they are large enough to raise scalability and heterogeneity issues [26], but not so large that indexing and the processing of queries using semantic technologies would require computational resources outside the scope of most research groups. Second, by using closed datasets we create controllable and reproducible settings in which all systems can be evaluated under the same conditions. Furthermore, the combination of these two different datasets allows to assess different aspects of semantic question answering systems.

DBpedia, on the one hand, requires the ability to scale to large data sources and to deal with incomplete and noisy data. For example, DBpedia contains heterogeneous terminology, primarily due to employing two partly overlapping namespaces for properties[15]. Also, entities are modelled at different levels of granularity, e.g., by means of a combination of simple DBpedia concepts and by means of single complex YAGO categories (such as `yago:CapitalsInEurope` and `yago:PresidentsOfTheUnitedStates`). While ontology-based question answering systems over restricted domains often interpret a question with respect to an unambiguous ontology, in the case of large open-domain ontologies such as DBpedia they encounter a wide range of ambiguous words—suddenly one query term can have multiple interpretations within the same ontology. These systems are therefore required to apply disambiguation or ranking algorithms. Furthermore, the DBpedia dataset is incomplete and contains modelling errors.

[11] `http://dbpedia.org`

[12] `http://www.mpi-inf.mpg.de/yago-naga/yago/`

[13] `http://musicbrainz.org`

[14] `http://musicontology.com`

[15] The `ontology` namespace comprises of properties modelled in the hand-crafted DBpedia ontology, while the `property` namespace comprises of automatically extracted properties and thus contains a considerable amount of noise.

Question answering systems thus have to be able to cope with this incompleteness and lack of rigor of the ontology, including missing domain and range information for properties, undefined entity types, complex semantic entity labels, redundant properties within the same dataset (such as `birthPlace` and `placeOfBirth`) or even modelling errors (e.g., incorrect property range).

MusicBrainz, on the other hand, offers a small and clean ontology, but requires the ability to adapt to a specific (and sometimes peculiar) domain-dependent modelling of the data. For instance, in the MusicBrainz ontology, the properties `beginDate` and `endDate` relate a date with different kinds of entities. In the case of a person, these relations refer to the day of birth or death. In the case of a group, the dates represent the date when a group was founded or broke up. And when the artist is related to a blank node that is linked to a group, then the begin and end date of this blank node indicate when an artist joined or left the band. Another example is the use of the property `releaseType` to identify the type of an instance (e.g., `TypeAudiobook`) instead of using the relation `rdf:type`. Thus in order to query this dataset, some domain-specific configurations, interactivity or learning mechanisms are required to map the meaning of natural language expressions to concepts in the ontology.

For QALD-2, the RDF export of the MusicBrainz data is no longer based on the MusicBrainz ontology but follows the BBC Music data model, i.e., relies mainly on the Music Ontology [32]. This makes the data available in terms of a more standard vocabulary, while the domain-specificity is preserved. That is, the MusicBrainz dataset can serve to test the ability of a question answering system to query homogeneous, high-quality domain ontologies with specific domain-dependent vocabulary, structure and modelling conventions.

3.2. Gold standard queries

User questions of varying complexity were provided in order to evaluate a system's ability to serve as a natural language interface to the above mentioned datasets, going beyond the expressivity of current keyword-based search engines. In the context of QALD-1, each dataset was made available together with a set of 50 training and 50 test questions each. For QALD-2, both QALD-1 sets have been combined to build a new training set, provided together with a newly created test set, leading to 100 training and 100 test questions for DBpedia, and 100 training and 50 test questions for MusicBrainz. Also, a few out-of-scope questions were added to each question set, i.e., questions to which the datasets do not contain the answer, in order to test the ability of participating systems to judge whether a failure to provide an answer lies in the dataset or the system itself. In addition, we provided a small set of questions that could only be answered by combining information from both datasets, DBpedia and MusicBrainz, thus testing a system's ability to combine several linked information sources when searching for an answer.[16]

All questions were annotated with corresponding SPARQL queries. Both questions and queries were hand-crafted. Some of the queries were picked from the PowerAqua query log, but since the main focus was not on quantity but rather on covering a wide range of challenges involved in mapping natural language to SPARQL, aiming at reflecting real user questions, and in order not to bias the results towards any particular system, most of the questions were generated by students not familiar with the functionalities of particular question answering systems.

The collected results were aimed to be complete and the best possible answers given the data. But of course answers are noisy and incomplete as linked data sources are, and validating the correctness and trust of these sources is out of the scope of this challenge.

Since the questions are often linguistically complex (containing prepositions, quantifiers, conjunctions, and so on), they are tailored to systems based on natural language and penalise keyword-based approaches. In order to avoid this, we annotated all QALD-2 questions with keywords. Additionally, systems were allowed to reformulate the query (e.g., inserting quotes to identify named entities, or translating the questions into a controlled language), as long as the changes were documented. This way we wanted to encourage also other relevant methods that can benefit from the datasets, e.g., methods for dynamic ontology matching, word sense disambiguation, fusion and ranking technologies, to report their results.

[16] We also invited participants to contribute questions on their own, as part of a participant's challenge, in order to allow them to point to challenges that we as organizers were not aware of. But unfortunately, this opportunity was not used by the participants.

The appendix shows examples of queries for both datasets. The entire query sets and datasets are available at `http://www.purl.org/qald/home`.

3.3. Evaluation and infrastructure

As briefly stated above, the task is to extract a list of correct answers (resources or literals) from each of the two provided RDF datasets, given a natural language question. In order to access the datasets, they can either be downloaded or queried by means of a provided SPARQL endpoint. Evaluation takes place with respect to the same SPARQL endpoint[17] (and not the official DBpedia endpoint, for example), in order to ensure invariable and therefore comparable results.

Training questions for each dataset were made available to the participants a couple of months in advance of the deadline for submitting results; test questions were then released two weeks in advance of the deadline.

The training questions are annotated with corresponding SPARQL queries and query results retrieved from the SPARQL endpoint. Annotations are provided in a proprietary XML format shown in Figure 1. The overall document is enclosed by a tag that specifies an ID for the question set, expressing whether the questions refer to DBpedia or MusicBrainz, and whether they are for the training or test phase. Also, each of the questions in the question set has an ID, and moreover specifies the natural language question, the corresponding SPARQL query, and the answers to the query. The answers can be either a literal (boolean, date, number or string) or a list of resources, for which both the URI as well as the English label (if it exists) are specified. For QALD-2, we annotated each question with additional metadata in the form of keywords extracted from the natural language question, and attributes indicating i) the answer type, ii) whether the question relies on classes and properties not contained in the DBpedia ontology (`onlydbo`), and iii) whether it requires aggregation or not, i.e., any SPARQL construct that goes beyond pure triple pattern matching, such as counting and filters.

Submission of results by participating systems was required in the same XML format. For all questions, the ID was obligatory. Beyond that, systems were free to specify either a SPARQL query or the answers—and in the case of returning resources could decide whether to return the URI, the label or both. The reason for requiring that all submissions comply with the same XML format was to facilitate the automatic comparison of the answers provided by the system with the ones provided by the gold standard XML document.

Participating systems were evaluated in terms of precision and recall, both on the level of single questions as well as on the level of the whole question set. With respect to a single question q, recall is defined as the ratio of the number of correct answers provided by the system to the number of gold standard answers, and precision is defined as the ratio of the number of correct answers provided by the system to the number of all answers provided by the system:

$$\text{Recall}(q) = \frac{\text{number of correct system answers for } q}{\text{number of gold standard answers for } q}$$

$$\text{Precision}(q) = \frac{\text{number of correct system answers for } q}{\text{number of system answers for } q}$$

For example, if a system returns `Mongolia` and nothing else as answer for the question depicted in Figure 1, it achieves a precision of 100 %, as the returned answer is among the gold standard answers, but has a recall of only 50 %, as it failed to return `Russia`. If it had provided `Mongolian People's Republic` or `Mongol country`, the answer would not have been counted as right, as it does not match the gold standard answer `Mongolia`. The questions were designed such that the data points to unique answers (e.g. the countries listed for the resource `Yenisei river` are only `Mongolia` and `Russia`, and not further variants), but this issue has to be taken into account in future evaluations.

