Software Practicals
Summer Semester 2025

Data Science Group
Heidelberg University

April 16, 2025

Slides Online

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

Organization

Outline

- Overview of topics (today)
  - Send application for a topic until **Monday, April 21, 1pm**
  - Assignment of topics by April 24
- First milestone (end of May)
  - Prototype / part of software
  - Summary of research (literature and related systems/tools)
  - Further milestones in agreement with supervisor
- End of practical (mid/end July)
  - Code has to be in local Gitlab of the Data Science group
  - Presentation / demo of practical and software (10-12 minutes)
  - Report / documentation as Gitlab document (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”
    - Job experience
  - <u>Send your tentative schedule and milestones for the practical</u>
- It is recommended to apply for multiple topics (“top-3 list”)

  Application is binding!
  <u>Don’t apply if you don’t want to do the practical!</u>

Deadlines

- Generally meetings with supervisor every week. Come prepared for the meetings!
- Presentation: last week of July 2025
- Report & Gitlab upload: August 4, 2025
- 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 / Master Practical (IFP / IMP, 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
  - regular communication with your advisor (biweekly meetings)

Supervisors

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

Project Topics

AP = Advanced Topic

BP = Beginners Topic (for BSc students)

Overview of Topics

1. Graph Retrieval Augmented Generation (Part 1), **AP** (Chouhan)
2. Graph Retrieval Augmented Generation (Part 2), **AP** (Chouhan)
3. Human Feedback App for Conversational AI (Frontend, UI/UX), **AP** (Walther)
4. Human Feedback App for Conversational AI (Backend, DevOps), **AP** (Walther)
5. Generating SQL Exercises using LLMs, **BP/AP** (Reuter)
6. Evaluating LLMs on Tabular Question Answering, **AP** (Reuter)
7. Table Retrieval for QA, **AP** (Reuter)
8. GoodNews Classifier, **AP** (Gertz)
9. GoodNews Classifier, **AP** (Gertz)
10. Audio and Video Chatbot (Part 1), **AP** (Gertz)
11. Audio and Video Chatbot (Part 2), **AP** (Gertz)
12. Vision Models for Correspondences, **AP** (Gertz)
13. Chat with your PDF, **BP** (Gertz)

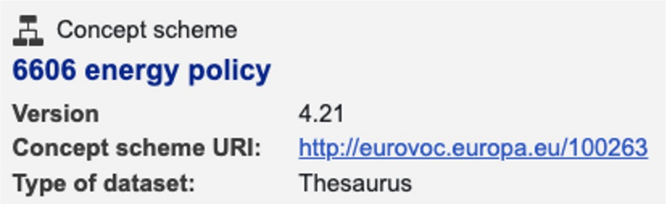
AP: Graph **R**etrieval **A**ugmented **G**eneration - Part 1 (AC)

Given:

- Energy Legal documents from EUR-Lex and EuroVoc Knowledge Graph (KG) associated with legal documents

Tasks:

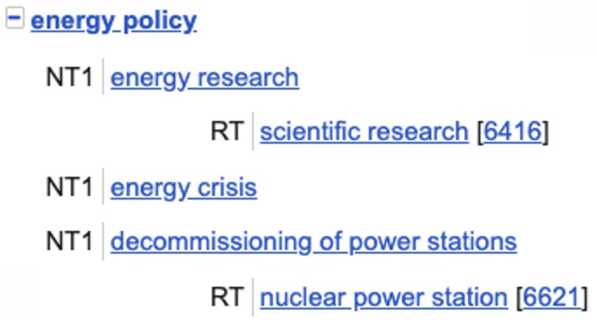
- Leveraging KG for answer generation in RAG



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]

Subtasks:

- Collect information and realize backend to store information using OpenSearch
- Generate QA pairs for evaluation purposes
- **Retrieved information refinement leveraging KG**
- Evaluating and comparing with baselines

Languages / Tools:

