QAnswer: A Question Answering prototype bridging the gap between a considerable part of the LOD cloud and end-users

Dennis Diefenbach
Université de Lyon, CNRS UMR 5516
Laboratoire Hubert Curien
Saint-Etienne, France
dennis.diefenbach@univ-st-etienne.fr

Pedro Henrique Migliatti
Universidade Federal de São Carlos, MaLL
São Carlos, Brasil
619744@comp.ufscar.br

Omar Qawasmeh
Université de Lyon, CNRS UMR 5516
Laboratoire Hubert Curien
Saint-Etienne, France
omar.alqawasmeh@univ-st-etienne.fr

Vincent Lully
Sorbonne Université
Paris, France
vincent.lully@sorbonne-universite.fr

Kamal Singh
Université de Lyon, CNRS UMR 5516
Laboratoire Hubert Curien
Saint-Etienne, France
kamal.singh@univ-st-etienne.fr

Pierre Maret
Université de Lyon, CNRS UMR 5516
Laboratoire Hubert Curien
Saint-Etienne, France
pierre.maret@univ-st-etienne.fr

ABSTRACT

We present QAnswer, a Question Answering system which queries at the same time 3 core datasets of the Semantic Web, that are relevant for end-users. These datasets are Wikidata with Lexemes, LinkedGeodata and Musicbrainz. Additionally, it is possible to query these datasets in English, German, French, Italian, Spanish, Pourtuguese, Arabic and Chinese. Moreover, QAnswer includes a fallback option to the search engine Qwant when the answer to a question cannot be found in the datasets mentioned above. These features make QAnswer as the first prototype of a Question Answering System over a considerable part of the LOD cloud.

CCS CONCEPTS

• **Information systems → Web search engines**; **Information retrieval**; *Resource Description Framework (RDF).*

KEYWORDS

Question Answering, Semantic Web, Multilingual, Multi-Knowledge-Base

ACM Reference Format:
Dennis Diefenbach, Pedro Henrique Migliatti, Omar Qawasmeh, Vincent Lully, Kamal Singh, and Pierre Maret. 2019. QAnswer: A Question Answering prototype bridging the gap between a considerable part of the LOD cloud and end-users. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3308558.3314124

1 INTRODUCTION

In the last decade, some new datasets adhering to Semantic Web standards were published on the Web. This growth can be seen by looking at the Linked Open Data Cloud[1] (LOD cloud), which collects datasets that have been published using Semantic Web Technologies. Though in 2007, the LOD cloud contained 12 datasets, it now contains 1,231[2] datasets. The LOD cloud does not only contain a lot of datasets, but at the same time some of these datasets are very large. A dump of the LOD cloud called LOD-a-lot[3] contains around 5 TB of structured data in uncompressed N-triples format. While Semantic Web standards are designed to make data to be machine comprehensible, they do not allow, at the same time, easy accessibility of the data to non experts and in particular to end-users. Question Answering (QA) is seen as the technology that can allow to bridge this gap between Semantic Web data and end-users. While the industry presents QA solutions on their proprietary Knowledge Base (such as Google and Baidu), no system reached a point which allows querying a substantial part of the LOD cloud.
Table 1 shows a list of QA systems which we are aware of, that are available online and query part of the LOD cloud. We indicate the datasets which they are able to query and the languages which are supported. For a general list and an overview of QA systems over KB, we refer to [4]. QAnswer represents a breakthrough since it allows to query many more datasets, at the same time, in real-time and in many more languages. The main algorithm behind QAnswer is described in [3]. In the following text, we show how the algorithm in [3] was implemented to create a first prototype of a QA System over the Semantic Web and which are the improvements over previous works.

2 DESCRIPTION

The algorithm behind QAnswer is described in [3]. It was shown that it has the following distinctive features:

- **Multilingual**, it supports multiple languages. In the previous work, it was shown that the algorithm works for English, German, French, Italian, Spanish, Portuguese. Moreover, the algorithm can easily be adapted to new languages.
- **Robust**, users ask questions using keywords, natural language questions and even malformed questions, i.e., syntactically wrong questions. The algorithm is robust enough to deal with all these scenarios, but not to spelling mistakes.

