Question Answering and Chatbots

7th Practical exercise – IR-based QA

Aleksandr Perevalov

aleksandr.perevalov@hs-anhalt.de

December 1, 2021

Hochschule Anhalt

Anhalt University of Applied Sciences

Plan for today

Plan for today

- Demo session;

Plan for today

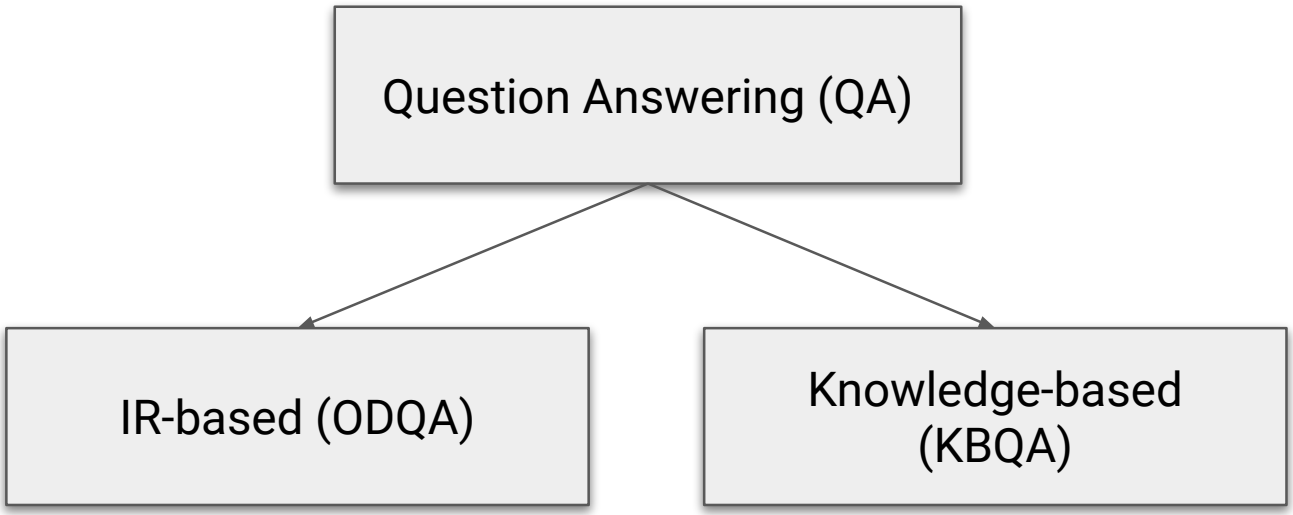
- Demo session;
- Discussion on the projects;

Plan for today

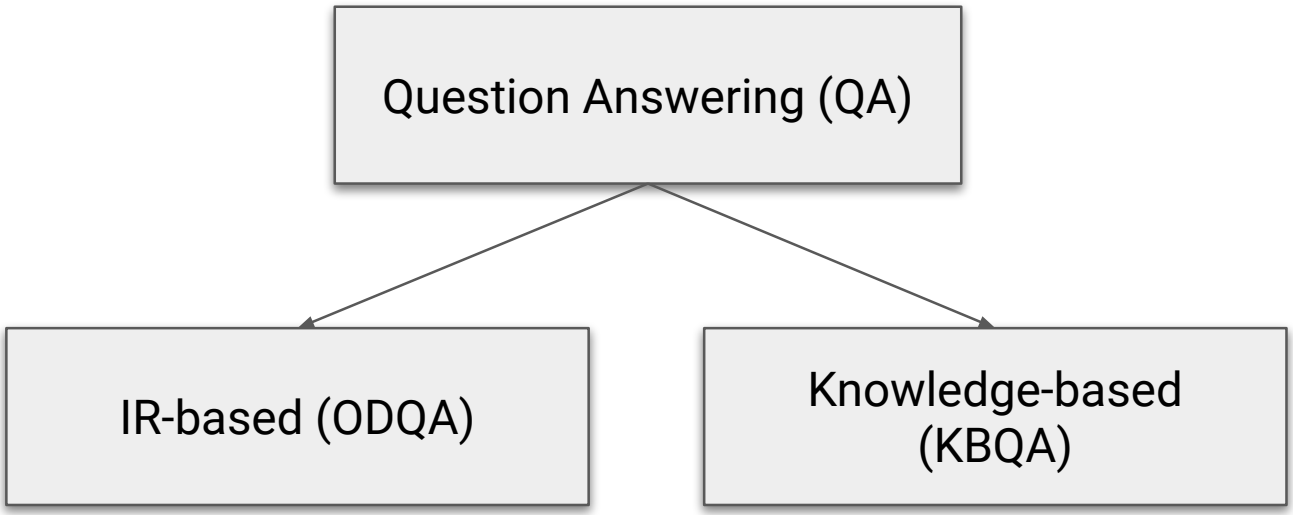
- Demo session;
- Discussion on the projects;
- Introduction of task 7.

