Question Answering and Chatbots

6th Practical exercise – Evaluation in Question Answering over Knowledge Graphs

Aleksandr Perevalov

aleksandr.perevalov@hs-anhalt.de

November 17, 2021

Hochschule Anhalt
Anhalt University of Applied Sciences

Plan for today

Plan for today

- Demo Session;

Plan for today

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

Plan for today

- Demo Session;
- Review the task for the Exercise 6;
- Work on the tasks.

Let's start the demo!

Evaluation in Question Answering over Knowledge Graphs

The evaluation is done on two levels:

Evaluation in Question Answering over Knowledge Graphs

The evaluation is done on two levels:

- Component-level – classical Precision, Recall, F1 score (as we used in Exercise 2);

Evaluation in Question Answering over Knowledge Graphs

The evaluation is done on two levels:

- Component-level – classical Precision, Recall, F1 score (as we used in Exercise 2);
- System-level – will discuss it today;

Evaluation in Question Answering over Knowledge Graphs

The evaluation is done on two levels:

- Component-level – classical Precision, Recall, F1 score (as we used in Exercise 2);
- System-level – will discuss it today;
- (User level).

Evaluation in Question Answering over Knowledge Graphs

We divide answers to relevant (correct) or non-relevant (incorrect) – binary decision.

Evaluation in Question Answering over Knowledge Graphs

We divide answers to relevant (correct) or non-relevant (incorrect) – binary decision.
The retrieved answers are compared to "gold standard" or "ground truth" ones.

Evaluation in Question Answering over Knowledge Graphs

We divide answers to relevant (correct) or non-relevant (incorrect) – binary decision.
The retrieved answers are compared to "gold standard" or "ground truth" ones.
An answer is relevant if it addresses an *information need* of a question.

Evaluation in Question Answering over Knowledge Graphs

We divide answers to relevant (correct) or non-relevant (incorrect) – binary decision.
The retrieved answers are compared to "gold standard" or "ground truth" ones.
An answer is relevant if it addresses an *information need* of a question.
The metrics are divided into two classes:

- Evaluation of unranked answer sets;
- Evaluation of ranked answer sets.

Confusion Matrix, Precision, Recall, F1 – unranked

Confusion Matrix, Precision, Recall, F1 – unranked

	Relevant	Nonrelevant
Retrieved	True Positive (TP)	False Positive (FP)
Not retrieved	False Negative (FN)	True Negative (TN)

