burgerLinker - Civil Registries Linking Tool

Purpose

This tool is being developed to improve and replace the current LINKS software. Points of improvement are:

- extremely fast and scalable matching procedure (using Levenshtein automaton and HDT);
- considers all first names of the individuals with multiple first names in order to find a candidate match;
- blocking is not required (i.e. all candidate records can be considered for matching, with no restrictions on their registration date or location, and no requirements on blocking parts of their individual names);
- detected links contains detailed provenance metadata, and can be saved in different formats (CSV and RDF are covered in the current version);
- allows family and life course reconstruction (by computing the transitive closure over all detected links);
- open software.

To download the latest version of the tool click releases on the right of the screen.

Use case

Historians use archival records to describe persons' lives. Each record (e.g. a marriage record) just describes a point in time. Hence historians try to link multiple records on the same person to describe a life course. This tool focuses on "just" the linkage of civil records. By doing so, pedigrees of humans can be created over multiple generations for research on social inequality, especially in the part of health sciences where the focus is on gene-social contact interactions.

User profile

The software is designed for the so called "digital historians" (e.g. humanities scholars with basic command line skills) who are interested in using the Dutch civil registries for their studies, or for linking their data to it.

Data

In its current version, the tool cannot be used to match entities from just any source. The current tool is solely focused on the linkage of civil records, relying on the sanguineous relations on the civil record, modelled according to our Civil Registries schema. An overview of the Civil Registries schema is available as a PNG file, and you can browse it on Druid.

Previous work

So far, (Dutch) civil records have been linked by bespoke programming by researchers, sometimes supported by engineers. Specifically the IISG-LINKS program has a pipeline to link these records and provide them to the Central Bureau of Genealogy (CBG). Because the

number of records has grown over time and the IISG-LINKS takes an enormous amount of time (weeks) to LINK all records currently present, *burgerLinker* is designed to do this much faster (full sample takes less than 48 hours).

The Golden Agents project has brought about Lenticular Lenses a tool designed to link persons across sources of various nature. We have engaged with the Lenticular Lenses team on multiple occasions (a demo-presentation, two person-vocabulary workshops, and a specific between-teams-workshop). From those meetings we have adopted the ROAR vocabulary for work in CLARIAH-WP4. On the specific *burgerLinker* and lenticular lenses tool, however we found that the prerequisite in Lenticular Lenses to allow for heterogenous sources, conflicted with the *burgerLinker* prerequisite to be fast: one reason for it to be fast is the limited set of sources that *burgerLinker* allows for.

The only other set of initiatives that we are aware of are bespoke programming initiatives by domain specific researchers, with country and time specific rules for linking in for example R. These linkage tools are on the whole slow. What we did do is make our own rule set for linking modular, to allow in the future for country and time specific rule sets to be incorporated in *burgerLinker*.

Operating Systems

- This tool is tested on Linux and Mac OS.
- Windows users are advised to use the Docker image.

Installation requirements

- Only the JAVA Runtime Environment (JRE), which is free and installed on almost every computer these days.

Input requirements

- Only one RDF dataset, describing the civil registries that are modelled according to our simple Civil Registries schema. For efficient querying (i.e. lower memory usage with fast search), the matching tool requires the dataset to be compressed and given as an HDT file with its index. What is HDT?

The tool allows the conversion of any valid RDF file to HDT using the `--function convertToHDT` (see Example 2 below).

Output format

Two possible output formats to represent the detected links:

- CSV file (default if no output format is specified by the user)
- N-QUADS file (it can be specified in the parameters of the tool using `--format RDF`)

Main dependencies

This tool mainly rely on two open-source libraries:

- Levenshtein automata (MIT License)
- RDF-HDT (LGPL License)

Tool functionalities

Functionalities that are supported in the current version: (case insensitive)

- `ConvertToHDT`: compress an RDF dataset given as input to an HDT file that will be generated in the same directory. This function can also be used for merging two HDT files into one (see Example 3 below)
- `ShowDatasetStats`: display some general stats about the HDT dataset, given as input.
- `Within_B_M`: link *newborns* in Birth Certificates to *brides/grooms* in Marriage Certificates (reconstructs life course)
- `Between_B_M`: link *parents of newborns* in Birth Certificates to *brides & grooms* in Marriage Certificates (reconstructs family ties)
- `Between_M_M`: link *parents of brides/grooms* in Marriage Certificates to *brides & grooms* in Marriage Certificates (reconstructs family ties)
- `Closure`: compute the transitive closure of all detected links to get a unique identifier per individual. The output of this function is a new RDF dataset, where linked individuals are replaced by the same identifier in the civil registries dataset.

Tool parameters

Parameters that can be provided as input to the linking tool:

- `--function`: (required) one of the functionalities listed below
- `--inputData`: (required) path of the HDT dataset
- `--outputDir`: (required) path of the directory for saving the indices and the detected links
- `--maxLev`: (optional, default = 4) integer between 0 and 4, indicating the maximum Levenshtein distance per first or last name allowed for accepting a link
- `--fixedLev`: (optional, default = False) add this flag without a value (i.e. True) for applying the same maximum Levenshtein distance independently from the string lengths
- `--format`: (optional, default = CSV) one of the two Strings: 'RDF' or 'CSV', indicating the desired format for saving the detected links between certificates
- `--debug`: (optional, default = error) one of the two Strings: 'error' (only display error messages in console) or 'all' (show all warning in console)