UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Software Practicals
Summer Semester 2022

Database Systems Research Group
Heidelberg University
April 20, 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, April 25, 1pm**
  - Assignment of topics by April 27
- 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 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
  - List relevant course experience, including course grades
  - List other experience:
    - Side projects you are working on
    - “Anwendungsgebiet”
    - Job experience
  - Send your tentative schedule and milestones for the practical
  - Group work is not possible!
- It is recommended to apply for multiple topics (“top-3 list”)

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

Deadlines

- Generally meetings with supervisor every week
- Presentation: last week of July 2022
- Report & Gitlab upload: August 8, 2022
- 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
- Satya Almasian (SA)
  almasian@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

1. Re-implementation of Qsearch Pipeline, **AP** (Almasian)
2. Extension of Numerical Extractor, **BP** (Almasian)
3. Crawling a Cross-lingual Summarization Dataset, **BP/AP** (Aumiller)
4. Analysis of English Summarization Data, **BP/AP** (Aumiller)
5. Mapping the University Web-space, **AP** (Gertz)
6. Table Extraction from PDFs, AP (Gertz)
7. QA System for German Finance Texts, **AP** (Gertz)
8. Trend Exploration UI, **AP** (Ziegler)
9. Weighted Subgraph Prediction, **AP** (Ziegler)
10. Medical Thesaurus Aggregation **BP/AP** (Salazar)
11. Benchmarking SOTA Keyword Extraction **BP/AP** (Salazar)

AP: Re-implementation of Qsearch Pipeline (SA)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

- The paper "Qsearch: Answering Quantity Queries from Text"
- Data generation script and news articles from finance domain

Tasks:

- Re-implement the Qsearch pipeline to extract Quantity Facts
- Input: a quantity fact ("Cars with price less than 100k Euros")
- Output: top-k relevant sentences to the query from the corpus

Subtasks:

- Apply the generation script to news articles to generate tagged data
- A sequence tagging model to identify parts of QFact (partially given)
- Implement the scoring functions from the paper

Languages / Tools:

- Python; Pytorch (some knowledge using hugginface recommended)

AP: Extension of Numerical Extractor (SA)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

- Implementation of Numerical Extractor (finding numbers and units in the text and standrazing them)

Tasks:

- Extend the current implementation to financial data

Subtasks:

- Get familiar with the existing codebase
- Get familiar with dependency trees
- Extend the code with regex and dependency rules to detect numbers and units in text

Languages / Tools:

- Python; (some knowledge using spaCy recommended)

BP/AP: Crawling a Cross-lingual Summarization Dataset (DA)

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Given:

1. Dump of Wikipedia articles

Tasks:

- Extract content from Wikipedia articles in different languages and identify pairs of uneven length
- Implement/adapt keyphrase extraction pipeline for German texts

Subtasks:

- Analyze category distribution of extracted articles (person, city, ...)

Languages / Tools:

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

BP/AP: Analysis of English Summarization Data (DA)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

1. English Summarization Datasets (e.g., CNN/DailyMail, XSUM)
2. Method to analyze distribution in sentence alignments

Tasks:

- Analyze distribution of dataset and compute basic properties
- (AP) Obtain model predictions and perform error analysis

Subtasks:

- Integrate methods into existing Python library for reproducibility

Languages / Tools:

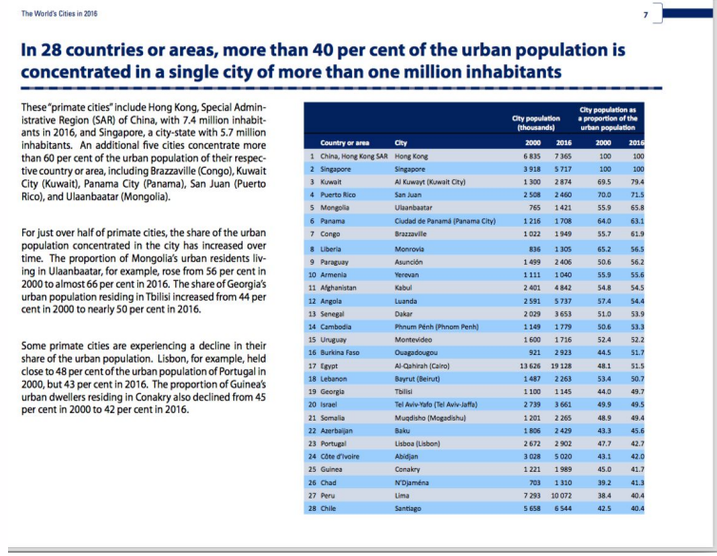
- Python; spaCy/NLTK beneficial

AP: Table Extraction fom PDFs (MG)

