

WDAqua-core0: A Question Answering Component for the Research Community

Dennis Diefenbach¹, Kamal Singh¹, Pierre Maret¹

Laboratoire Hubert Curien, Saint Etienne, France,
dennis.diefenbach,kamal.singh,pierre.maret@univ-st-etienne.fr

Abstract. We describe and present a new Question Answering (QA) component that can be easily used by the QA research community.

It can be used to answer questions over DBpedia and Wikidata. The language support over DBpedia is restricted to English, while it can be used to answer questions in 4 different languages over Wikidata namely English, French, German and Italian. Moreover it supports both full natural language queries as well as keyword queries.

We describe the interfaces to access and reuse it and the services it can be combined with. Moreover we show the evaluation results we achieved on the QALD-7 benchmark.

Keywords: Question Answering, Qanary, QALD

1 Introduction

Question answering (QA) is a very old research field in computer science. In the last two decades, thanks to the development of the Semantic Web, a lot of new structured data has become available on the web in the form of knowledge bases (KBs). Nowadays, there are KBs about media, publications, geography, life-science and more¹. The idea behind a QA system over KBs is to find the information, in a KB, requested by the user using natural language. This is generally addressed by translating a natural question to a SPARQL query that can be used to retrieve the desired information. We present here a QA component to answer questions over DBpedia and Wikidata that can answer both full and keyword natural language questions. It is integrated in the Qanary Ecosystem[4] so that first, it can be easily reused by the research community and second, it takes advantage of the services available in Qanary.

2 Related work

In the context of QA, a large number of systems have been developed in the last years. For example, more than twenty QA systems were evaluated against the QALD benchmark². While many systems are querying DBpedia, we are only aware of one system querying wikidata, namely *Platypus*³. Moreover most of the works address full natural language

¹ <http://lod-cloud.net>

² <http://qald.sebastianwalter.org>

³ <https://askplatyp.us/?>

questions while only few address keyword questions. One exception is *SINA*[7]. The fact that QA systems often reuse existing techniques lead to the idea of developing QA systems in a modular way. Four frameworks tried to achieve this goal: QALL-ME [5], openQA [6], the Open Knowledge Base and Question-Answering (OKBQA) challenge⁴ and Qanary [8,1,4]. We integrated our QA component into the Qanary Ecosystem since it makes it easily reusable by the research community and offers a series of off-the-shelf services related to QA systems.

3 Description of WDAqua-core0

Our SPARQL creation algorithm uses a combinatorial approach based on the semantics encoded in the underlying KB. The full details will be disclosed in an upcoming publication as this is only a challenge submission. In the following we briefly describe the capabilities of WDAqua-core0. WDAqua-core0 can answer questions on both DBpedia and Wikidata. Note that the Wikidata dump⁵ contains binary and non-binary relationships. An example of a non-binary relationships expressing that *the capital of Germany was Berlin from 1990* is expressed in two versions:

```
@prefix wd: <http://www.wikidata.org/entity/> .
@prefix p: <http://www.wikidata.org/prop/> .
@prefix ps: <http://www.wikidata.org/prop/statement/> .
@prefix wdt: <http://www.wikidata.org/prop/direct/> .
@prefix pq: <http://www.wikidata.org/prop/qualifier/> .
@prefix wds: <http://www.wikidata.org/entity/statement/> .

wd:Q183 rdfs:label "Germany"@en ;
wd:Q64 rdfs:label "Berlin"@en ;
wdt:P36 rdfs:label "capital"@en ;

#VERSION 1: reefied
wd:Q183 p:P36 wds:q183-7068B86F .
wds:q183-7068B86F a wikibase:Statement ,
    ps:P36 wd:Q64 ;
pq:P580 "1990-10-03T00:00:00Z"^^xsd:dateTime .

#VERSION 2: non-reefied
wd:Q183 wdt:P36 wd:Q64 ;
```

The first version uses properties with the namespaces `p` and `ps` while the second loses the temporal information and uses the namespace `wdt`. WDAqua-core0 is querying only the triples containing properties with namespace `wdt`. WDAqua-core0 can answer both keyword questions and questions in natural language. The complexity of the generated queries is limited to queries containing at most two triple patterns. The generated queries can be of type SELECT or ASK. The modifiers are limited to the COUNT operator. Thus, the questions with superlatives and comparatives can in general not be answered. Finally it supports English on DBpedia and 4 different language over Wikidata, namely English, French, German and Italian. The evaluation is shown in section 5.

