

# QALD-Mini-Project

Lukas Blübaum  
Nick Düsterhus  
Ralf Keller

University of Paderborn

*<https://github.com/LukasBluebaum/QALD-Mini-Project>*

July 19, 2018

# Overview

- 1 Task Description
- 2 Our Approach
- 3 Architecture
- 4 Question Preprocessing
- 5 Template Overview
- 6 Benchmarking

# Task Description

- Building a Question Answering Engine that is able to get an F-measure of at least 0.1
- Using DBpedia as knowledge base

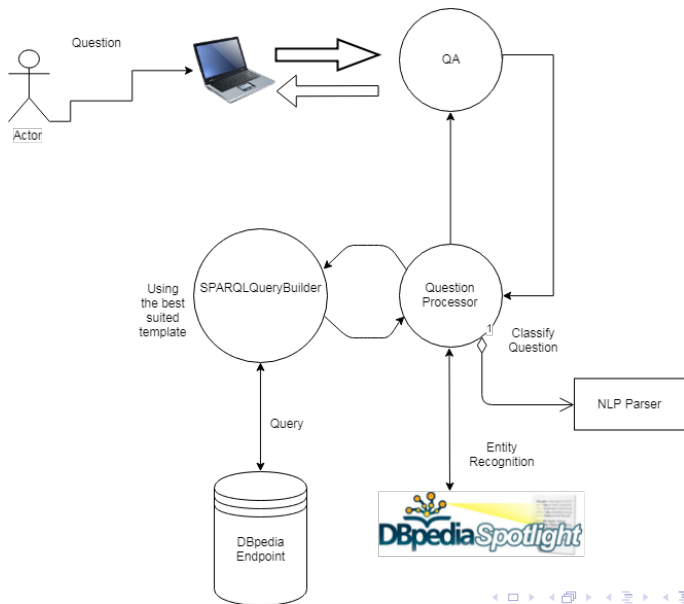
## Given

- Library qa.annotation (finding entities, properties, classes) and qa.common (load / store QALD Datasets)
- A wrapper to plug in GERBIL QA

# Our Approach

- Template based
- Classify question types and apply natural language processing to get important keywords
- Find entities, classes, and properties that match the question
- Build SPARQL query templates for the most common types of questions

# Simplified Procedure



- Three modes:
  - Run offline: tries to answer one committed question
  - Answer Dataset Questions: Load a QALD Dataset and save the result as a JSON-File (list of questions and their answers)
  - Can be used as a web service

- Using the GerbilQA-Benchmarking-Template
  - Web server via the Spring Framework
  - Takes an HTTP-POST request containing the question
- Passes the question to the QuestionEngine
- Returns a JSON-String containing the answer and the used query

- Determining which method to build a SPARQL query should be applied
- Classifies questions by their starting word
  - e.g. "When": we can conclude from that, that the given result should be from the datatype date or year
  - Distinguishes between ASK and SELECT clause



# Question Preprocessing

- Requesting Spotlight to get all named entities in the question
- Using the Stanford CoreNLP to find keywords that give us information about the relations from the question
- FindClasses/findProperties: Using IndexDBO\_classes from the qa.annotation library on nouns, verbs and adjectives

## Templates for different types of questions

- Boolean questions such as: "Do Prince Harry and Prince William have the same parents?"
  - List questions
  - Who, Which, When, Where
  - How (much/many)
- 
- Further differentiation which template to use based on the number of classes/entities and comparison words
  - Requests DBpedia endpoint using Apache Jena library

# Sparql Query Templates

## Example Query

### Example (most basic query)

- Question: "Who was the doctoral supervisor of Albert Einstein?"
- One Entity: Albert Einstein
- Doctoral supervisor maps to property dbo:doctoralAdvisor

Query  $\Rightarrow$  `SELECT DISTINCT ?answer WHERE {  
 ?answer a foaf:Person.  
 <http://dbpedia.org/resource/Albert_Einstein>  
 <http://dbpedia.org/ontology/doctoralAdvisor> ?answer .  
}`

