Question Answering over Linked Data (QALD-4)

Christina Unger[1], Corina Forascu[2], Vanessa Lopez[3], Axel-Cyrille Ngonga Ngomo[4], Elena Cabrio[5], Philipp Cimiano[1], and Sebastian Walter[1]

[1] CITEC, Bielefeld University, Germany
cunger@cit-ec.uni-bielefeld.de
cimiano@cit-ec.uni-bielefeld.de
swalter@techfak.uni-bielefeld.de
[2] Alexandru Ioan Cuza University of Iasi, Romania
corinfor@info.uaic.ro
[3] IBM Research, Dublin, Ireland
vanlopez@ie.ibm.com
[4] AKSW, University of Leipzig, Germany
ngonga@informatik.uni-leipzig.de
[5] INRIA Sophia-Antipolis Méditerrané, Cedex, France
elena.cabrio@inria.fr

1 Introduction

With the increasing amount of semantic data available on the web there is a strong need for systems that allow common web users to access this body of knowledge. Especially question answering systems have received wide attention, as they allow users to express arbitrarily complex information needs in an easy and intuitive fashion (for an overview see [4]). The key challenge lies in translating the users' information needs into a form such that they can be evaluated using standard Semantic Web query processing and inferencing techniques. Over the past years, a range of approaches have been developed to address this challenge, showing significant advances towards answering natural language questions with respect to large, heterogeneous sets of structured data. However, only few systems yet address the fact that the structured data available nowadays is distributed among a large collection of interconnected datasets, and that answers to questions can often only be provided if information from several sources are combined. In addition, a lot of information is still available only in textual form, both on the web and in the form of labels and abstracts in linked data sources. Therefore approaches are needed that can not only deal with the specific character of structured data but also with finding information in several sources, processing both structured and unstructured information, and combining such gathered information into one answer.

The main objective of the open challenge on *question answering over linked data*[6] [3] (QALD) is to provide up-to-date, demanding benchmarks that establishe a standard against which question answering systems over structured data can be evaluated and compared. QALD-4 is the fourth instalment of the QALD

[6] http://www.sc.cit-ec.uni-bielefeld.de/qald

open challenge, comprising three tasks: multilingual question answering, biomedical question answering over interlinked data, and hybrid question answering.

2 Task description

2.1 Task 1: Multilingual question answering

Task 1 is the core task of QALD and aims at all question answering systems that mediate between a user, expressing his or her information need in natural language, and semantic data. Given the English DBpedia 3.9 dataset[7] and a natural language question or set of keywords in one of seven languages (English, Spanish, German, Italian, French, Dutch, Romanian), the participating systems had to return either the correct answers, or a SPARQL query that retrieves these answers.

To get acquainted with the dataset and possible questions, a set of 200 training questions was provided. These questions were compiled from the QALD-3 training and test questions, slightly modified in order to account for changes in the DBpedia dataset. Later, systems were evaluated on 50 different test questions. These questions were mainly devised by the challenge organizers.

All training questions were manually annotated with keywords, corresponding SPARQL queries and with answers retrieved from the provided SPARQL endpoint. Annotations were provided in an XML format. Each of the questions specifies an ID for the question together with a range of other attributes explained below, the natural language string of the question in the seven languages, keywords in the same languages, a corresponding SPARQL query, as well as the answers this query returns. Along with a unique ID, the following attributes were specified for each question:

- `answertype` specifies the answer type, which can be one the following: `resource` (one or many resources, for which the URI is provided), `string` (a string value), `number` (a numerical value such as 47 or 1.8), `date` (a date provided in the format YYYY-MM-DD, e.g. 1983-11-02), `boolean` (either `true` or `false`).
- `aggregation` indicates whether any operations beyond triple pattern matching are required to answer the question (e.g., counting, filters, ordering).
- `onlydbo` is given only for DBpedia questions and reports whether the query relies solely on concepts from the DBpedia ontology.

Here is an example from the training set:

```
<question id="36" answertype="resource"
                  aggregation="false" onlydbo="false">

<string lang="en">
Through which countries does the Yenisei river flow?
</string>
```

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

```
<string lang="de">
Durch welche Länder fließt der Yenisei?
</string>
<string lang="es">
¿Por qué países fluye el río Yenisei?
</string>
...

<query>
PREFIX res: <http://dbpedia.org/resource/>
PREFIX dbp: <http://dbpedia.org/property/>
SELECT DISTINCT ?uri WHERE {
  res:Yenisei River dbp:country ?uri .
}
</query>

<answers>
<answer>
<uri>http://dbpedia.org/resource/Mongolia</uri>
</answer>
<answer>
<uri>http://dbpedia.org/resource/Russia</uri>
</answer>
</answers>
</question>
```

Of the 200 training questions, 38 questions require aggregation and 74 questions require namespaces other than from the DBpedia ontology. Of the 50 test questions, 15 questions require aggregation and 10 cannot be answered with the DBpedia ontology only. As an additional challenge, 12 training and 2 test questions are out of scope, i.e. they cannot be answered with respect to the dataset.

