EU H2020 Research and Innovation Project

HOBBIT – Holistic Benchmarking of Big Linked Data

Project Number: 688227 Start Date of Project: 01/12/2015 Duration: 36 months

Deliverable 6.1.1
First Version of the Question Answering Benchmark

Dissemination Level	Public
Due Date of Deliverable	Month 18, 31/05/2017
Actual Submission Date	Month 18, 15/05/2017
Work Package	WP 6
Task	T 6.1
Type	Report
Approval Status	Approved
Version	1.0
Number of Pages	14
Filename	D6.1.1_QA_Benchmark_V1.pdf

Abstract: This report describes the results of the tasks related to the first version of the question answering benchmark. It includes a description of the work that has been carried out as well as a detailed description of the different available experiments.

The information in this document reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided "as is" without guarantee or warranty of any kind, express or implied, including but not limited to the fitness of the information for a particular purpose. The user thereof uses the information at his/ her sole risk and liability.

Project funded by the European Commission within the H2020 Framework Programme

History

Version	Date	Reason	Revised by
0.0	25/04/2017	First Draft	Bastian Haarmann
0.1	28/04/2017	Internal review comments	Henning Petzka
0.2	08/05/2017	Peer review comments	Irini Fundulaki
1.0	15/05/2015	Approval	Bastian Haarmann

Author List

Organisation	Name	Contact Information
IAIS	Bastian Haarmann	bastian.haarmann@iais.fraunhofer.de

Executive Summary

This report describes the task 6.1 activities of the HOBBIT project for the first version of the Question Answering Benchmark. The main content of this document is a detailed description of the activities from July 1st, 2016 until the end of May 2017. The report includes information about the work that has been carried out by Fraunhofer IAIS and other participating partners. It also describes in detail the three experiment tasks of the QA Benchmark and walks through an example experiment.

This report ends with an outlook on the next steps toward the second version of the Question Answering Benchmark.

Table of Contents

List of Figures

List of Tables

1. Introduction

The past years have seen a growing amount of research on question answering over large-scale RDF data. At the same time, the growing amount of data has led to a heterogeneous data landscape. The general goal of a Question Answering (QA) system is to compute the correct answer to a natural language query given a number of structured datasets.

The benchmarking of QA systems results in a score that corresponds to the amount of correct answers from the participating system on the given natural language questions. The QA Benchmark makes it possible to rank question answering systems based on their performance and accuracy and to make statements about their excellence and quality. Previous work on benchmarking question answering systems include Yahoo! Semantic Search[1], GERBIL QA[2] and QALD[3].

In general, the underlying datasets are organized in factual triples including a subject, predicate and object of a so-called statement according to the RDF standard. A triple consists of an entity as subject which the factual statement is about, a predicate forming the attribute of the entity and an object as the value for the attribute. SPARQL can be used as a query language to gather answers from the dataset. Therefore, the main task of a QA system is to translate a natural language question into an appropriate SPARQL query which delivers the correct answer.

Since most open-source question answering systems operate on the dataset of DBpedia[4] and it is commonly used in QA challenges such as QALD, the first version of the Question Answering Benchmark in HOBBIT will only include questions concerning the DBpedia dataset. This is to ensure comparability among common QA systems and a consequent continue of previous benchmarking work such as GERBIL QA. Subsequent versions of this benchmark are to be extended by different datasets.

The Question Answering over Linked Data (QALD) challenges have taken place for six years and are planned to be continued. Question Answering systems can take part in the annual challenge and subscribe to different tasks such as a multilingual or a datacube task. QALD tasks might be changed, discontinued or renewed from year to year. For the continued tasks, a system can get the corresponding text questions and correct answers from all previous QALD editions for training purposes. The HOBBIT Question Answering Benchmark and its three tasks described here will be part of the forthcoming challenges QALD-7 and QALD-8.

2. Choke Points and Goals

When a question answering system is provided with a natural language query and expected to find the corresponding answer in one or more accessible datasets, it is clear that these queries must be factoid in nature and exclude any opinion-based or tentative traits. Beyond this, there are numerous choke points for a system to be mindful of.