Global precision and recall values were defined as the average mean of the precision and recall values of all single questions. Additionally, the global F-measure was computed in the familiar fashion:

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Note that for the global question set we did not take into account the set of all gold standard questions but only those questions for which the participating system provided answers. This penalises

[17] Some systems reported difficulties connecting to the SPARQL endpoint provided for the challenge, due to a limited server timeout, which was not sufficient for executing some of the systems' SPARQL queries.

```
<question id="36" answertype="resource" aggregation="false" onlydbo="false">
  <string>Through which countries does the Yenisei river flow?</string>
  <keywords>Yenisei river, flow through, country</keywords>
  <query>
   PREFIX res:  <http://dbpedia.org/resource/>
   PREFIX dbp:  <http://dbpedia.org/property/>
   SELECT DISTINCT ?uri ?string WHERE {
        res:Yenisei_River dbp:country ?uri .
        OPTIONAL { ?uri rdfs:label ?string . FILTER (lang(?string) = "en") }
   }
   </query>
  <answers>
    <answer>
     <uri>http://dbpedia.org/resource/Mongolia</uri>
     <string>Mongolia</string>
    </answer>
    <answer>
     <uri>http://dbpedia.org/resource/Russia</uri>
     <string>Russia</string>
    </answer>
  </answers>
</question>
```

Figure 1: A query example from the QALD-2 DBpedia training set in the specified XML format.

systems that have a high coverage but provide a lot of incorrect answers, compared to systems that have lower coverage but provide answers with higher quality. This, however, has been the source of many discussions and thus was decided to be changed in the third instantiation of the challenge. Also, when reporting evaluation results in the next session, we will include F-measure values computed over the total number of questions.

For all submissions, these metrics were computed automatically by an evaluation tool, to which results could be submitted online. During training and test phases, each participating system was allowed to submit their results as often as desired in order to experiment with different configurations of their system.

During the training phase, evaluation results were returned immediately, while during the test phase the results were returned only after submission was closed.

If participants were submitting a paper, they were also encouraged to report performance, i.e., the average time their system takes to answer a query.

4. Evaluation results

In this section we report on results of both QALD evaluations. In total, seven question answering systems participated in the test phase of the challenge; three of them in QALD-1: FREyA, covering both datasets, PowerAqua covering DBpedia, and SWIP covering MusicBrainz; and four systems in QALD-2: SemSeK, Alexandria, MHE and QAKis, all covering the DBpedia question set.

4.1. Overview of evaluated systems

The systems that participated in the evaluation represent an array of approaches to question answering over linked data. In the following we will briefly discuss their main characteristics.

PowerAqua [27] performs question answering over structured data on the fly and in an open domain scenario, not making any particular assumption about the vocabulary or structure of the dataset, thus being able to exploit the wide range of ontologies available on the Semantic Web. PowerAqua follows a pipeline architecture. The user query is first transformed into *query triples* of the form ⟨subject, property, object⟩ by means of linguistic processing (not covering comparisons and superlatives, which occur in some of the QALD questions). At the next step, the query triples are passed on to a mapping component that identifies suitable semantic resources in various ontologies that are likely to describe the query terms (including a WordNet search in order to find synonyms, hypernyms, derived words and meronyms). Given these semantic resources, a set of ontology triples that jointly cover the user query is derived. Finally, because each resulting triple may lead to only partial answers, they need to be combined into a complete answer. To this end, the various interpretations produced in different ontologies are merged and ranked.

PowerAqua was evaluated on the DBpedia question set. It accesses the DBpedia ontology through a local version of Virtuoso[18] as backend, providing efficient query and full text searches. To generate answers on the fly and in real time, PowerAqua uses iterative algorithms and filter and ranking heuristics to obtain the most precise results first, as a compromise between performance and precision/recall.

FREyA [14] allows users to enter queries in any form. In a first step it generates a syntactic parse tree in order to identify the answer type. The processing then starts with a lookup, annotating query terms with ontology concepts using an ontology-based gazetteer. If there are ambiguous annotations, the user is engaged in a clarification dialog. In this case, the user's selections are saved and used for training the system in order to improve its performance over time. Next, on the basis of the ontological mappings, triples are generated, taking into account the domain and range of the properties. Finally the resulting triples are combined to generate a SPARQL query.

FREyA is the only system that was evaluated using both datasets, thereby proving its portability. In order to perform the ontology-based lookup, FREyA automatically extracts and indexes ontological lexicalizations, which requires scanning through the whole RDF data. Thus, the initialization of the system can take considerable time for large datasets—50.77 hours in the case of DBpedia.

SWIP [12] was evaluated on MusicBrainz and did not use the provided natural language questions as input but rather a translation of them into a semi-formal keyword-based language.[19] The system transforms the keyword-based input representation into a semantic graph query as follows: First, the keywords are transformed into concepts and a set of ranked patterns that are semantically close to those concepts are identified. These patterns are chosen from a set of predefined patterns that have been generated by experts beforehand on the basis of typical user queries. The system then asks the users to choose the query pattern that best expresses the meaning of a particular input keyword. From the chosen representations the final query graph is generated.

FREyA and SWIP are the only of the participating systems that rely on manual intervention at run time, where FREyA can also run without any manual intervention (with better results when granted some training beforehand).

QAKiS [9] is a question answering system over DBpedia that focuses on bridging the gap between natural language expressions and labels of ontology concepts by means of the WikiFramework repository. This repository was built by automatically extracting relational patterns from Wikipedia free text, that specify possible lexicalizations of properties in the DBpedia ontology. For example one of the natural language patterns that express the relation `birthDate` is `was born on`. For QALD-2, QAKiS focused on a subset of the DBpedia training and test questions, namely simple questions that contain one named entity that is connected to the answer via one relation. First, QAKiS determines the answer type as well as the type of the named entity, and next matches the resulting typed question with the patterns in the WikiFramework repository, in order to retrieve the most likely relation, which is then used to build a SPARQL query.

Although the coverage of the system is still quite low, the approach of using a pattern repository (as also done by the TBSL system, see 4.3 below) represents a promising tool for bridging the lexical gap between natural language expressions and ontology labels.

SemSek [1] is a question answering system that also focuses on matching natural language expressions to ontology concepts. It does so by means of three steps: a linguistic analysis, query annotation, and a semantic similarity measure. Query annotation mainly looks for entities and classes in a DBpe-

[18] `http://www.openlinksw.com`

[19] The exact input that was used is documented in the evaluation reports for QALD-1 and QALD-2, that can be accessed at `http://www.purl.org/qald/qald-1` and `http://www.purl.org/qald/qald-2`.

dia index that match the expressions occuring in the natural language question. This process is guided by the syntactic parse tree provided by the linguistic analysis. Starting from the most plausible of the identified resources and classes, SemSek retrieves an ordered list of terms following the dependency tree. In order to match these terms to DBpedia concepts, SemSek then involves two semantic similarity measures, one being Explicit Semantic Analysis based on Wikipedia, and one being a semantic relatedness measure based on WordNet structures.

SemSek thus mainly relies on semantic relatedness as an important tool in order to match natural language expressions with ontology labels in a vocabulary independent way. Similar means were also exploited, e.g., by Freitas et. al [18] (see 4.3 below), who additionally use graph exploration techniques, which offer a way to build SPARQL queries without prior knowledge about the modelling of the data, similar to graph matching algorithms, as used in **MHE** (Multi-Hop Exploration of Entity Graph).