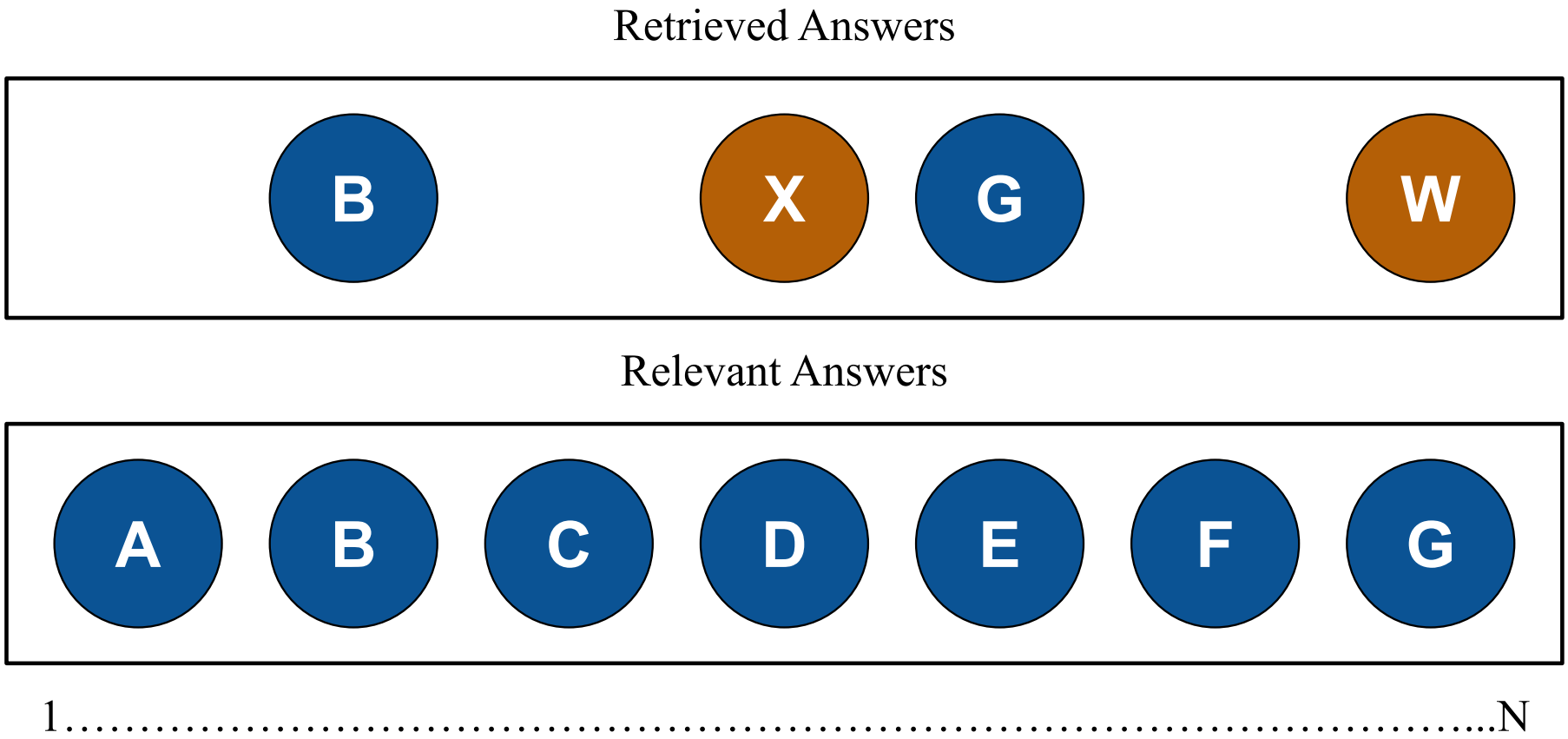
Confusion Matrix, Precision, Recall, F1 – unranked

	Relevant	Nonrelevant
Retrieved	True Positive (TP)	False Positive (FP)
Not retrieved	False Negative (FN)	True Negative (TN)

$$Precision = \frac{TP}{TP+FP} = \frac{\#relevant\ retrieved}{\#retrieved} \quad Recall = \frac{TP}{TP+FP} = \frac{\#relevant\ retrieved}{\#relevant}$$

