On the scalability of the QA system WDAqua-core1

Dennis Diefenbach[1], Kamal Singh[1], Pierre Maret[1]

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

Abstract. Scalability is an important problem for Question Answering (QA) systems over Knowledge Bases (KBs). Current KBs easily contain hundreds of millions of triples and all these triples can potentially contain the information requested by the user.
In this publication, we describe how the QA system WDAqua-core1 deals with the scalability issue. Moreover, we compare the scalability of WDAqua-core1 with existing approaches.

Keywords: Question Answering, Knowledge Bases, Scalability, WDAqua-core1

1 Introduction

Question Answering (QA) over Knowledge Bases (KBs) is a field in computer science that tries to build a system able to search in a KB the information requested by a user using natural language. For example if a user asks: "What is the capital of Congo?" a QA system over Wikidata should be able to generate the following SPARQL query that retrieves the desired answer:

```
PREFIX wde: <http://www.wikidata.org/entity/>
PREFIX wdp: <http://www.wikidata.org/prop/direct/>
SELECT ?o where {
    wde:Q974 wdp:P36 ?o .
}
```

Scalability refers to the problem of answering a question in a time that is acceptable by the user. This problem mainly depends on the size of the data to query. Note that a KB like Wikidata contains more than 2 billion triples, which corresponds to roughly 250 Gb in an ntriples dump. On the other side, a user nowadays expects response times of around 1 second. Since a QA system is running on a server, in addition to the time required to compute the answer, it is also necessary to add the overhead of the network requests, and the retrieval and rendering of the information related to the answer. This means that very easily scalability becomes a problem for QA systems.
In this publication, we describe how the QA system WDAqua-core1 tackles this problem.

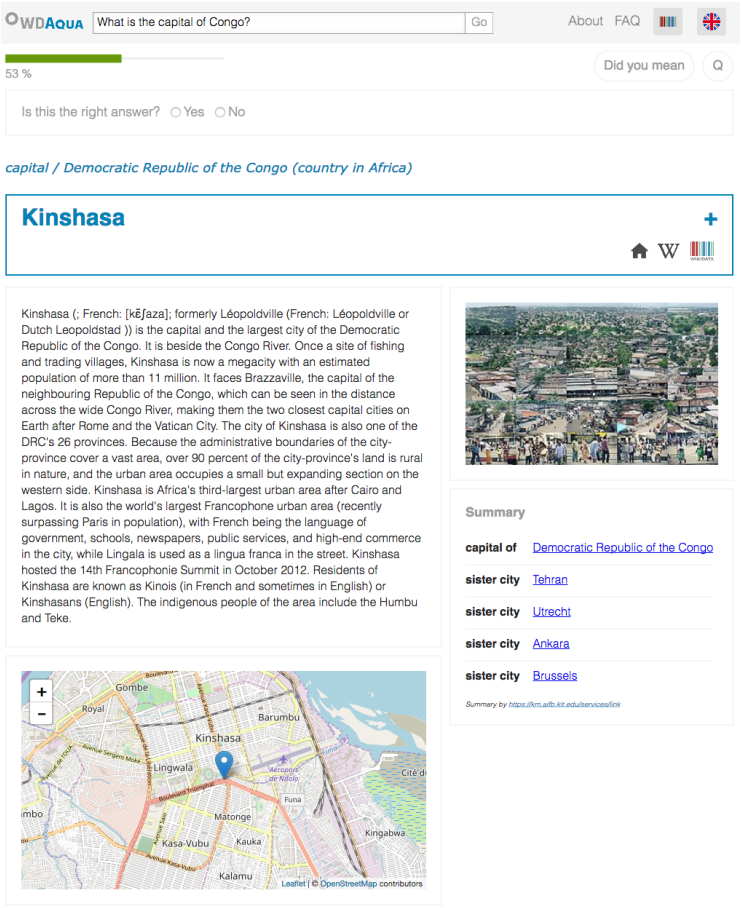

Fig. 1. Screenshot of Trill [3] using WDAqua-core1 as a back-end QA system for the question "What is the capital of Congo?". The answer is given in 0.974s. In this case Wikidata is queried which is roughly 250 Gb ntriples dump.