[1] http://krisztianbalog.com/2011-03-03/yahoo-semantic-search-challenge/ retrieved on 02/05/2017

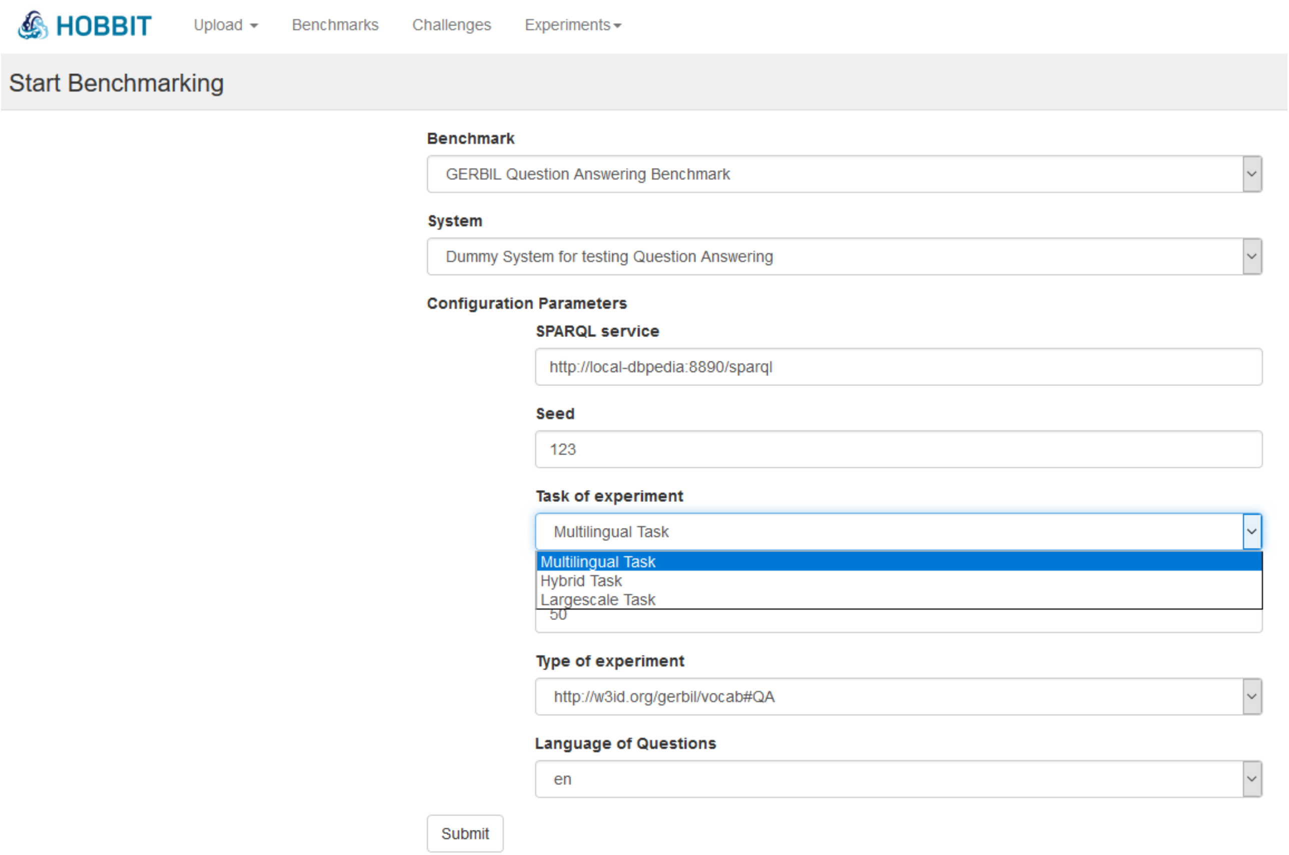
[2] http://gerbil-qa.aksw.org/gerbil/config/ retrieved on 02/05/2017

[3] http://qald.sebastianwalter.org/ retrieved on 02/05/2017

[4] http://wiki.dbpedia.org/ retrieved on 02/05/2017

Some important choke points include common NLP[5] problems like word sense ambiguity, such as bass (sound) vs. bass (fish), and vague queries like „*near London*“. Moreover, some natural language queries might require lexical knowledge (astronauts vs. kosmonauts vs. taikonauts, depending on nationality) or knowledge on word derivation (i.e. „*tall*“ refers to a property named „*height*“; „*communist country*“ means Country governmentType communism).

The goal of WP 6.1 is to compile a benchmark for QA systems including well defined metrics for evaluation. The performance and accuracy measurements are going to be carried out on the HOBBIT platform.



