Question Answering and Chatbots

6th Practical exercise – Tests

Aleksandr Perevalov

aleksandr.perevalov@hs-anhalt.de

October 4, 2021

Hochschule Anhalt
Anhalt University of Applied Sciences

Plan for today

Plan for today

- Review the task for the Exercise 6;

Plan for today

- Review the task for the Exercise 6;
- Demo Session;

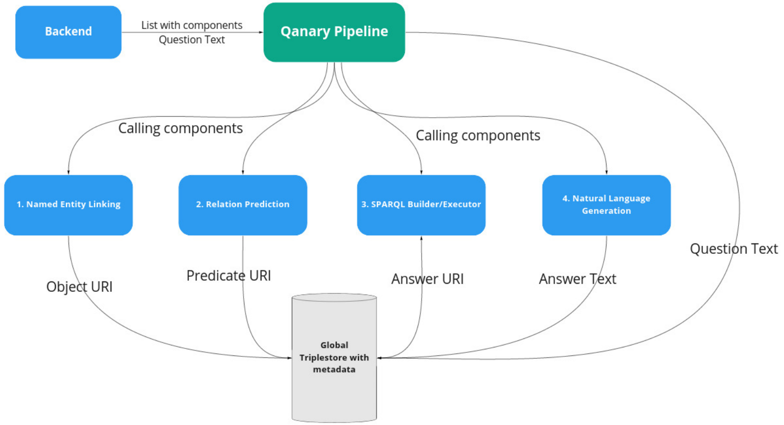
Plan for today

- Review the task for the Exercise 6;
- Demo Session;
- Questions;

Plan for today

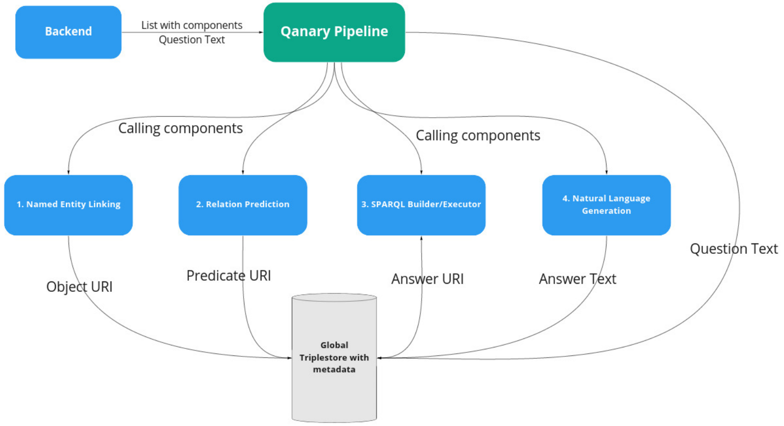
- Review the task for the Exercise 6;
- Demo Session;
- Questions;
- Introduction to the Exercise 7 (Deploying).

Exercise 6 – the Task



Based on the test data for each variant:

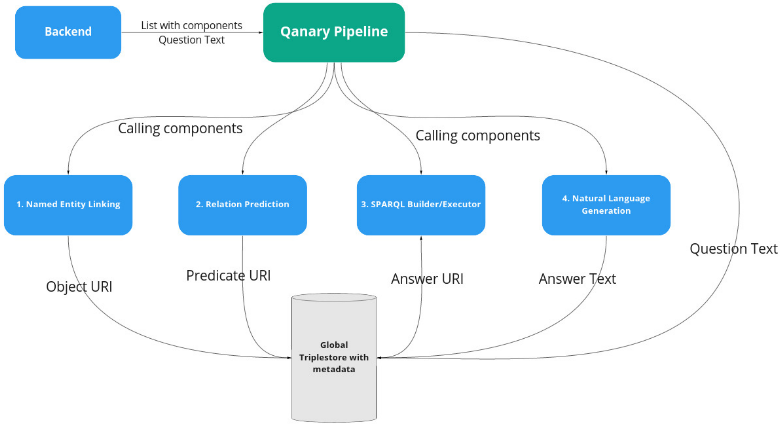
Exercise 6 – the Task



Based on the test data for each variant:

- Implement End-to-End Question Answering test;

