# Benchmarking Question Answering Systems

Ricardo Usbeck[1], Michael Röder[1], Christina Unger[2], Michael Hoffmann[1], Christian Demmler[1], Jonathan Huthmann[1], and Axel-Cyrille Ngonga Ngomo[1]

[1] AKSW Group, University of Leipzig, Germany
usbeck|roeder|ngonga@informatik.uni-leipzig.de
[2] CITEC,University of Bielefeld
cunger@cit-ec.uni-bielefeld.de

**Abstract.** The need for making the Semantic Web better accessible for lay users and the uptake of interactive systems and smart assistants for the Web have spawned a new generation of RDF-based question answering systems. However, comparing the quality of these systems, repeating the published experiments or running on the same datasets remains a complex and time-consuming task. Thus, we extended the GERBIL benchmarking framework to support the fine-grained evaluation of question answering systems. In this paper, we describe the evaluation paradigm underlying our extension. In addition, we present the current implementation of the solution including different measures, datasets and pre-implemented systems as well as possibilities to work with novel formats for interactive and non-interactive benchmarking of question answering systems. One particular feature of our framework lies in its provision of diagnostics, through which developers are provided with insights pertaining to the weakness and strengths of their systems. Therewith, we provide an open benchmarking suite that can potentially speed up the development of future systems.

## 1 Introduction

The Web of Data has grown to contain billions of facts pertaining to a large variety of domains. While this wealth of data can be easily accessed by experts, it remains difficult to use for non-experts [4, 17]. This need has led to the development of a large number of question answering (QA) and keyword search tools for the Web of Data [14, 15]. As benchmarking has been credited with the more rapid advancement of research, many campaigns and challenges have evolved around the QA research field (see Section 2) since the first question answering system [5]. However, evaluation datasets, measures and QA system processes are hardly documented nor is there a continuous overview of existing frameworks or test beds.

Thus, we extended the GERBIL evaluation framework [18] so as to address the needs of the QA community for citable, comparable and extensible in-depth benchmarking of QA systems. Here, we follow the FAIR principles (Findable, Accesible, Interoperatable, Re-Usable)[3] by enabling the linking of experiment results via W3ID[4] URIs and offering a public SPARQL endpoint at http://gerbil-qa.aksw.org/

[3] https://www.force11.org/group/fairgroup/fairprinciples

[4] https://w3id.org/

sparql. An overview of GERBIL is given in Figure 1. Our framework supports both online systems and file-based evaluation campaigns over a large variety of datasets.Our contributions include:

- The provision of citable, stable experiment URIs and descriptions, which are both human and machine-readable[5].
- Enabling the comparison against 4 existing QA systems, on 21 datasets (QALD-1 to QALD-6 and NLQ).
- Allowing for the upload of system results on datasets as well as for the connection to Web-service-based systems on the fly.
- Metrics for benchmarking QA systems as well as QA sub-experiment types to improve the diagnostics process.
- The support of several formats for the interactive communication of QA systems via Web-service calls.

A demo of the QA benchmarking system is available at http://gerbil-qa.aksw.org/gerbil/. Furthermore, we made the datasets, utilities and the source code openly available and extensible.[6]

