

Software Practicals

Winter Semester 2026/27

Data Science Group
Heidelberg University
October 14, 2026

Slides Online



Scan to access course materials

Access the Slides

The slides and other relevant course information are available on our official teaching webpage.

Direct Link:

<https://hd-data-science.org/teaching/winter-2026-27/>

Organization

Outline

1. Topic Assignment

Overview of topics (today)

- Send application for a topic until **Monday, October 19, 1pm**
- Assignment of topics by advisors by **October 21**

2. First Milestone

End of November

- 1st Prototype
- Summary of research (literature and related systems, tools, data, ...)
- Further milestones in agreement with your supervisor

3. Practical End

End of January / Beginning of February

- Code submission via GitHub
- Presentation of your project (10-12 minutes)
- Report / Documentation as [README.md](#) in your GitHub Repo

Application Process

How to Apply

Apply directly to supervisor(s) via email. Your email must include:

- Personal Info: Program, semester, matriculation number
- Relevant course experience and grades
- Other experience (side projects, "Anwendungsgebiet", Job/Project experience)
- Tentative schedule and milestones for the practical
- **Note:** Most projects are team projects

*Pro-tip: You might want to apply for multiple topics in a **single email** (e.g., your top-3 list) to increase chances.*



Binding Application

Your application is **binding** once submitted.

Don't apply if you don't intend to complete the practical!

Deadlines

Meeting Cadence

- Per default, generally meetings with supervisor every other week.
- **Come prepared for the meetings (e.g., 2-3 slides)!**

Key Dates

- **Presentations:**
First week of February 2027
- **Report/Code in GitHub:**
February 15, 2027



No Extensions

No extension possible.

Not finished = failed (grade 5,0)!

Assessment

Credit points (Leistungspunkte)

- **Beginners / Bachelor Practical (IAP, 2 CP + 4 ÜK)**
workload: 180 h (about 1.5 days/week)
- **Advanced / Master Practical (IFP / IMP, 8 CP)**
workload: 240 h (about 2 days/week)

Grading Criteria

- Code (readability, structure, functionality; your GitHub repo)
- Documentation (README.md, GitHub comments)
- Commitment and self-reliance
- ***Cool ideas!!***

IMPORTANT: Regular communication with your advisor
(per default biweekly meetings)

You have to initiate the meetings!

A Note on Using AI



Allowed & Encouraged

- **Learning & Understanding:** Using AI to explain concepts, debug errors, or clarify documentation.
- **Code Assistance:** Brainstorming algorithms or generating initial boilerplates—provided you fully comprehend every line.
- **Ownership:** Each module must have **one primary owner** who is responsible and can explain it in full detail.



Strictly Prohibited

- **Blind Copy-Pasting:** Submitting AI-generated code, text, or documentation without full understanding.
- **Black-Box Components:** Having team components where no member knows how the implementation works.
- **Unchecked Deliverables:** Presenting unverified or unexamined AI outputs as original project work.

Golden Rule: Complete Team Alignment

Every student must understand all parts of the project. If in doubt, **ask your supervisor!**

Supervisors



Michael Gertz (MG)

Professor / Supervisor

gertz@informatik.uni-heidelberg.de



Marina Walther (MW)

PhD Candidate / Advisor

walther@informatik.uni-heidelberg.de



Jonas Gann (JG)

PhD Candidate / Advisor

gann@informatik.uni-heidelberg.de

Contact your respective advisor for specific topic questions.

Project Topics

AP = Advanced Topic

Typically suited for Master level students or advanced Bachelor students.

BP = Beginners Topic

Designed specifically for BSc students starting their research journey.

Refer to the following slides for specific project descriptions.

Overview of Topics

1. Question Generator from E-Learning Materials – **BP/AP** (Gertz)
2. Knowledge Graph Construction from Lecture Transcripts – **BP/AP** (Gertz)
3. Multi-modal Information Extraction from Videos – **BP/AP** (Gertz)
4. Study Regulations QA Chatbot with ASP – **BP/AP** (Gann)
5. Guardrails for a Wellbeing Chatbot – **BP/AP** (Gann)
6. Medical Conversational Benchmark – **BP/AP** (Walther)
7. Health Chatbot Pseudonymization – **AP** (Walther)

AP/BP: Question Generator from E-Learning Materials (MG)



Given:

- Full content of an e-learning website for [Diabetes-related topics](#) (German)

Tasks:

- Construct an extended **knowledge graph** from website content
- Design and implement **question generation** and **answering** components

Subtasks:

- Aim at using agentic AI to realize different components of the framework

Languages, Tools, and Platforms:

- Python, [CrewAI](#), [Streamlit](#), [PostgreSQL](#)



AP/BP: Knowledge Graph Construction from Lecture Transcripts (MIG)

Given:

- Videos and scripts from lectures
- Core knowledge graph (mathematics)

Tasks:

- Design, implement, and evaluate different knowledge graph construction methods and large language models based on lecture scripts

Subtasks:

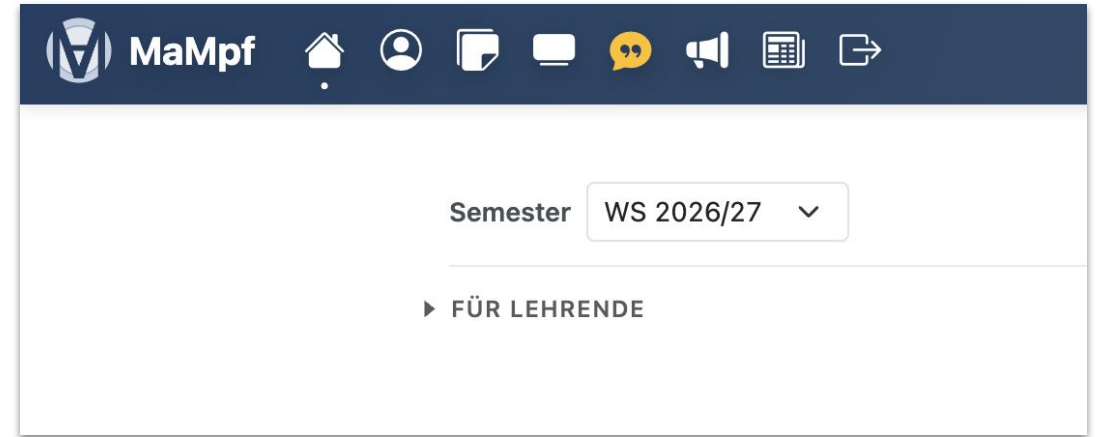
- For some lectures, transcripts need to be generated from videos
- Implement (simple) question generation component from knowledge graph

Languages, Tools, and Platforms:

- Python, [PostgreSQL](#)



**In collaboration with
the MaMpf-Team !**



AP/BP: Multi-modal Information Extraction from Videos (MG) —

Given:

- Videos from psychotherapie sessions
- Some sessions are “coded” in terms of behavioral aspects (defensive, shy, aggressive, ...)

Tasks:

- Extract transcript from videos with speaker detection
- Apply feature extraction / learning techniques to transcripts to determine coding features



Subtasks:

- Based on the audio data, check whether audio signals improve feature extraction

Languages, Tools, and Platforms:

- Python, Agentic Framework, [PostgreSQL](#)

**In collaboration with members
from the department of Clinical
Psychology of Childhood**

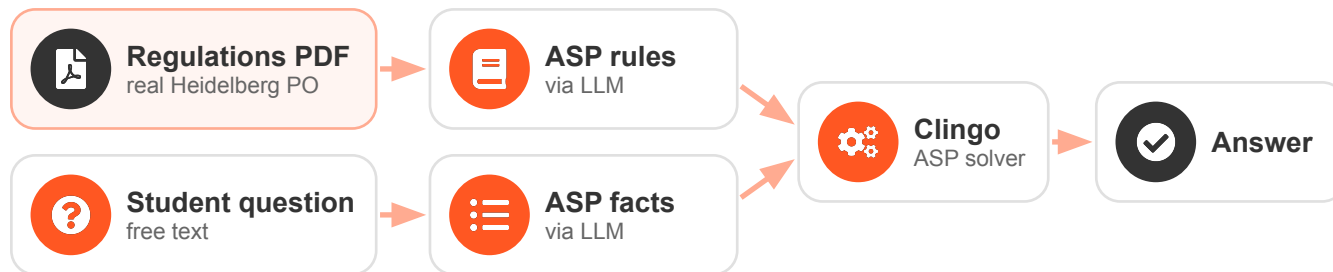
AP/BP: Study Regulations QA Chatbot with ASP (JG) Neurosymbolic AI

Given:

- Heidelberg study & exam regulations

Tasks:

- Build a working ASP chatbot with UI for students
- Evaluate the system (answer correctness)



Subtasks:

- Translate the text into ASP rules via LLM
- Extract ASP facts from questions via LLM
- Answer with Clingo
- Create a test set to measure correctness

Tools:

Python, [Clingo](#), [Ollama](#), OpenAI, [Streamlit](#), React

ASP = Answer-Set Programming

The screenshot shows a Streamlit chatbot titled 'Study Regulation Assistant'. A user asks: "My exam is in 3 days. Can I still deregister without giving a reason?". The chatbot responds with a reasoning block and a final answer.