[1] https://lod-cloud.net

[2] stand December 2018

[3] http://lod-a-lot.lod.labs.vu.nl

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3314124

QA system	Lang	KBs	Url
Qakis[2]	en	DBpedia	http://qakis.org/
Quint[1]	en	Freebase	https://gate.d5.mpi-inf.mpg.de/quint/quint
AskNow[5]	en	DBpedia	http://asknowdemo.sda.tech
Frankenstein[8]	en	DBpedia	http://frankenstein.qanary-qa.com
gAnswer [10]	en	DBpedia	http://ganswer.gstore-pku.com/
Ask Platypus[9]	en, fr, es	Wikidata	https://askplatyp.us
QAnswer	en, fr, de, es, pt, zh, ar	Wikidata, LinkedGeoData, MusicBrainz, DBpedia, DBLP	http://qanswer.eu/qa

Table 1: List of Question Answering systems that query part of the LOD cloud and that are available online.

- **Real-time**, the algorithm can answer questions in real-time, i.e., on existing benchmarks like QALD and SimpleQuestions an average run-time of 2 seconds per query is realistic.
- **Low hardware footprint**, the algorithm uses specific indexes that guarantee low disk and memory footprint. Our demo will show that QAnswer can query 700Gb of n-triples and can be run on a (standard) laptop having 4 cores and 16 Gb of RAM.
- **Portable**, making question answering over a new dataset can be difficult. Some approaches need a lot of training data, other are not designed to be portable at all. Our system is designed such that any new dataset can be used as a base for a new QA system.
- **Multi-Knowledge-base**, the algorithm allows to query multiple Knowledge-bases at the same time.
- **Precision and Recall**, the algorithm was tested on multiple benchmarks and can compete with most of the existing approaches [3].

We are using the algorithm described in [3] to query what we believe are the three more significant datasets in the LOD cloud for end-users: Wikidata, LinkedGeoData and MusicBrainz. While, DBpedia and Freebase can also be queried, we consider these datasets as outdated since both are not maintained anymore.

We now describe the improvements over previous works brought by QAnswer.

- Non-european languages: While it was shown that the algorithm in [3] can be used to answer questions in multiple languages, it was only tested over European languages. Recently, we could also successfully apply it to two non-european languages, namely Chinese and Arabic. Example requests that can now be addressed are:

Figure 1: Screenshot of QAnswer for فيلم ذيب.

  - زوجه باراك اوباما, i.e., asking for the wife of Obama in Arabic.
  - "巴黎的博物馆", i.e., searching in Chinese for the museums in Paris.

  A screenshot of an example can be seen in Figure 1.

- LinkedGeoData: One of the largest open databases for geographical information is OpenStreetMap (OSM). It collects geographical information thanks to more than 2 million registered users. It contains information about streets, buildings, points of interest (like shops, restaurants, museums, fountains), cities, regions and many more. The data is natively stored in a PostgreSQL database with PostGIS extension. LinkedGeoData is the RDF extract of OSM data. The last public extraction was performed in 2015. We extracted a new export, covering entire Europe, and made the dataset queriable by QAnswer. Note that there is no online demo querying geo-spatial data and we are aware of only one work tackling this problem, which is presented in [7]. Some example requests that can be answered using this dataset are:
  - "Steinwenden", i.e. searching for a city
  - "Worms Renzstraße", i.e. searching for a street in a city
  - "give me fountains in Saarbrücken", i.e. searching for a point of interest in a city

  A screenshot of an example can be seen in Figure 2.