MHE is a method for retrieving entities from an entity graph given an input query in natural language. It was developed by Marek Ciglan at the Institute of Informatics at the Slovak Academy of Sciences. The method relies on query annotation, where parts of the query are labeled with possible mappings to the given knowledge base. The annotations comprise entities and relations, and were generated by means of a gazetteer, in order to expand relations with synonyms, and a Wikifier tool, in order to annotate entities. From those annotations, MHE constructs possible sub-graphs as query interpretation hypotheses and matches them against the entity graph of DBpedia.

Alexandria[20] [42] is a German question answering system over a domain ontology that was built primarily with data from Freebase, parts of DBpedia, and some manually generated content, and contains information on persons, locations, works etc., as well as events, including temporal ones, and n-ary relations between entities. Alexandria addresses the task of mapping natural language questions to SPARQL queries as a graph mapping problem. The syntactic structure of the question is represented by a dependency tree. Then, first the natural language tokens are mapped to ontology concepts based on a hand-crafted lexicon for properties and an index for named entity recognition. Here disambiguation choices can be (but do not have to be) provided by the user and are stored for later lookup. Second, the edges of the dependency parse tree are aggregated into a SPARQL graph pattern, by means of a compositional process. The modeling of n-ary relations in the ontology schema allows Alexandria to match simple linguistic expressions with complex triple patterns.

It is noteworthy that, since Alexandria so far only covers German, the QALD-2 questions were first translated into German. Also, since Alexandria relies on its own ontology schema, the evaluation with respect to the QALD-2 gold standard suffers from data mismatches.

4.2. Results

The results for both datasets for each participating system are shown in Table 1 for QALD-1 and Table 2 for QALD-2, listing the global precision, recall and F-measure values for the whole question set. The column *answered* states for how many of the questions the system provided an answer, *right* specifies how many of these questions were answered with an F-measure of 1, and *partially* specifies how many of the questions were answered with an F-measure strictly between 0 and 1. A detailed listing of precision, recall and F-measure results for each question are available at `http://www.purl.org/qald/home`. We also indicate the global F-measure values in brackets as they would be if computed over the total number of questions and not only the number of questions that were processed by the system (cf. Section 3.3 above).

Additionally, FREyA reports an average performance of 36 seconds using the DBpedia dataset, PowerAqua reports in [27] an average of 20 seconds for a scalability evaluation based on DBpedia and other semantic data, and Alexandria reports an average of less than 20 miliseconds for their algorithm and in-memory SPARQL processing.

4.3. Overview and results of non-participant systems

In addition to the above mentioned systems, other systems such as C-Phrase, Treo, TBSL and BELA did not take part in the online evaluation but used the datasets and questions for their own evaluation purposes.

Granberg and Minock [19] report on the adaptation of their system **C-Phrase** to the MusicBrainz

[20] `http://alexandria.neofonie.de`

DBpedia:

	Total	Answered	Right	Partially	Precision	Recall	F-measure
FREyA	50	43	27	10	0.63	0.54	0.58 (0.5)
PowerAqua	50	46	24	13	0.52	0.48	0.5 (0.46)

MusicBrainz:

	Total	Answered	Right	Partially	Precision	Recall	F-measure
FREyA	50	41	33	1	0.8	0.66	0.71 (0.59)
SWIP	50	35	28	2	0.8	0.56	0.66 (0.46)

Table 1: Results for each of the participating systems in QALD-1

DBpedia:

	Total	Answered	Right	Partially	Precision	Recall	F-measure
SemSeK	100	80	32	7	0.44	0.48	0.46 (0.36)
Alexandria	100	25	5	10	0.43	0.46	0.45 (0.11)
MHE	100	97	30	12	0.36	0.4	0.38 (0.37)
QAKiS	100	35	11	4	0.39	0.37	0.38 (0.13)

Table 2: Results for each of the participating systems in QALD-2

dataset, finally achieving a coverage of 45 out of the 50 training questions for MusicBrainz. The C-Phrase authoring tool can in principle be used for any domain. However, the adaptation has to be performed manually for each database; Granberg and Minock state that half the training set can be authored within 90 minutes. Because the system is built for relational databases, it uses the data in the original MusicBrainz POSTGRESQL database instead of the provided RDF data.

Freitas et al. [17] report that their system **Treo** achieves a precision of 0.395 and a recall of 0.451 on the DBpedia training question set. Also, they report an average time of 728 seconds to answer a question. This low performance stems from the strategy employed in Treo: Instead of exploring the dataset using SPARQL queries, the query mechanism uses sequences of dereferenced URIs to navigate through the data online. The query processing starts by determining pivot entities in the natural language question, which can potentially be mapped to instances or classes and thus can serve as an entry point for the spreading activation search in the linked data web. Starting from these pivot entities, the algorithm navigates through the neighbouring nodes, computing the semantic relatedness between query terms and vocabulary terms encountered in the exploration process. It returns a set of ranked triple paths from the pivot entity to the final resource representing the answer, ranked by the average of the relatedness scores over each triple path.

TBSL [36] is a question answering system that focuses on transforming natural language questions into SPARQL queries in such a way that the query is a faithful representation of the semantic structure of the question. To this end, TBSL first produces a SPARQL template that mirrors the internal structure of the question and that is then instantiated with URIs by means of statistical entity identification and predicate detection. It achieves a precision of 0.61 and a recall of 0.63 on the QALD-1 training question set.

The main goal of the system **BELA** [40] was to explore the contribution of several of the above mentioned techniques for mapping natural language expressions to ontology labels. To this end, BELA builds on a pipeline that iteratively constructs SPARQL query hypotheses as interpretations of a given natural language question, factoring in more and more expensive processing mechanisms.

BELA starts with a simple index lookup, then exploits string similarity, next involves a WordNet-based lexical expansion, and finally computes semantic similarity based on Explicit Semantic Analysis. Each of these steps increase the results of the system on the DBpedia training and test questions, and in sum outperform most of the above mentioned systems. BELA also identifies the gap between the linguistic structure of the natural language question and the underlying structure of the data as one of the major challenges to be addressed in question answering over linked data.

4.4. Discussion

The results are encouraging, showing that current systems performing question answering over linked data can deliver answers to quite complex information needs expressed in natural language, using heterogeneous semantic data. At the same time, the training and test questions were challenging enough to show that there is still plenty room for improvement.

Cimiano and Minock [11] present an analysis of characteristic problems involved in the task of mapping natural language to formal queries. Most of these problems were also encountered in the QALD challenge:

- **Lexical gap:**
  The gap between the vocabulary of the user and that of the ontology cannot always be bridged by the use of string distance metrics or generic dictionaries such as WordNet [29]. For instance, the question *In which country does the Nile start?* requires mapping *start* to the ontological property `sourceCountry`, and for the question *Who is the mayor of New York City?* the expression *mayor* needs to be matched with the property `leaderName`.

  59 % of the QALD-2 DBpedia training questions and 65 % of the QALD-2 DBpedia test questions require more than string similarity and WordNet expansion in order to bridge the lexical gap (cf. [40]).

- **Lexical ambiguities:**
  Lexical ambiguities arise if one word can be interpreted in different ways, i.e., it can refer to different entities or concepts. For example, the name *Lincoln* can refer to a range of different entities (Abraham Lincoln, other people called Lincoln, a fair amount of cities, a mountain, a band, a movie, a novel, and so on). The correct answer can only be obtained by using the contextually relevant mapping.

  Considering only DBpedia resources, 25 % of the expressions used in the QALD-2 questions to refer to these resources are ambiguous.

- **Light expressions:**
  A lot of semantically light expression such as the verbs *to be* and *to have*, and prepositions *of* and *with* either refer to an ontological property in a massively underspecified way (e.g., in *Give me all movies with Tom Cruise*, the proposition *with* needs to be mapped to the ontological property `starring`) or do not correspond to any property at all.

  Around 10 % of the MusicBrainz questions and slightly more than 20 % of the DBpedia questions contain semantically light expressions.

