Praktikum Wirtschaftsinformatik
26. November 2025
Prof. Dr. Mirjam Minor
M.Sc. Tolga Tel

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

In this Exercise Sheet, we will use Generative AI to extract information from texts by using structured outputs and by annotating an existing descriptive text by using its process model. We will compare both methods of these methods.
At last we will also try to change our initial generation to include annotation. Each exercise is a separate Pipeline.
You can find all the documentation on https://haystack.deepset.ai/

Deadline: 08.12.2025, 10am

Exercise 1 (Structured Outputs) — 9 Points

a. **Read** the Haystack example on the use of structured outputs (https://haystack.deepset.ai/tutorials/28_structured_output_with_loop)
b. **Define** your preferred datastructure for the extraction of tasks. (For example a list of task names). **(1 Point)**
c. **Use** the component "OutputValidator", that is defined in the tutorial in your own code by using your datastructure-template as a variable. **(1 Point)**
d. **Rewrite** the prompt to instruct an LLM to extract **task names** from a BPMN description. **(3 Points)**
e. **Replace** the given generator with the **OllamaChatGenerator** that was used in the last exercise. If Llama3.1 does not produce a valid JSON output, Llama3.2 might help. **(1 Point)**
f. **Run** this pipeline for 3 different **descriptive texts** or BPMN models and verify the results using precision and recall scores **manually** (Hint: Sometimes increasing "max_runs_per_component" isn't as heplful as just rerunning the pipeline. When testing, Llama3.2 felt more consistent in producing these structured outputs). Present the scores in your presentation. **(3 Points)**

Exercise 2 (Annotation) — 6 Points

a. **Create** a Pipeline for a plain LLM without a document store or retrieval. It should contain following components: **(3 Points)**
    1. A (Chat)PromptBuilder that receives a query containing a BPMN model and its textual description (no examples). It instructs the LLM to annotate the textual description with BPMN tags like <bpmn:task>Task name</bpmn:task>.
    2. A generator that works with the BPMN model.
b. **Run** this pipeline for 3 different BPMN-text pairs. Verify the results (Is each BPMN element tagged correctly?) using precision and recall scores **manually.** Present the scores in your presentation. **(3 Points**)

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Exercise 3 (Annotation-Generation) **5+3* Points**

a. **Create** a Pipeline similar to the last exercise sheet (Just one generator) with a document store containing BPMN-text pairs. It should contain following components: **(3 Points)**
   1. A (Chat)PromptBuilder that receives a query containing a BPMN model and an **example BPMN-text pair**. It instructs the LLM to create a textual description with BPMN tags like <bpmn:task>Task name</bpmn:task>.
   2. A matching generator.

b. **Run** this pipeline for 2 different BPMN models. Verify the results using precision and recall scores **manually**. Present the scores in your presentation. **(2 Points)**

c. **Replace** the BPMN-text pairs in the document store with BPMN-annotated-text pairs (already containing tags like <bpmn:task>Task name</bpmn:task> in the descriptive text) and **rerun** this pipeline. Compare your results with the results from exercise 3b) using precision and recall scores **manually**. Present the scores in your presentation. **(3* Points)**