UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

# Software Practicals
# Winter Semester 2022/23

Database Systems Research Group
Heidelberg University
October 19, 2022

# Slides Online

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

# Organization

# Outline



- Overview of topics (today)
  - Send application for a topic until **Monday, October 24, 1pm**
  - Assignment of topics by October 26
- 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/end February)
  - Code has to be in local Gitlab of the database group
  - Presentation / demo of practical and software (10-12 minutes)
  - Report / documentation as Gitlab document (README.md)

# Application

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

- Apply directly to supervisor via mail
  - Program of study, matriculation number, “Anwendungsgebiet”
  - List relevant course experience, including course grades
  - List other experience, such as side projects you are working on or job experience
  - Give your tentative schedule and milestones for the practical
  - Group work is not possible!
- It is recommended to apply for multiple topics (“top-3 list”); clearly indicate whether you are applying for a beginners or advanced practical.

Application is binding!

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

# Deadlines



- Generally meetings with supervisor every week
- Presentation: mid February 2023
- Report & Gitlab upload: end of February 2023
- No extension possible

Not finished = failed (grade 5,0)!

# Assessment

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

- 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

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386


- Michael Gertz (MG)
  gertz@informatik.uni-heidelberg.de
- Nicolas Reuter (NR)
  reuter@informatik.uni-heidelberg.de
- Dennis Aumiller (DA)
  aumiller@informatik.uni-heidelberg.de
- Jayson Salazar (JS)
  salazar@informatik.uni-heidelberg.de
- John Ziegler (JZ)
  ziegler@informatik.uni-heidelberg.de

# Project Topics

AP = Advanced Topic

BP = Beginners Topic (for BSc students)

# Overview of Topics

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

1. Semi-Automated Filtering for Datasets, **BP/AP** (Aumiller)
2. Generating Sentence Alignments for Paraphrasing, **BP/AP** (Aumiller)
3. “Kleine Anfragen” of the German Bundestag, **AP** (Gertz)
4. Synonyms for Information Retrieval in OpenSearch, **AP** (Gertz)
5. Twitter Query Expansion, AP (Gertz)
6. Search Optimization, **AP** (Ziegler)
7. Unified Logging Layer, **AP** (Ziegler)
8. Medical Thesaurus Aggregation **BP/AP** (Salazar)
9. Integration of PyCoNet and FHIRPACK **BP/AP** (Salazar)
10. Comparison of Data Labeling Tools, **BP** (Reuter)

# **BP/AP:** Semi-Automated Filtering of Datasets (DA)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

**Given:**

1. Large text datasets for summarization (several 100k samples)
2. Filtering function with several unspecified parameters

**Tasks:**

- Iteratively show a subselection of samples in interface
- Refine parameters based on user judgments of shown samples
- Allow custom filters to be added “on the fly” (*AP must, BP optional)*

**Subtasks:**

- Integrate backend code into existing data processing library
- Optimize speed of existing routines (e.g., through multiprocessing)

**Languages / Tools:**

- Python; prior experience with data processing helpful

# **BP/AP:** Generating Sentence Alignments for Paraphrasing (DA)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

**Given:**

1. Document-aligned corpora for various tasks
2. Models for sentence similarity

**Tasks:**

- Extract sentence/paragraph pairs that are most similar
- Evaluate the quality of alignments and improve the model/data
- (AP:) Analyze performance of existing alignment methods