2 Approach

In this section, we are going to describe what is the main idea behind the algorithm used by WDAqua-core1. We would describe it using an example. Imagine the QA system receives the question: "What is the capital of Congo?". WDAqua-core1 takes into consideration every n-gram in the question and maps it to potential meanings. For example "what is" can correspond to "What Is... (album by Richie Kotzen)" and capital can refer to the expected property "capital" but also to "Capital (German business magazine)", "Capital Bra (German rapper)" and so on. For this example question, 76 different meanings are taken into consideration. All these meanings are used to generate SPARQL queries that are possible interpretations of the questions. For this example question, we gener-

ate 292 possible interpretations. Once all interpretations are generated, they are ranked and the first ranked query is executed and the answer is retrieved. The approach clearly needs to be implemented efficiently to achieve acceptable time performance. For further details, we refer to [4]. A running demo of the approach can be found at:

`www.wdaqua.eu/qa`

A screen-shot of Trill [3], a reusable front-end for QA systems, using in the back-end WDAqua-core1 can be found in Figure 1.

3 Scalability

In this section, we are going to describe which are the key elements that improve the scalability of the approach.

- To find the potential meanings of all n-grams in a question we rely on a Lucene Index of the labels of the targeted KB. It is characterized by high efficiency and low memory footprint.
- To construct all possible SPARQL queries out of the potential meanings we use an efficient algorithm. It is described in [4]. To achieve this we rely on HDT (Header Dictionary Triples) as an index. HDT is used because it allows fast breath search operations over RDF graphs. In fact RDF graphs are stored in HDT like adjecency lists. This is an ideal data structure to perform breath search operations. HDT is also characterized by low memory footprint.
- The current implementation uses as few HTTP requests as possible, to reduce the overhead they generate.

Above allows us to run a QA system over DBpedia and Wikidata with a memory footprint of only 26 Gb. Above ideas in particular allow the system to scale horizontally with the size of a given KB.

4 Experiments

Table 4 summarizes the existing QA solutions evaluated over the QALD benchmarks. It also indicates the average run time of all the queries per given task, if indicated in the corresponding publication. This clearly depends on the underlying hardware, but independently from that, it provides a hint on the scalability of the proposed approach. Note that, for most of the systems, no information is provided about the performance in terms of execution time. One of the main reasons is that these QA systems do not exist as one pipeline, but often authors write sub-results of sub-pipelines into files and arrive step-wise at the final answer. For the few systems that provided the performance in terms of execution time, this varies, depending on the question, from around 1 second for gAnswer [20] to more than 100 seconds for some questions in the case of SINA [15]. WDAqua-core1 obtains results similar to the best systems. It is an entire pipeline and available on line.

QA system	P	R	F	Time
QALD-3				
WDAqua-core1	**0.58**	**0.46**	**0.51**	**1.08s**
gAnswer [20]∗	0.40	0.40	0.40	0.971s
RTV [8]	0.32	0.34	0.33	-
Intui2 [5]	0.32	0.32	0.32	-
SINA [15]∗	0.32	0.32	0.32	≈10-20s
DEANNA [18]∗	0.21	0.21	0.21	≈1-50s
SWIP [13]	0.16	0.17	0.17	-
Zhu et al. [19]∗	0.38	0.42	0.38	-
QALD-4				
Xser [17]	0.72	0.71	0.72	-
WDAqua-core1	**0.56**	**0.30**	**0.39**	**0.46s**
gAnswer [20]	0.37	0.37	0.37	0.972 s
CASIA [10]	0.32	0.40	0.36	-
Intui3 [6]	0.23	0.25	0.24	-
ISOFT [12]	0.21	0.26	0.23	-
Hakimov [9]∗	0.52	0.13	0.21	-

QA system	P	R	F	Time
QALD-5				
Xser [17]	0.74	0.72	0.73	-
WDAqua-core1	**0.56**	**0.41**	**0.47**	**0.62s**
AskNow[7]	0.32	0.34	0.33	-
QAnswer[14]	0.34	0.26	0.29	-
SemGraphQA[2]	0.19	0.20	0.20	-
YodaQA[1]	0.18	0.17	0.18	-
QuerioDali[11]		0.48	?	?
QALD-6				
UTQA [16]	0.82	0.69	0.75	-
UTQA [16]	0.76	0.62	0.68	-
UTQA [16]	0.70	0.61	0.65	-
WDAqua-core1	**0.62**	**0.40**	**0.49**	**0.93s**
SemGraphQA [2]	0.70	0.25	0.37	-
QALD-7				
WDAqua-core1	**0.63**	**0.32**	**0.42**	**0.47s**

Table 1. Table comparing WDAqua-core1 with other QA systems evaluated over QALD-3 (over DBpedia 3.8), QALD-4 (over DBpedia 3.9), QALD-5 (over DBpedia 2014), QALD-6 (over DBpedia 2015-10), QALD-7 (over DBpedia 2016-04). We indicated the average running times of a query if the corresponding publication contained the information. These evaluation are performed on different hardware, but still give a good idea about scalability.