- Lexemes: In the previous work, the algorithm presented in [3] was used to query Wikidata. When Wikidata was new, it contained only Q-items, i.e., resources describing a thing or an idea, but not the word describing it. Since 2018, Wikidata was extended to also store new information such as words, phrases and sentences, in many languages. This information is stored in new types of entities, called Lexemes (L), Forms

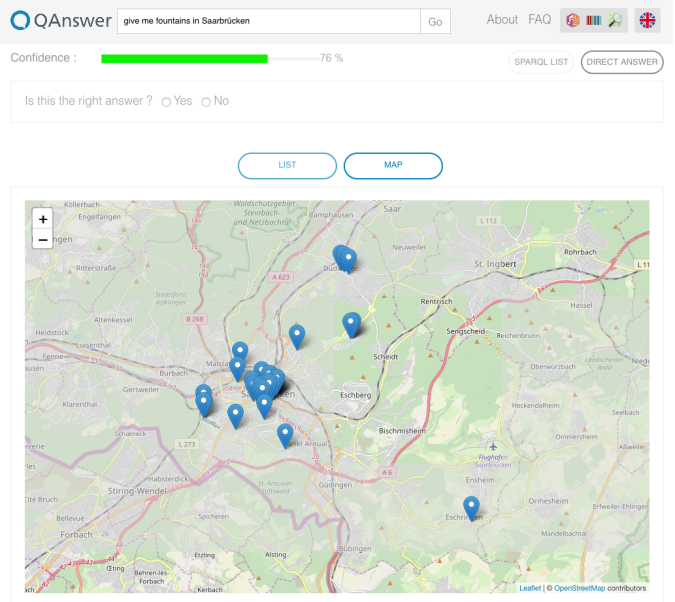

Figure 2: Screen shot of QAnswer for "give me fountains in Saarbrücken"

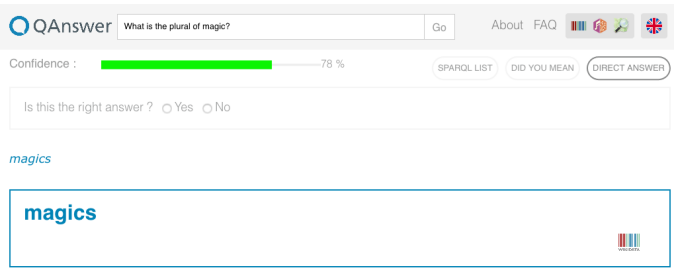

Figure 3: Screen shot of QAnswer for "What is the plural of magic?"

(F) and Senses (S). The Lexeme extension can be seen as a structured representation of the Wiktionary[4]. There exists an earlier attempt to semantify wiktionary by DBpedia project[5]. This project is unfortunately not maintained anymore and the extraction framework does not work anymore due to changes in the structure of Wiktionary. While the Lexemes in Wikidata still represent a small portion of the information contained in Wiktionary, we believe that the active Wikidata community will be able to fill this gap. Example questions that can be answered using this data are:

- "what is the pronunciation of magic?", i.e. searching for the pronunciation of a word
- "What is the plural of magic?", i.e., searching for forms of a word

A screenshot of an example can be seen in Figure 3.

[4]https://www.wiktionary.org
[5]https://wiki.dbpedia.org/wiktionary-rdf-extraction

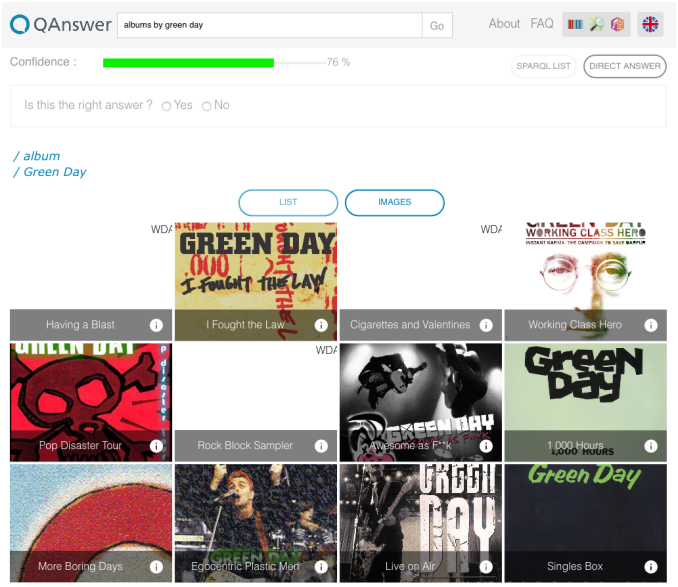

Figure 4: Screen shot of QAnswer for "albums by Green Day"