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Given:

- Collection of PDFs (finance reports etc.)
- Tools for extracting table data from PDFs



The World's Cities in 2016 7

In 28 countries or areas, more than 40 per cent of the urban population is concentrated in a single city of more than one million inhabitants

These "primate cities" include Hong Kong, Special Administrative Region (SAR) of China, with 7.4 million inhabitants in 2016, and Singapore, a city-state with 5.7 million inhabitants. An additional five cities concentrate more than 60 per cent of the urban population of their respective country or area, including Brazzaville (Congo), Kuwait City (Kuwait), Panama City (Panama), San Juan (Puerto Rico), and Ulaanbaatar (Mongolia).

For just over half of primate cities, the share of the urban population concentrated in the city has increased over time. The proportion of Mongolia's urban residents living in Ulaanbaatar, for example, rose from 56 per cent in 2000 to almost 66 per cent in 2016. The share of Georgia's urban population residing in Tbilisi increased from 44 per cent in 2000 to nearly 50 per cent in 2016.

Some primate cities are experiencing a decline in their share of the urban population. Lisbon, for example, held close to 48 per cent of the urban population of Portugal in 2000, but 43 per cent in 2016. The proportion of Guinea's urban dwellers residing in Conakry also declined from 45 per cent in 2000 to 42 per cent in 2016.

	Country or area	City	City population (thousands) 2000	City population (thousands) 2016	City population as a proportion of the urban population 2000	City population as a proportion of the urban population 2016
1	China, Hong Kong SAR	Hong Kong	6 835	7 365	100	100
2	Singapore	Singapore	3 918	5 717	100	100
3	Kuwait	Al Kuwayt (Kuwait City)	1 300	2 874	69.5	79.4
4	Puerto Rico	San Juan	2 508	2 460	70.0	71.5
5	Mongolia	Ulaanbaatar	765	1 421	55.9	65.8
6	Panama	Ciudad de Panamá (Panama City)	1 216	1 708	64.0	63.1
7	Congo	Brazzaville	1 022	1 949	55.7	61.9
8	Liberia	Monrovia	836	1 305	65.2	56.5
9	Paraguay	Asunción	1 499	2 406	50.6	56.2
10	Armenia	Yerevan	1 111	1 040	55.9	55.6
11	Afghanistan	Kabul	2 401	4 842	54.8	54.5
12	Angola	Luanda	2 591	5 737	57.4	54.4
13	Senegal	Dakar	2 029	3 653	51.0	53.9
14	Cambodia	Phnum Pénh (Phnom Penh)	1 149	1 779	50.6	53.3
15	Uruguay	Montevideo	1 600	1 716	52.4	52.2
16	Burkina Faso	Ouagadougou	921	2 923	44.5	51.7
17	Egypt	Al-Qahirah (Cairo)	13 626	19 128	48.1	51.5
18	Lebanon	Bayrut (Beirut)	1 487	2 263	53.4	50.7
19	Georgia	Tbilisi	1 100	1 145	44.0	49.7
20	Israel	Tel Aviv-Yafo (Tel Aviv-Jaffa)	2 739	3 661	49.9	49.5
21	Somalia	Muqdisho (Mogadishu)	1 201	2 265	48.9	49.4
22	Azerbaijan	Baku	1 806	2 429	43.3	45.6
23	Portugal	Lisboa (Lisbon)	2 672	2 902	47.7	42.7
24	Côte d'Ivoire	Abidjan	3 028	5 020	43.1	42.0
25	Guinea	Conakry	1 221	1 989	45.0	41.7
26	Chad	N'Djaména	703	1 310	39.2	41.3
27	Peru	Lima	7 293	10 072	38.4	40.4
28	Chile	Santiago	5 658	6 544	42.5	40.4

Tasks:

- Develop target model in which extracted table data is to be mapped.
- For different types of tables in PDFs, evaluate the extraction quality of different tools (e.g., tabula-py or Camelot)

Subtasks:

- Realize backend to store extracted data using OpenSearch.