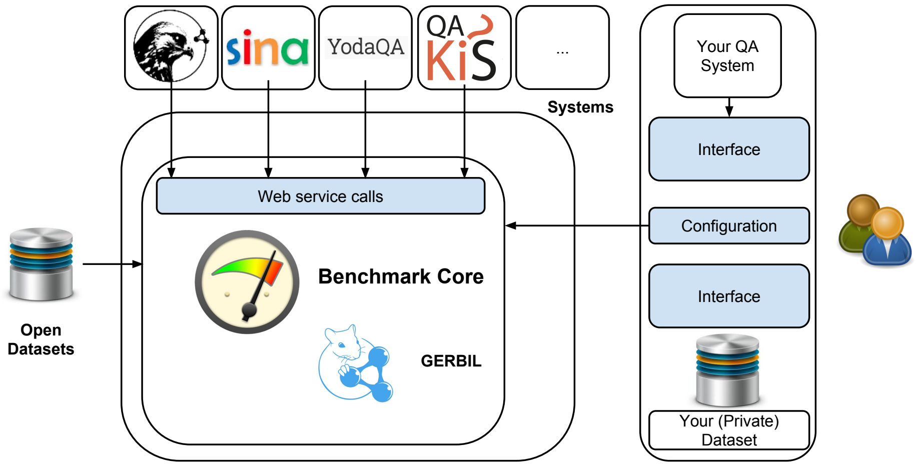


**Fig. 1.** Overview of the GERBIL QA Benchmarking platform.

[5] GERBIL uses the recently proposed DataID [3] ontology that combines VoID [1] and DCAT [8] metadata with Prov-O [7] provenance information and ODRL [9] licenses to describe datasets.

[6] https://github.com/AKSW/NLIWOD and https://github.com/AKSW/gerbil/tree/QuestionAnswering

## 2 Question Answering Benchmarking Campaigns

Since 1998, the TREC conference, especially the QA track [19], aims at providing domain-independent evaluations over large, unstructured corpora. This seminal campaign pushed research projects forwards over the course of its more than ten implementations. The latest TREC-QA tackles the field of live QA[7] where systems answer real-life, real-time questions of users submitted to a popular community-based Question and Answer sites.

Next to that, the BioASQ series [13] challenges semantic indexing as well as QA systems on biomedical data and is currently at its fifth implementation. Here, systems have to work on RDF as well as textual data to present matching triples as well as text snippets. Moreover, the OKBQA[8] is primarily an open QA platform powered by several Korean research institutes but they also released the NLQ datasets within their 3rd hackathon[9]. This dataset is answerable purely by Wikipedia respectively by DBpedia using SPARQL.

The well-known QALD (Question Answering over Linked Data) [15] campaign, currently running in its 6th instantiation, is a diverse evaluation series including 1) RDF-based, 2) hybrid, i.e., RDF and textual data, 3) statistical as well as 4) multi knowledge base and 5) music-domain-based benchmarks. Thus, we will use the QALD datasets and format as a base for our benchmarking suite, since they are adopted by more than 20 QA systems since 2011 [6]. So far, yearly QALD events enable participants to upload XML or JSON-based system answers to previously uploaded files on the QALD website. Our platform will allow us to a) use curated, updated benchmark datasets (e.g., via github) instead of once-uploaded-static files, b) refer specific experiments to specific versions of datasets and c) implement wrappers for QA systems respectively using REST interfaces in an interactive manner to benchmark QA systems online and in real-time (see example below). We refer the interested reader to our dataset project homepage[10] to read up more or add novel datasets.

## 3 Datasets

In its current version, our framework supports 20 QALD campaign datasets[12] as well as the OKBQA NLQ shared task 1[13] listed in table 1. It is important to note that no evaluation campaign, especially QALD and OKBQA, offers endpoints for all knowledge bases, i.e., developers and end users have to setup their own knowledge base (KB) endpoint for the respective version. Adding supplementary datasets to GERBIL is easy. One can either add it to our project repository and write a dataset wrapper in Java or one can upload a dataset as a file via our Web-interface for only one particular experiment. Note that the first option enables other users to benchmark with this dataset and can