# Sparql Query Templates

## Comparison

- Predefined comparison enum for questions containing superlatives or comparatives

```
package utils;

import java.util.ArrayList;

public enum Comparison {
    LONG("http://dbpedia.org/ontology/length" ),
    LONGER("http://dbpedia.org/ontology/length", "DESC"),
    LONGEST("http://dbpedia.org/ontology/length", "DESC"),
    OLD("http://dbpedia.org/ontology/openingYear",http://dbpedia.org/ontology/birthDate"),
    OLDER("http://dbpedia.org/ontology/openingYear,http://dbpedia.org/ontology/birthDate", "DESC"),
    OLDEST("http://dbpedia.org/ontology/openingYear,http://dbpedia.org/ontology/birthDate", "DESC"),
    TALL("http://dbpedia.org/ontology/height"),
    TALLER("http://dbpedia.org/ontology/height", "DESC"),
    TALLEST("http://dbpedia.org/ontology/height", "DESC"),
    SHORT("http://dbpedia.org/ontology/height"),
    SHORTER("http://dbpedia.org/ontology/height", "ASC"),
    SHORTEST("http://dbpedia.org/ontology/height", "ASC"),
    HIGH("http://dbpedia.org/ontology/elevation"),
    HIGHER("http://dbpedia.org/ontology/elevation,http://dbpedia.org/property/higher", "DESC"),
    HIGHEST("http://dbpedia.org/ontology/elevation,http://dbpedia.org/property/highest", "DESC"),
    SMALL("http://dbpedia.org/ontology/areaTotal"),
    SMALLER("http://dbpedia.org/ontology/areaTotal", "ASC"),
    SMALLEST("http://dbpedia.org/ontology/areaTotal", "ASC"),
    LARGE ("http://dbpedia.org/ontology/areaTotal"),
    LARGER("http://dbpedia.org/ontology/areaTotal", "DESC"),
    LARGEST("http://dbpedia.org/ontology/areaTotal", "DESC"),
    BIG("http://dbpedia.org/ontology/areaTotal"),
    BIGGER("http://dbpedia.org/ontology/areaTotal", "DESC"),
    BIGGEST("http://dbpedia.org/ontology/areaTotal", "DESC");

    private String order;
    private ArrayList<String> uri = new ArrayList<String>();

    Comparison(String pURI){
        String[] uris = pURI.split(",");
        for(String u: uris) {
            uri.add(u);
        }
    }

    Comparison(String pURI, String pOrder) {
        this.order = pOrder;
        String[] uris = pURI.split(",");
        for(String u: uris) {
            uri.add(u);
        }
    }

    public String getURI(int i) {
        return uri.get(i);
    }

    public String getOrder() {
        return this.order;
    }
}
```

# Sparql Query Templates

## Example Query

### Example (largest)

- Question: "What is the largest country in the world?"
- Zero Entities
- One Class: Country
- Comparison enum: largest ( dbo:areaTotal )
- Order: DESC
- Query  $\Rightarrow$ 

```
SELECT ?answer WHERE {  
  ?answer rdf:type <http://dbpedia.org/ontology/Country>.  
  ?answer <http://dbpedia.org/ontology/areaTotal> ?area .}  
ORDER BY DESC(?area) LIMIT 1 OFFSET 0
```

# Benchmarking

## QALD8 Test

### GERBIL Experiment

Experiment URI: <http://gerbil-qa.aksw.org/gerbil/experiment?id=201807180000> and <http://w3id.org/gerbil/qa/experiment?id=201807180000>

Type: QA

Matching: Me - strong entity match

Annotator	Dataset	Language		Micro F1	Micro Precision	Micro Recall	Macro F1	Macro Precision	Macro Recall	Error Count	avg millis/doc	Macro F1 QALD	Timestamp	GERBIL version
QA (uploaded)	QALD8 Test Multilingual	en		0,42	0,75	0,2917	0,3831	0,3862	0,3821	0	0	0,5271	2018-07-18 00:04:09	0.2.3
QA (uploaded)	QALD8 Test Multilingual	en	Answer Type	1	1	1	1	1	1	0			2018-07-18 00:04:09	0.2.3
QA (uploaded)	QALD8 Test Multilingual	en	C2KB	0,5031	0,5714	0,4494	0,4862	0,4951	0,4919	0			2018-07-18 00:04:09	0.2.3
QA (uploaded)	QALD8 Test Multilingual	en	P2KB	0,4524	0,5429	0,3878	0,4309	0,4431	0,4358	0			2018-07-18 00:04:09	0.2.3
QA (uploaded)	QALD8 Test Multilingual	en	RE2KB	0,3571	0,4286	0,3061	0,3659	0,3618	0,374	0			2018-07-18 00:04:09	0.2.3

- <http://gerbil-qa.aksw.org/gerbil/experiment?id=201807160000>

# Benchmarking

## QALD8 Train

### GERBIL Experiment

Experiment URI: <http://gerbil-qa.aksw.org/gerbil/experiment?id=201807180003> and <http://w3id.org/gerbil/qa/experiment?id=201807180003>

Type: QA

Matching: Me - strong entity match

Annotator	Dataset	Language		Micro F1	Micro Precision	Micro Recall	Macro F1	Macro Precision	Macro Recall	Error Count	avg millis/doc	Macro F1 QALD	Timestamp	GERBIL version
QA (uploaded)	QALD8 Train Multilingual	en		0,0407	0,0433	0,0384	0,2784	0,2805	0,3261	0	0	0,4504	2018-07-18 22:35:43	0.2.3
QA (uploaded)	QALD8 Train Multilingual	en	Answer Type	1	1	1	1	1	1	0			2018-07-18 22:35:43	0.2.3
QA (uploaded)	QALD8 Train Multilingual	en	C2KB	0,4339	0,5754	0,3483	0,4	0,4519	0,3894	0			2018-07-18 22:35:43	0.2.3
QA (uploaded)	QALD8 Train Multilingual	en	P2KB	0,3164	0,4398	0,2471	0,2793	0,3196	0,2756	0			2018-07-18 22:35:43	0.2.3
QA (uploaded)	QALD8 Train Multilingual	en	RE2KB	0,2411	0,3351	0,1882	0,2177	0,2306	0,2245	0			2018-07-18 22:35:43	0.2.3

- <http://gerbil-qa.aksw.org/gerbil/experiment?id=201807170000>