- **Complex queries:**
  In addition to lexical and structural ambiguities, the QALD question sets included information needs that can only be expressed using complex queries containing aggregation functions, comparisons, superlatives, and temporal reasoning. 24 % of the QALD-2 questions were of this nature.

Lexical ambiguities were handled well by open-domain approaches, mainly exploiting the ontology semantics and the user query context combined with ranking algorithms (in the case of PowerAqua) and clarification dialogs (in the case of FREyA). Only in very few cases, properties were mapped to the wrong entity, e.g., for the query *Since when is Tom Araya a member of Slayer*, FREyA gave the birthday of Tom Araya instead of the date when he joined the band [14].

Most failures occured when trying to bridge the lexical gap or were due to the complexity of the queries in general, where complexity had the following sources:

- **Aggregating functions**, e.g., counting, as in *How many bands broke up in 2010?*, sometimes combined with ordering, as in *Which countries have more than two official languages?*, or with superlatives, as in *How many members does the largest group have?*

- **Comparisons**, like in *Who recorded more singles than Madonna?* and *Which bridges are of the same type as the Manhattan Bridge?*

- **Superlatives**, e.g., *Which rock album has the most tracks?* and *What is the highest mountain?*
- **Temporal reasoning**, as in *Which artists have their 50th birthday on May 30?* and *Who was born on the same day as Frank Sinatra?*

Note that queries containing linguistic superlatives or comparatives are not considered complex queries if no mechanisms are required to understand the comparison within the ontology. For example, in *What is the highest place of Karakoram?* the superlative is directly mapped to the ontological property `highestPlace`, such that no aggregation is needed. This mapping process is very different in terms of complexity from the query *What mountain is the highest after the Annapurna*, where *highest* should be mapped to the ontological property `elevation` and the values for all mountains first need to be filtered, so that only the ones with less elevation than the elevation of the Annapurna are kept, and then need to be sorted in descending order, so the first one can be picked as answer.

Failures when trying to bridge the lexical gap often arise from the heterogeneity with respect to different levels of granularity at which entities are modelled. Complex conjunctive ontological terms, such as the YAGO category `PresidentsOfTheUnitedStates`, are notoriously difficult to handle. In total, only 14 % of the DBpedia questions are translated into purely conjunctive categories. For example, the question *When did Germany join the EU* is expressed by the following triple:

```
res:Germany dbp:accessioneudate ?date.
```

That is, the verb phrase *join the EU* maps to the single ontological property `accessioneudate`, which has no domain or range defined in DBpedia.

Another difficulty that often keeps question answering systems from achieving 100 % recall is that DBpedia contains many redundant classes and properties that have a similar or overlapping meaning but are modelled with different URIs (e.g., `dbo:President` and `yago:President` as well as the more specific `yago:PresidentsOfTheUnitedStates`). Thus, there might be more than one ontological correspondent for a natural language term, leading to partial answers. For instance, the query *Which companies are in the computer software industry?* required to find not only companies with the property `industry` "computer software", but also "computer hardware, software" and "computer software and engineering".

Most of the mentioned difficulties arise from the heterogeneity of the data available on the Semantic Web and therefore need to be addressed by all systems striving for successful question answering over linked data. One of the aims of the QALD challenge is to highlight these difficulties and invite question answering systems to present their solutions. The evaluation results indicate that the provided question sets achieve a good balance between complexity and feasibility, i.e., the problem was challenging enough but nevertheless participating systems could obtain decent results to allow for meaningful comparison.

4.5. Comparison of results based on question and answer types

User questions in the context of questions answering over RDF data are in principle not different from the questions used in TREC or CLEF challenges, as it is not the information needs that differ but the means to answer them. Nevertheless, the QALD questions were mainly designed to incorporate the challenges that arise from retrieving answers from RDF data sources, as discussed in the previous section.

In order to compare how the different systems perform according to the different challenges, we annotated the QALD-2 questions according to the complexity required to transform the natural language question into the appropriate SPARQL query, especially with respect to the use of aggregation functions such as counting, unions and filters, as well as the combination of schemas in addition to the DBpedia ontology. We now use these annotations to quantitatively assess the current coverage and shortcomings for the participating question answering systems. To this end, we exploit the average F-measure of the systems as a measure indicating to which degree each annotation type affects the system's ability to find the correct answers. The average F-measure for a given annotation is then calculated by considering the F-measures for all questions containing this annotation, regardeless of whether the system did or did not provide an answer to the question.

As all QALD-2 participants only submitted results for the DBpedia question set, we cannot make any statements regarding MusicBrainz and the fusion questions across DBpedia and MusicBrainz. The results with respect to the question type are shown in Table 3. The column *aggregation* refers to queries for which aggregation functions are required, e.g. questions with comparisons (often requiring filters) or superlatives (requiring ordering of results), and *how many* question (requiring counting). The column *other schema* shows the results for all queries that require the use of other schema than the DBpedia ontology. This comprises the DBpedia property namespace as well as YAGO and FOAF. Finally, we call *list* queries those which do not contain aggregations or do not require the use of other schema. From the 100 test questions, 73 are list questions, 18 require aggregations and 44 require the use of other schema. Note that some queries require both the use of other schema and aggregation functions to find the appropriate answers. Also note that the results for Alexandria in the last column are not very meaningful, as it did not use DBpedia as a data basis.

In order to answer list questions, the systems needed to bridge the lexical gap, to handle lexical ambiguities, and in the case of other schemas being relevant, also to deal with the heterogeneity of the data sources—YAGO, for example, includes rather complex conjunctive ontological terms. To answer aggregation queries the systems needed to be able to detect the linguistic constructions used in the natural language questions to denote aggregations, such as superlatives and comparisons, and to construct corresponding SPARQL queries using unions, filters, ordering and counting predicates. In Table 3, we can see that the systems were doing much better on list questions than on the other question types, and especially struggled with aggregation questions. In fact, only half of the systems, MHE and Alexandria, were able to handle to some extent aggregation questions and queries requiring other schema. This shows both that the task of answering questions using linked data is not straightforward, and that the complexity of the QALD-2 questions is very high considering the current performance of the systems, therefore leaving quite a lot of space for improvement and further challenges.

Table 4 shows the results with respect to the type of the answer that the questions expect, independent of whether they require aggregations or other schema, comprising the following answer types: numbers or a count of the result answers (15 queries), literals or strings (3 queries), booleans (8 queries), dates (3 queries) and a resource or a list of resources (70 queries). Here, MHE is the only QALD-2 participant that provided answers to all types of questions, performing best on string and date questions. All systems provided answers for numeral and resource queries, which make up the biggest part of the question set, covering 85 % of all questions. The lower F-measure on those questions stems from the fact that when retrieving a list of resources, some answers may be missing or incorrect and thus cause a decrease of the overall score.

5. Conclusions and future evaluations

The importance of interfaces that bridge the gap between the end user and Semantic Web data have been widely recognised [8]. QALD is the first public challenge aiming at providing a common benchmark for evaluating the success of question answering systems in answering information needs by taking into account data available on the Semantic Web.

For the QALD challenge, we simplified the evaluation process as much as possible. Participants ran their system locally and submitted the results in an XML file via an online form. An alternative approach is for participants to implement a wrapper, so their system can be run on a central server, as has been done for the SEALS semantic search evaluation in 2010. The main advantage is that performance across systems can be fairly measured. However, it also introduces a major overhead for both participants and organizers. In particular, it involves several infrastructural issues and challenges. Current systems are based on very different infrastructures, e.g., they may use the provided SPARQL endpoint or not, they might be based on different semantic database servers (such as Jena, Sesame, or Virtuoso), they may require indexes to optimize performance, as well as different configurations and libraries (such as GATE, gazetteers, or WordNet). For the QALD challenges we therefore designed an evaluation with the goal of facilitating participation as much as possible.