Languages / Tools:

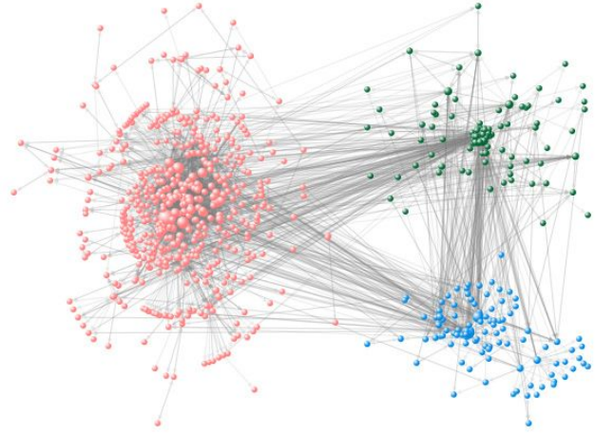
- Python; OpenSearch

AP: Mapping the University Web-space (MG)

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Given:

- Crawl of HD university websites, plus metadata, stored in OpenSearch (~200k web pages)



Tasks:

- Construct network from crawled data
- Perform different network analysis tasks (e.g., centrality, community detection)

Subtasks:

- Realize backend to store network information, e.g., using Neo4j

Languages / Tools:

- Python; Neo4j

AP: QA System for German Finance Texts (MG)

Given:

- Collection of German finance news
- German Language Models (GermanBert, GermanQuAD GermanDPR) from deepset.ai.

deepset

Tasks:

- Develop set of QA pairs specific to finance news dataset
- Evaluate performance on base model(s)
- Fine-tune model(s) and again evaluate performance

Subtasks:

- Familiarize yourself with Haystack pipelines and Language Models

Languages / Tools:

- Python; Haystack; huggingface

AP: Trend Exploration UI (JZ)

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Given:

- REST API to access temporal trends represented as networks

Task:

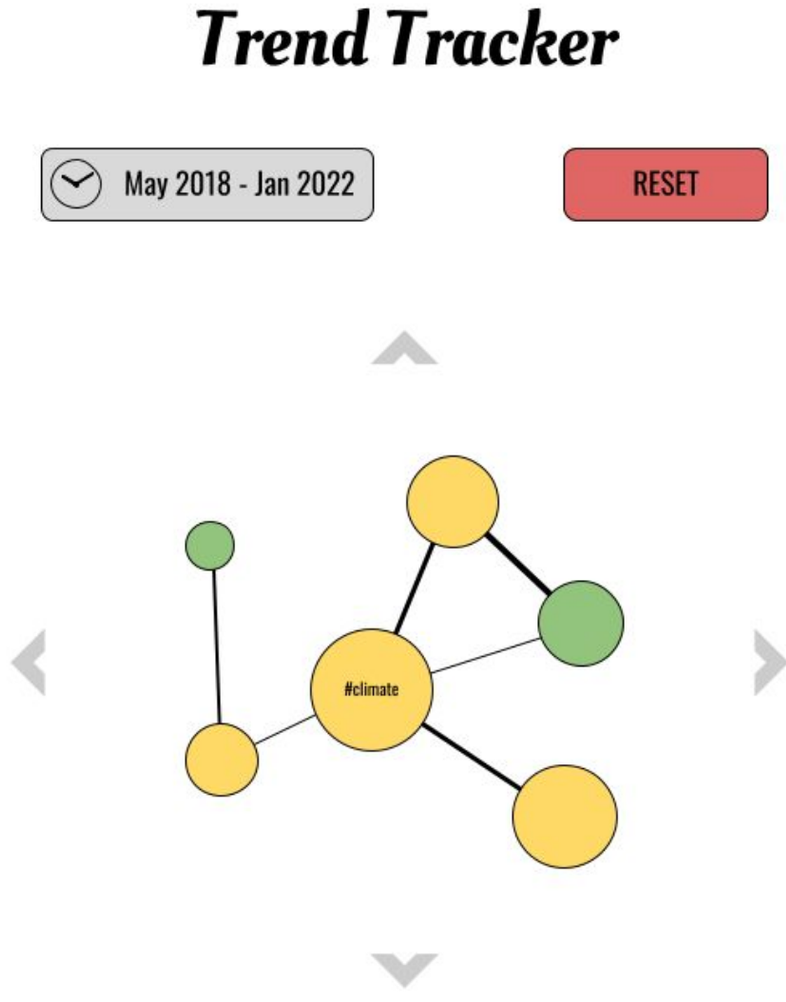
- Implementation of UI to explore trends

Subtasks:

- Handle API access
- Visualization of networks

Languages / Tools:

- Frontend framework (e.g., React)
- TypeScript



AP: Weighted Subgraph Prediction (JZ)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

- Public temporal network datasets (e.g., SNAP)
- Twitter dataset managed at database group
- Possibility to write MSc thesis in the following (!!!)

