

QAKiS: an Open Domain QA System based on Relational Patterns

Elena Cabrio¹, Julien Cojan^{1*}, Alessio Palmero Aprosio^{2,3},
Bernardo Magnini², Alberto Lavelli², and Fabien Gandon¹

¹ INRIA, 2004 Route des Lucioles, Sophia Antipolis, France
{elena.cabrio, julien.cojan, fabien.gandon}@inria.fr

² FBK-Irst, Via Sommarive 18, Povo-Trento, Italy
{aprosio, magnini, lavelli}@fbk.eu

³ Università degli Studi di Milano, Via Comelico 39/41, Milano, Italy

Abstract. We present QAKiS, a system for open domain Question Answering over linked data. It addresses the problem of question interpretation as a relation-based match, where fragments of the question are matched to binary relations of the triple store, using relational textual patterns automatically collected. For the demo, the relational patterns are automatically extracted from Wikipedia, while DBpedia is the RDF data set to be queried using a natural language interface.

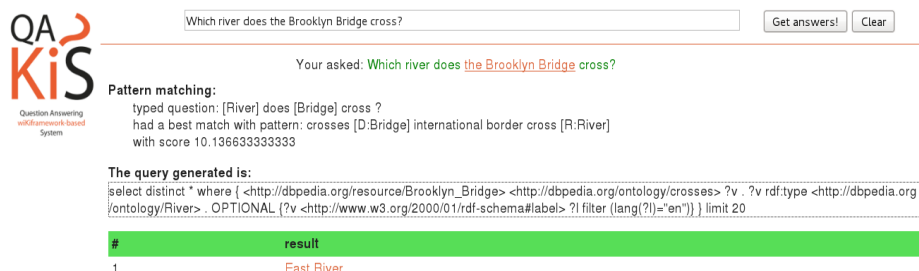
1 Introduction

To enhance users interactions with the web of data, query interfaces providing a flexible mapping between natural language expressions, and concepts and relations in structured knowledge bases are becoming particularly relevant. This demonstration presents QAKiS (Question Answering wiKiframework-based System), that allows end users to submit a query to an RDF triple store in English and obtain the answer in the same language, hiding the complexity of the non intuitive formal query languages involved in the resolution process. At the same time, the expressiveness of these standards is exploited to scale to the huge amounts of available semantic data. In its current implementation, QAKiS addresses the task of QA over structured Knowledge Bases (KBs) (e.g. DBpedia) where the relevant information is expressed also in unstructured form (e.g. Wikipedia pages). Its major novelty is to implement a relation-based match for question interpretation, to convert the user question into a query language (e.g. SPARQL). Most of the current approaches (for an overview, see [2]) base this conversion on some form of flexible matching between words of the question and concepts and relations of a triple store, disregarding the relevant context around a word, without which the match might be wrong. QAKiS tries instead first to establish a matching between fragments of the question and relational textual patterns automatically collected from Wikipedia. The underlying intuition is that a relation-based matching would provide more precision with respect to matching on single tokens, as done by current QA systems.

* Acknowledges the French Minister of Culture for supporting the DBpedia Fr project.

2 QAKiS system description

QAKiS demo⁴ (Fig. 1) is based on Wikipedia for patterns extraction. DBpedia is the RDF data set to be queried using a natural language interface.



The screenshot shows the QAKiS interface. At the top left is the logo 'QAKiS' with the tagline 'Question Answering with framework-based System'. A search bar contains the question 'Which river does the Brooklyn Bridge cross?'. To the right are buttons for 'Get answers!' and 'Clear'. Below the search bar, the user's question is repeated: 'Your asked: Which river does the Brooklyn Bridge cross?'. Under 'Pattern matching:', it shows a typed question '[River] does [Bridge] cross ?' and a match with pattern 'crosses [D:Bridge] international border cross [R:River]' with a score of 10.136633333333. Below this, 'The query generated is:' is followed by a SPARQL query: 'select distinct * where { <http://dbpedia.org/resource/Brooklyn_Bridge> <http://dbpedia.org/ontology/crosses> ?v . ?v rdf:type <http://dbpedia.org/ontology/River> . OPTIONAL { ?v <http://www.w3.org/2000/01/rdf-schema#label> ?l filter (lang(?l)='en') } } limit 20'. At the bottom, a table with a green header shows one result: '# result' and '1 East River'.

Fig. 1. QAKiS demo interface. The user can either write a question (or select among a list of examples) and click on *Get Answers!*. QAKiS outputs: *i*) the user question (the recognized Named Entity (NE) is linked to its DBpedia page), *ii*) the generated typed question (see Section 2.1), *iii*) the pattern matched, *iv*) the SPARQL query sent to the DBpedia SPARQL endpoint, and *v*) the answer (below the green rectangle *results*).

QAKiS makes use of relational patterns (automatically extracted from Wikipedia and collected in the WikiFramework repository [3]), that capture different ways to express a certain relation in a given language. For instance, the relation **crosses**(**Bridge**,**River**) can be expressed in English, among the others, by the following relational patterns: [**Bridge** crosses the **River**] and [**Bridge** spans over the **River**]. Assuming that there is a high probability that the information in the Infobox is also expressed in the same Wikipedia page, the WikiFramework establishes a 4-step methodology to collect relational patterns in several languages for the DBpedia ontology relations (similarly to [1],[4]): *i*) a DBpedia relation is mapped with all the Wikipedia pages in which such relation is reported in the Infobox; *ii*) in such pages we collect all the sentences containing both the domain and the range of the relation; *iii*) all sentences for a given relation are extracted and the domain and range are replaced by the corresponding DBpedia ontology classes; *iv*) the patterns for each relation are clustered according to the lemmas between the domain and the range, and sorted according to their frequency.