$$F1 = \frac{2*Precision*Recall}{Precision+Recall}$$

Confusion Matrix, Precision, Recall, F1 – unranked



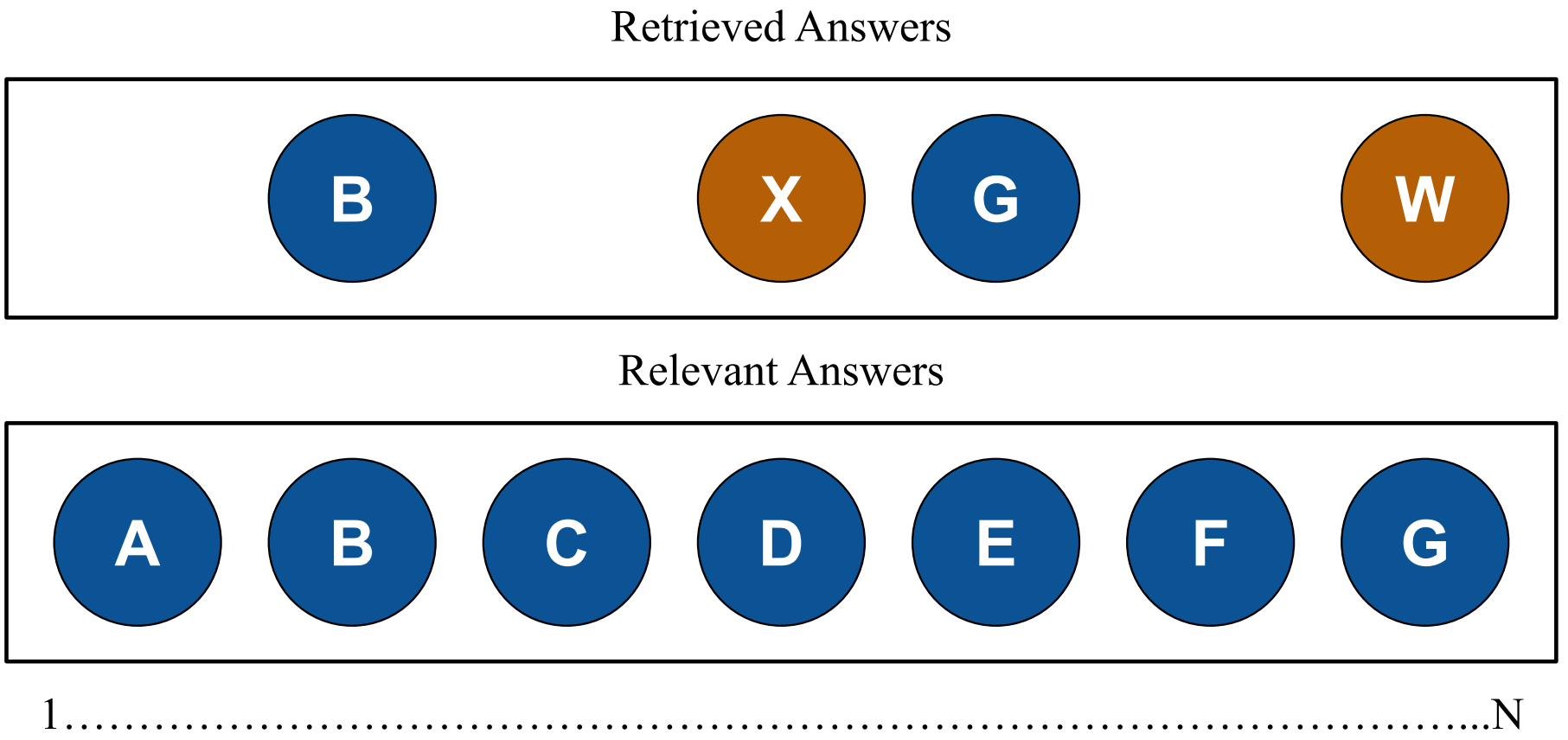
$$Precision = \frac{\#relevant\ retrieved}{\#retrieved} = \frac{2}{4};\ Recall = \frac{\#relevant\ retrieved}{\#relevant} = \frac{2}{7};$$

Precision@k, Recall@k, Reciprocal Rank – ranked

$$Precision@k = \frac{\#relevant\ retrieved\ @k}{\#retrieved\ @k}\ Recall@k = \frac{\#relevant\ retrieved\ @k}{\#relevant};$$

$RR = \frac{1}{rank}$, where *rank* refers to the position of the first relevant answer.

Precision@k, Recall@k, Reciprocal Rank – ranked



$$Precision@5 = \frac{\#relevant\ retrieved\ @5}{\#retrieved\ @5} = \frac{2}{4}\ Recall@5 = \frac{\#relevant\ retrieved\ @5}{\#relevant} = \frac{2}{7};$$

$$RR = \frac{1}{rank} = \frac{1}{1}$$

User utility measures

User utility measures

- Questions:
  - Who is the target audience of the systems?
  - How diverse is the actual audience?
  - Can different user groups use the system with the same efficiency (language, age, ability)?
  - What is the satisfaction level?