[7] https://sites.google.com/site/trecliveqa2016/

[8] http://www.okbqa.org

[9] http://2015.okbqa.org/nlq

[10] https://github.com/AKSW/NLIWOD/tree/master/qa.datasets

[12] http://qald.sebastianwalter.org/

[13] http://3.okbqa.org/nlq

**Table 1.** Build-in datasets and their features.

| Dataset | #Questions | Knowledge Base |
|---|---|---|
| NLQ shared task 1 | 39 | DBpedia 2015-04 |
| QALD1_Test_dbpedia | 50 | DBpedia 3.6 |
| QALD1_Train_dbpedia | 50 | DBpedia 3.6 |
| QALD1_Test_musicbrainz | 50 | MusicBrainz[11] (dump 2011) |
| QALD1_Train_musicbrainz | 50 | MusicBrainz (dump 2011) |
| QALD2_Test_dbpedia | 99 | DBpedia 3.7 |
| QALD2_Train_dbpedia | 100 | DBpedia 3.7 |
| QALD3_Test_dbpedia | 99 | DBpedia 3.8 |
| QALD3_Train_dbpedia | 100 | DBpedia 3.8 |
| QALD3_Test_esdbpedia | 50 | DBpedia 3.8 es |
| QALD3_Train_esdbpedia | 50 | DBpedia 3.8 es |
| QALD4_Test_Hybrid | 10 | DBpedia 3.9 + long abstracts |
| QALD4_Train_Hybrid | 25 | DBpedia 3.9 + long abstracts |
| QALD4_Test_Multilingual | 50 | DBpedia 3.9 |
| QALD4_Train_Multilingual | 200 | DBpedia 3.9 |
| QALD5_Test_Hybrid | 10 | DBpedia 2014 + long abstracts |
| QALD5_Train_Hybrid | 40 | DBpedia 2014 + long abstracts |
| QALD5_Test_Multilingual | 49 | DBpedia 2014 |
| QALD5_Train_Multilingual | 300 | DBpedia 2014 |
| QALD6_Train_Hybrid | 49 | DBpedia 2015-10 + long abstracts |
| QALD6_Train_Multilingual | 333 | DBpedia 2015-10 |

thus spark the generation of new datasets. The input format supported by the upload are JSON and XML files in the QALD format.

## 4 Systems

The first release of GERBIL QA contains 4 pre-implemented systems—wrapped via Java—, capable of answering hybrid and multilingual questions as well as keyword queries. These systems are:

1. HAWK [16], the first hybrid source QA system which processes RDF as well as textual information to answer one input query.
2. SINA [12], a keyword and natural language query search engine which exploits the structure of RDF graphs to implement an explorative search approach.
3. YodaQA [2], a modular, open-source, hybrid approach built on top of the Apache UIMA framework[14].
4. QAKIS [4], an agnostic QA system grounded in ontology-relation matches.

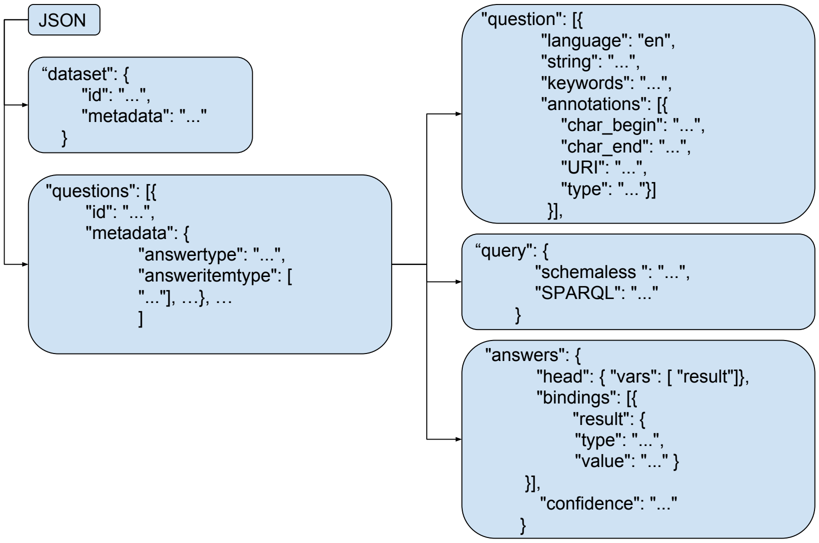
Currently, GERBIL QA supports the addition of three types of systems, implemented as wrapper or Web-interface using QALD JSON or XML as result format.

