UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Software Practicals
Winter Semester 2018/19

Database Systems Research Group
Heidelberg University
October 18, 2018

Organization

Slides Online

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

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

Outline

- Overview of topics (today)
  - send application for a topic until Monday, October 22, 13:00
  - assignment of topics by October 25
- First milestone (end of November)
  - prototype/part of software
  - summary of research (literature and related systems/tools)
  - further milestones in agreement with supervisor
- End of practical (beginning of February)
  - code (SVN / build-script / comments)
  - report (~ 10 pages) as pdf or wiki documentation
  - presentation/demo of practical and software (10-15 minutes)

Organizational issues

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

- Application
  - by email directly to supervisor
  - brief list of relevant courses / prior knowledge
  - schedule and milestones for the practical
  - group work is not possible
  - application is binding (don't apply if you don't want to do the practical)
- Deadlines
  - presentation: planned for second week in February 2019
  - report: end of February 2019
  - no extension possible
  - not finished = failed (grade 5,0)

Assessment

- Credit points (Leistungspunkte)
  - Beginners Practical (IAP, 6 ECTS) [Bachelor students]
    - workload: 180 h (~1 ½ days/week)
  - Advanced Practical (IFP, 8 ECTS / 6 ECTS)
    - workload: 240 h (~2 days/week)
- Grading based on
  - code (readability, structure, functionality)
  - documentation (README, comments)
  - report
  - commitment and self-reliance
  - cool ideas!!
- IMPORTANT
  - talk to / communicate with your advisor

Topics

Overview of Topics

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

1. Collecting Facebook Postings using ELK Stack, **BP/AP** (Gertz)
2. Collecting Twitter Politics Postings using ELK Stack, **AP** (Gertz)
3. Exploration and Analysis of Twitter/Facebook Data, **2 APs** (Gertz)
4. Evaluating Network-based Entity Linking, **AP** (Spitz)
5. Collection and Analysis of Time-Varying Open Data Graphs, **2 APs** (Lackner)
6. Comparison of Visualization Frameworks for Time-Varying Graphs, **BP** (Lackner)
7. Tracking Changes in Dynamic Information Networks, **BP/AP** (Lackner)

ELK Stack

Elasticsearch: search engine based on Lucene, NoSQL, RESTful Web interface

Logstash: open source, server-side data processing pipeline

Kibana: open source data visualization plugin for Elasticsearch

BP/AP: Exploration of Facebook Postings (Gertz)

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Given:

1. Pipeline to extract postings (parties & politicians) from Facebook
2. Storage framework of postings in MongoDB

Tasks:

- Build data collection and storage pipeline using ELK stack
- Monitoring and analysis GUI of postings

Subtasks:

- Rewrite pipeline (MongoDB API → Elasticsearch API)
- Develop GUI components for data analysis using Kibana

Languages / Tools:

- Python; Elasticsearch for data storage; Kibana for analysis

AP: Analysis of Twitter Postings (Gertz)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

1. Pipeline to collect Twitter posts (TWIPA)
2. File-based storage framework and GUI for filtering/export

Tasks:

- Collect tweets from parties and politicians (user-specified lists)
- Develop and implement simple monitoring components

Subtasks:

- Python-based pipeline using Elasticsearch API
- GUI components for analysis using Kibana

Languages / Tools:

- Python; Elasticsearch for data storage; Kibana for analysis

2 APs: Exploration of Facebook / Twitter Data (Gertz)

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Given:

Collections of time-stamped Tweets / Facebook postings

Tasks:

- Both: Named Entity Recognition (German), persons, locations, ...
- Facebook: co-occurrence network of entities and terms
- Twitter: hashtag analysis and visualization