2.2 Task 2: Biomedical question answering over interlinked data

Also for the life sciences, linked data plays a bigger and bigger role. Already a tenth of the Linked Open Data cloud[8] consists of biomedical datasets. Especially biomedical data is distributed among a large collection of interconnected datasets, and answers to questions can often only be provided if information from several sources are combined. Task 2 therefore focuses on interlinked data. Given the following three biomedical datasets and a natural language question or set of keywords in English, the participating systems had to return either the correct answers or a SPARQL query that retrieves the answers.

- SIDER, describing drugs and their side effects
  `http://sideeffects.embl.de`
- Diseasome, encompassing description of diseases and genetic disorders
  `http://wifo5-03.informatik.uni-mannheim.de/diseasome/`

[8] `http://lod-cloud.net`

– Drugbank, describing FDA-approved active compounds of medication `http://www.drugbank.ca`

The training question set comprised 25 questions over those datasets. All training questions were provided in an XML format similar to the one used for Task 1. Since the focus of the task is on interlinked data, most of the questions require the integration of information from at least two of those datasets. Here is an example query (ommitting prefix definitions), representing the question What are the side effects of drugs used for Tuberculosis?

```
SELECT DISTINCT ?x
WHERE {
        disease:1154 diseasome:possibleDrug ?v2 .
        ?v2 rdf:type drugbank:drugs .
        ?v3 owl:sameAs ?v2 .
        ?v3 sider:sideEffect ?x .
}
```

Note that the drugs used for Tuberculosis are retrieved from Diseasome, and their side effects are retrieved from SIDER. The link between the relevant resources in these datasets (bound to `?v2` and `?v3`) is established using the OWL property `sameAs`.

Later, participating systems were evaluated on 25 similar test questions.

2.3 Task 3: Hybrid question answering

A lot of information is still available only in textual form, both on the web and in the form of labels and abstracts in linked data sources. Task 3 therefore focuses on the integration of both structured and unstructured information in order to gather answers. Given English DBpedia 3.9, containing both RDF data and free text available in the DBpedia abstracts, and a natural language question or keywords, participating systems had to retrieve the correct answer(s).

A set of 25 training questions was provided in an XML format that is very similar to the one used for Tasks 1 and 2. However, for this task, not only the RDF triples are relevant, but also the English abstracts, related to a resource by means of the property `abstract`.

All questions are annotated with a pseudo query and the correct answers. The pseudo query is like an RDF query but can contain free text as subject, property, or object of a triple. This free text is marked as `text:"..."`. Here is an example pseudo query for the question Give me the currencies of all G8 countries:

```
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?uri
WHERE {
  ?x text:"member of" text:"G8" .
  ?x dbo:currency ?uri .
}
```

This pseudo query contains two triples: One is an RDF triple retrieving the currency of a country, which is information that is only available as RDF data, not in the abstracts. And the other contains free text reducing the list of countries to those that are a member of the G8, which is information that is contained only in the abstracts, not in the RDF data. For example, the abstract for Canada contains the following sentence:

```
Canada is a recognized middle power and a member of many
    international institutions, including the G7, G8, G20,
    NATO, NAFTA, OECD, WTO, Commonwealth of Nations,
    Francophonie, OAS, APEC, and the United Nations.
```

3 Evaluation measures

The results submitted by participating systems were automatically compared to the gold standard results and evaluated with respect to precision and recall. For each question q, precision, recall and F-measure were computed as follows:

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

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

$$F\text{-}Measure(q) = \frac{2 * Precision(q) \times Recall(q)}{Precision(q) + Recall(q)}$$

On the basis of these measures, overall precision and recall values as well as an overall F-measure value were computed as the average mean of the precision, recall and F-measure values for all questions. In the results reported below, precision, recall and F-measure values refer to the averaged values.

4 Participating systems

Eight teams participated in QALD-4: four teams from Europe (one from France, one from Germany and two from Romania) and four teams from Asia (three from China and one from South Korea). Six participants took part in Task 1, three participants took part in Task 2, and there was one participant for Task 3 that later withdrew his submission. In the following, we give some details on those participating systems that are also described in working note papers.