Tasks:

- Comparison of different time-series prediction models
- Application to temporal (sub)graphs

Subtasks:

- Research on time-series prediction tools (e.g., Facebook Prophet, AWS Gluon)
- Preparation of temporal network(s) as multivariate time-series
- Evaluation of tools on dataset, i.e., extracted weighted subgraphs

Languages / Tools:

- Python; some ML knowledge

BP(AP) Benchmarking SOTA Keyword Extraction (JS)

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Given:

- Paper containing a thorough review of interesting graph-based keyword extraction methods
- Dataset containing 50,000 scientific articles

Tasks:

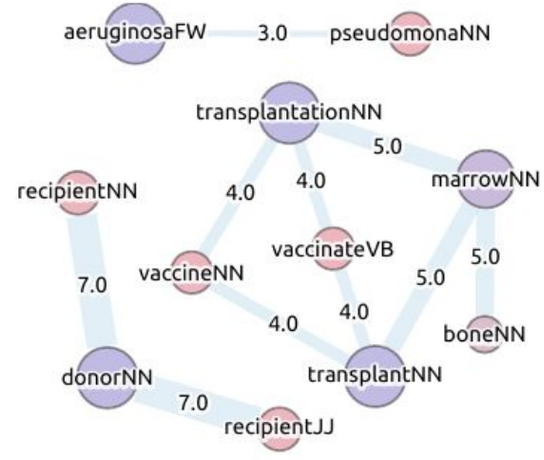
- Implement a Python module that performs preprocessing, tokenization, graph construction and keyword extraction on an input plain-text document tuple **<title, abstract, keywords>**
- Run at least three of the methods described against each other

Languages / Tools:

- Python, Pandas, PostgreSQL

TITLE: hemoadsorption corrects hyperresistinemia restores anti-bacterial neutrophil function ①

ABSTRACT: mounting evidence suggests sepsis-induced morbidity mortality due immune activation immunosuppression ② resistin inflammatory cytokine uremic toxin ③ septic hyperresistinemia plasma resistin associated greater disease severity worse outcomes exacerbated concomitant acute kidney injury ④ septic hyperresistinemia disturbs actin polymerization neutrophils leading impaired neutrophil migration crucial first-line mechanism host defense bacterial infection ⑤ experimental objective study effects hyperresistinemia f-actin-dependent neutrophil defense mechanisms particular intracellular bacterial clearance generation reactive oxygen species ⑥ also sought examine effects hemoadsorption hyperresistinemia neutrophil dysfunction ⑦ thirteen patients septic shock six control patients analyzed serum resistin levels effects neutrophil migration ⑧

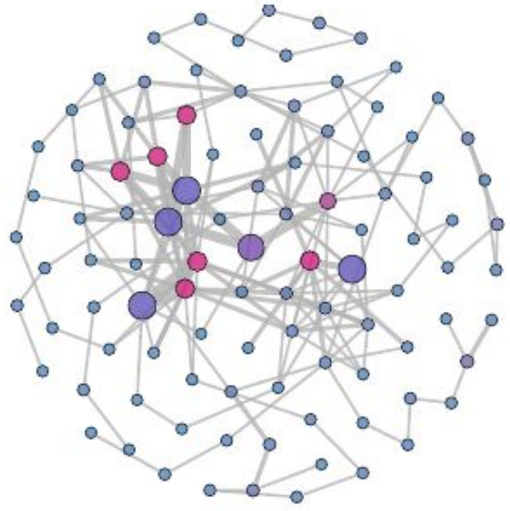


BP(AP) Medical Thesaurus Aggregation (JS)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

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



Tasks:

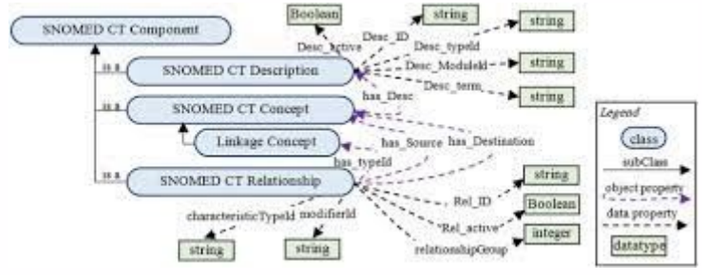
- Acquire and analyze the structure, contents and programmatic interaction of each thesaurus.
- Write a Python CLI tool that links and translates keywords to the respective entities

MeSH

SNOMED International

Languages / Tools:

- Python, Pandas, PostgreSQL



Slides Online

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