Future evaluation campaigns of the QALD series can develop in several directions. On the one hand, we want to extend the evaluation to new datasets and questions. Since participants of future challenges will have access to all training and test data from previous challenges, we are increasingly creating a public test corpus to facilitate standardized evaluations and comparison across systems to

DBpedia:

	List	Aggregation	Other schema
SemSeK	0.49	0	0.32
MHE	0.44	0.14	0.17
QAKiS	0.17	0	0.06
Alexandria	0.13	0.04	0.07

Table 3: F-measure average for each of the participating systems in QALD-2 by question type

DBpedia:

	Boolean	Date	Numeral	Resource	String
SemSeK	0	0.67	0.33	0.38	0.67
MHE	0.38	0.67	0.47	0.3	0.67
QAKiS	0	0	0.13	0.15	0
Alexandria	0.25	0	0.03	0.11	0

Table 4: F-measure average for each of the participating systems in QALD-2 by answer type

progress on this field and foster research in this novel and challenging area. On the other hand, future challenges can focus on extending the current evaluation methodologies to assess further aspects of question answering systems, such as performance and usability.

However, both evaluation challenges showed that participants were all more familiar and interested in finding better ways to query large linked data collections covering heterogeneous schema and domains, such as DBpedia, rather than domain-specific homogeneous datasets following a particular schema, such as MusicBrainz.

The third challenge will focus on multilinguality as one aspect of querying linked data collections. Multilinguality has become an issue of major interest for the Semantic Web community, as both the number of actors creating and publishing open data in languages other than English as well as the amount of users that access this data and speak native languages other than English is growing substantially. In particular, DBpedia is becoming inherently language-independent as the English DBpedia contains multilingual labels and versions of DBpedia in other languages are prepared and published, such as the Spanish[21] and the French[22] DBpedia. Thus we will extend our evaluation by providing questions and open domain datasets in different languages in order to also evaluate multilingual and non-English based question answering systems.

Another very important aspect is addressing the *linked* in linked data, i.e., the ability to answer questions across sources, as already introduced in QALD-2, eventually scaling to the whole linked data cloud.

One limitation of the QALD challenges arises from the fact that they focus on structured data only and do not consider the possibility of participants incorporating information retrieval techniques, for example extracting results from literals such as DBpedia abstracts. It is quite likely that future question answering systems build on a hybrid approach, combining structured and unstructured data sources. This scenario is very appealing but will require much more sophisticated methods of evaluation.

Appendix

In the following we list a few examples of queries together with their SPARQL annotation, as used in the QALD-2 training phase.[23]

[21] http://es.dbpedia.org

[22] http://wimmics.inria.fr/projects/dbpedia/

[23] Using the following prefixes for DBpedia:

DBpedia

1. Who is the daughter of Bill Clinton married to?

```
SELECT DISTINCT ?uri ?string WHERE {
  res:Bill_Clinton dbo:child ?child .
  ?child dbp:spouse ?string .
  ?uri rdfs:label ?string .
}
```

2. Which actors were born in Germany?

```
SELECT DISTINCT ?uri ?string WHERE {
  ?uri rdf:type dbo:Actor .
 { ?uri dbo:birthPlace res:Germany . }
  UNION
 { ?uri dbo:birthPlace ?city .
  ?city rdf:type yago:StatesOfGermany . }
  OPTIONAL { ?uri rdfs:label ?string .
             FILTER (lang(?string) = ’en’) }
}
```

3. Which caves have more than 3 entrances?

```
SELECT ?uri ?string WHERE {
  ?uri rdf:type dbo:Cave .
  ?uri dbo:numberOfEntrances ?entrance .
  FILTER (?entrance > 3) .
  OPTIONAL { ?uri rdfs:label ?string .
             FILTER (lang(?string) = ’en’) }
}
```

4. Who produced the most films?

```
SELECT DISTINCT ?uri ?string WHERE {
  ?film rdf:type dbo:Film .
  ?film dbo:producer ?uri .
  OPTIONAL { ?uri rdfs:label ?string .
             FILTER (lang(?string) = ’en’) }
} ORDER BY DESC(COUNT(?film)) LIMIT 1
```

MusicBrainz

5. Give me all live albums by Michael Jackson.

```
SELECT DISTINCT ?album ?title WHERE {
  ?album rdf:type mm:Album .
  ?album mm:releaseType mm:TypeLive .
  ?album dc:title ?title .
  ?album dc:creator ?artist .
  ?artist dc:title ’Michael Jackson’ .
}
```

6. Did the Sex Pistols already break up?

```
ASK WHERE {
  ?artist dc:title ’Sex Pistols’.
  ?artist mm:endDate ?endDate.
  FILTER (bound(?endDate))
}
```

7. In which bands did Kurt Cobain play?

```
SELECT DISTINCT ?band ?title WHERE {
  ?artist dc:title ’Kurt Cobain’ .
  ?artist ar:memberOfBand ?bandinstance.
  ?bandinstance ar:toArtist ?band .
  ?band dc:title ?title .
}
```

8. Give me all Thrash Metal albums.

```
SELECT DISTINCT ?album ?name WHERE {
  ?album rdf:type mo:Record .
  ?album dc:description ?tag .
  ?album dc:title ?name .
  FILTER regex(?tag,"thrash metal","i")
}
```

Questions across datasets

Finally, we list an example of the seven additional questions we provided that require information from both datasets to be answered.

9. In which country was the singer of the Drunken Lullabies by Flogging Molly born?

```
SELECT DISTINCT ?uri ?string WHERE {
  ?album rdf:type mo:Record .
  ?album dc:title ’Drunken Lullabies’ .
  ?album mo:singer ?mb_singer .
  ?album foaf:maker ?mb_band .
  ?mb_band foaf:name ’Flogging Molly’ .
  ?dbp_singer owl:sameAs ?mb_singer .
  ?dbp_singer dbo:birthPlace ?city .
  ?city dbo:country ?uri .
  ?uri rdf:type dbo:Country .
  OPTIONAL { ?uri rdfs:label ?string .
             FILTER (lang(?string) = ’en’) }
}
```

- res for http://dbpedia.org/resource/
- dbo for http://dbpedia.org/ontology/
- dbp for http://dbpedia.org/property/

And the following prefixes for MusicBrainz:

- mm for http://musicbrainz.org/mm/mm-2.1
- ar for http://musicbrainz.org/ar/ar-1.0#
- dc for http://purl.org/dc/elements/1.1/

References

[1] N. Aggarwal, P. Buitelaar, A System Description of Natural Language Query over DBpedia, in: *Proc. of Interacting with Linked Data (ILD 2012)* [37] (2012), 96–99. `http://ceur-ws.org/Vol-913/08-ILD2012.pdf`

[2] L. Androutsopoulos, Natural Language Interfaces to Databases – An Introduction, *Journal of Natural Language Engineering* **1** (1995), 29–81.

[3] K. Balog, M. Ciglan, R. Neumayer, W. Wei and K. Nørvåg, NTNU at SemSearch 2011, in: *Proc. of the 4th International Semantic Search Workshop (SEMSEARCH '11) at the World Wide Web conference 2011* (2011).

[4] A. Bernstein, E. Kaufmann, C. Kaiser and C. Kiefer, Ginseng: A Guided Input Natural Language Search Engine, in: *Proc. of the 15th Workshop on Information Technologies and Systems (WITS 2005)* (2006), 45–50.

[5] C. Bizer, T. Heath and T. Berners-Lee, Linked Data—The Story So Far, *International Journal on Semantic Web and Information Systems* **5** (2009), 1–22.

[6] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak and S. Hellmann, DBpedia: A Crystallization Point for the Web of Data, *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* **7** (2009), 154–165.

[7] R. Blanco, H. Halpin, D. Herzig, P. Mika, J. Pound, H.S. Thompson and T.-T. Duc, Entity Search Evaluation over Structured Web Data, in: *Proc. of the 1st International Workshop on Entity-Oriented Search at SIGIR 2011* (2011).