[14] `https://uima.apache.org/`

(1) Basic QA systems accept a question as a string via their Web interface. They return a single or a list of answers. The simplicity of this approach allows measuring the QA performance of a system but does not allow for a deeper analysis of the behavior of the system. (2) More advanced types of systems return not only answers but also the SPARQL query it executed. Therewith, it allows for the extraction of information that allow benchmarking the system in depth, i.e, enables benchmarking w.r.t. every sub-experiment except answer type comparison (see Section 5). Finally, (3) we propose to implement a system using an extended QALD JSON-schema to represent answers of a QA system to support the full benchmark set, see Figure 2. This elaborated format includes 1) a knowledge base version, 2) questions in multiple languages and also described via keywords, 3) annotations of the question w.r.t. RDF resources and properties, 4) meta-information like answer type and answer item type, 5) a schema-less[15] as well as a SPARQL query and 6) answers from the KB formated compliant with the W3C JSON-RDF standard[16] as well as confidence scores for further evaluations.

We aim to standardize this and extend this format for natural language interfaces (see Section 6).

In addition to supporting integrated systems, our platform offers uploading a result file containing the answers in QALD's XML or JSON format. This enables developers to benchmark their system without setting up a web-service endpoint under a public address. Within the main GERBIL platform, experiments and log files remain private until published, i.e., companies and interested parties can test their systems online without fearing premature publication.



```
“dataset": {
    "id": "...",
    "metadata": "..."
  }
```

```
"questions": [{
    "id": "...",
    "metadata": {
        "answertype": "...",
        "answeritemtype": [
        "..."], ...}, ...
        ]
```

```
"question": [{
        "language": "en",
        "string": "...",
        "keywords": "...",
        "annotations": [{
            "char_begin": "...",
            "char_end": "...",
            "URI": "...",
            "type": "..."}]
        }],
```

```
“query": {
        "schemaless ": "...",
        "SPARQL": "..."
    }
```

```
"answers": {
        "head": { "vars": [ "result"]},
        "bindings": [{
            "result": {
            "type": "...",
            "value": "..." }
      }],
        "confidence": "..."
    }
```

**Fig. 2.** Structural overview of the extended QALD JSON format to enable all 5 sub-experiments.

[15] https://sites.google.com/site/eswcsaq2015/documents

[16] https://www.w3.org/TR/sparql11-results-json/

## 5 Experiments

**Experiments and Matchings** In addition to being able to benchmark whole QA systems, GERBIL QA allows measuring the performance of common components of QA systems (named entity recognition, entity linking, etc.). We use the term *sub-experiments* to denote experiments for benchmarking such sub-components. We designed and implemented 5 sub-experiments inspired by past evaluation campaigns. The data necessary to carry out these sub-experiments can be provided via the extended QALD JSON-schema proposed before. For four of the following five sub-experiments, the needed data can also be derived from the SPARQL query that might be returned by the QA system via the second model.

**Question Answering.** The first experiment is the tradition experiment as described by evaluation campaigns like OKBQA and QALD. It aims to measure the capability of a system to answer questions correctly. A system's answer and the corresponding gold standard answer are regarded as set of URIs and literals and the traditional precision, recall and F-measure used for evaluation.

**Resource to Knowledge Base (C2KB).** This sub-experiment aims at the identification of all resources that are relevant for the given question. It is known from GERBIL [18] as *Concept to KB*. The evaluation calculates the measures precision, recall and F-measure based on the comparison of the expected resource URIs and the URIs returned by the QA system. Instead of a simple string comparison we make use of an advanced meaning matching implementation offered by GERBIL and explained in [11].

**Properties to Knowledge Base (P2KB).** This sub-experiment is a special form of the C2KB sub-experiment type. For this experiment, the system has to identify all properties that are relevant for the given question but follows the process of the C2KB experiment.