Start Benchmarking

Benchmark: GERBIL Question Answering Benchmark

System: Dummy System for testing Question Answering

Configuration Parameters

SPARQL service: http://local-dbpedia:8890/sparql

Seed: 123

Task of experiment: Multilingual Task

- Multilingual Task
- Hybrid Task
- Largescale Task

50

Type of experiment: http://w3id.org/gerbil/vocab#QA

Language of Questions: en

Figure 1: Selection of experiment task in the Question Answering Benchmark

Included tasks are going to tackle multilingual question answering over DBpedia, such that answers can be retrieved from an RDF data repository given an information need expressed in a variety of languages (including English, German, Dutch, French, Spanish, Italian, Romanian, Persian and Hindi), hybrid question answering that requires the integration both from RDF and textual data sources and large-scale question answering including a massive amount of automatically derived questions taking into account not only a system’s accuracy on answers but also the time needed to retrieve these answers.

The goal of this workpackage does not include development, evaluation or benchmarking of user interfaces, neither user surveys.

[5] Natural language processing

3. Work performed and results

The aim for the first version of the Question Answering Benchmark in HOBBIT is to assist the assessment of QA systems with a fixed set of natural language questions and their respective SPARQL queries. Systems can be assessed in three tasks, each tackling a different number of choke points.

As Key Performance Indicators, we set the usual suspects: precision, recall, F1-score and, in the large-scale task, we record the systems' time for successfully answered questions while constantly increasing the number of issued questions.

We list specific steps of the progress that was made since the start of this workpackage and sum up the current state before going into details of each task.

Work Point
We collected the main choke points of question answering, that is, the difficulties that arise for a system in answering natural language questions based on structured datasets.
We explored our possibilities to develop benchmarking scenarios that extend already existing benchmarks.
We investigated the characteristics of several HOBBIT datasets for their suitability as an underlying question answering database.
We compiled 100 new gold standard test questions for the multilingual and hybrid task including their SPARQL queries and their correct answers. InfAI delivered the translations for the multilingual questions.
We identified 150 question-query pairs from previous challenges suitable for being a template for generating a large amount of new questions and queries.
We replaced the instance data (mostly named entities) in the template question-query pairs by instance data of the same class and generated more than 2 million new valid gold standard question-query-pairs for the large-scale task.
We wrote a task generator, a data generator module and the benchmark controller module and integrated them into the HOBBIT platform. The modules select the questions, send them to the system, gather the gold standard answer and sends the answer to the evaluation module.
InfAI set up a private DBpedia SPARQL endpoint for task 3 and connected the GERBIL QA evaluation module to the HOBBIT platform.
We collected the main choke points of question answering, that is, the difficulties that arise for a system in answering natural language questions based on structured datasets. Further, we set the KPI's.
We explored our possibilities to develop benchmarking scenarios that extend already existing benchmarks.

Table 1: Major work points carried out by IAIS and other participating partners

In the following, we give a more detailed description of the three tasks we compiled.

3.1 Task 1: Multilingual Question Answering

The Multilingual Question Answering task consists of 50 natural language questions with each query available in eight different languages. Available languages include English, German, French, Spanish, Italian, Dutch, Romanian, and Farsi. Additionally, besides the complete question there are keywords available for the systems.

Example:

en: Are penguins endangered?	penguins, endangered
de: Sind Pinguine vom Aussterben bedroht?	Pinguine, Aussterben, Bedrohung
it: I pinguini sono una specie in pericolo?	pinguini, specie, pericolo

Together with question and keywords, the corresponding DBpedia SPARQL query and the correct answer are available to the evaluation component. The DBpedia query for the sample question above would be

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/
ASK WHERE {
          ?uri dbo:family dbr:Penguin .
          ?uri dbo:conservationStatus “EN“ .
}
```

This query executed on the DBpedia public SPARQL endpoint[6] returns the answer *true* .

This task is a straightforward question answering challenge. Each QA system is able to choose a language in which it performs the natural language analysis and the SPARQL query conversion best. Possible analysis problems in one language can be compensated by making additional sense from a different language.

Tackled choke points of this task include all of the mentioned choke points in chapter 2.

3.2 Task 2: Hybrid Question Answering

The Hybrid Question Answering task is different from the straight-forward task 1 in such a way that the QA system not only needs to retrieve one piece of information from the database but also a second piece of information that is only available in the textual abstract and not in the triplified data. Only the combination of both the structured and the unstructured information leads to the correct answer.

Example:

[6] http://dbpedia.org/sparql/ retrieved on 02/05/2017

Give me the names of all of the people who signed the American Declaration of Independence.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?uri WHERE {
          ?uri rdf:type dbo:person .
          ?uri text:\"signed\" .
            {
                     ?uri text:\"American Declaration of Independence\" .
            } UNION {
                     ?uri text:\"Declaration of Independence\" .
            }
}
```