Let's start the demo!

Recap: Paradigms of Question Answering

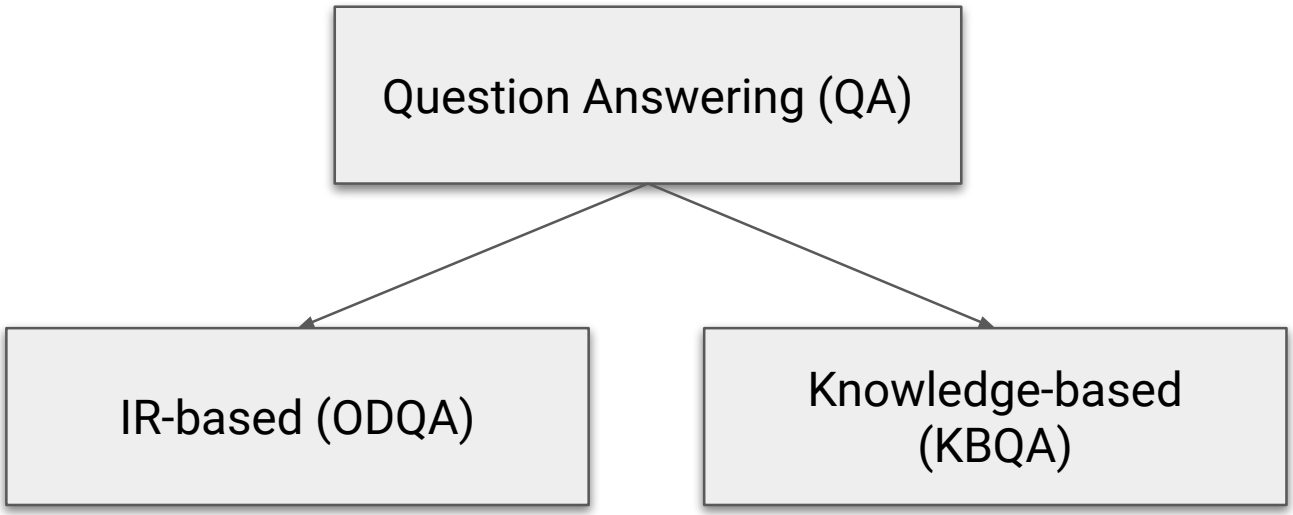


Recap: Paradigms of Question Answering



- **IR-based QA (ODQA)** – works with unstructured data;

Recap: Paradigms of Question Answering



- **IR-based QA (ODQA)** – works with unstructured data;
- **Knowledge-based QA** – works with structured data.

IR-based Question Answering

IR-based QA systems *work with unstructured data* (e.g., web pages, text documents). These systems execute *two steps* in order to retrieve the answer:

IR-based Question Answering

IR-based QA systems *work with unstructured data* (e.g., web pages, text documents). These systems execute *two steps* in order to retrieve the answer:

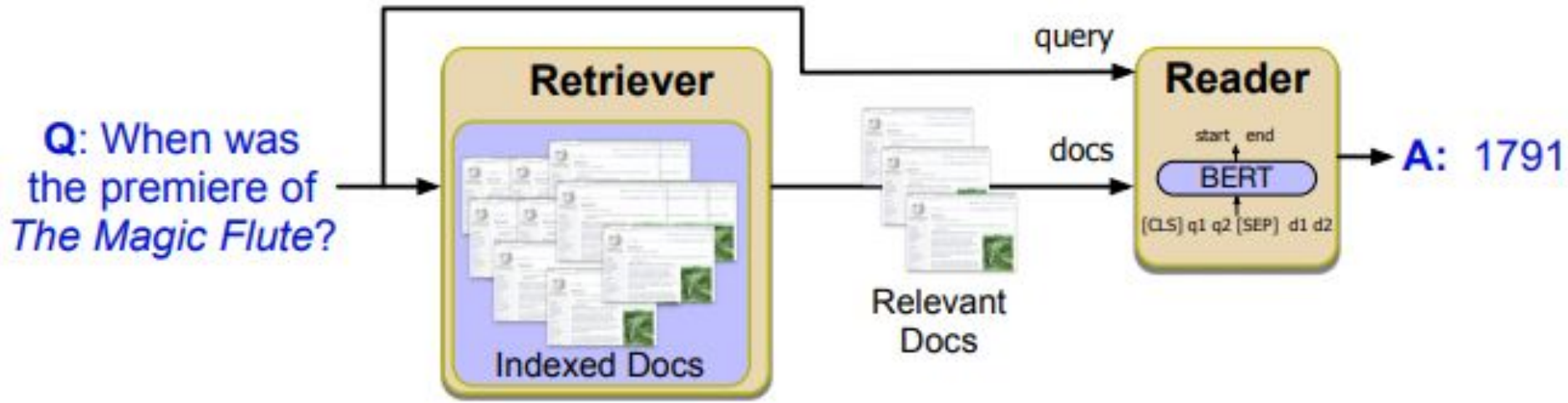
1. Retrieve the most relevant document ("passage") among the others (*retriever step*);