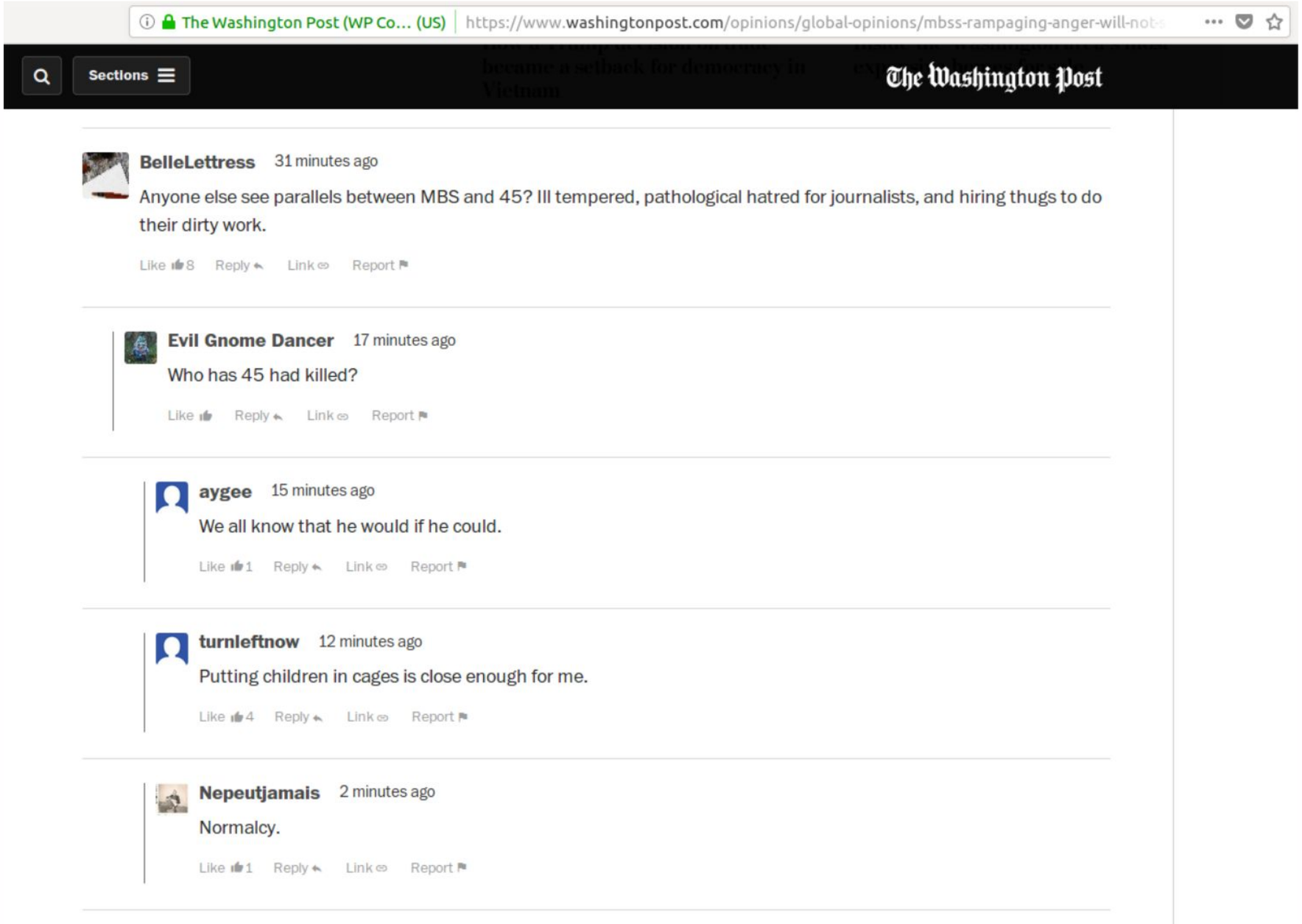
Subtasks:

- Diverse GUIs on top of Elasticsearch

Languages / Tools:

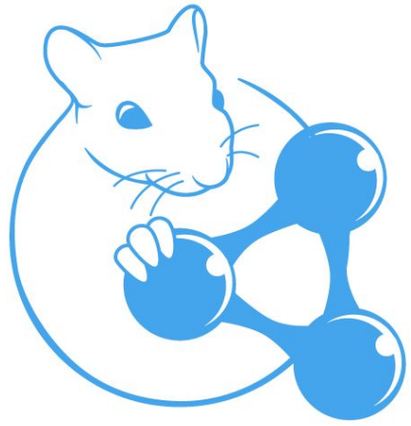
- Python; Elasticsearch for data storage
- ReactiveSearch (https://opensource.appbase.io/reactivesearch/)
- ReactiveMaps (https://opensource.appbase.io/reactivemaps/)

Entity Recognition and Linking



AP: Evaluating Network-based Entity Linking (Spitz)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386



Given:

1. Framework for entity recognition and linking in texts
2. GERBIL evaluation interface [1, 2]

Task:

Evaluate our entity linking framework with GERBIL

Subtasks:

- Design and implement a Web service for our framework
- Run thorough evaluations on GERBIL and tweak the framework

Languages / Tools / Knowledge:

- Good Java programming skills (mandatory)
- Meticulous and diligent work ethic (mandatory)
- Experience with RESTful communication / NLP (helpful)

2 APs: Collection and Analysis of Time-Varying Open Data Graphs (Lackner)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

1. Relational dataset with time information (e.g., from [1] or [2])
2. Existing methods for analyzing time-varying graphs

Tasks:

- Collect data and construct a time-varying graph
- Apply methods, analyze evolution and interpret results

Subtasks:

- Decide on a specific dataset (*include idea(s) in your application!*)
- Optionally: Implement additional methods

Languages / Tools:

- Python; MongoDB; knowledge in Java is a plus.

BP: Comparison of Visualization Frameworks for Time-Varying Graphs (Lackner)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

Time-varying graph datasets with 10^2 ... 10^5 vertices.

Task:

Create survey comparing existing visualization frameworks for time-varying graphs.

Subtasks:

- Determine which frameworks are suitable (*Gephi?, Graphviz?, …*)
- Import dataset into frameworks (write conversion scripts)
- Compare frameworks with regards to performance, usability, ...

Languages / Tools / Knowledge:

- Python; knowledge in Bash is a plus

BP/AP: Tracking Changes in Dynamic Information Networks (Lackner)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

1. Dynamic network topics dataset [1] based on news articles
2. Existing libraries for community detection

Task: Based on [2], implement (a subset of) methods to analyze the network dynamics (e.g., forming of new communities, splitting of communities, size transitions, …)

Subtasks:

- Understand the paper and get familiar with existing code
- Implement (a subset of) methods proposed in [2]
- Apply the methods to snapshots of the network topics dataset
- Evaluate the quality of results / Implement interactive visualization

Languages / Tools:

- Python; knowledge in Java is a plus.

Supervisors

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

- Michael Gertz (MG)

  gertz@informatik.uni-heidelberg.de

- Andreas Spitz (AS)

  spitz@informatik.uni-heidelberg.de

- Sebastian Lackner (SL)

  lackner@informatik.uni-heidelberg.de

Slides Online

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

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