The fact that someone is a person is represented in the triplified data and can be retrieved by “?uri rdf:type dbo:person .“ However, the fact that a person signed the American Declaration of Independence is not represented in the data but mentioned in the textual abstract of the resource. Retrieving the words “*signed*“ and “*Declaration of Independence*“ or “*American Declaration of Independence*“ in the abstracts of all resources that also have the class “*person*“ leads to the correct answer:

```
<http://dbpedia.org/resource/John_Hancock>
<http://dbpedia.org/resource/John_Adams>
<http://dbpedia.org/resource/Benjamin_Franklin>
...
<http://dbpedia.org/resource/Thomas_Jefferson>
<http://dbpedia.org/resource/Lyman_Hall>
<http://dbpedia.org/resource/George_Walton>
```

In this task there are no keywords provided along with the questions. Tackled choke points include all of the mentioned choke points and the challenging fact that part of the information must be extracted from natural language text.

3.3 Task 3: Large-Scale Question Answering

The Large-scale Question Answering task adds another dimension to the evaluation. Systems are not only scored according to their correct answers but also with respect for needed time. The scope is to find out how many questions a system can answer in one minute. The task comprises a massive amount of more than 2 million questions and their respective SPARQL queries. During an experiment in this task, the amount of issued questions is permanently increasing. The answering time for the system between issued question sets is always one minute. The experiment starts with a question set of one question. After one minute, the platform issues the second question set containing two questions to the system and waits another one minute before issuing the next question set containing three questions, and so forth. After a while, QA systems are exposed to a massive question set and will eventually fail to answer all questions in the set before receiving the next one.

While the questions for tasks 1 and 2 were handcrafted, questions for this task had to be generated automatically. Therefore, we took 150 suitable questions and their SPARQL queries from previous QALD challenges. We automatically analyzed the so-called instance data in the question. Instance data consist of entities that have a DBpedia data entry. In most cases, these entities are named entities.

For example, for the question

How many scientists graduated from University of Hamburg?

```
PREFIX res: <http://dbpedia.org/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT (COUNT(DISTINCT ?uri) AS ?count) WHERE {
          ?uri rdf:type dbo:Scientist .
          ?uri dbo:almaMater res:University_of_Hamburg .
}
```

we identified the entity "*University of Hamburg*" in both the natural language question and the SPARQL query. Then we replaced the entity by a placeholder representing the DBpedia entity type, in this case *$UNIVERSITY*. Next, we replaced the placeholders in all 150 template questions by all available DBpedia entities of the same type, executed the SPARQL query on the DBpedia SPARQL endpoint and saved those question-query pairs that returned an answer.

This way, we derived similar questions to the templates. From 150 templates we generated more than 2 million questions. For the example above, other generated questions include

How many scientists graduated from University of Basel?

How many scientists graduated from University of Florence?

How many scientists graduated from University of Lima?