Xser [8] takes as input a natural language question in English, and retrieves an answer in two steps. First the user query is linguistically analyzed in order to detect predicate argument structures through a semantic parser. Second the query is instantiated with respect to the knowledge base. Besides the DAG dependency parsing it relies on a structured prediction approach implemented

using a Collins-style hidden perceptron. The system requires training data but among all participants obtained the highest precision and recall values.

gAnswer [9] is a graph-driven question answering system that processes questions in two stages. First, based on the dependency parse of the question, a graph is build that represents the semantic structure of the question. Second, this graph is matched with subgraphs in the RDF dataset. Disambiguation takes place when evaluating subgraph matches. The system achieves real-time performance, requiring an average of 972 miliseconds to answer a question.

CASIA [7] proposes an algorithm based on Markov Logic Networks for learning a joint model for detecting phrases, mapping phrases to semantic items, and grouping semantic items into a graph. As a result, each step can be subject to global optimization. The system does not yet process questions which contain numbers and aggregation operations (such as filters, comparisons, or ordering), but shows very promising results on non-aggregation questions. It makes use of the Stanford Named Entity Recognizer, the PATTY and ReVerb resources, as well as *thebeast* tool[9] for weight learning and MAP inferencing.

Intui3 [1] accepts as input a natural language question and constructs its interpretation using syntactic and semantic cues in the question and a target triple store. First, the question is syntactically analyzed and chunked, and the named entities are identified. Then each chunk receives one or more interpretation depending on its type and on additional semantic and syntactic information available for that chunk. The interpretation of the question is then constructed by combining the interpretations assigned to each chunk, based on a set of combination rules that are attached to each type of interpretation. Finally, the question interpretation is mapped to a corresponding SPARQL query, which is then run against a SPARQL endpoint to retrieve the answers.

ISOFT [6] follows a template-based approach for transforming natural language questions into SPARQL queries. Based on a linguistic analysis of the input question, query templates and slots are determined, which are then filled by searching for appropriate concepts in the knowledge base, based on string similarity and Explicit Semantic Analysis.

The Faculty of Computer Science at Alexandru Ioan Cuza University of Iasi, Romania, participated with two systems (**RO_FII**), one tackling question answering over DBpedia and one tackling interlinked biomedical datasets.[10] The former builds on Quepy[11], a Python tool for transforming natural language questions into SPARQL or MQL queries. The latter comprises three components, based on Service Oriented Architecture principles: a text annotator that receives the question in plain text and returns a list of compound words annotated with POS tags and lemmas (using Standford Core NLP), the triple builder that builds a list of triples given a list of keywords and URIs (currently assembled manu-

[9] `http://code.google.com/p/thebeast`

[10] The former was built by Andrei Micu, the latter was built by Claudiu Epure, both supervised by Adrian Iftene.

[11] `http://quepy.machinalis.com`

ally), and a query builder that builds the final SPARQL query on the basis of the list of annotated words and the list of triples.

GFMed [5] follows a controlled natural language approach for biomedical question answering. It builds on a Grammatical Framework[12] (GF) grammar for the biomedical datasets DrugBank, Diseasome, and SIDER. GF is a special-purpose programming language for writing multilingual grammars. For GFMed, an abstract syntax for the biomedical domain, spanning the concepts of the three datasets, as well as two concrete syntaxes that provide linearizations of those concepts, one for English and one for SPARQL, were created manually. In addition, a lexicon for both languages that covers all resource names was automatically constructed based on their labels. The resulting grammar allows to transform English questions into SPARQL queries by parsing the English input, yielding an abstract syntax representation that can then be linearized using the SPARQL concrete syntax. The approach can deal with complex questions and achieves a very high precision, but as any controlled language its coverage is limited, especially it does not easily scale to other domains.

POMELO [2] POMELO operates (in contrast to most other approaches) on frames. First, the RDF dataset is converted to frames: The predicates of the RDF triples are mapped to frame predicates while the subjects and objects are mapped to core frame elements. Each question is transformed to a SPARQL query using a four-step approach: First, the query is annotated with semantic and linguistic information using the converted resources. For example, numbers and resources from different datasets are tagged as such. Thereafter, in the question abstraction step, argument and predicate descriptions are used to generate a query template. This template is completed by using `owl:sameAs` and used to construct a SPARQL query skeleton, which is finally used to generate a final SPARQL query.

5 Results

Tables 1 and 2 report on the results obtained by the participating systems on Tasks 1 and 2, respectively. The column *proc.* 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 *part.* specifies how many of the questions were answered with an F-measure strictly between 0 and 1.