[8] P. Buitelaar, T. Declerck, N. Calzolari and A. Lenci, Language Resources and the Semantic Web, *Proc. of the ELSNET/ENABLER Workshop, Paris, France* (2003).

[9] E. Cabrio, A. Palmero Aprosio, J. Cojan, B. Magnini, F. Gandon, A. Lavelli, QAKiS @ QALD-2. In: Proceedings of Interacting with Linked Data (ILD 2012) [37] (2012), 87–95. `http://ceur-ws.org/Vol-913/07-ILD2012.pdf`

[10] P. Cimiano, P. Haase and J. Heizmann, Porting Natural Language Interfaces between Domains—An Experimental User Study with the ORAKEL System, in: *Proc. of the International Conference on Intelligent User Interfaces* (2007), 180–189.

[11] P. Cimiano and M. Minock, Natural Language Interfaces: What's the Problem?—A Data-driven Quantitative Analysis, in: *Proc. of the International Conference on Applications of Natural Language to Information Systems (NLDB 2009)* (2009), 192–206.

[12] C. Comparot, O. Haemmerle and N. Hernandez, An easy way of expressing conceptual graph queries from keywords and query patterns, in: *Conceptual Structures: From Information to Intelligence, 18th International Conference on Conceptual Structures (ICCS 2010), Kuching, Sarawak, Malaysia, July 26-30, 2010, Proceedings*, M. Croitorou, S. Ferre, D. Lukose, eds., Springer, LNCS 6280 (2010), 84–96.

[13] D. Damljanovic, M. Agatonovic and H. Cunningham, Natural Language interface to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction, in: *Proc. of the European Semantic Web Conference, Heraklion, Greece*, Springer (2010).

[14] D. Damljanovic, M. Agatonovic and H. Cunningham, FREyA: an Interactive Way of Querying Linked Data using Natural Language, in: *Proc. of 1st Workshop on Question Answering over Linked Data (QALD-1) at the 8th Extended Semantic Web Conference (ESWC 2011)* (2011).

[15] M. Fernandez, V. Lopez, E. Motta, M. Sabou, V. Uren, D. Vallet and P. Castells, Using TREC for cross-comparison between classic IR and ontology-based search models at a Web scale, in: *Proc. of the Semantic search workshop, collocated with the 18th International World Wide Web Conference, Madrid, Spain* (2009).

[16] P. Forner, D. Giampiccolo, B. Magnini, A. Peñas, A. Rodrigo and R. Sutcliffe, Evaluating Multilingual Question Answering Systems at CLEF, in: *Proc. of the Conference on Language Resources and Evaluation (LREC), Malta* (2010).

[17] A. Freitas, J. Gabriel de Oliveira, S. O'Riain, E. Curry and J.J.C. Pereira da Silva, Treo: Combining Entity-Search, Spreading Activation and Semantic Relatedness for Querying Linked Data, in: *Proc. of 1st Workshop on Question Answering over Linked Data (QALD-1) at the 8th Extended Semantic Web Conference (ESWC 2011)* (2011).

[18] A. Freitas, J.G. Oliveira, S. O'Riain, E. Curry, J.C.P. Da Silva, Querying linked data using semantic relatedness: a vocabulary independent approach, in: *Proc. of the 16th International Conference on Applications of Natural Language to Information Systems (NLDB '11)* (2011).

[19] J. Granberg and M. Minock, A Natural Language Interface over the MusicBrainz Database, in: *Proc. of 1st Workshop on Question Answering over Linked Data (QALD-1) at the 8th Extended Semantic Web Conference (ESWC 2011)* (2011).

[20] H. Halpin, D. Herzig, P. Mika, R. Blanco, J. Pound, H.S. Thompson and T.-T. Duc, Evaluating Ad-Hoc Object Retrieval, in: *Proc. of the International Workshop on Evaluation of Semantic Technologies (IWEST 2010) at the 9th ISWC* (2010).

[21] E. Kaufmann, A. Bernstein and R. Zumstein, Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs, in: *Proc. of the 5th International Semantic Web Conference, Athens, USA*, Springer, LNCS 4273 (2006), 980–981.

[22] E. Kaufmann and A. Bernstein, How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users?, in: *Proc. of the 6th International Semantic Web Conference, Busan, Korea*, Springer, LNCS 4825 (2007), 281–294.

[23] E. Kaufmann, A. Bernstein and L. Fischer, NLP-Reduce: A "naive" but domain-independent Natural Language Interface for Querying Ontologies, in: *Proc. of the 4th European Semantic Web Conference, Innsbruck*, Springer (2007).

[24] X. Liu, C.-L. Yao and H. Fang, A study of Semantic Search in SemSearch 2011, in: *Proc. of the 4th International Semantic Search Workshop (SEMSEARCH '11) at the International World Wide Web Conference* (2011).

[25] V. Lopez, V. Uren, E. Motta and M. Pasin, AquaLog: An ontology-driven question answering system for organizational semantic intranets, *Journal of Web Semantics: Science Service and Agents on the World Wide Web* **5** (2007), 72–105.

[26] V. Lopez, A. Nikolov, M. Sabou, V. Uren, E. Motta and M. d'Aquin, Scaling up Question-Answering to Linked Data, in: *Proc. of the 17th Knowledge Engineering and Knowledge Management by the Masses (EKAW)*,

Springer (2010), 193–210.

[27] V. Lopez, M. Fernandez, E. Motta and N. Stieler, PowerAqua: Supporting Users in Querying and Exploring the Semantic Web, *Semantic Web Journal* (2011).

[28] V. Lopez, V. Uren, M. Sabou and E. Motta, Is Question Answering fit for the Semantic Web? A Survey, *Semantic Web* **2** (2011), 125–155.

[29] G. A. Miller, WordNet: A Lexical Database for English, *Communications of the ACM* **38** (1995), 39–41.

[30] M. Minock, C-Phrase: A System for Building Robust Natural Language Interfaces to Databases, *Journal of Data and Knowledge Engineering* **69** (2010), 290–302.

[31] L. Nixon, R. García-Castro, S. Wrigley, M. Yatskevich, C. Santos and L. Cabral, The State of Semantic Technology Today—Overview of the First SEALS Evaluation Campaigns, in: *Proc. of the 7th International Conference on Semantic Systems (I-SEMANTICS)* (2011).

[32] Y. Raimond, S. Abdallah, M. Sandler and F. Giasson, The music ontology, in: *Proc. of the International Conference on Music Information Retrieval* (2007), 417–422.

[33] S. Shah and G. Arora, Information Retrieval on semantic data—Semsearch 2011 List Search Track system description, *Information Retrieval* **2** (2011).

[34] T. Strzalkowski and S. Harabagiu, *Advances in Open Domain Question Answering*, Springer (2006).

[35] L.R. Tang and R.J. Mooney, Using multiple clause constructors in inductive logic programming for semantic parsing, in: *Proc. of the 12th European Conference on Machine Learning (ECML-2001)* (2001), 466–477.

[36] C. Unger, L. Bühmann, J. Lehmann, A.-C. Ngonga Ngomo, D. Gerber, P. Cimiano: SPARQL Template-Based Question Answering, in: *Proc. of the 22nd International World Wide Web Conference (WWW 2012)* (2012).

[37] C. Unger, P. Cimiano, V. Lopez, E. Motta, P. Buitelaar, R. Cyganiak (eds.), Proceedings of Interacting with Linked Data (ILD 2012), workshop co-located with the 9th Extended Semantic Web Conference, May 28, 2012, Heraklion, Greece (2012). `http://ceur-ws.org/Vol-913`

[38] V. Uren, M. Sabou, E. Motta, M. Fernandez, V. Lopez, and Y. Lei, Reflections on five years of evaluating semantic search systems, *International Journal of Metadata, Semantics and Ontologies* **5** (2010), 87–98.