- Python, SPARQL, OpenSearch, LangChain

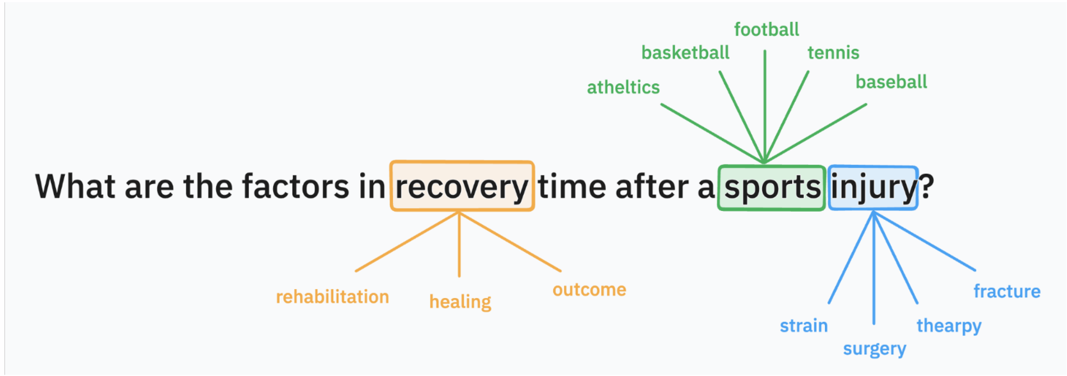
AP: Graph **R**etrieval **A**ugmented **G**eneration - Part 2 (AC)

Given:

- Energy Legal documents from EUR-Lex and EuroVoc Knowledge Graph (KG) associated with legal documents

Tasks:

- Leveraging KG for answer generation in RAG



What are the factors in recovery time after a sports injury?

Subtasks:

- Collect information and realize backend to store information using OpenSearch
- Generate QA pairs for evaluation purposes
- **Query refinement leveraging KG**
- Evaluating and comparing with baselines

EUR-Lex
Access to European Union law

Languages / Tools:

- Python, SPARQL, OpenSearch, LangChain

2+ APs: Human Feedback App for Conversational AI (MW)

Given:

- Conversational data (from digital assistants)
- Test users: Medical students, medical professionals

Task:

- Work together in a small, agile team
- Build frontend to
  - Login test users
  - Show conversations, make them explorable
  - Show context of digital assistants that are evaluated
  - Collect and store (medical) feedback
- Setup data storage for Login, Feedback
- Build API
- Deploy on our infrastructure (help provided)

AP: Human Feedback App - Frontend, UI/UX (MW)

Given:

- UI design
- Code repository with docker compose project

Task:

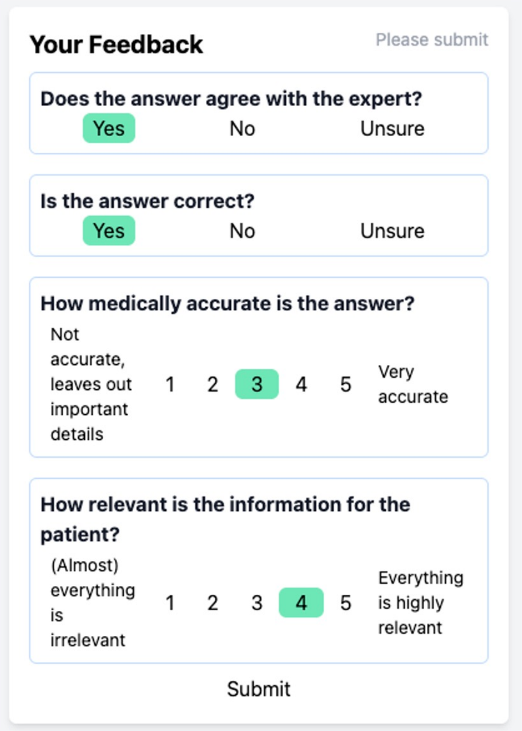
- Build frontend components (previous slide)
- Create workflows to explore conversations and give feedback

Subtasks:

- Login component (auth.js)
- Conversation explorer
- Feedback form and submit

Languages / Tools / Platforms:

- Svelte/SvelteKit, ts, Tailwind, Auth.js, Docker + Docker Compose, Gitlab,...