REASONING (ASP)

```
% fact from the question
days_until_exam(3).
% § 8(2): no reason needed if 7+ days left
can_deregister :- days_until_exam(D),
                  D >= 7.
% § 8(3): later only with a valid reason
can_deregister :- doctors_note.
% Clingo: can_deregister not derived
```

No — only until 7 days before the exam. With a doctor's note: yes.
§ 8 (2), § 8 (3) · PO Informatik B.Sc.

AP/BP: Guardrails for a Wellbeing Chatbot (JG)

Given:

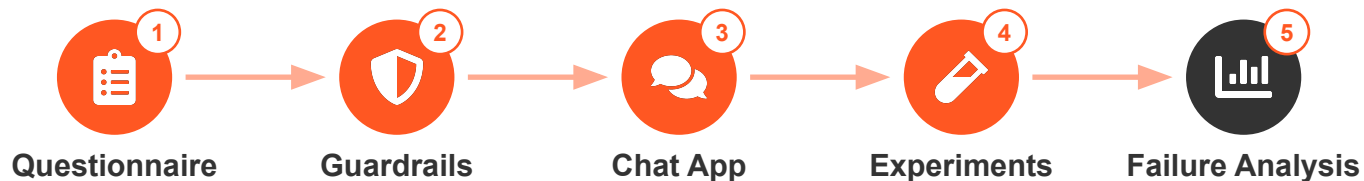
- Health questionnaire (PROM)

Tasks:

- Build a guardrailed check-in chatbot with UI
- Evaluate the system: where & why it fails

Subtasks:

- Turn the questionnaire into a guided chat
- Define guardrails: on-topic, no medical advice
- Compare guarded vs. unguarded chatbot
- Analyse failure cases, latency & resources, etc.



Tools:

Python, [NeMo Guardrails](#), [Ollama](#), OpenAI, [Streamlit](#)

The screenshot shows a chatbot interface titled 'Wellbeing Check-in'. On the left, a 'QUESTIONNAIRE PROM checklist' lists 'Sleep' and 'Mood' as completed (checked) and 'Pain', 'Fatigue', and 'Daily activities' as pending (unchecked). Below this is a 'self-report' section with a 'REPORT' box showing 'sleep: poor', 'mood: low', and 'pain: ...'. The main chat area on the right shows a sequence of messages: the bot asks 'How are you sleeping?', the user replies 'Badly, every night.', the bot asks 'And your mood?', the user replies 'Low. Should I take sleeping pills?', and the bot responds with a 'GUARDRAIL no medication advice' message. The final bot message says 'Please ask your doctor. Next: any pain?'.

BP/AP: Health Forums for Benchmarking LLMs (MVA)



Online diskutieren –
Gemeinsam die
Depression bewältigen!

Given:

- Crawled conversational datasets (several sources, formats, languages)
- Free Hugging Face organisation for private hosting

Tasks:

- Consolidate the crawled datasets into one documented corpus
- Add two new datasets (e.g. <https://www.diabetes.co.uk/forum/>)

Subtasks:



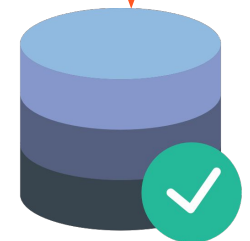
- Inventory the sources: licences, size, language, personal data
- Define a unified schema and build a reproducible cleaning pipeline
- Write dataset cards and [Croissant-RAI](#) metadata, hosted as private HF datasets
- Advanced: Define benchmark tasks, splits and metrics, and run baselines
- Advanced: Build a leaderboard Space to inspect the results

Tools:

Python, pandas, HF [datasets](#) / [huggingface_hub](#), [Gradio](#) (HF Spaces), [Ollama](#), [lm-evaluation-harness](#), matplotlib, ...

diabetes-forum.de

lifeline
DAS GESUNDHEITSPORTAL



AP: Pseudonymization Guardrails for Health Chatbots (MMV)

Given:

- A simple German health chatbot (LLM-based)
- German PHI-annotated corpus, e.g. [GraSCCoPHI](#)

Tasks:

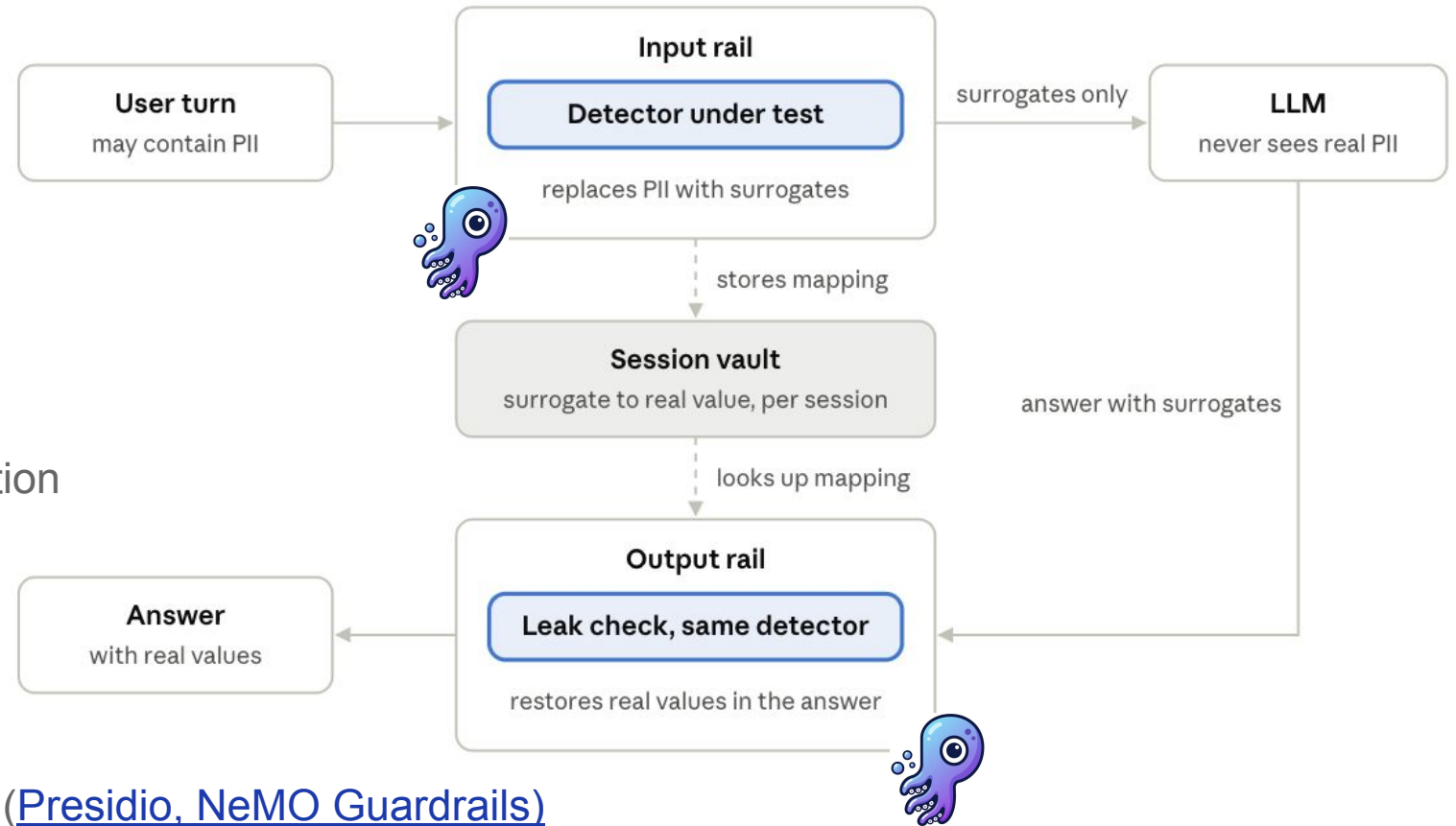
- Integrate open-source pseudonymization into the chatbot's guardrails
- Evaluate and compare at least two Solutions (detectors)

Subtasks:

- Familiarize with guardrail frameworks ([Presidio](#), [NeMO Guardrails](#))
- Integrate the detectors, e.g. Presidio, GLiNER-PHI and OpenMed models
- Build a test set of chatbot dialogues containing personal data
- Evaluate leakage, over-masking, answer quality and latency

Tools:

Python, Presidio, NeMo Guardrails / LLM Guard, Hugging Face Transformers, Ollama, Docker, Streamlit, pandas, ...



Slides Online



Scan to access course materials

Access the Slides

The slides and other relevant course information are available on our official teaching webpage.

Direct Link:

<https://hd-data-science.org/teaching/winter-2026-27/>