- Musicbrainz: Musicbrainz is one of the largest music databases that is open and available online. The data is natively stored in a relational database (i.e. a PostgreSQL database). In previous work, it was already possible to query MusicBrainz, but it relied on the dump provided by the following RML mappings: https://github.com/LinkedBrainz/MusicBrainz-R2RML. We improved the mappings in different ways. The most radical change was related to how some information is organized. For example in the original LinkedBrainz dump every album appears multiple times (every time for each publication year and country where it was released). The same problem appears for songs. In particular the dump didn't allow to recognize that the different publications of the albums and songs were in fact referring to the same album and song. This means that when asking for albums or songs, many duplicates appeared, a behaviour that is not expected by users and also not explicable for them. We therefore restructured the export to avoid this. Other changes represent a better coverage to external links like to Wikipedia, Wikidata and Social Media. Moreover, the dump is enriched with the links to the covers of albums from http://coverartarchive.org. Finally, we enriched the artists, albums and songs with links to youtube so that users can effectively hear the pieces they are querying. Example requests where Musicbrainz is used to answer some queries are:
  - "albums by green day", i.e., searching for albums of a band
  - "record label from turin", i.e. searching for record labels in a city,
  - "songs blink-182", i.e., searching for songs of a band

  A screenshot of an example can be seen in Figure 4.

- Qwant: While the Semantic Web grew very fast in the last years, it contains only a fraction of the information contained

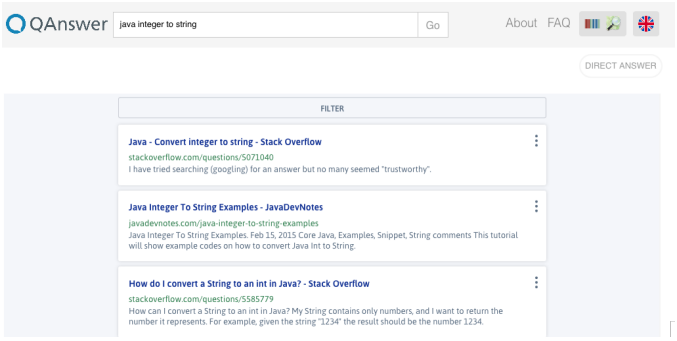

Figure 5: Screen shot of QAnswer for "java integer to string"

in the web. Whereas, most of the information is stored in a non-structured format, i.e., in the form of HTML pages. This information is typically accessed with traditional information retrieval techniques [6].

We therefore integrated, a fall back option, which is a traditional search engine, namely Qwant[6]. This means that when we cannot find the information requested by the users in one of the underlying datasets, we search for an answer using Qwant. Note that this represents a new problem in the area of QA over Knowledge Bases. Current benchmarks only contain questions whose answer can be found in the underlying knowledge base. There are a few exceptions in the QALD benchmark where less than 1% of the questions are not answerable. However, in a real scenario many of the questions asked by an end-user are not contained in the underlying Knowledge-Base. This means that recognizing when a question should not be answered (because it is not answerable considering the underlying knowledge-base) is an important and not well studied problem. Technically speaking this comes down to the fact that in current benchmarks only the macro F-measure is evaluated and the micro F-measure is generally ignored.

We chose Qwant as a fall-back option since it does not rely on a Knowledge Graph to provide direct answers. Moreover, the main distinctive feature as compared to other existing search engines is that Qwant does not track users and does not personalize search results and therefore users are not trapped in a filter bubble. Example requests where Qwant is used as a fall-back are:

- "How many legs has a horse?"
- "With how many degrees should I cook a pizza?"
- "java integer to string"

A screenshot of an example can be seen in Figure 5.