5 Conclusion

We have described the scalability problem of QA system over KB. Moreover, we have described how scalability is addressed in WDAqua-core1. Finally, we have compared the runtime performance of WDAqua-core1 with existing QA solutions evaluated over the popular QALD benchamrk series. The presented results show that WDAqua-core1 has a runtime performance that can compete with all evaluated system.

Note: There is a Patent Pending for the presented approach. It was submitted the 18 January 2018 at the EPO and has the number EP18305035.0.

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

References

1. Baudiš, P., Šedivỳ, J.: QALD challenge and the YodaQA system: Prototype notes

2. Beaumont, R., Grau, B., Ligozat, A.L.: SemGraphQA@QALD-5: LIMSI participation at QALD-5@CLEF. CLEF (2015)
3. Diefenbach, D., Amjad, S., Both, A., Singh, K., Maret, P.: Trill: A reusable front-end for qa systems. In: ESWC P&D (2017)
4. Diefenbach, D., Both, A., Singh, K., Maret, P.: Towards a question answering system over the semantic web (2018), arXiv:1803.00832
5. Dima, C.: Intui2: A prototype system for question answering over linked data. Proceedings of the Question Answering over Linked Data lab (QALD-3) at CLEF (2013)
6. Dima, C.: Answering natural language questions with Intui3. In: Conference and Labs of the Evaluation Forum (CLEF) (2014)
7. Dubey, M., Dasgupta, S., Sharma, A., Höffner, K., Lehmann, J.: AskNow: A Framework for Natural Language Query Formalization in SPARQL. In: International Semantic Web Conference. Springer (2016)
8. Giannone, C., Bellomaria, V., Basili, R.: A HMM-based approach to question answering against linked data. Proceedings of the Question Answering over Linked Data lab (QALD-3) at CLEF (2013)
9. Hakimov, S., Unger, C., Walter, S., Cimiano, P.: Applying semantic parsing to question answering over linked data: Addressing the lexical gap. In: Natural Language Processing and Information Systems. Springer (2015)
10. He, S., Zhang, Y., Liu, K., Zhao, J.: CASIA@ V2: A MLN-based Question Answering System over Linked Data. Proc. of QALD-4 (2014)
11. Lopez, V., Tommasi, P., Kotoulas, S., Wu, J.: Queriodali: Question answering over dynamic and linked knowledge graphs. In: International Semantic Web Conference. pp. 363–382. Springer (2016)
12. Park, S., Shim, H., Lee, G.G.: ISOFT at QALD-4: Semantic similarity-based question answering system over linked data. In: CLEF (2014)
13. Pradel, C., Haemmerlé, O., Hernandez, N.: A semantic web interface using patterns: the SWIP system. In: Graph Structures for Knowledge Representation and Reasoning. Springer (2012)
14. Ruseti, S., Mirea, A., Rebedea, T., Trausan-Matu, S.: QAnswer-Enhanced Entity Matching for Question Answering over Linked Data. CLEF (2015)
15. 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)
16. Pouran-ebn veyseh, A.: Cross-Lingual Question Answering Using Profile HMM & Unified Semantic Space. In: ESWC (2016), *to appear*
17. Xu, K., Feng, Y., Zhao, D.: Xser@ QALD-4: Answering Natural Language Questions via Phrasal Semantic Parsing (2014)
18. Yahya, M., Berberich, K., Elbassuoni, S., Ramanath, M., Tresp, V., Weikum, G.: Natural language questions for the web of data. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics (2012)
19. Zhu, C., Ren, K., Liu, X., Wang, H., Tian, Y., Yu, Y.: A Graph Traversal Based Approach to Answer Non-Aggregation Questions Over DBpedia. arXiv preprint arXiv:1510.04780 (2015)
20. Zou, L., Huang, R., Wang, H., Yu, J.X., He, W., Zhao, D.: Natural language question answering over RDF: a graph data driven approach. In: Proceedings of the 2014 ACM SIGMOD international conference on Management of data. ACM (2014)