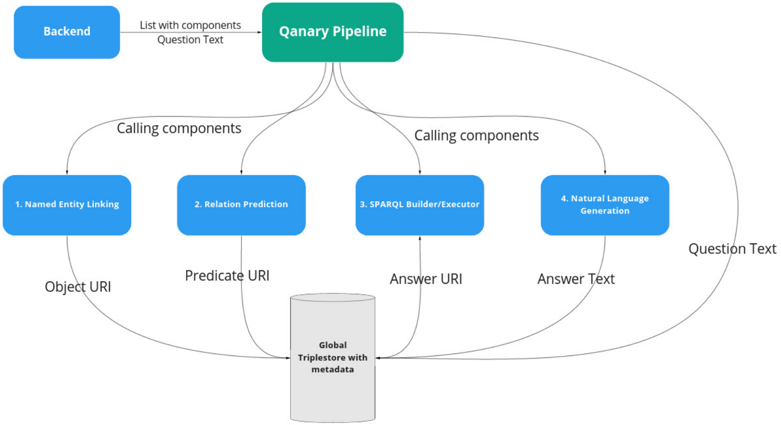
Exercise 6 – the Task



Based on the test data for each variant:

- Implement End-to-End Question Answering test;
- Implement tests for all components;

Exercise 6 – the Task



Based on the test data for each variant:

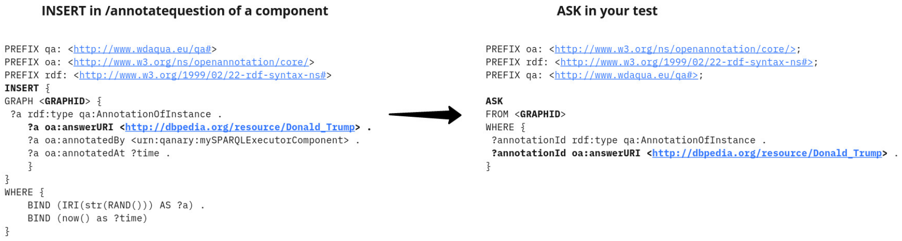
- Implement End-to-End Question Answering test;
- Implement tests for all components;
- Create a test for your machine learning models.

Exercise 6 – difficulties

Exercise 6 – difficulties

- How to get the Answer URI to implement End-To-End test?

Exercise 6 – difficulties



INSERT in /annotatequestion of a component

```
PREFIX qa: <http://www.wdaqua.eu/qa#>
PREFIX oa: <http://www.w3.org/ns/openannotation/core/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
INSERT {
GRAPH <GRAPHID> {
 ?a rdf:type qa:AnnotationOfInstance .
    ?a oa:answerURI <http://dbpedia.org/resource/Donald_Trump> .
    ?a oa:annotatedBy <urn:qanary:mySPARQLExecutorComponent> .
    ?a oa:annotatedAt ?time .
    }
}
WHERE {
    BIND (IRI(str(RAND())) AS ?a) .
    BIND (now() as ?time)
}
```

ASK in your test

```
PREFIX oa: <http://www.w3.org/ns/openannotation/core/>;
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>;
PREFIX qa: <http://www.wdaqua.eu/qa#>;

ASK
FROM <GRAPHID>
WHERE {
 ?annotationId rdf:type qa:AnnotationOfInstance .
 ?annotationId oa:answerURI <http://dbpedia.org/resource/Donald_Trump> .
}
```

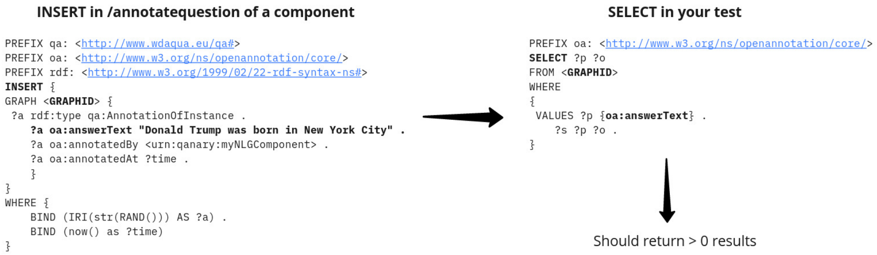
Figure: Test a component when we know the result

- How to get the Answer URI to implement End-To-End test?

Exercise 6 – difficulties

- How to test NLG (templates are selected randomly)?