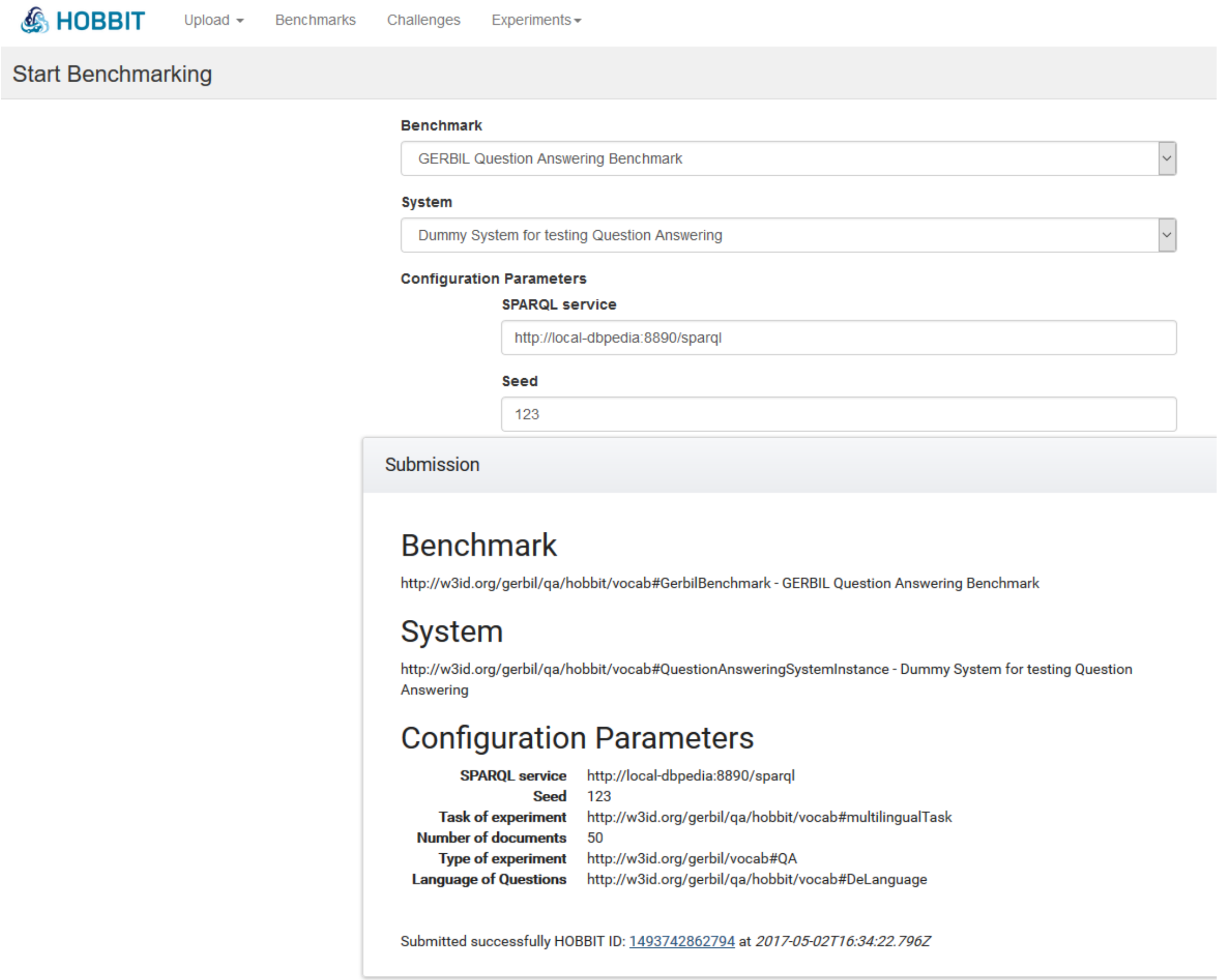
All answers of all tasks are generated by the task generator at runtime. Therefore, we set up a private local DBpedia SPARQL endpoint, so firstly we protect the public endpoint from massive querying.

Secondly, we match the systems‘ answers against the correct gold standard answers retrieved at the same time from the same underlying database which might change from time to time. When, for example, a new American president is in office and the database has been updated, this might lead to a wrong answer evaluation if we saved all correct answers as a gold standard at compilation time.

4. Example Experiment

In this chapter, we walk through an example system benchmarking with the first version of the HOBBIT Question Answering Benchmark.

The Question Answering test system we used is a simple system that always returns the answer *false* for every issued question. The QA system undergoes the multilingual QA task in this walkthrough example (as depicted in figure 1 on page 8). The following figure shows the summary of the settings before the experiment is executed.



Start Benchmarking

Benchmark
GERBIL Question Answering Benchmark

System
Dummy System for testing Question Answering

Configuration Parameters

SPARQL service
http://local-dbpedia:8890/sparql

Seed
123

Submission

Benchmark

http://w3id.org/gerbil/qa/hobbit/vocab#GerbilBenchmark - GERBIL Question Answering Benchmark

System

http://w3id.org/gerbil/qa/hobbit/vocab#QuestionAnsweringSystemInstance - Dummy System for testing Question Answering

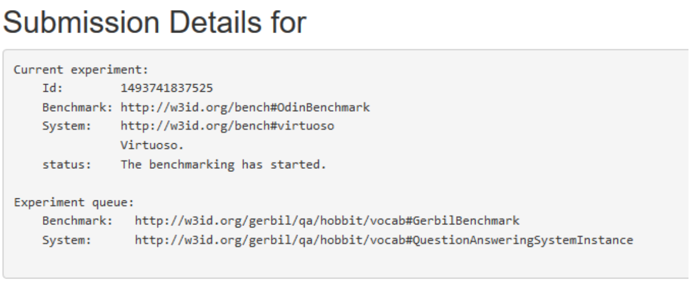
Configuration Parameters

SPARQL service	http://local-dbpedia:8890/sparql
Seed	123
Task of experiment	http://w3id.org/gerbil/qa/hobbit/vocab#multilingualTask
Number of documents	50
Type of experiment	http://w3id.org/gerbil/vocab#QA
Language of Questions	http://w3id.org/gerbil/qa/hobbit/vocab#DeLanguage