IR-based Question Answering

IR-based QA systems *work with unstructured data* (e.g., web pages, text documents). These systems execute *two steps* in order to retrieve the answer:

1. Retrieve the most relevant document ("passage") among the others (*retriever step*);
2. Find the answer in the retrieved document by *extracting a textual span from it* (*reader step/reading comprehension*).

Retriever-Reader Architecture



https://web.stanford.edu/~jurafsky/slp3/23.pdf

Approaches for the Retriever step

The *retriever* typically uses IR-based approaches to select the relevant documents:

Approaches for the Retriever step

The *retriever* typically uses IR-based approaches to select the relevant documents:

- Keyword-based search (word count);

Approaches for the Retriever step

The *retriever* typically uses IR-based approaches to select the relevant documents:

- Keyword-based search (word count);
- (1) Index the documents, (2) encode a question, (3) score the documents;

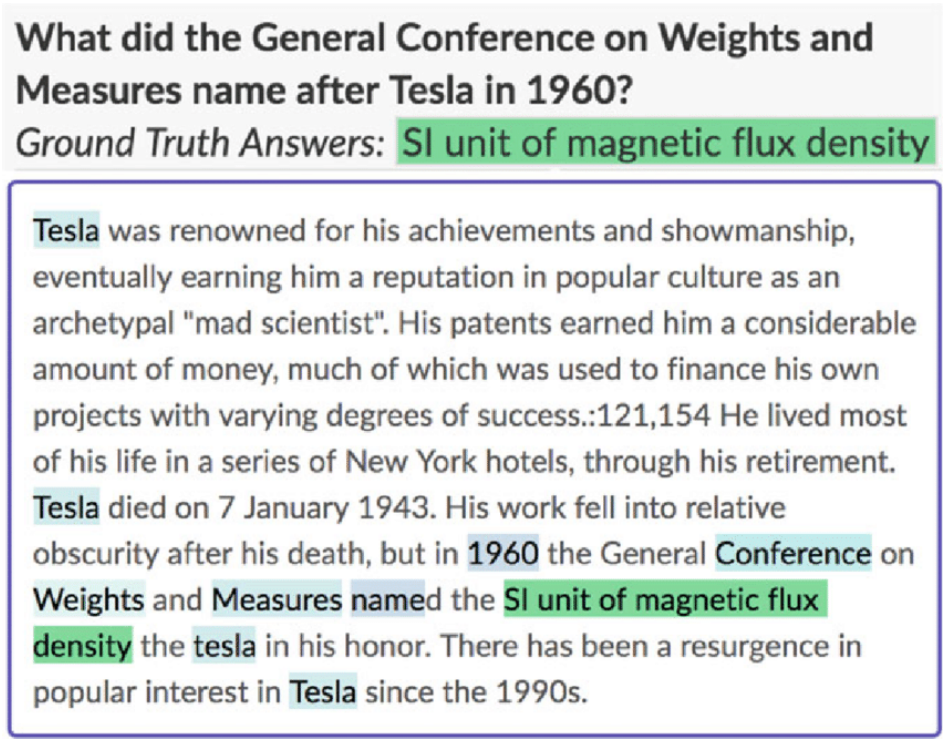
Approaches for the Retriever step

The *retriever* typically uses IR-based approaches to select the relevant documents:

- Keyword-based search (word count);
- (1) Index the documents, (2) encode a question, (3) score the documents;
- But, we can try something else e.g., look for abstracts of named entities in a knowledge graph.