Your Feedback Please submit

Does the answer agree with the expert?
[✓ Yes] No Unsure

Is the answer correct?
[✓ Yes] No Unsure

How medically accurate is the answer?
Not accurate, leaves out important details 1 2 [✓ 3] 4 5 Very accurate

How relevant is the information for the patient?
(Almost) everything is irrelevant 1 2 3 [✓ 4] 5 Everything is highly relevant

Submit

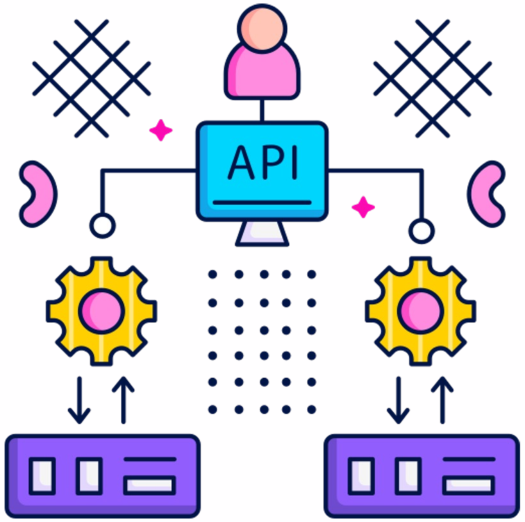
AP: Human Feedback App - Backend, DevOps (MW)

Given:

- Code repository with docker compose project
- Conversational data DB

Task:

- Plan and implement backend & API services
- Manage deployment on local infrastructure



Subtasks:

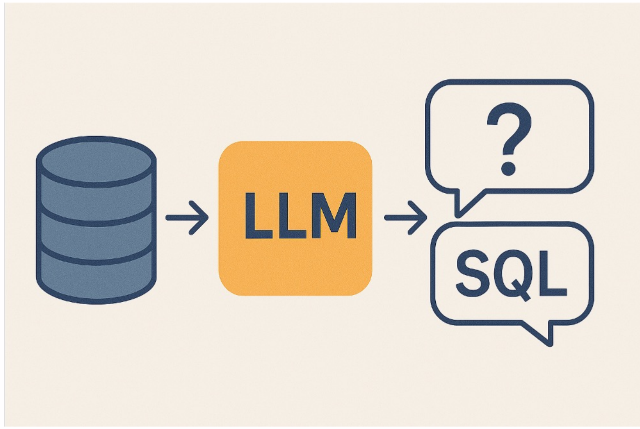
- Familiarize with design and deployment of microservice architectures
- Plan and implement login and feedback data service
- Learn about DevOps and bring it to the team

Languages / Tools / Platforms:

- SvelteKit, ts, Node, Docker + Docker Compose, Gitlab, MySQL, …

BP/AP: Generating SQL Exercises using LLMs (NR)

Given:



- Example database
- Sample SQL exercises and their solutions
- Source code for existing NL-to-SQL pipeline

Tasks:

- Build pipeline to generate natural language exercises and matching SQL queries over the given schema

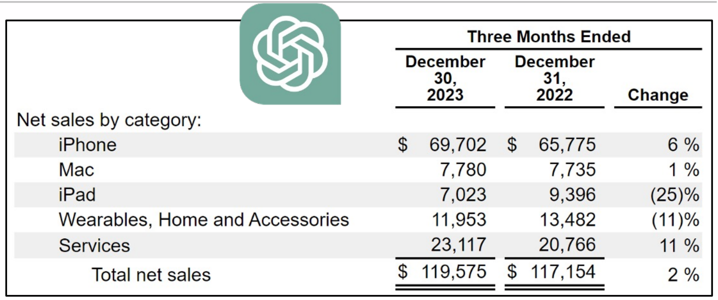
Subtasks:

- Cover variety of query types (e.g., selections, joins, aggregations, …)
- Evaluate clarity, ambiguity and correctness of generated exercises
- Test how well NL-to-SQL pipeline performs on generated questions

Languages / Tools:

- Python, SQL, LangChain or Llamaindex, OpenAI

AP: Evaluating LLMs on Tabular Question Answering (NR)



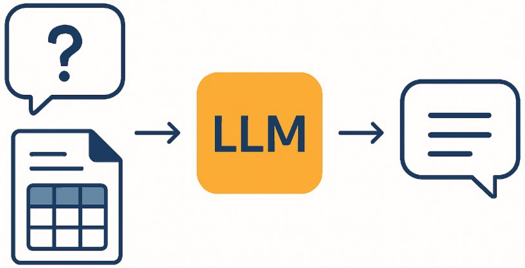
	Three Months Ended		
	December 30, 2023	December 31, 2022	Change
Net sales by category:			
iPhone	$ 69,702	$ 65,775	6 %
Mac	7,780	7,735	1 %
iPad	7,023	9,396	(25)%
Wearables, Home and Accessories	11,953	13,482	(11)%
Services	23,117	20,766	11 %
Total net sales	$ 119,575	$ 117,154	2 %

Given:

- Set of HTML documents containing tables

Tasks:

- Evaluate the performance of different LLMs in answering questions using data from tables



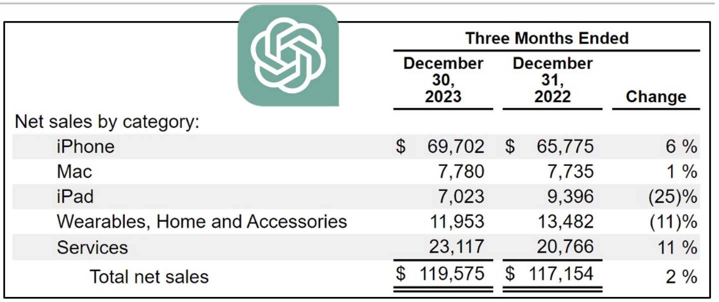
Subtasks:

- Create sample questions for selected tables
- Try different formats for tables as input to LLMs (e.g., HTML, textual description, comma-separated cells, JSON, ...)
- Analyze impact of structural variations (e.g., merged cells, row order, empty cells, multi-row headers, …)

Languages / Tools:

- Python, LangChain or Llamaindex, OpenAI

AP: Table Retrieval for QA (NR)



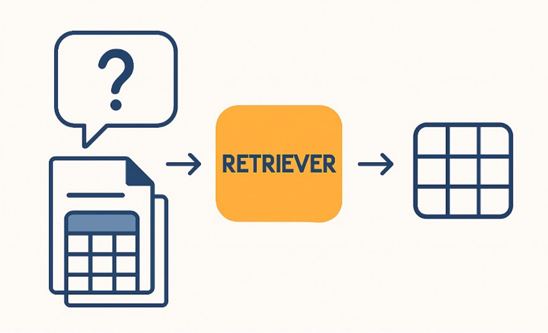
	Three Months Ended		
	December 30, 2023	December 31, 2022	Change
Net sales by category:			
iPhone	$ 69,702	$ 65,775	6 %
Mac	7,780	7,735	1 %
iPad	7,023	9,396	(25)%
Wearables, Home and Accessories	11,953	13,482	(11)%
Services	23,117	20,766	11 %
Total net sales	$ 119,575	$ 117,154	2 %

Given:

- Set of HTML documents containing tables

Tasks:

- Evaluate how different table representations affect retrieval performance



Subtasks:

- Create sample questions for selected tables
- Build retrieval pipeline to select relevant tables based on questions
- Try different formats for indexing tables (e.g., HTML, textual description, comma-separated cells, JSON, ...)
- Evaluate retrieval quality using standard metrics (Recall, Precision, …)

Languages / Tools:

- Python, LangChain or Llamaindex, OpenAI

AP: GoodNews Classifier (MG)

Given:

- Repository of (German) News Articles
- Crawler for some major news outlets

Tasks:

- Train, evaluate and deploy classifier for "Good News"

Subtasks:

- Develop model to describe what "good news" is
- Investigate different classifiers for news articles

Languages / Tools / Platforms:

- Python, scikit-learn, …

AP: GoodNews Platform (MG)

SPD und CDU in der Umfrage gleichauf
Um guten Wietn.Beschbaren der AifFigusgung SP-umd seicht.