User utility measures

- Questions:
  - Who is the target audience of the systems?
  - How diverse is the actual audience?
  - Can different user groups use the system with the same efficiency (language, age, ability)?
  - What is the satisfaction level?
- Metrics:
  - Time per question (TpQ);
  - Rate of return (RoR);
  - Difference between e.g., RoR_{20yrs} *vs* RoR_{60yrs}
  - Survey analysis;
  - Feedback analysis.

Exercise 6 – the task

- Using data from exercise 4, retrieve gold standard answers;

Exercise 6 – the task

- Using data from exercise 4, retrieve gold standard answers;
- Run the questions from the test dataset on your system and collect all the graph ids from Qanary;

Exercise 6 – the task

- Using data from exercise 4, retrieve gold standard answers;
- Run the questions from the test dataset on your system and collect all the graph ids from Qanary;
- Using the graph ids from the previous step to build a SPARQL SELECT query (or several queries) to measure:
  - For each component: Execution time, Confusion Matrix, Precision, Recall, and F1 Score;
  - For the system: Confusion Matrix, Precision, Recall, F1 Score, Precision@k (where k=1,5,10), Reciprocal Rank;

Exercise 6 – the task

- Using data from exercise 4, retrieve gold standard answers;
- Run the questions from the test dataset on your system and collect all the graph ids from Qanary;
- Using the graph ids from the previous step to build a SPARQL SELECT query (or several queries) to measure:
  - For each component: Execution time, Confusion Matrix, Precision, Recall, and F1 Score;
  - For the system: Confusion Matrix, Precision, Recall, F1 Score, Precision@k (where k=1,5,10), Reciprocal Rank;
- Create a .csv file that incorporates the aforementioned metrics for each question (graph id);

Exercise 6 – the task

- Using data from exercise 4, retrieve gold standard answers;
- Run the questions from the test dataset on your system and collect all the graph ids from Qanary;
- Using the graph ids from the previous step to build a SPARQL SELECT query (or several queries) to measure:
  - For each component: Execution time, Confusion Matrix, Precision, Recall, and F1 Score;
  - For the system: Confusion Matrix, Precision, Recall, F1 Score, Precision@k (where k=1,5,10), Reciprocal Rank;
- Create a .csv file that incorporates the aforementioned metrics for each question (graph id);
- Create tables with confusion matrices for all the components and the system.

Exercise 6 – the task

- Using data from exercise 4, retrieve gold standard answers;
- Run the questions from the test dataset on your system and collect all the graph ids from Qanary;
- Using the graph ids from the previous step to build a SPARQL SELECT query (or several queries) to measure:
  - For each component: Execution time, Confusion Matrix, Precision, Recall, and F1 Score;
  - For the system: Confusion Matrix, Precision, Recall, F1 Score, Precision@k (where k=1,5,10), Reciprocal Rank;
- Create a .csv file that incorporates the aforementioned metrics for each question (graph id);
- Create tables with confusion matrices for all the components and the system.
- Optional: Create visualizations based on your .csv report;

Exercise 6 – the task

- Using data from exercise 4, retrieve gold standard answers;
- Run the questions from the test dataset on your system and collect all the graph ids from Qanary;
- Using the graph ids from the previous step to build a SPARQL SELECT query (or several queries) to measure:
  - For each component: Execution time, Confusion Matrix, Precision, Recall, and F1 Score;
  - For the system: Confusion Matrix, Precision, Recall, F1 Score, Precision@k (where k=1,5,10), Reciprocal Rank;
- Create a .csv file that incorporates the aforementioned metrics for each question (graph id);
- Create tables with confusion matrices for all the components and the system.
- Optional: Create visualizations based on your .csv report;
- Optional: Propose a set of metrics to quantify user utility of your system s.t. it is possible to evaluate the user "happiness".

Let's do 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 ODQA system.