Approaches for the Reader step

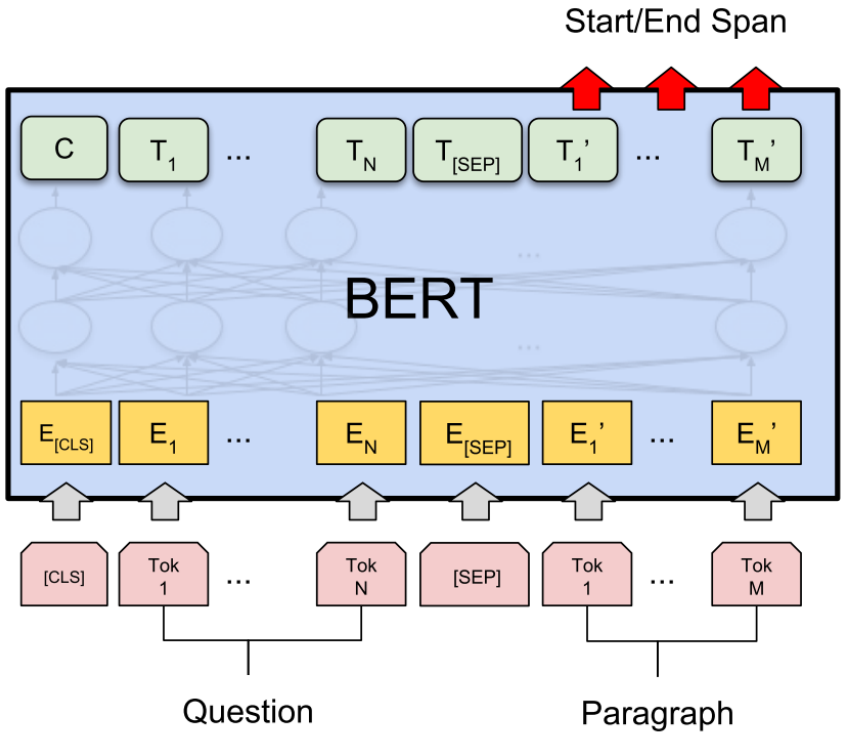
The most common approach is the *machine reading comprehension* (MRC) model



Liu, Zhiyuan. Lin, Yankai. Sun, Maosong. (2020). Document Representation.

Approaches for the Reader step

One of the examples: BERT for question-answering



Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding."

Exercise 7

Task 1

Task 2

Exercise 7

Task 1

- Using the data from Exercise 4 and DBpedia SPARQL endpoint, *retrieve the truth answer labels* for each of the questions (rdfs:label);

Task 2

Exercise 7

Task 1

- Using the data from Exercise 4 and DBpedia SPARQL endpoint, *retrieve the truth answer labels* for each of the questions (rdfs:label);
- Run the questions from the test dataset on your system and collect all the graph IDs returned from Qanary pipeline e.g., (1) NEL, (2) Retriever, (3) Reader – RoBERTa-For-QA).

Task 2

Exercise 7

Task 1

- Using the data from Exercise 4 and DBpedia SPARQL endpoint, *retrieve the truth answer labels* for each of the questions (rdfs:label);
- Run the questions from the test dataset on your system and collect all the graph IDs returned from Qanary pipeline e.g., (1) NEL, (2) Retriever, (3) Reader – RoBERTa-For-QA).
- Prepare a .csv table with the statistics.

Task 2

Exercise 7

Task 1

- Using the data from Exercise 4 and DBpedia SPARQL endpoint, *retrieve the truth answer labels* for each of the questions (rdfs:label);
- Run the questions from the test dataset on your system and collect all the graph IDs returned from Qanary pipeline e.g., (1) NEL, (2) Retriever, (3) Reader – RoBERTa-For-QA).
- Prepare a .csv table with the statistics.

Task 2

- Work with your project team members on a first sketch of the system that should fulfill your project.

Exercise 7

Task 1

- Using the data from Exercise 4 and DBpedia SPARQL endpoint, *retrieve the truth answer labels* for each of the questions (rdfs:label);
- Run the questions from the test dataset on your system and collect all the graph IDs returned from Qanary pipeline e.g., (1) NEL, (2) Retriever, (3) Reader – RoBERTa-For-QA).
- Prepare a .csv table with the statistics.

Task 2

- Work with your project team members on a first sketch of the system that should fulfill your project.
- Collect 10+ questions and the corresponding answers that should be answered in German and English.

Exercise 7

Task 1

- Using the data from Exercise 4 and DBpedia SPARQL endpoint, *retrieve the truth answer labels* for each of the questions (rdfs:label);
- Run the questions from the test dataset on your system and collect all the graph IDs returned from Qanary pipeline e.g., (1) NEL, (2) Retriever, (3) Reader – RoBERTa-For-QA).
- Prepare a .csv table with the statistics.

Task 2

- Work with your project team members on a first sketch of the system that should fulfill your project.
- Collect 10+ questions and the corresponding answers that should be answered in German and English.
- Create a sketch of the QA components (processing steps).

Any questions?

Let's start the work.

0 Introduction;
1 NER & NEL;
2 Question classification & Web service/API;
3 SPARQL queries over Knowledge Graphs;
4 Simple KGQA system – based on exercises 0, 1, 2, 3;
5 Qanary Framework – component oriented approach;
6 Evaluation of QA systems;
7 **Simple IR-based/ODQA system**.