# RecordLinker

RecordLinker is a novel record linking method. The basic steps to generating Unique Historical Individuals are described below. The process is described in detail afterwards.

1. Preprocessing data in pairs.py  
   The input data provided by ELO is processed into pairs.
2. Generating links in recordlinker.py  
   Similar pairs are found by comparing names. The person references in these similar pairs are linked.
3. Processing links to Unique Historical Individuals in "Unique Individuals/individuals.py"   
   The linked person references are grouped to find all person references that refer to the same person.

## Pairs

In pairs.py, the following input files are read and processed:

1. Geboorte.csv
2. Huwelijk.csv
3. Overlijden.csv

These files contain birth, marriage and death certificates. Each certificate mentions people. These people can be divided into pairs. The pairs describe a man and a woman. The following types of pairs are possible:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Person 1** | Person 2 | Certificate type |
| **1** | father | mother | birth |
| **2** | groom | bride | marriage |
| **3** | father of groom | mother of groom | marriage |
| **4** | father of bride | mother of bride | marriage |
| **5** | deceased | partner | death |
| **6** | father of deceased | mother of deceased | death |

(The sex of the deceased and their partner is not known. This is important for the order of the name comparison)

Also, for some types of pairs, the name of a child is provided. Children need to be incorporated into the RecordLinker to find the birth certificate of Unique Historical Individuals. The pairs with children become as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Person 1** | Person 2 | Child | Certificate type |
| **1** | father | mother | child born | birth |
| **2** | groom | bride | - | marriage |
| **3** | father of groom | mother of groom | groom | marriage |
| **4** | father of bride | mother of bride | bride | marriage |
| **5** | deceased | partner | - | death |
| **6** | father of deceased | mother of deceased | deceased | death |

The result of pairs.py is a .csv file with all pairs. It contains the following fields:

|  |  |
| --- | --- |
|  |  |
| **year** | Year the event took place: yyyy |
| **first\_letters** | First letter of last name man + first letter of last name woman |
| **pair** | Type of pair (integer)  **1:** parents of newborn  **2:** groom and bride  **3:** parents of groom  **4:** parents of the bride  **5:** deceased and partner  **6:** parents of deceased |
| **man** | Name of the man |
| **woman** | Name of the woman |
| **child** | Name of the child  "" if no child is provided |
| **age** | Either the age of the bride or groom, depending on the type of pair:  age of bride if pair type is 1 or 3  age of groom if pair type is 2  -1 if age is unknown or pair type is not 1, 2 or 3 |
| **uuid** | uuid of the certificate |
| **man\_uuid** | uuid of the person reference of the man |
| **woman\_uuid** | uuid of the person reference of the woman |
| **child\_uuid** | uuid of the person reference of the child  "" if no child is provided |

Note that for pair type 5 (deceased and partner) the sex is unknown. The deceased will be treated as the man and their partner as the woman. This causes an issue in the linker. The order of the names affects the Levenshtein distance. The RecordLinker has a function to minimize this impact.

## Name Processing

The input data contains fields for first name, prefix, last name, age, place of birth, date of birth, occupation, place of residence and comments for each person reference. The names are processed in the following way:

1. Only the first name and last name fields are used (no prefixes)
2. Each first name and last name are cleaned up individually
3. The cleaning up removes capital letters, removes accents and other diacritics, only keeps symbols in the alphabet and replaces
   1. ch for g
   2. c for k
   3. z for s
   4. ph for f
   5. ij for y
4. Orders first names alphabetically and appends the last name (no spaces)

**Example:**

Mariä Anña van 't Schip → annamariasgip

Also, the first letter of the last name is saved (used to optimize algorithm)

## RecordLinker

Afbeelding met Lettertype, tekst, Graphics, schermopname

Automatisch gegenereerde beschrijving

Afbeelding met tekst, schermopname

Automatisch gegenereerde beschrijving

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving