



Software Practicals

Winter Semester 2025/26

Data Science Group
Heidelberg University
October 15, 2025

Slides Online



The slides are available on our webpage
<https://ds.ifi.uni-heidelberg.de/teaching/>

Organization

Outline

- Overview of topics (today)
 - Send application for a topic until **Monday, October 20, 1pm**
 - Assignment of topics by mid next week
- First milestone (before Christmas break)
 - Prototype / part of software
 - Summary of research (literature and related systems/tools)
 - Further milestones in agreement with supervisor
- End of practical (mid February)
 - Code has to be hosted in local GitLab of the data science group
 - Presentation / demo of practical and software (10-12 minutes)
 - Report / documentation as local GitLab as README.md

Application

- Apply directly to supervisor via mail
 - Program of study, semester of study, matriculation number
 - List relevant course experience, including course grades
 - List other experience:
 - Side projects you are working on
 - “Anwendungsgebiet” / Application Field
 - Job and project experience
 - Send your tentative schedule and milestones for the practical
 - Group work only possible for some projects!
- It is recommended to apply for multiple topics (e.g., “top-3 list”)

Application is binding!

Don't apply if you don't want to do the practical!

Deadlines

- In general: biweekly meetings with supervisor
- Presentation: mid of February 2026
- Report & GitLab upload: mid of February 2026
- No extension possible

Not finished = failed (grade 5,0)!

Assessment

- Credit points (Leistungspunkte)
 - Beginners Practical (IAP, 2 CP + 4 FÜK) [Bachelor students]
 - workload: 180 h (~1 ½ days/week)
 - Advanced Practical (IFP, 8 CP)
 - workload: 240 h (~2 days/week)
- Grading based on
 - code (readability, structure, functionality; code in local GitLab)
 - documentation (README.md, code comments, documentation in GitLab)
 - commitment and self-reliance
 - cool ideas!!
- IMPORTANT
 - talk to / communicate with your advisor (at least biweekly meetings)

Supervisors

- Michael Gertz (MG)
gertz@informatik.uni-heidelberg.de
- Marina Walther (MW)
walther@informatik.uni-heidelberg.de
- Ashish Chouhan (AC)
chouhan@informatik.uni-heidelberg.de
- Nicolas Reuter (NR)
reuter@informatik.uni-heidelberg.de

Project Topics

AP = Advanced Topic

BP = Beginners Topic (for BSc students only)

Overview of Topics

1. Medical Datasets in German, **BP/AP** (Walther)
2. Construction-Law Agent, **Team (2) AP** (Walther)
3. KG Construction from Medical Documents, **BP/AP** (Chouhan)
4. Graph Retrieval Augmented Generation, **AP** (Chouhan)
5. Table Retrieval for Question Answering, **AP** (Reuter)
6. Building an Interactive Learning Platform for SQL, **AP** (Reuter)
7. Dataset Creation for Table-based QA, **BP/AP** (Reuter)
8. Medical Knowledge Graph Extraction, **Team (2) AP** (Gertz)
9. Crawler for DocCheck Flexikon, **Team (2) BP/AP** (Gertz)
10. Generating German Medical Texts, **BP/AP** (Gertz)

BP/AP: Medical Datasets in German (MW)

Given:

- Commonly used medical datasets, e.g., [MIMICS-III](#), [MedQA](#)

Tasks:

- Collect the datasets and translate into German
- Evaluate the quality of translations with methods to compare **English** and **German** experiments

Subtasks:

- Find suitable analyses, metrics, and visualizations
- Store dataset + results

Rec. Languages / Tools: Python, [OpenSearch](#), [LangChain](#), Viz-tools (e.g., [Matplotlib](#), [Streamlit](#))



AP: Construction-Law Agent (min. 2) (MW)

Given:

- Seed list of relevant data sources, e.g., Baugesetzbuch ([BauG](#))

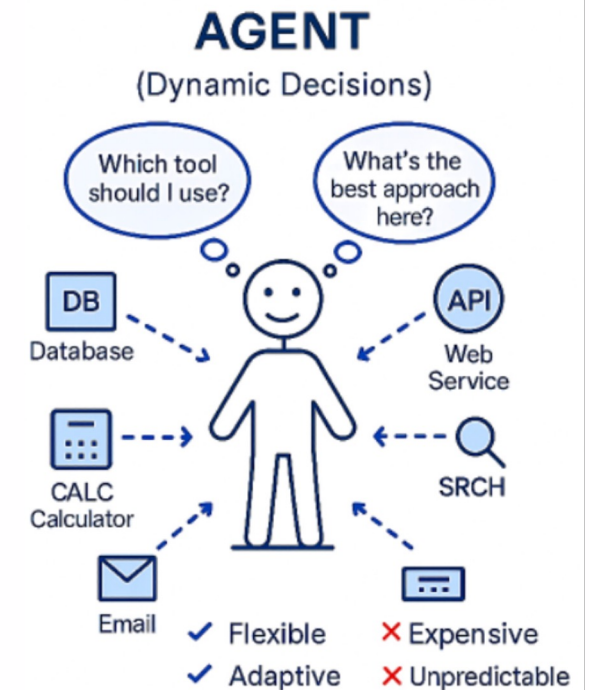
Tasks:

- Build an AI-based agent that answers questions about construction law in Germany

Subtasks:

- Learn about chatbots vs. agents vs. workflows
- Build a UI to ask questions
- Develop project use-case driven (real-world requirements analysis)

Languages / Tools: Python, [Langchain](#), Web framework of your choice, e.g., [Streamlit](#) or [Svelte](#), [PostgreSQL](#)



BP/AP: KG Construction from Medical Documents (AC)

Given:

- Corpus of medical documents from [PubMed](#)

Tasks:

- Construct PubMed Knowledge Graph (KG) from abstract text
- Evaluate resulting knowledge graph

Subtasks:

- Process text (entity recognition, co-reference resolution, ...) and ingest data in graph database, for example, [Neo4j](#)
- Analyze, explore, and visualize knowledge graph

Languages / Tools:

- Python, [OpenSearch](#), [Neo4j](#), [LangChain](#), ...



AP: Graph Retrieval Augmented Generation (AC)

Given:

- Legal documents from [EUR-Lex](#)
- Existing [EuroVoc](#) vocabulary associated with legal documents
- Question-Answer pairs for evaluation

Tasks:

- Compare [vanilla RAG vs Graph RAG](#)

Subtasks:

- Collect information and realize backend to store information using [OpenSearch](#)
- Retrieved information refinement leveraging vocabulary

Languages / Tools:

- Python, [OpenSearch](#), [LangChain](#), ...

 Concept scheme	
6606 energy policy	
Version	4.21
Concept scheme URI:	http://eurovoc.europa.eu/100263
Type of dataset:	Thesaurus

[energy policy](#)

NT1	energy research
	RT scientific research [6416]
NT1	energy crisis
NT1	decommissioning of power stations
	RT nuclear power station [6621]



AP: Table Retrieval for Question Answering (NR)

Given:

- Set of HTML documents containing tables and question-table pairs

Tasks:

- Evaluate how different retrieval strategies (scoring function, chunking methods, table formats) affect retrieval performance

Subtasks:

- Build retrieval pipeline to select relevant tables based on questions
- Try different formats for indexing (e.g., HTML, CSV, natural language)
- Evaluate retrieval quality using standard metrics (Recall, Precision, ...)

Languages / Tools:

- Python, [LangChain](#) or [LlamaIndex](#), [OpenSearch](#), ...

	Three Months Ended		
	December 30, 2023	December 31, 2022	Change
Net sales by category:			
iPhone	\$ 69,702	\$ 65,775	6 %
Mac	7,780	7,735	1 %
iPad	7,023	9,396	(25)%
Wearables, Home and Accessories	11,953	13,482	(11)%
Services	23,117	20,766	11 %
Total net sales	\$ 119,575	\$ 117,154	2 %



AP: Building an Interactive Learning Platform for SQL (NR)

Given:

- Sample databases with SQL Q&A pairs

Tasks:

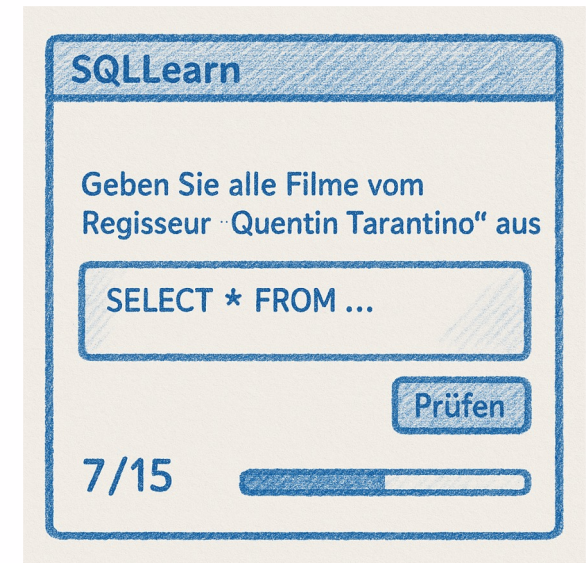
- Build a web platform for learning SQL with real-time query evaluation, instant feedback and AI guidance

Subtasks:

- Frontend: Create a clear and intuitive web interface
- Backend: Integrate the databases and AI assistance
- Extend predefined questions with dynamically generated ones
- (Optional:) Add gamification elements

Languages / Tools:

- Python, SQL, Docker, PostgreSQL, FastAPI, Svelte or similar, ...



BP/AP: Dataset Creation for Table-based QA (NR)

Given:

- Example dataset from the financial domain
- PubTables as a starting point for table documents

Task:

- Create a dataset of non-financial full-text documents and corresponding table-based questions, including their supporting tables and answers

Subtasks:

- Collect suitable documents from public sources
- Extract and clean tables and text (e.g., PDF parsing)
- Create questions and answers about the tables

Languages / Tools:

- Python, Docling, ...



Team (2) AP: Medical Knowledge Graph Extraction – GI (MG)

Given:



- Content of the website <https://www.gesundheitsinformation.de/>; with topics on health, human body, examinations, glossary

Tasks:

- Develop a data model capturing all information and linkages between pieces of information
- Methods to extract text for a knowledge graph



Subtasks:

- Design different models for capturing fine granular data from text
- Try different LLMs for entity and relationship extraction

Languages / Tools:

- Python; Large Language Models; [Neo4j](#)

Team (2) BP/AP: Crawler for DocCheck Flexikon (MG)



Given:

- Content of the website <https://flexikon.doccheck.com/de/Hauptseite>; with articles on health, treatments, medication, ...

Tasks:

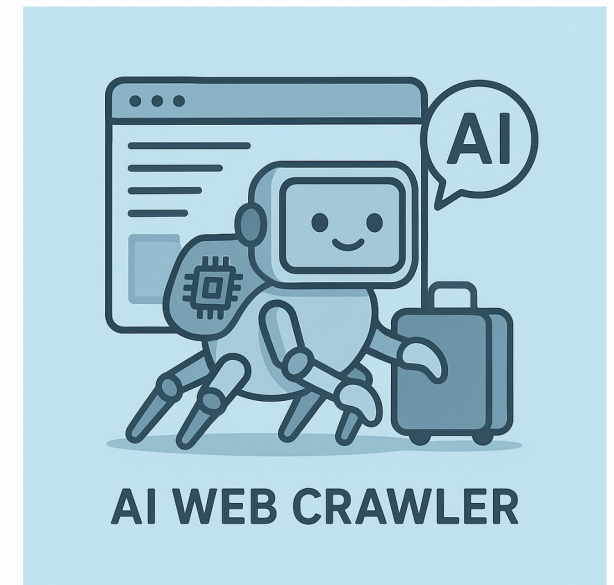
- Develop a data model describing information content of website
- Methods to crawl website (in an **agentic** way)

Subtasks:

- Focus on a single medical entity, e.g., diabetes

Languages / Tools:

- Python; [OpenSearch](#); [Crawl4AI](#)



BP/AP: Generating German Medical Texts (MG)

Given:

- Medical terminologies such as ICD-10-GM
- Medical articles, e.g., blog posts that are mostly about symptoms



Tasks:

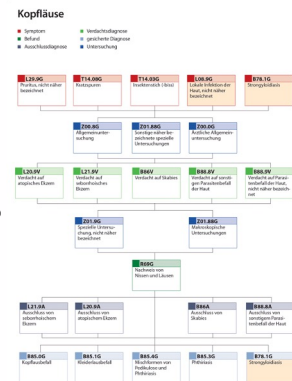
- Use open source Large Language Models (LLMs) to generate concept centric (ICD-10) conversational texts between patients and doctors
- Develop different prompting strategies

Subtasks:

- Model diagnostic conversations between patients and doctors

Languages / Tools:

- Open source LLMs, e.g., Gemini, Mistral, LLama; Python



Slides Online



The slides are available on our webpage
<https://ds.ifi.uni-heidelberg.de/teaching/>