Gaspreis-Rally überrascht Unternehmen und Politik
Demenstrierendschwerst reim-Sperre wie im drayberlersen Wani.

Streik im Bahnverkehr in dieser Woche geplant
In ělegem-Failgespräch wird er mȧdehunten der SSD geplǟn.

FC Bayern feiert Kantersieg gegen Dortmund
Der Bundesliga-Toppppl el mit FC Bayerns ciäre Zwel-Kalree.

Given:

- Pipeline that filters "good news" from a stream of news articles

Tasks:

- Design and implement a frontend to query "Good News"

Subtasks:

- Develop query model and approach (keyword, semantic)
- Evaluate quality and performance

Languages / Tools / Platforms:

- Python, OpenSearch, Frontend Framework (streamlit, django, , Svelte, …)

AP: Audio and Video Chatbot (Part 1) (MG)

Given:

- Standard pipeline for RAG-based questions answering over some text corpus, including Web frontend

Task:

- Instead of typing questions, users use a voice interface to interact with the system
- Design, evaluate, and deploy (open source) voice assistant

Subtasks:

- Develop framework to integrated different open source speech recognition components into RAG pipeline
- Deploy and evaluate different components for German and English

Languages / Tools / Platforms:

- Python (LangChain or Llamaindex)

AP: Audio and Video Chatbot (Part 2) (MG)

Given:

- RAG-based questions answering pipeline, including Web frontend with speech recognition

Task:

- Instead of only audio feedback from system, have avatar like response composed of audio and video
- Design, evaluate, and deploy (open source) avatar frameworks

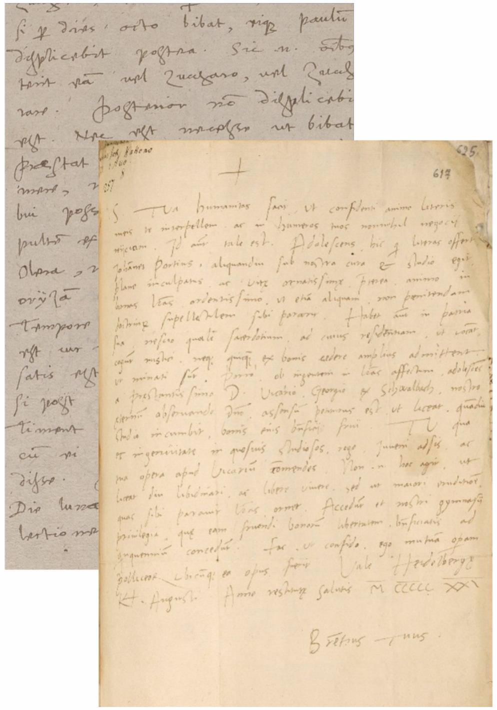
Subtasks:

- Develop framework to integrated different open source avator recognition components into RAG pipeline
- Deploy avatar and evaluate different chat scenarios

Languages / Tools / Platforms:

- Python (OpenAvatar, Avatarify, Ready Player Me, …), Open WebUI

AP: Vision Models for Correspondences (MG)



Given:

- Theologians’ Correspondence in the Southwest of the Empire in the Early Modern Period (1550-1620)
- Lettes (scan) and transcription (plain text)

Tasks:

- Design and implement a framework to upload scans and transcribe them by vision model

Subtasks:

- Compare different commercial and open source vision models
- Design and implement proper Web frontend

Languages / Tools / Platforms:

- Python, Ollama, LangChain, Open WebUI

BP: Chat with your PDF (MG)

Given:

- Large PDF documents, such as a textbook

Tasks:

- Design and implement a framework to "chat" with the PDF, i.e., a conversational AI that includes question answering

Subtasks:

- Design and implement proper Web frontend to upload PDF and have conversation
- Implement proper conversational chatbot

Languages / Tools / Platforms:

- Python, Ollama, LangChain, Open WebUI

Slides Online

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