UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Software Practicals
Summer Semester 2021

Database Systems Research Group
Heidelberg University
April 14, 2021

Slides Online

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

Organization

Outline

- Overview of topics (today)
  - Send application for a topic until **Monday, April 19, 1pm**
  - Assignment of topics by April 21/22
- First milestone (mid/end 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 in local Gitlab
  - Presentation / demo of practical and software (10-12 minutes)
  - Report / documentation as local Wiki document

Application

- Apply directly to supervisor(s) via mail
  - List relevant course experience, including course grades
  - List other experience:
    - Side projects
    - “Anwendungsgebiet”
    - Job experience
  - Send tentative schedule and milestones for the practical
  - Group work is not possible
- It is recommended to apply for multiple topics (2-3)

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

Deadlines

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

- In general weekly meetings with supervisor
- Presentation: last week in July 2021
- Report & Gitlab upload: August 10, 2021
- No extension possible

<u>Not finished = failed (grade 5,0)!</u>

Assessment

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

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

Supervisors

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

- Michael Gertz (MG)
  gertz@informatik.uni-heidelberg.de
- Satya Almasian (SA)
  almasian@informatik.uni-heidelberg.de
- Dennis Aumiller (DA)
  aumiller@informatik.uni-heidelberg.de
- Philip Hausner (PH)
  hausner@informatik.uni-heidelberg.de
- John Ziegler (JZ)
  ziegler@informatik.uni-heidelberg.de

Project Topics

Overview of Topics

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

1. German Keyphrase Extraction, **BP/AP** (Aumiller)
2. RNV Monitor Frontend, **BP/AP** (Aumiller/Hausner)
3. Active Learning for Annotations, **AP**, (Almasian/Hausner)
4. HTML Tag Embedding, **AP**, (Hausner)
5. Date Understanding in Word Embeddings , **AP**, (Almasian)
6. Data Extraction from Federal Agency for Civic Education, **AP** (Gertz)
7. Migration and Exploration of Historical Letters Texts, **BP/AP** (Gertz)
8. Pattern Recognition for Fine Notices / Document Scans, 2 **APs** (Gertz)
9. Continuous Postgres Monitoring, **AP,** (Ziegler)
10. Graph Stream-based temporal Network Centralities, **AP,** (Ziegler)

BP = Beginner Practical

AP = Advanced Practical

BP/AP: German Keyphrase Extraction (DA)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

1. Court Decision from Bundesverwaltungsgericht (BVerwG) including Keyphrase, see, e.g. [1]
2. Python implementations of several keyphrase extraction algorithms

Tasks:

- Crawl texts and keyphrases from BVerwG website
- Implement/adapt keyphrase extraction pipeline for German texts

Subtasks:

- Investigate extraction quality with common metrics

Languages / Tools:

- Python; spaCy; RegEx; Web Crawling; German beneficial

AP: Active Learning for Annotations (SA/PH)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

1. English news articles corpus (already pre-processed)
2. Labeling tool (label-studio)

Tasks:

- Create a general active learning framework, compatible with different models and label requirements
- Test the framework with the task of sequence labeling for numerical information in news articles

Subtasks:

- Create the initial model with any Deep Learning framework
- Create a small test set for evaluation
- Get familiar with the labeling platform and integrate active learning

Languages / Tools:

- Python, SciKit-Learn, docker (beginner knowledge required), some Deep Learning framework

BP/AP: RNV Monitor Frontend (DA/PH)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

- Elasticsearch database with data about the RNV
- Insights from previous practicals

Tasks:

- Create a web frontend to investigate delays of RNV public transport
- Visualize data with plots and a map

Subtasks:

- Become familiar with previous work
- Build a web page to display data
- Work on further intuitive visualizations

Languages / Tools:

- HTML, CSS, Javascript

AP: HTML Tag Embeddings (PH)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

- Set of News Pages
- Paper about Node2Vec [Link]

Tasks:

- Create low-dimensional vector embeddings for HTML tags
- Similar to Word2Vec embeddings

Subtasks:

- Create graph structures from web pages
- Implement method similar to Node2Vec
- Investigate and identify a suitable set of parameters

Languages / Tools:

- Python, Gensim, Graphs, optional: Web Crawling, ideally Word2Vec

AP: Date Understanding in Word Embeddings (SA)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

- Formulation of different date comparison tasks

Tasks:

- Test the Word2Vec understanding of date information
- Generate data for date comparison

Subtasks:

- Defining different tasks for date comprehension and testing them on the embeddings, examples:
  - Comparison between two dates
  - Identifying most recent time in a list
  - Understand different date formats (2020-02-02, 02 Feb 2020)

Languages / Tools:

- Python, Pytorch

AP: Federal Agency for Civic Education (MG)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

- Website of the Federal Agency for Civic Education (Bundeszentrale für politische Bildung)

Tasks:

- Build pipeline to (1) extract text data (by nested categories) from website and (2) record text data and metadata in Elasticsearch
- Periodically check for updates on Website

Subtasks:

- Design schema for document and metadata storage in Elasticsearch
- Design and implement IR Style query interface using Kibana

Languages / Tools:

- Python, Elasticsearch, Kibana, Web crawling

BP/AP: Historical Letter Texts (MG)

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

HEIDELBERGER AKADEMIE DER WISSENSCHAFTEN
Akademie der Wissenschaften des Landes Baden-Württemberg

Given:

- Project "Theologenbriefwechsel" at the Akademie der Wissenschaften; database with > 6,000 letters

Tasks:

- Migrate document data from MongoDB to Solr. (BP)
- Implement search interface on top of Solr using existing frontend infrastructure for letters' text data. (AP)

Subtasks:

- Design schema mapping; develop continuous migration pipeline from MongoDB to Solr
- Design and implement IR Style query interface on top of Solr

Languages / Tools:

- Python, Solr, [MongoDB, JSON]

AP: Pattern Recognition for Fine Notices (MG)

Given:

Sample fine notices (speeding) and related documents as non-OCRed PDFs (e.g., calibration certificates)

Tasks:

- Identify, extract, and record information segments (text) from documents

Subtasks:

- Become familiar with Tesseract OCR and OCRmyPDF
- Implement text extraction and data management pipeline

Languages / Tools:

- Python; tesseract; German language preferred but not required

AP: Continuous Postgres Monitoring (JZ)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

1. Running Postgres (PG) instance
2. GraphQL API in front of PG
3. PGAdmin set up

Tasks:

- Set up a service that continuously checks the performance of the DB
- Finding slow queries
- Notification via alerts

Subtasks:

- Get familiar with PG monitoring and according internal statistics
- Learn how to query GraphQL API

Languages / Tools:

- Postgres, PGAdmin
- SQL, Unix-Shell

AP: Graph Stream-based temporal Network Centralities (JZ)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

1. Dataset of a graph stream
2. Some reference implementations in Python
3. Research publications

Tasks:

- Implementation of temporal network centrality measures working on graph streams

Subtasks:

- Get familiar with graph stream model
- Read into temporal network centralities
- Implementation of measure(s) as SQL function(s)

Languages / Tools:

- Postgres, PGAdmin, SQL

Slides Online

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