⁴ <http://www.okbqa.org/>

⁵ <https://dumps.wikimedia.org/wikidatawiki/entities/>, https://www.mediawiki.org/wiki/Wikibase/Indexing/RDF_Dump_Format

4 Integration in Qanary

Qanary is a framework to integrate QA components with the goal to make existing research in the QA field reusable. The QA component presented here is integrated into Qanary. A running version is registered into the Qanary service running under:

`http://www.wdaqua.eu/qanary`

In particular the component can be executed through RestFul interfaces. To run the service over a new question the RestFul interface under:

`http://www.wdaqua.eu/qanary/startquestionansweringwithtextquestion`

can be used. Besides the generated answer, the top-30 generated queries can also be retrieved.

The integration into Qanary allows the combination of WDAqua-core0 with the other components and services that are already integrated into Qanary. In particular it can be combined with a speech recognition component and a language detection component. Additionally it can be used together with a number of services that are constructed around Qanary. These include a reusable front-end called Trill [2]. A demo of Trill that in the back-end uses WDAqua-core0 can be found under www.wdaqua.eu/qa. Figure 1 shows a screen-shot of Trill. Moreover WDAqua-core0 can be used together with some interfaces for user-feedback that are integrated into Trill [3]. One such feedback-interface can be seen in Figure 2. As a consequence WDAqua-core0 can be used by end-users and can for example be used to drive forward research in the domain of human-computer interaction. Finally Qanary has an interface that allows QA pipelines to be evaluated using Gerbil for QA ⁶. This means that WDAqua-core0 can be evaluated by the research community at all time especially when new benchmarks arise.

5 Evaluation over QALD-7

In this section we show the results of WDAqua-core0 over QALD-7 task 1 and task 4. We evaluate both over the keyword and the full-natural language questions.

Moreover, we extended task 4 and introduced a new type of multilingual QA benchmark. QALD-7 task 1 requires to answer questions in multiple languages using data contained in the English DBpedia. In particular taking the Italian DBpedia to answer the Italian questions of QALD-7 task 1 does not work in general. The fact that the Italian questions must be answered using the English dataset, forces the systems to use translations. Instead we translate the questions of the QALD-7 task 4 into French, German and Italian and try to answer them using Wikidata. This is fundamentally different since in Wikidata the knowledge is the same and only the labels change. In particular a translation is not required, one can answer the Italian questions using an Italian dataset. The global (or macro) precision, recall and F-measure achieved over QALD-7 can be found in table 1. Note that WDAqua-core0 does not use a machine learning algorithm so there is not a problem of over-fitting the dataset.

⁶ <http://gerbil-qa.aksw.org>

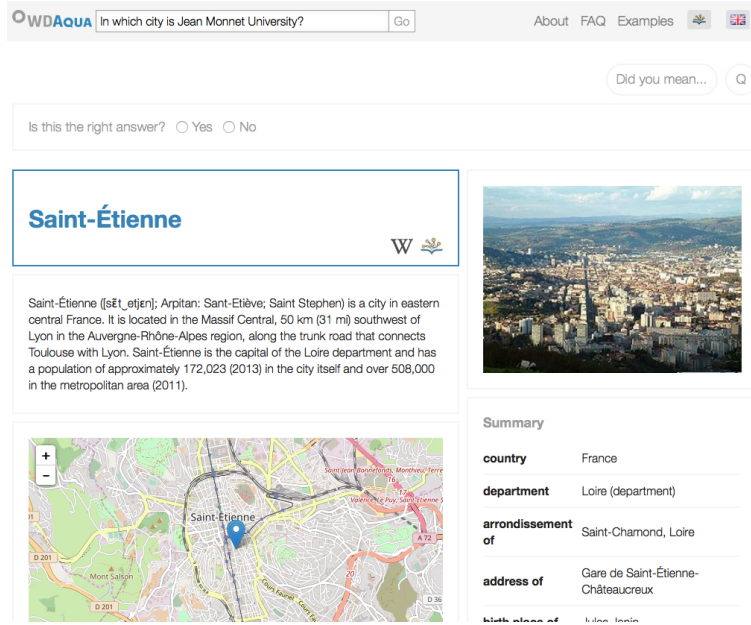


Fig. 1. Screenshot of Trill using in the back-end WDAqua-core0 for the question “In which city is Jean Monnet University?”