**Subtasks:**

- Integrate methods into existing Python library for reproducibility

**Languages / Tools:**

- Python; experience with spaCy or other NLP tools helpful

# **AP:** Kleine Anfragen of the German Bundestag (MG)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

## Given:

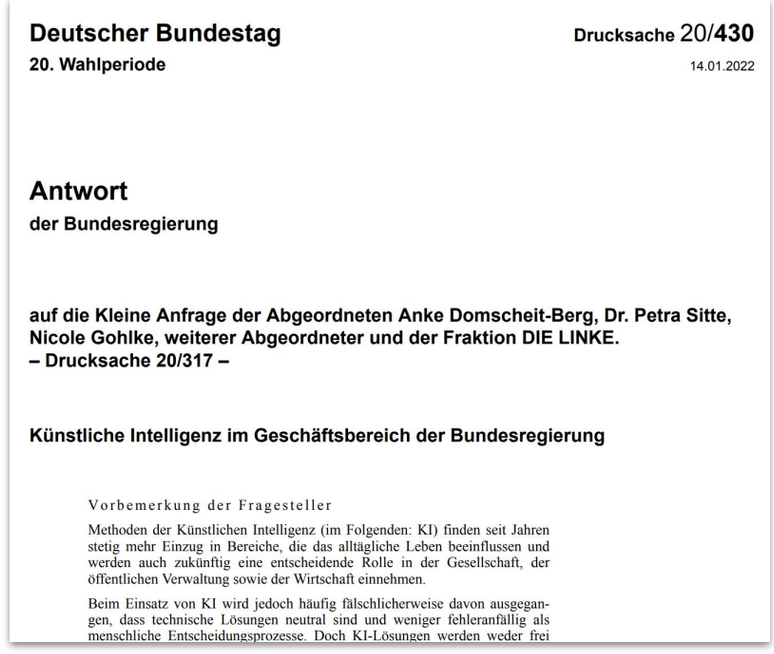
- “Kleine Anfragen” (plus answers) on the Website of the German Bundestag in the form of PDF files, see, e.g., https://dip.bundestag.de/suche?term=Kleine%20Anfrage&rows=25

## Tasks:

- Crawl PDFs and extract document text data
- Store extracted text data plus metadata in OpenSearch for Information Retrieval

## Languages / Tools:

- Python; OpenSearch



**Deutscher Bundestag** **Drucksache 20/430**
**20. Wahlperiode** 14.01.2022

**Antwort**
**der Bundesregierung**

**auf die Kleine Anfrage der Abgeordneten Anke Domscheit-Berg, Dr. Petra Sitte, Nicole Gohlke, weiterer Abgeordneter und der Fraktion DIE LINKE.**
**– Drucksache 20/317 –**

**Künstliche Intelligenz im Geschäftsbereich der Bundesregierung**

Vorbemerkung der Fragesteller

Methoden der Künstlichen Intelligenz (im Folgenden: KI) finden seit Jahren stetig mehr Einzug in Bereiche, die das alltägliche Leben beeinflussen und werden auch zukünftig eine entscheidende Rolle in der Gesellschaft, der öffentlichen Verwaltung sowie der Wirtschaft einnehmen.

Beim Einsatz von KI wird jedoch häufig fälschlicherweise davon ausgegangen, dass technische Lösungen neutral sind und weniger fehleranfällig als menschliche Entscheidungsprozesse. Doch KI-Lösungen werden weder frei

# **AP:** Synonyms for Information Retrieval in OpenSearch

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

## Given:

- Diverse lists of synonyms for German and English on the Web
- IR systems such as OpenSearch/ElasticSearch that allow for integration of synonyms


## Tasks:

- Develop software pipeline to integrate synonym data into OpenSearch
- Perform different IR-tasks to check for proper usage of synonyms, ideally for both English and German

## Languages / Tools:

- Python; OpenSearch

**Note:** this is closely related to the task of query rewriting → Thesis

# **AP:** Twitter Query Expansion (MG)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

## Given:

- Large collection of German Tweets from politicians
- Tweets and extracted metadata are managed in ElasticSearch and Postgres, resp.

## Tasks:

- Develop framework to efficiently compute likely completions of a query based on a search prefix
- First consider hashtag-based completions, second consider keyword-based completion (e.g., named entities)

## Languages / Tools:

- Python; ElasticSearch

# **AP:** Search Optimization (JZ)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

## Given:

- Elasticsearch with stored tweets and news articles
- Metadata of tweets, statistics about hashtags, usernames, …
- Search logs

## Task:

- Improve information retrieval to reach higher relevance
- Add additional data sources to be considered: see this blog post

## Subtasks:

- Explore available data sources, extend retrieval component
- Evaluate results

## Languages / Tools:

- Elasticsearch, Postgres, TypeScript (Node.js)

# **AP:** Unified Logging Layer (JZ)

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

**Given:**

- Elasticsearch + Opensearch instances
- Multiples services that depend on each other: DB, API, …
- Logs of services are partially collected in Elasticsearch

**Task:**

- Improve observability of services by collecting and analyzing logs

**Subtasks:**

- Inventory of logs
- Forwarding of logs, centralized storage, schema unification
- Analysis + alerts

**Languages / Tools:**

- ELK stack, Linux basics

# **BP(AP)** Integration of PyCoNet and FHIRPACK (JS)

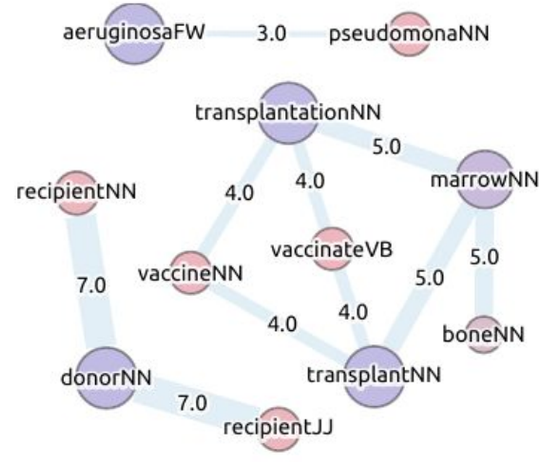
UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

## Given:

- PyCoNet, a prototype Python package aimed towards the generation of co-occurrence networks from raw text.
- FHIRPACK, an open source FHIR data Python processing toolkit.