The results in Task 1 are comparable to results achieved in earlier challenges, with an average F-measure of 0.33, showing that the level of complexity of the questions is still very demanding. But what has changed with respect to earlier challenges is that question answering systems have become more versatile: There is no particular type of questions anymore that systems struggle with, rather most of them can handle all answer types as well as aggregation. The biggest problem, however, remains the matching of natural language questions to correct vocabulary elements. For example, the questions that all systems struggled with

[12] `http://www.grammaticalframework.org`

Table 1. Results for Task 1: Multilingual question answering over DBpedia

	Total	Proc.	Right	Part.	Recall	Precision	F-measure
Xser	50	40	34	6	0.71	0.72	**0.72**
gAnswer	50	25	16	4	0.37	0.37	**0.37**
CASIA	50	26	15	4	0.40	0.32	**0.36**
Intui3	50	33	10	4	0.25	0.23	**0.24**
ISOFT	50	28	10	3	0.26	0.21	**0.23**
RO_FII	50	50	6	0	0.12	0.12	**0.12**

Table 2. Results for Task 2: Biomedical question answering over interlinked data

	Total	Proc.	Right	Part.	Recall	Precision	F-measure
GFMed	25	25	24	1	0.99	1.0	**0.99**
POMELO	25	25	19	3	0.87	0.82	**0.85**
RO_FII	25	25	4	0	0.16	0.16	**0.16**

are surprisingly simple with respect to the linguistic structure and the structure of the target query:

- How deep is Lake Placid?
  ```
  SELECT ?n WHERE {
  res:Lake_Placid_(Texas) dbo:depth ?n . }
  ```
- Which spaceflights were launched from Baikonur?
  ```
  SELECT ?uri WHERE {
  ?uri dbo:launchPad res:Baikonur_Cosmodrome . }
  ```

6 Future perspectives

QALD-4, the fourth edition of the QALD challenge, has attracted a higher number of participants than previous editions, showing that there is a growing interest among researchers to provide end users with an intuitive and easy-to-use access to the huge amount of data present on the Semantic Web. Although one of the aspects of Task 1 was multilinguality, all participating systems worked on English data only. This shows that the multilingual scenario is not yet broadly addressed, although it is starting to attract attention. Similarly, research teams start to look at hybrid question answering, although Task 3 did not have participating systems yet.

In future challenges, we want to emphasize further aspects of question answering over linked data, such as including statistical question answering (e.g. How much money was spent for public transport in Berlin in 2014?), introducing spoken language in addition to written language (an aspect that is interesting especially for search engines) as well as dialogue-based interaction into the challenge, allowing the system to ask for feedback or clarification, as well as the user to refer to previous questions and answers, thus moving to question answering

systems that can exploit the previous interaction context in interpreting new questions.

References

1. Corina Dima. Answering natural language questions with Intui3. In *CLEF 2014 Working Notes Papers*, 2014.
2. Thierry Hamon, Natalia Grabar, Fleur Mougin, and Frantz Thiessard. Description of the POMELO system for the task 2 of QALD-2014. In *CLEF 2014 Working Notes Papers*, 2014.
3. Vanessa Lopez, Christina Unger, Philipp Cimiano, and Enrico Motta. Evaluation question answering over linked data. *Journal of Web Semantics*, in press.
4. Vanessa Lopez, Victoria S. Uren, Marta Sabou, and Enrico Motta. Is question answering fit for the semantic web?: A survey. *Semantic Web*, 2(2):125–155, 2011.
5. Anca Marginean. GFMed: Question answering over biomedical linked data with Grammatical Framework. In *CLEF 2014 Working Notes Papers*, 2014.
6. Seonyeong Park, Hyosup Shim, and Gary Geunbae Lee. ISOFT at QALD-4: Semantic similarity-based question answering system over linked data. In *CLEF 2014 Working Notes Papers*, 2014.
7. He Shizhu, Zhang Yuanzhe, Kang Liu, and Jun Zhao. CASIA@V2: A MLN-based question answering system over linked data. In *CLEF 2014 Working Notes Papers*, 2014.
8. Kun Xu, Yansong Feng, and Dongyan Zhao. Answering natural language questions via phrasal semantic parsing. In *CLEF 2014 Working Notes Papers*, 2014.
9. Lei Zou, Ruizhe Huang, Haixun Wang, Jeffrey Xu Yu, Wenqiang He, and Dongyan Zhao. Natural langauge question answering over RDF – a graph data driven approach. In *Proceedings of SIGMOD 2014*, 2014.