**Relation to Knowledge Base (RE2KB).** This sub-experiment focuses on the triples that have to be extracted from the question and are needed to generate the SPARQL query that would retrieve the correct answers. These triples can contain resources, variables and literals. The evaluation of this sub-experiment calculates precision, recall and F-measure based on the comparison of the expected triples and the triples returned by the QA system. For achieving a true positive, a returned triple has to match an expected triple. Two triples are counted as matching if they contain the same resources at the same positions. If they contain variables, the positions of the variables have to be the same but the variable names are ignored. If they contain a literal, the value of the literal has to be the same.

**Answer Type (AT).** The identification of the answer type is an important part of a QA system. We distinguish 4 different answer types extracted from the QALD benchmarking campaign [15], i.e., date, number, string and list of resources. For every question a single answer type is expected for which the F-measure is calculated. Note that this sub experiment can only generate meaningful results if the extended QALD JSON schema is used.

**Answer Item Type to Knowledge Base (AIT2KB).** The answer item types are the `rdf:type` information of the returned resources. Precision, recall and F-measure are calculated based on the set of expected types. If the expected answer set of a question does not contain resources the set of answer item types is expected to be empty.

**Metrics** Our platform uses micro- as well as macro-precision, recall and F-measure [18]. However, GERBIL offers the implementation of additional metrics [11]. Thus, it would be possible to use a hierarchical F-measure, e.g., for the AIT2KB sub-experiment [10].

In addition to these result-focused metrics, our system measures the performance of live systems in two ways. First, it computes the average time a system needs to generate a response. Second, the number of errors the system returns or that occur during the communication with the system are counted.

For example, this experiment[17] describes a stable URL of an experiment with four QA systems, three pre-implemented as well as an uploaded QALD XML, on two datasets, namely QALD-5-train multilingual and hybrid. The uploaded HAWK file suggests an improvement over the pre-implemented HAWK system. The pre-implemented HAWK system however performs better on hybrid questions than on plain English questions. Systems like YODA, which do only provide answers without a SPARQL query cannot be analysed sufficiently. However, systems implementing method (2), i.e., also providing a SPARQL query, can be analysed towards there performance in the sub-experiments. For instance, HAWKs entity recognition abilities (C2KB task) outperform the other systems in that respect.

## 6 Conclusion & Future work

We present the first online, live benchmarking system for question answering approaches. Our platform strives to speed up the development process by offering diverse datasets, systems and interfaces to generate repeatable and citable experiments with in-depth analytics of a system's performance. By these means, we hope to speed up the process of QA development.

A known limitation is our focus on RDF-based systems (RDF resource matching, required SPARQL query for sub-experiments) which we seek to circumvent in the future by using a standard to let interfaces communicate the needed information with demanding a SPARQL query within the result set.

In near-future developments, we will add additional metrics (hierachical f-measure), novel datasets, more systems as well as unify the way of matching system answers with gold standard answers and thus pushing a fast-pace, open science movement. Furthermore, we will add this benchmarking platform to the HOBBIT project[18] for a wider spread of our activities. Finally, we will bring this development to the W3C community group of Natural Language Interfaces for the Web of Data to standardize system interfaces and allow for an even easier and concise benchmarking.[19]

**Acknowledgements** This work has been supported by Eurostars projects DIESEL (E!9367) and QAMEL (E!9725) as well as the European Union's H2020 research and innovation action HOBBIT (GA 688227). We thank the QANARY team for inspiring discussions. Furthermore, we want to thank Jin-Dong Kim for his thoughts on the novel QA format.