[39] E. Voorhees, Evaluation by Highly Relevant Documents, in: *Proc. of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2001), 74–82.

[40] S. Walter, C.Unger, P. Cimiano, D. Bär, Evaluation of a layered approach to question answering over linked data, in: *Proc. of the 11th International Semantic Web Conference (ISWC 2012)* (2012).

[41] C. Wang, M. Xiong, Q. Zhou, and Y. Yu, PANTO: A portable Natural Language Interface to Ontologies, in: *Proc. of the 4th European Semantic Web Conference, Innsbruck, Austria*, Springer (2007), 473–487.

[42] M. Wendt, M. Gerlach, H. Düwiger: Linguistic Modeling of Linked Open Data for Question Answering. In: Proceedings of Interacting with Linked Data (ILD 2012) [37] (2012), 75–86. `http://ceur-ws.org/Vol-913/06-ILD2012.pdf`

[43] S. Wrigley, K. Elbedweihy, D. Reinhard, A. Bernstein and F. Ciravegna, Evaluating semantic search tools using the SEALS Platform, in: *Proc. of the International Workshop on Evaluation of Semantic Technologies (IWEST 2010) at the International Semantic Web Conference* (2010). Campaign 2010 results at: `http://www.sealsproject.eu/seals-evaluation-campaigns/semantic-searchtools/results-2010`