Exercise 6 – difficulties


INSERT in /annotatequestion of a component
SELECT in your test
Should return > 0 results

```
PREFIX qa: <http://www.wdaqua.eu/qa#>
PREFIX oa: <http://www.w3.org/ns/openannotation/core/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
INSERT {
GRAPH <GRAPHID> {
 ?a rdf:type qa:AnnotationOfInstance .
    ?a oa:answerText "Donald Trump was born in New York City" .
    ?a oa:annotatedBy <urn:qanary:myNLGComponent> .
    ?a oa:annotatedAt ?time .
    }
}
WHERE {
    BIND (IRI(str(RAND())) AS ?a) .
    BIND (now() as ?time)
}
```

```
PREFIX oa: <http://www.w3.org/ns/openannotation/core/>
SELECT ?p ?o
FROM <GRAPHID>
WHERE
{
 VALUES ?p {oa:answerText} .
    ?s ?p ?o .
}
```

Figure: Test a component when the exact result is unknown

- How to test NLG (templates are selected randomly)?

Exercise 6 – difficulties

- What metrics should I use?

Exercise 6 – difficulties

- Precision @ k, k = **1**, 2, 3, ...: how many relevant items is present in first k answers of your system;
- Precision, Recall, F1 Score: metrics for classification;
- Accuracy: metric for the components tests.

- What metrics should I use?

Any questions?

Let's start the demo!

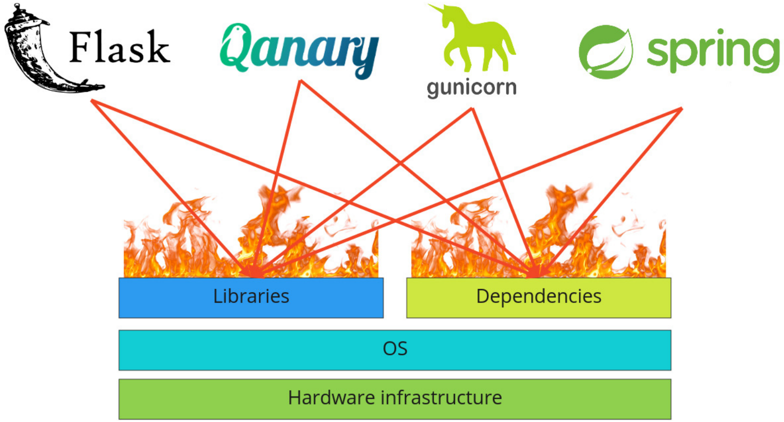
Deploying QA system

Deploying QA system

Motivation: We want our system to work in any environment: local PC, server of the University, server of the Project Sponsor etc.

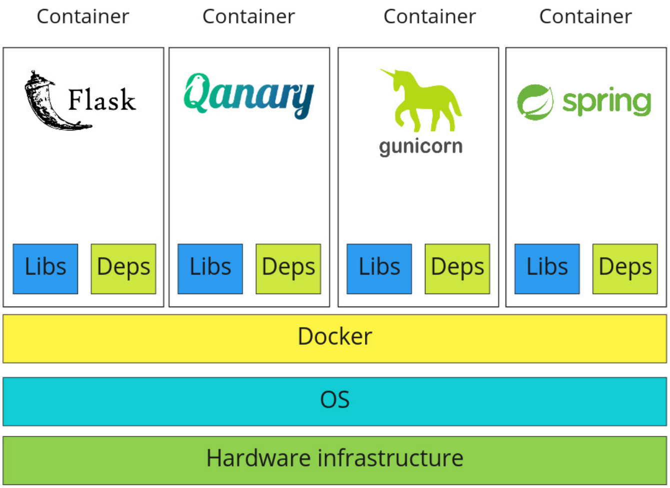
Deploying QA system

Motivation: We want our system to work in any environment: local PC, server of the University, server of the Project Sponsor etc.
Problem: Every environment has its own setup: hardware, OS, libraries and dependencies



Deploying QA system

Solution: put each service into a separate container using Docker.



Plan for the Exercise 7: Deploying

Work in teams:

- Get access to a server;
- Set up a repository for your solution and clone files on the server;
- Create Dockerfile(s) for your components;
- Create Dockerfiles for Backend and Frontend;
- Start all Dockerfiles using docker-compose;
- Show demo;

1. SPARQL;
2. Work with Natural Language (NER);
3. Question classification;
4. Back-end and Front-end;
5. Simple QA system and Qanary Framework;
6. **Tests for QA system**;
7. Deploying QA system;