2.1 System architecture

QAKiS is composed of two main modules (Fig. 2): *i*) the **query generator** takes the user question as input, generates the typed questions, and then generates the SPARQL queries from the retrieved patterns; *ii*) the **pattern matcher** takes as input a typed question, and retrieves the patterns (among those stored in the pattern repository) matching it with the highest similarity.

⁴ Available at <http://dbpedia.inria.fr/qakis/>

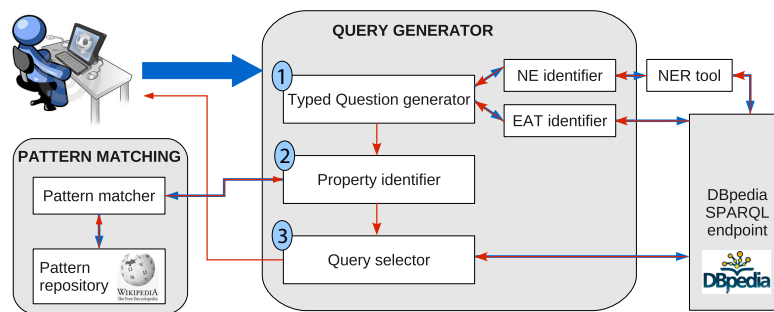


Fig. 2. QAKiS workflow

The current version of QAKiS targets questions containing a NE related to the answer through one property of the ontology, as *Which river does the Brooklyn Bridge cross?*. Each question matches a single pattern (i.e. one relation).

Expected Answer Type (EAT) and NE identification. Before running the *pattern matcher* component, we identify the target of the question with a NER tool. We apply the Stanford Core NLP NE Recognizer together with a set of strategies based on the comparison with the labels of the instances in the DBpedia ontology. We plan to test the use of other NER tools in the future. At the same time, simple heuristics are applied to infer the EAT from the question keyword, e.g. if the question starts with “When”, the EAT is [Date] or [Time], with “Who”, the EAT is [Person] or [Organisation] and so on.

Typed questions generation. We generate a *typed question* by replacing the question keywords (e.g. who, where) and the NE by the types and supertypes. Given the question “Who is the husband of Amanda Palmer?” 9 typed questions are generated, since *i)* both [Person] or [Organisation] (subclasses of [owl:Thing]) are considered as EAT, and *ii)* [MusicalArtist], [Artist] and [owl:Thing] are the types of the NE *Amanda Palmer*.

WikiFramework pattern matching. The typed questions are lemmatized, tokenized, and stopwords are removed. A Word Overlap algorithm is then applied to match such typed questions with the patterns for each relation. A similarity score is provided for each match: the highest represents the most likely relation.

Query selector. A set of patterns (max. 5) is retrieved by the pattern matcher component for each typed question, and sorted by decreasing matching score. For each of them, one or two SPARQL queries are generated, either *i)* `select ?s where{?s <property> <NE>}`, *ii)* `select ?s where{<NE> <property> ?s}` or *iii)* both, according to the compatibility between their types and the property domain and range. Such queries are then sent to the SPARQL endpoint for answer retrieval. If the query produces no results, we try with the next pattern, until a satisfactory query is found or no more patterns are retrieved.

2.2 Experimental evaluation

Table 1 reports QAKiS’s results on the QALD-2 data sets⁵ (DBpedia track). For the demo, we focused on code optimization reducing QAKiS average processing time per question from 15 to 2 sec., w.r.t. the version used for the challenge.

	<i>Precision</i>	<i>Recall</i>	<i>F-measure</i>	<i># answered</i>	<i># right answ.</i>	<i># partially right</i>
train	0.476	0.479	0.477	40/100	17/40	4/40
test	0.39	0.37	0.38	35/100	11/35	4/35

Table 1. QAKiS performances on DBpedia data sets (participation to QALD-2)

Most of QAKiS’ mistakes concern wrong relation assignment (i.e. wrong pattern matching). We plan to replace the Word Overlap algorithm with approaches considering the syntactic structure of the question. Another issue concerns questions ambiguity, i.e. the same surface forms can in fact refer to different relations in the DBpedia ontology. We plan to cluster relations with several patterns in common, to allow QAKiS to search among all the relations in the cluster.

The partially correct answers concern questions involving more than one relation: the actual version of the algorithm detects indeed only one of them. We plan to target questions as *Give me all people that were born in Vienna and died in Berlin* in a short time, since the two relations are easily separable. On the contrary, we need more complex strategies to answer questions with nested relations.

3 Future perspectives

We are currently considering to publish the WikiFramework relational patterns as RDF triples, organized according to a newly defined RDF vocabulary (similarly to [1]). We are also planning improvements on: *i*) the WikiFramework pattern extraction algorithm, following [4]; *ii*) the question-pattern matching algorithm; *iii*) the system coverage, addressing boolean and n-relation questions. We are also exploring QAKiS applicability in real application scenarios.

References

1. Gerber, D., Ngonga Ngomo, A.C. (2011), Bootstrapping the Linked Data Web, in 1st Workshop on Web Scale Knowledge Extraction @ ISWC 2011.
2. Lopez V., Uren V., Sabou, M., Motta, E. (2011), Is Question Answering fit for the Semantic Web?: a Survey, in Semantic Web journal, vol. 2(2), pp. 125-155.
3. Mahendra, R., Wanzare, L., Bernardi, R., Lavelli, A., Magnini, B. (2011), Acquiring Relational Patterns from Wikipedia: A Case Study, in Proc. of LTC2011.
4. Wu F., Weld, D.S. (2010), Open information extraction using Wikipedia, in Proc. of ACL2010, pp. 118-127.

⁵ <http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/>