[17] `http://w3id.org/gerbil/qa/experiment?id=201605010001`

[18] `http://project-hobbit.eu/`

[19] `https://www.w3.org/community/nli/`

## References

1. K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing linked datasets with the void vocabulary, 2011. http://www.w3.org/TR/void/.
2. P. Baudiš and J. Šedivý. *CLEF'15*, chapter Modeling of the Question Answering Task in the YodaQA System, pages 222–228. Springer International Publishing, 2015.
3. M. Brümmer, C. Baron, I. Ermilov, M. Freudenberg, D. Kontokostas, and S. Hellmann. DataID: Towards semantically rich metadata for complex datasets. In *I-SEMANTICS*, 2014.
4. E. Cabrio, J. Cojan, F. Gandon, and A. Hallili. Querying Multilingual DBpedia with QAKiS. In *ESWC*, pages 194–198, 2013.
5. B. F. Green Jr, A. K. Wolf, C. Chomsky, and K. Laughery. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, pages 219–224. ACM, 1961.
6. K. Höffner, S. Walter, E. Marx, J. Lehmann, A.-C. Ngonga Ngomo, and R. Usbeck. Overcoming Challenges of Semantic Question Answering in the Semantic Web. *Submitted to Semantic Web Journal*, 2016.
7. T. Lebo, S. Sahoo, D. McGuinness, K. Belhajjame, J. Cheney, D. Corsar, D. Garijo, S. Soiland-Reyes, S. Zednik, and J. Zhao. PROV-O: The PROV Ontology, 2013. http://www.w3.org/TR/prov-o/.
8. F. Maali, J. Erickson, and P. Archer. Data Catalog Vocabulary (DCAT), 2014. http://www.w3.org/TR/vocab-dcat/.
9. M. McRoberts and V. Rodríguez-Doncel. Open Digital Rights Language (ODRL) Ontology, 2014. http://www.w3.org/ns/odrl/2/.
10. M. Röder, R. Usbeck, and A.-C. Ngonga Ngomo. Developing a Sustainable Platform for Entity Annotation Benchmarks. In *ESWC Developers Workshop 2015*, 2015.
11. M. Röder, R. Usbeck, and A.-C. Ngonga Ngomo. Gerbil's new stunts: Semantic annotation benchmarking improved. Technical report, Leipzig University, 2016.
12. S. Shekarpour, E. Marx, A.-C. N. Ngomo, and S. Auer. Sina: Semantic interpretation of user queries for question answering on interlinked data. *Journal of Web Semantics*, 2014.
13. G. Tsatsaronis, M. Schroeder, G. Paliouras, Y. Almirantis, I. Androutsopoulos, E. Gaussier, P. Gallinari, T. Artieres, M. R. Alvers, M. Zschunke, et al. BioASQ: A challenge on large-scale biomedical semantic indexing and Question Answering. In *2012 AAAI Fall Symposium Series*, 2012.
14. C. Unger, C. Forascu, V. Lopez, A. N. Ngomo, E. Cabrio, P. Cimiano, and S. Walter. Question answering over linked data (QALD-4). In *CLEF*, pages 1172–1180, 2014.
15. C. Unger, C. Forascu, V. Lopez, A. N. Ngomo, E. Cabrio, P. Cimiano, and S. Walter. Question answering over linked data (QALD-5). In *CLEF*, 2015.
16. R. Usbeck, A. N. Ngomo, L. Bühmann, and C. tina Unger. HAWK – Hybrid Question Answering Using Linked Data. In *The Semantic Web. Latest Advances and New Domains - 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31 - June 4, 2015. Proceedings*, pages 353–368, 2015.
17. R. Usbeck and A.-C. Ngonga Ngomo. HAWK@QALD5 – Trying to answer hybrid questions with various simple ranking techniques. In *CLEF 2015 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings (CEUR-WS.org/Vol-1391)*, 2015.
18. R. Usbeck, M. Röder, A.-C. Ngonga Ngomo, C. Baron, A. Both, M. Brümmer, D. Ceccarelli, M. Cornolti, D. Cherix, B. Eickmann, P. Ferragina, C. Lemke, A. Moro, R. Navigli, F. Piccinno, G. Rizzo, H. Sack, R. Speck, R. Troncy, J. Waitelonis, and L. Wesemann. GERBIL – General Entity Annotation Benchmark Framework. In *24th WWW conference*, 2015.
19. E. M. Voorhees et al. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82, 1999.