Univ.-Prof. Dr.-Ing. Matthias Boehm
Technische Universität Berlin
Faculty IV - Electrical Engineering and Computer Science
Berlin Institute for the Foundations of Learning and Data (BIFOLD)
Big Data Engineering (DAMS Lab) Group

1 DIA WiSe2023: Exercise – Entity Resolution of Publication Data

Published: Oct 19, 2023 (last update: Oct 19)
Deadline: Feb 02, 2023, 11.59pm

This exercise is an alternative to the DIA programming projects, and aims to provide practical experience in the development of data engineering and ML pipelines. The task is to construct an Entity Resolution (ER) pipeline for deduplication of research publication datasets. You may use any programming language(s) of your choosing, and utilize existing open-source ML frameworks and libraries. The expected result is a zip archive named `DIA_Exercise_<student_ID>.zip` (replace `<student_ID>` by your student ID) of max 5 MB, containing:

- The source code used to solve the individual sub-tasks
- A PDF report of up to 8 pages (10pt), including the names of all team members, a brief summary of how to run your code, and a description of the solutions to the individual sub-tasks.

Data: Obtain the DBLP and ACM datasets from here and here respectively. A description of the data can be found at `https://www.aminer.org/citation`.

Grading: This exercise can be pursued in teams of 1 to 3 persons (one submission, scale quality expectations). The overall grading is a *pass/fail* for the entire team. Exercises with $\geq 50/100$ points are a pass, and with $\geq 90/100$ points we receive 5 extra points in the exam.

1.1 Data Acquisition and Preparation (20/100 points)

Obtain the datasets mentioned above. The datasets are in text format. As a prerequisite for Entity Resolution and model training, extract paper ID, paper title, author names, publication venue, and year of publication from the datasets. Collect all the publications published between 1995 to 2004 in VLDB and SIGMOD venues. To filter the venues, check if the publication venue contains the strings "SIGMOD" or "VLDB" (case insensitive). Store the extracted entries in two CSV files, `DBLP_1995_2004.csv` and `ACM_1995_2004.csv`.

Expected Results: Code for data preparation. The report should discuss the implementation details of converting the custom text format to CSV.

1.2 Entity Resolution Pipeline (45/100 points)

Construct an ER pipeline to match the entries from `DBLP_1995_2004.csv` and `ACM_1995_2004.csv` that refer to the same publication. The pipeline should include the following steps:

- **Blocking:** Use a blocking scheme to assign the entries to one or more buckets based on blocking keys. Example blocking methods are structured keys, n-gram blocking, as well as hash- or sort-based blocking (e.g., by year ranges).
- **Matching:** For all pairs of entities in a bucket, apply a similarity function to determine if they refer to the same entity (if above a certain similarity score threshold). Write all the matched pairs

in a file, Matched_Entities.csv. To verify the ER match quality, you need a baseline. Apply the same similarity function on all pairs of the datasets and use the result as your baseline. Calculate the precision, recall, and F-measure of the matches generated by your ER pipeline compared to the baseline. Experiment with different blocking schemes, similarity functions, and similarity score thresholds to improve match quality and reduce execution time.

- **Clustering:** Once you are happy with the matches, group together all the identified matches in clusters such that all entities within a cluster refer to the same entity. Finally, resolve the unmatched entities with a single version in the datasets and write them back to disk.

Expected results: Code for the ER pipeline, as well as descriptions of the techniques and baselines used. The report must include the match quality measures and the execution time improvement of the ER pipeline compared to the baseline.

1.3 Data-Parallel Entity Resolution Pipeline (35/100 points)

Reimplement your entity resolution pipeline on top of a data-parallel computation framework such as Apache Spark, Apache Flink, or Dask. This data-parallel pipeline should produce the same results as your local pipeline (validated by comparing Matched_Entities.csv of both pipelines). Furthermore, in order to validate the scalability of your pipeline, please replicate each dataset 1 to 10 times with minor modifications of each entity's attributes, and plot the resulting execution time (x-asis: replication factor, y-axis: runtime in seconds).

Expected Results: Code for the data-parallel ER pipeline, a description of the data-parallel pipeline, and the plot of the runtime experiment.