## Tasks:

- Create a FHIRPACK transformer and loader so text data present in patient records can be processed with PyCoNet, text co-occurrence networks generated and stored accordingly.




## Languages / Tools:

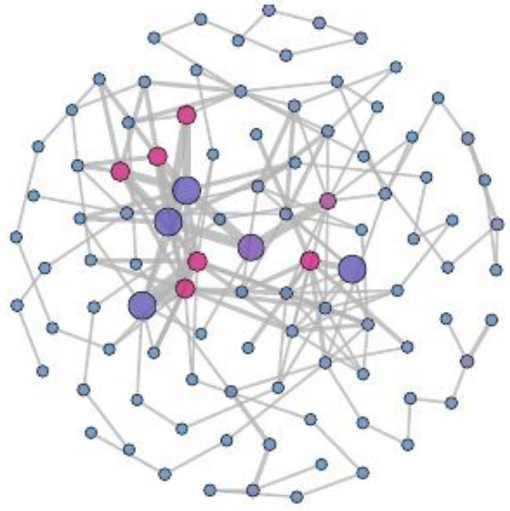
- Python, Pandas, NetworkX, PostgreSQL

# **BP(AP)** Medical Thesaurus Aggregation (JS)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

## Given:

- Access to important Medical Vocabularies like SNOMED-CT, UMLS, RxNorm, MESH, LOINC Medical Ontologies



## Tasks:

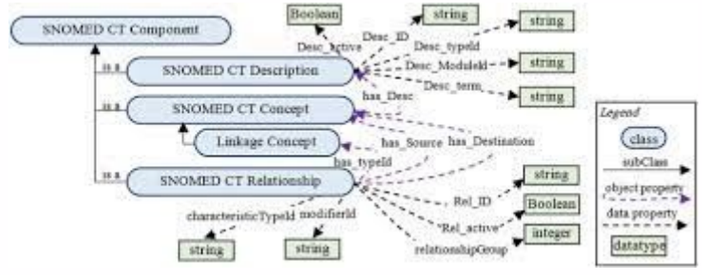
- Acquire and analyze the structure, contents and programmatic interaction of each thesaurus.
- Write a Python CLI tool that links and translates keywords to the respective entities

MeSH

SNOMED International

## Languages / Tools:

- Python, Pandas, PostgreSQL



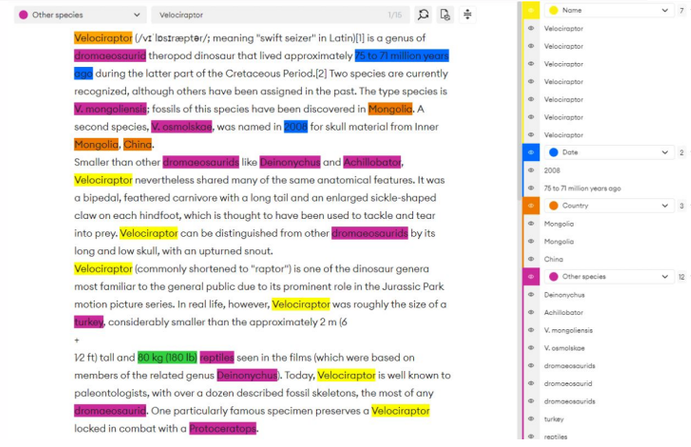
# **BP** Comparison of Data Labeling Tools (NR)

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

## **Given:**

Collection of data labeling tools for

- text: tagging of parts-of-speech, named entities, classification
- images: e.g., segmentation, object detection, classification




Dataset (HTML, PNG, CSV files) for a specific data annotation task

## **Tasks:**

1. Make a selection of tools and compare them.
2. Are there tools that are objectively best for a task?
3. Is there a tool with which the given annotation task can be performed?
4. Find a solution to implement the given annotation task.

## **Languages / Tools:**

- Depending on the result of task 3, Docker, Python

# Slides Online

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