QALD Challenge and the YodaQA System: Prototype Notes

Petr Baudiš and Jan Šedivý

Dept. of Cybernetics, Czech Technical University,
Technická 2, Praha, Czech Republic
baudipet@fel.cvut.cz

Abstract. We briefly outline the YodaQA open domain question answering system and its initial application on the Question Answering over Linked Data challenge QALD5 on CLEF2015. Since YodaQA is focused on QA over unstructured data and has been only minimally modified for QALD, it can clearly serve as just a baseline for this task.

Keywords: Question answering, linked data, natural language processing.

1 Introduction

The YodaQA system for open domain factoid English question answering has been published recently. [1] [2] The system is fully open source, modular pipeline inspired by the IBM Watson DeepQA system [4].

QALD [7] is a series of evaluation campaigns on multilingual question answering primarily over linked data, for the 2015 run also extended with hybrid questions that require integration of both linked data and information stored in unstructured text snippets.

The working notes are structured as follows. In Sec. 2, we briefly outline the YodaQA system in its original form. In Sec. 3, we discuss the changes of the system for the biomedical domain. In Sec. 4, we review the system performance.

2 YodaQA Summary

The YodaQA pipeline is implemented mainly in Java, using the Apache UIMA framework [5]. Detailed technical description of the pipeline is included in a technical report [1].

The system maps an input question to ordered list of answer candidates in a pipeline fashion, encompassing the following stages:

- **Question Analysis** extracts natural language features from the input and produces in-system representations of the question. We currently build just a naive representation of the question as bag-of-features. The most important characterization of the question is a set of clues (keywords, keyphrases and *concept clues* that crisply match enwiki titles) and possible lexical answer types.

- **Answer Production** generates a set of candidate answers based on the question, typically by performing a **Primary Search** in the knowledge bases according to the question clues and either directly using search results as candidate answers or filtering relevant passages from these (the **Passage Extraction**) and generating candidate answers from picked passages (the **Passage Analysis**).
  Answers from text passages (fetched from English Wikipedia) are produced by a simple strategy of considering all named entities and noun phrases as candidates. Answers from structured knowledge bases are produced by considering triples with subject being a concept clue; the (direct) objects are the candidate answers, with their lexical types pre-seeded as the predicate labels. The DBpedia [6] `ontology` (curated) and `property` (raw infobox) namespaces and the Freebase [3] RDF dump are used as data sources.
- **Answer Analysis** generates various answer features based on detailed analysis. Most importantly, this concerns lexical type determination and coercion to question type. Other features include distance from clues in passages or text overlap with clues.
- **Answer Merging and Scoring** consolidates the set of answers, removing duplicates and using a machine learned classifier (logistic regression) to score answers by their features.

3 YodaQA Domain Adaptation

We made only minimal adjustments to the end-to-end pipeline for the QALD task. The changes are available within the public open source code base (`https://github.com/brmson/yodaqa`) in the `d/clef2015-qald` branch.

We did some technical modifications to be able to process the dataset in the QALD XML format and return resource identifiers instead of strings as answers when applicable.[1] As another minor change, we enhanced our question analysis for imperative and otherwise specifically phrased questions which were uncommon in our TREC-based open domain dataset.

The QALD task requires exact answer matches whereas our typical evaluation scenario that simply requires that the gold standard answer is a sub-string of the produced answers answer (e.g. "the red color" would be acceptable for gold standard "red" in our scenario). Therefore, we modify our system to require exact matches during training, and disable a heuristic in answer analysis which attempts to find a *focus word* in the answer and run analysis (like title lookup, type coercion) on it instead of the whole answer.

Our system is currently designed to answer just factoid single-answer questions, while the QALD challenge also includes boolean questions and many questions that require a list of multiple answers. In case the question contains one of words *Give, List, Show*, the top 15 generated answers (no matter the confidence)

[1] Full-text based answers had DBpedia resource identifier generated based on their originating Wikipedia article. Answers based solely on Freebase were ignored when a resource identifier was required.

Pipeline	Recall	Accuracy-at-1	MRR
final (ref)	54.8%	24.2%	0.290
final (eval)	42.4%	18.6%	0.239
open domain	79.3%	32.6%	0.420

Fig. 1. Benchmark results of various pipeline variants on the test split of the dataset (ref) and the competition results (eval). Recall counts questions where at least one of the generated answers is member of the gold standard set. Accuracy-at-1 counts questions where the first answer is member of the gold standard set. MRR is the Mean Reciprocal Rank $|Q| \cdot \sum_{q \in Q} 1/r_q$.

are output; otherwise, only the top generated answer is output. Since we implemented no text entailment algorithm yet, we simply use a fixed *true* answer for all since it was much more common in the training dataset. We further made use of the supplied answer type information to filter only parseable float answers in case *number* was specified. We did not handle the *date* type in any special way.

4 Results

To evaluate the performance of our system, we split the (randomly reshuffled) reference QALD5 training dataset to a local dev/train set (164 questions) and a test set (157 questions); we did not distinguish the hybrid questions in any way, but ignored the "out of scope" questions. For comparison, we also include baseline version performance[2] on the "curated" factoid open domain dataset [2].

The results are summarized in Table 1. Note that the performance criteria are computed more optimistically in our framework compared to the QALD rules, as explained in the table caption — it woudl be sufficient to return only one of the correct answers.

The YodaQA system is currently focused on information extraction from unstructured text[3] rather than advanced graph based extraction methods and reasoning. This choice clearly handicaps it in this challenge. We aim to add more advanced structured query capabilities for the next year's version of YodaQA.

References

1. Baudiš, P. YodaQA: A Modular Question Answering System Pipeline. In *POSTER 2015 - 19th International Student Conference on Electrical Engineering.*
2. Baudiš, P., and Šedivý, J. Modeling of the Question Answering Task in the YodaQA System. In *CLEF 2015, short paper (submitted, in review)* (2015).

[2] This performance is done using the open domain metric which permits gold standard substrings, see above.

[3] Our motivation is easy adaptation to closed domains where structured data may be expected to be scarce.

3. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data* (2008), ACM, pp. 1247–1250.
4. Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J., et al. Building watson: An overview of the deepqa project. *AI magazine 31*, 3 (2010), 59–79.
5. Ferrucci, D., and Lally, A. UIMA: An architectural approach to unstructured information processing in the corporate research environment. *Nat. Lang. Eng. 10*, 3-4 (Sept. 2004), 327–348.
6. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web* (2014).
7. Lopez, V., Unger, C., Cimiano, P., and Motta, E. Evaluating question answering over linked data. *Web Semantics Science Services And Agents On The World Wide Web 21* (2013), 3–13.