Submitted successfully HOBBIT ID: 1493742862794 at *2017-05-02T16:34:22.796Z*

Figure 2: Summary of configuration parameters before experiment execution

The system was set for the multilingual QA task with an amount of 50 questions in German. Once the experiment has been started by the user it will line up in the HOBBIT experiment queue and eventually be executed when previous experiments are completed.



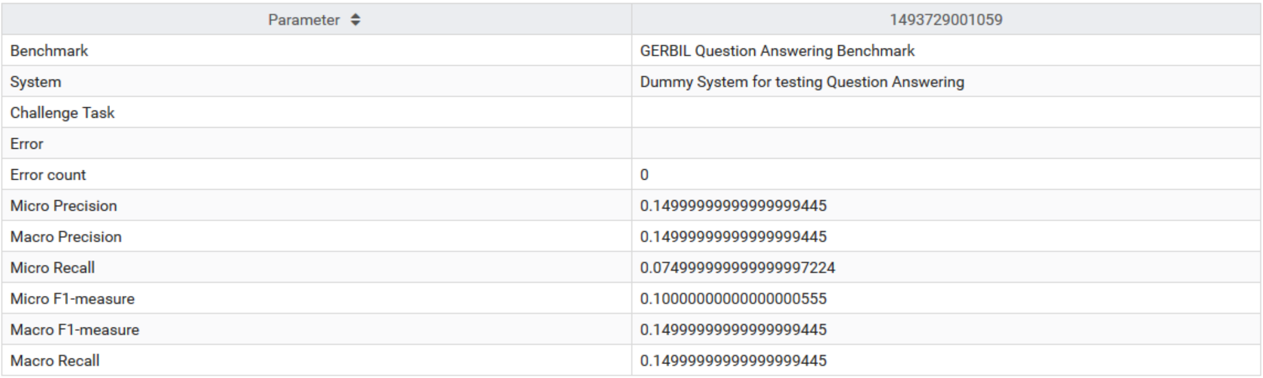
Submission Details for

```
Current experiment:
    Id:        1493741837525
    Benchmark: http://w3id.org/bench#OdinBenchmark
    System:    http://w3id.org/bench#virtuoso
               Virtuoso.
    status:    The benchmarking has started.

Experiment queue:
    Benchmark:   http://w3id.org/gerbil/qa/hobbit/vocab#GerbilBenchmark
    System:      http://w3id.org/gerbil/qa/hobbit/vocab#QuestionAnsweringSystemInstance
```

Figure 3: Running Multilingual QA experiment

Once the experiment has finished the results are displayed.



Parameter ◆	1493729001059
Benchmark	GERBIL Question Answering Benchmark
System	Dummy System for testing Question Answering
Challenge Task	
Error	
Error count	0
Micro Precision	0.14999999999999999445
Macro Precision	0.14999999999999999445
Micro Recall	0.074999999999999997224
Micro F1-measure	0.10000000000000000555
Macro F1-measure	0.14999999999999999445
Macro Recall	0.14999999999999999445

Figure 4: Results for the Question Answering test system in a Multilingual QA experiment

In the example experiment, the system scored a low F1-measure when always returning the answer *false* for every question. Every finished experiment is repeatable and gets a citable URL.

5. Next Steps

We are already making an impact on the research community: the QALD-7 challenge is ongoing and competing systems are being benchmarked with the Question Answering Benchmark described here. The winners will be announced at the end of the ESWC conference in Slovenia. We are also preparing for the QALD-8 challenge which will take place in the context of the ISWC 2017 in Vienna and allow wider participation by removing the strict conference requirement for competing system to turn in a paper.

Having concluded the first phase of Task 6.1 and provided a fully functioning benchmark environment for QA systems, we can now look at our current standing point and plan for the next version of the HOBBIT QA benchmark. We already have several ideas for improvement, including the possibility to choose different datasets to the introduction of new KPIs and questions. We hope the QA community will greatly accept this first QA Benchmark version and provides us with suggestions for the upcoming second version.