3 DEMO

A demo of the current version can be found under:

http://qanswer.eu/qa

Moreover, by clicking on the above examples you will be redirected to the online demo.

[6]https://www.qwant.com

4 CONCLUSION

We presented QAnswer a QA system which queries 3 key Semantic Web datasets at the same time, namely, Wikidata with lexemes, LinkedGeoData and Musicbrainz. These datasets contain a huge amount of information like books, films, persons, music, streets, points of interest and many more. The data can be queried using natural language in 8 different languages, namely English, German, French, Italian, Spanish, Pourtuguese, Arabic and Chinese. In particular two non-European languages, Arabic and Chinese, are included. All together this represents a first prototype of a QA system querying a considerable part of the Semantic Web.

This represents also a step towards new challenges in this area like: correctly selecting which Knowledge Base to query, new scalability scenarios, making use of links between the datasets to deal with redundant information, studying the impact of the dataset quality on the QA performance, and studying the problem of not answering a question.

In future, we would like to make the algorithm and infrastructure publicly available via web APIs so that new RDF datasets can be indexed and accessed using natural language. We believe that this work will further boost the publication of RDF data and therefore the expansion of the Semantic Web.

REFERENCES

[1] Abdalghani Abujabal, Rishiraj Saha Roy, Mohamed Yahya, and Gerhard Weikum. 2017. QUINT: Interpretable Question Answering over Knowledge Bases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 61–66.

[2] Elena Cabrio, Julien Cojan, Alessio Palmero Aprosio, Bernardo Magnini, Alberto Lavelli, and Fabien Gandon. 2012. QAKiS: an open domain QA system based on relational patterns. In *Proceedings of the 2012th International Conference on Posters & Demonstrations Track-Volume 914*.

[3] Dennis Diefenbach, Andreas Both, Kamal Deep Singh, and Pierre Maret. 2018. Towards a Question Answering System over the Semantic Web. *Semantic Web Journal* (2018).

[4] Dennis Diefenbach, Vanessa López, Kamal Deep Singh, and Pierre Maret. 2018. Core techniques of question answering systems over knowledge bases: a survey. *Knowl. Inf. Syst.* 55, 3 (2018), 529–569. https://doi.org/10.1007/s10115-017-1100-y

[5] Mohnish Dubey, Sourish Dasgupta, Ankit Sharma, Konrad Höffner, and Jens Lehmann. 2016. AskNow: A Framework for Natural Language Query Formalization in SPARQL. In *ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Proceedings*. 300–316. https://doi.org/10.1007/978-3-319-34129-3_19

[6] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*.

[7] Dharmen Punjani, Ellie Nagaishi, Andreas Both, Manolis Koubarakis, Ioannis Angelidis, Konstantina Bereta, Themis Beris, Dimitris Bilidas, Kelly A. Martin, Nikolaos Karalis, Christian Lange, D Pantazi, Costas Papaloukas, and George Stamoulis. 2018. Template-Based Question Answering over Linked Geospatial Data. In *GIR'18*.

[8] Kuldeep Singh, Arun Sethupat Radhakrishna, Andreas Both, Saeedeh Shekarpour, Ioanna Lytra, Ricardo Usbeck, Akhilesh Vyas, Akmal Khikmatullaev, Dharmen Punjani, Christoph Lange, et al. 2018. Why Reinvent the Wheel: Let's Build Question Answering Systems Together. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1247–1256.

[9] Thomas Pellissier Tanon, Marcos Dias de Assunção, Eddy Caron, and Fabian M. Suchanek. 2018. Demoing Platypus - A Multilingual Question Answering Platform for Wikidata. In *The Semantic Web: ESWC 2018 Satellite Events - ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers*. 111–116. https://doi.org/10.1007/978-3-319-98192-5_21

[10] Lei Zou, Ruizhe Huang, Haixun Wang, Jeffrey Xu Yu, Wenqiang He, and Dongyan Zhao. 2014. Natural language question answering over RDF: a graph data driven approach. In *SIGMOD 2014, Snowbird, UT, USA*. 313–324. https://doi.org/10.1145/2588555.2610525