[44] C. Hallet, D. Scott and R.Power, Composing Questions through Conceptual Authoring, in *Computational Linguistics* **33**, (2007), 105–133.

```
This is pdfTeX, Version 3.1415926-1.40.10 (Web2C 2009) (format=latex 2011.5.3)  21 MAY 2013
15:43
entering extended mode
 %&-line parsing enabled.
**evaluatingqald.tex
(./evaluatingqald.tex
LaTeX2e <2009/09/24>
Babel <v3.8l> and hyphenation patterns for english, usenglishmax, dumylang, noh
yphenation, german-x-2009-06-19, ngerman-x-2009-06-19, ancientgreek, ibycus, ar
abic, basque, bulgarian, catalan, pinyin, coptic, croatian, czech, danish, dutc
h, esperanto, estonian, farsi, finnish, french, galician, german, ngerman, mono
greek, greek, hungarian, icelandic, indonesian, interlingua, irish, italian, ku
rmanji, latin, latvian, lithuanian, mongolian, mongolian2a, bokmal, nynorsk, po
lish, portuguese, romanian, russian, sanskrit, serbian, slovak, slovenian, span
ish, swedish, turkish, ukenglish, ukrainian, uppersorbian, welsh, loaded.
(c:/texlive/2009/texmf/tex/latex/aries/elsarticle.cls
Document Class: elsarticle 2009/09/17, 1.20b: Elsevier Ltd
\@bls=\dimen102
(c:/texlive/2009/texmf-dist/tex/latex/base/article.cls
Document Class: article 2007/10/19 v1.4h Standard LaTeX document class
(c:/texlive/2009/texmf-dist/tex/latex/base/size10.clo
File: size10.clo 2007/10/19 v1.4h Standard LaTeX file (size option)
)
\c@part=\count79
\c@section=\count80
\c@subsection=\count81
\c@subsubsection=\count82
\c@paragraph=\count83
\c@subparagraph=\count84
\c@figure=\count85
\c@table=\count86
\abovecaptionskip=\skip41
\belowcaptionskip=\skip42
\bibindent=\dimen103
) (c:/texlive/2009/texmf-dist/tex/latex/graphics/graphicx.sty
Package: graphicx 1999/02/16 v1.0f Enhanced LaTeX Graphics (DPC,SPQR)
(c:/texlive/2009/texmf-dist/tex/latex/graphics/keyval.sty
Package: keyval 1999/03/16 v1.13 key=value parser (DPC)
\KV@toks@=\toks14
) (c:/texlive/2009/texmf-dist/tex/latex/graphics/graphics.sty
Package: graphics 2009/02/05 v1.0o Standard LaTeX Graphics (DPC,SPQR)
(c:/texlive/2009/texmf-dist/tex/latex/graphics/trig.sty
Package: trig 1999/03/16 v1.09 sin cos tan (DPC)
) (c:/texlive/2009/texmf-dist/tex/latex/latexconfig/graphics.cfg
File: graphics.cfg 2009/08/28 v1.8 graphics configuration of TeX Live
)
Package graphics Info: Driver file: dvips.def on input line 91.
(c:/texlive/2009/texmf-dist/tex/latex/graphics/dvips.def
File: dvips.def 1999/02/16 v3.0i Driver-dependant file (DPC,SPQR)
))
\Gin@req@height=\dimen104
\Gin@req@width=\dimen105
```

```
) (c:/texlive/2009/texmf-dist/tex/latex/psnfss/pifont.sty
Package: pifont 2005/04/12 PSNFSS-v9.2a Pi font support (SPQR)
LaTeX Font Info:    Try loading font information for U+pzd on input line 63.
(c:/texlive/2009/texmf-dist/tex/latex/psnfss/upzd.fd
File: upzd.fd 2001/06/04 font definitions for U/pzd.
)
LaTeX Font Info:    Try loading font information for U+psy on input line 64.
(c:/texlive/2009/texmf-dist/tex/latex/psnfss/upsy.fd
File: upsy.fd 2001/06/04 font definitions for U/psy.
))
\c@tnote=\count87
\c@fnote=\count88
\c@cnote=\count89
\c@ead=\count90
\c@author=\count91
\@eadauthor=\toks15
\c@affn=\count92
\absbox=\box26
\keybox=\box27
\Columnwidth=\dimen106
\space@left=\dimen107
\els@boxa=\box28
\els@boxb=\box29
\leftMargin=\dimen108
\@enLab=\toks16
\@sep=\skip43
\@@sep=\skip44
(./evaluatingqald.spl) (c:/texlive/2009/texmf-dist/tex/latex/natbib/natbib.sty
Package: natbib 2009/07/16 8.31 (PWD, AO)
\bibhang=\skip45
\bibsep=\skip46
LaTeX Info: Redefining \cite on input line 694.
\c@NAT@ctr=\count93
)
\splwrite=\write3
\openout3 = `evaluatingqald.spl'.

(c:/texlive/2009/texmf-dist/tex/latex/geometry/geometry.sty
Package: geometry 2008/12/21 v4.2 Page Geometry
(c:/texlive/2009/texmf-dist/tex/generic/oberdiek/ifpdf.sty
Package: ifpdf 2009/04/10 v2.0 Provides the ifpdf switch (HO)
Package ifpdf Info: pdfTeX in pdf mode not detected.
) (c:/texlive/2009/texmf-dist/tex/generic/oberdiek/ifvtex.sty
Package: ifvtex 2008/11/04 v1.4 Switches for detecting VTeX and its modes (HO)
Package ifvtex Info: VTeX not detected.
)
\Gm@cnth=\count94
\Gm@cntv=\count95
\c@Gm@tempcnt=\count96
\Gm@bindingoffset=\dimen109
\Gm@wd@mp=\dimen110
\Gm@odd@mp=\dimen111
```

```
\Gm@even@mp=\dimen112
\Gm@dimlist=\toks17
(c:/texlive/2009/texmf-dist/tex/xelatex/xetexconfig/geometry.cfg)) (c:/texlive/
2009/texmf-dist/tex/latex/base/fleqn.clo
File: fleqn.clo 1998/08/17 v1.1c Standard LaTeX option (flush left equations)
\mathindent=\dimen113
)
\appnamewidth=\dimen114
) (c:/texlive/2009/texmf-dist/tex/latex/amsmath/amsmath.sty
Package: amsmath 2000/07/18 v2.13 AMS math features
\@mathmargin=\skip47
For additional information on amsmath, use the `?' option.
(c:/texlive/2009/texmf-dist/tex/latex/amsmath/amstext.sty
Package: amstext 2000/06/29 v2.01
(c:/texlive/2009/texmf-dist/tex/latex/amsmath/amsgen.sty
File: amsgen.sty 1999/11/30 v2.0
\@emptytoks=\toks18
\ex@=\dimen115
)) (c:/texlive/2009/texmf-dist/tex/latex/amsmath/amsbsy.sty
Package: amsbsy 1999/11/29 v1.2d
\pmbraise@=\dimen116
) (c:/texlive/2009/texmf-dist/tex/latex/amsmath/amsopn.sty
Package: amsopn 1999/12/14 v2.01 operator names
)
\inf@bad=\count97
LaTeX Info: Redefining \frac on input line 211.
\uproot@=\count98
\leftroot@=\count99
LaTeX Info: Redefining \overline on input line 307.
\classnum@=\count100
\DOTSCASE@=\count101
LaTeX Info: Redefining \ldots on input line 379.
LaTeX Info: Redefining \dots on input line 382.
LaTeX Info: Redefining \cdots on input line 467.
\Mathstrutbox@=\box30
\strutbox@=\box31
\big@size=\dimen117
LaTeX Font Info:    Redeclaring font encoding OML on input line 567.
LaTeX Font Info:    Redeclaring font encoding OMS on input line 568.
\macc@depth=\count102
\c@MaxMatrixCols=\count103
\dotsspace@=\muskip10
\c@parentequation=\count104
\dspbrk@lvl=\count105
\tag@help=\toks19
\row@=\count106
\column@=\count107
\maxfields@=\count108
\andhelp@=\toks20
\eqnshift@=\dimen118
\alignsep@=\dimen119
\tagshift@=\dimen120
```

```
\tagwidth@=\dimen121
\totwidth@=\dimen122
\lineht@=\dimen123
\@envbody=\toks21
\multlinegap=\skip48
\multlinetaggap=\skip49
\mathdisplay@stack=\toks22
LaTeX Info: Redefining \[ on input line 2666.
LaTeX Info: Redefining \] on input line 2667.
) (c:/texlive/2009/texmf-dist/tex/latex/tools/dcolumn.sty
Package: dcolumn 2001/05/28 v1.06 decimal alignment package (DPC)
(c:/texlive/2009/texmf-dist/tex/latex/tools/array.sty
Package: array 2008/09/09 v2.4c Tabular extension package (FMi)
\col@sep=\dimen124
\extrarowheight=\dimen125
\NC@list=\toks23
\extratabsurround=\skip50
\backup@length=\skip51
)) (c:/texlive/2009/texmf-dist/tex/latex/enumitem/enumitem.sty
Package: enumitem 2009/05/18 v2.2 Customized lists
\enit@toks=\toks24
\labelindent=\skip52
\enit@outerparindent=\dimen126
\enitdp@desc=\count109
) (c:/texlive/2009/texmf-dist/tex/latex/ltxmisc/url.sty
\Urlmuskip=\muskip11
Package: url 2006/04/12  ver 3.3  Verb mode for urls, etc.
) (c:/texlive/2009/texmf-dist/tex/latex/xcolor/xcolor.sty
Package: xcolor 2007/01/21 v2.11 LaTeX color extensions (UK)
(c:/texlive/2009/texmf-dist/tex/latex/latexconfig/color.cfg
File: color.cfg 2007/01/18 v1.5 color configuration of teTeX/TeXLive
)
Package xcolor Info: Driver file: dvips.def on input line 225.
Package xcolor Info: Model `cmy' substituted by `cmy0' on input line 1337.
Package xcolor Info: Model `RGB' extended on input line 1353.
Package xcolor Info: Model `HTML' substituted by `rgb' on input line 1355.
Package xcolor Info: Model `Hsb' substituted by `hsb' on input line 1356.
Package xcolor Info: Model `tHsb' substituted by `hsb' on input line 1357.
Package xcolor Info: Model `HSB' substituted by `hsb' on input line 1358.
Package xcolor Info: Model `Gray' substituted by `gray' on input line 1359.
Package xcolor Info: Model `wave' substituted by `hsb' on input line 1360.
(c:/texlive/2009/texmf-dist/tex/latex/graphics/dvipsnam.def
File: dvipsnam.def 1999/02/16 v3.0i Driver-dependant file (DPC,SPQR)
)) (c:/texlive/2009/texmf-dist/tex/latex/footmisc/footmisc.sty
Package: footmisc 2009/09/15 v5.5a a miscellany of footnote facilities
\FN@temptoken=\toks25
\footnotemargin=\dimen127
\c@pp@next@reset=\count110
Package footmisc Info: Declaring symbol style bringhurst on input line 855.
Package footmisc Info: Declaring symbol style chicago on input line 863.
Package footmisc Info: Declaring symbol style wiley on input line 872.
Package footmisc Info: Declaring symbol style lamport-robust on input line 883.
Package footmisc Info: Declaring symbol style lamport* on input line 903.
Package footmisc Info: Declaring symbol style lamport*-robust on input line 924
.
) (./evaluatingqald.aux)
\openout1 = `evaluatingqald.aux'.

LaTeX Font Info:    Checking defaults for OML/cmm/m/it on input line 24.
LaTeX Font Info:    ... okay on input line 24.
LaTeX Font Info:    Checking defaults for T1/cmr/m/n on input line 24.
LaTeX Font Info:    ... okay on input line 24.
LaTeX Font Info:    Checking defaults for OT1/cmr/m/n on input line 24.
LaTeX Font Info:    ... okay on input line 24.
LaTeX Font Info:    Checking defaults for OMS/cmsy/m/n on input line 24.
LaTeX Font Info:    ... okay on input line 24.
LaTeX Font Info:    Checking defaults for OMX/cmex/m/n on input line 24.
LaTeX Font Info:    ... okay on input line 24.
LaTeX Font Info:    Checking defaults for U/cmr/m/n on input line 24.
LaTeX Font Info:    ... okay on input line 24.
*geometry auto-detecting driver*
*geometry detected driver: dvips*
-------------------- Geometry parameters
paper: user defined
landscape: --
twocolumn: true
twoside: --
asymmetric: --
h-parts: 64.75394pt, 468.0pt, 64.75394pt
v-parts: 111.52342pt, 622.0pt, 111.52342pt
hmarginratio: 1:1
vmarginratio: 1:1
lines: --
heightrounded: --
bindingoffset: 0.0pt
truedimen: --
includehead: --
includefoot: --
includemp: --
driver: dvips
-------------------- Page layout dimensions and switches
\paperwidth  597.50787pt
\paperheight 845.04684pt
\textwidth  468.0pt
\textheight 622.0pt
\oddsidemargin  -7.51605pt
\evensidemargin -7.51605pt
\topmargin  -22.74657pt
\headheight 50.0pt
\headsep    12.0pt
\footskip   18.0pt
\marginparwidth 4.0pt
\marginparsep   10.0pt
```

```
\columnsep  24.0pt
\skip\footins  24.0pt plus 2.0pt minus 12.0pt
\hoffset 0.0pt
\voffset 0.0pt
\mag 1000
\@twocolumntrue
(1in=72.27pt, 1cm=28.45pt)
-----------------------

! LaTeX Error: File `introduction.tex' not found.

Type X to quit or <RETURN> to proceed,
or enter new name. (Default extension: tex)

Enter file name:
! Emergency stop.
<read *>

l.80 \input{introduction}
                         ^^M
*** (cannot \read from terminal in nonstop modes)


Here is how much of TeX's memory you used:
 3042 strings out of 493849
 37750 string characters out of 3152231
 106947 words of memory out of 3000000
 6290 multiletter control sequences out of 15000+200000
 10077 words of font info for 36 fonts, out of 3000000 for 9000
 714 hyphenation exceptions out of 8191
 32i,8n,35p,510b,219s stack positions out of 5000i,500n,10000p,200000b,50000s
No pages of output.
```