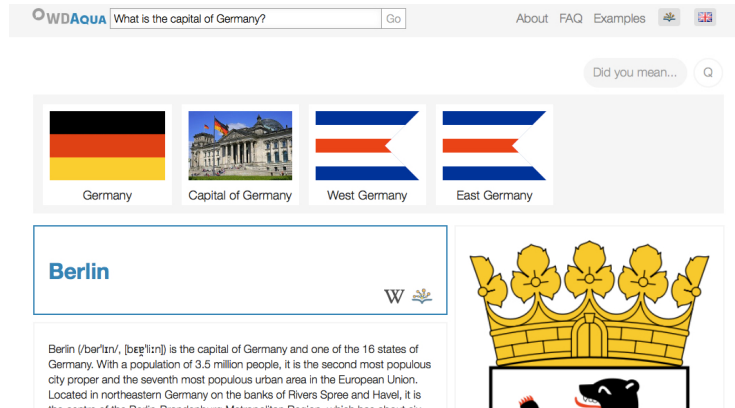


Fig. 2. Snapshot of the disambiguation interface for the question: “What is the capital of Germany?”. By clicking on “Did you mean” several entities, the question might referred to, are shown. These include the actual “Federal Republic of Germany” but also the “Capital of Germany” (as an entity), “West Germany”, “East Germany”, “Allied-Occupied Germany” and others. By clicking on the entity, the question is interpreted differently and a new answer is presented, e.g., if the user clicks on “West Germany”, the answer “Bonn” is computed.

Task	Dataset	Language	Type	Precision	Recall	F-measure
1	DBpedia	en	full	0.488	0.535	0.511
1	DBpedia	en	keywords	0.374	0.407	0.390
4	Wikidata	en	full	0.320	0.323	0.322
4	Wikidata	en	keywords	0.280	0.280	0.280
4	Wikidata	fr	full	0.123	0.131	0.127
4	Wikidata	fr	keywords	0.326	0.365	0.344
4	Wikidata	de	full	0.220	0.264	0.240
4	Wikidata	de	keywords	0.301	0.349	0.325
4	Wikidata	it	full	0.164	0.184	0.173
4	Wikidata	it	keywords	0.188	0.207	0.197

Table 1. The table shows the results of WDAqua-core0 over the QALD-7 training set.

6 Conclusion

We have presented a QA component integrated into the Qanary Ecosystem that can be easily reused by the QA community. In particular it can be used to push forward research in directions like the integration of speech recognition systems with QA systems and the interaction with users.

We have evaluated the component against QALD-7 in multiple aspects. We have shown the performance over both DBpedia and Wikidata with respect to keyword and full-natural language queries. Moreover, we have introduced a new type of multilingual QA benchmark that does not require translation but where the questions and the KB are in the same language. We have shown our results over this new type of multilingual QA benchmark.

Acknowledgments Parts of this work received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 642795, project: Answering Questions using Web Data (WDAqua).

References

- Both, A., Diefenbach, D., Singh, K., Shekarpour, S., Cherix, D., Lange, C.: Qanary – a methodology for vocabulary-driven open question answering systems. In: ESWC 2016 (2016)
- Diefenbach, D., Amjad, S., Both, A., Singh, K., Maret, P.: Trill: A reusable front-end for qa systems. In: ESWC P&D (2017)
- Diefenbach, D., Hormozi, N., Amjad, S., Both, A.: Introducing feedback in qanary: How users can interact with qa systems. In: ESWC P&D (2017)
- Diefenbach, D., Singh, K., Both, A., Cherix, D., Lange, C., Auer, S.: The Qanary Ecosystem: getting new insights by composing Question Answering pipelines. In: ICWE (2017)
- Ferrández, Ó., Spurk, C., Kouylekov, M., Dornescu, I., al.: The QALL-ME framework: A specifiable-domain multilingual Question Answering architecture. J. Web Sem.
- Marx, E., Usbeck, R., Ngonga Ngomo, A., Höffner, K., Lehmann, J., Auer, S.: Towards an open question answering architecture. In: SEMANTiCS (2014)
- Shekarpour, S., Marx, E., Ngomo, A.C.N., Auer, S.: Sina: Semantic interpretation of user queries for question answering on interlinked data. Web Semantics: Science, Services and Agents on the World Wide Web 30 (2015)
- Singh, K., Both, A., Diefenbach, D., Shekarpour, S.: Towards a message-driven vocabulary for promoting the interoperability of question answering systems. In: ICSC 2016 (2016)