Universiteit Leiden
The Netherlands

Informatics

A Novel Technique for Life Course Reconstruction using Historical Record Linkage

Tijmen ter Beek

Supervisors:
Richard van Dijk & Hendrik Jan Hoogeboom (LIACS)
Ellen Gehring & Leida van Hees & Cor de Graaf (ELO)

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

August 27, 2023

Abstract

A unique historical individual is a person who has lived. Its life course can be reconstructed by finding all documented events that mention this person. Erfgoed Leiden en Omstreken has digitised approximately 750 thousand certificates. These certificates describe the events of birth, marriage or death. A record linking algorithm can compare the names on certificates to find what names likely describe the same person. In this thesis, the BurgerLinker algorithm is compared to the RecordLinker algorithm. The BurgerLinker is an existing method whilst the RecordLinker is created as a part of this thesis. The methods are compared based on the quality of life course reconstruction. The quality metrics used in this thesis are group size, life course duration, life course status and coverage. Large group sizes can be an indication of low-quality life courses. Analysis of life course duration can show deviations from the expected results. The status of a life course is determined by a set of rules. For example, a life course is deemed incorrect if it contains two births. The RecordLinker results are found to contain fewer incorrect life courses. The RecordLinker also linked more certificates and produced more complete life courses.

Contents

1 Introduction

1.1 Erfgoed Leiden en Omstreken

Erfgoed Leiden en Omstreken (ELO) is a non-profit, regional, cultural heritage agency. ELO specialises in archiving, monuments, archaeology and construction history [Leic]. The main goal of the organisation is to preserve and promote the cultural heritage of Leiden and the surrounding areas. ELO manages approximately 18 km of archives, containing records of 27 former municipalities [Leib]. This archive contains a vast collection of photographs, maps, certificates and other materials that are used to document the cultural heritage of the region.

In the last decades, the organisation has been working with volunteers to digitise their collection. A portion of the digitised information is freely available on their website. Currently, more than 7.000.000 person references are listed on the website. These person references describe people as documented on birth, marriage, divorce and death certificates. Digitisation contributes to ELO performing the task of securing the region's heritage by improving data redundancy [Leia]. This is especially important when the collection contains fragile documents that have obtained damage over centuries.

Currently, the digitised records allow online users to query their relatives. This feature can help users construct their own family tree. In addition, users can review information online. This functionality can point the staff of ELO to mistakes in the data and help improve their data quality. Also, digitisation allows for potential genealogy research that was not possible previously or would be too expensive.

1.2 Definitions

Words in bold point to that word being defined in this Section.

Documented events
Events like birth, marriage, divorce and death.

Certificate
A historical record that describes a documented event and the persons it involved. The schematic diagrams of the types of certificates can be found in Figure 2.

Person reference
A mention of a person on a certificate. The certificate describes a life event, creating a snapshot of a person's life.

Pair
The person reference of a man and a woman that are related. The groom and bride is an example of a pair. The pairs are used for generating links in the RecordLinker algorithm.

Unique historical individual
A person who has lived.

Life Course Reconstruction
The reconstruction of the life of a unique historical individual by finding all person references that mention this individual. This group of person references can be displayed as a life course on a timeline.

Link
Two person references that mention the same unique historical individual. Produced by a linking algorithm.

Levenshtein distance
The least amount of character insertions, deletions or substitutions to change one string into another. The lower the Levenshtein distance, the more similar two strings are.

uuid
A universally unique identifier for certificates or person references.

1.3 Problem

Although the digitization of the records improves ease of access and enables the ability to query the data, the data is not yet ready for computerised genealogy research. Currently, the information only contains **person references**. The certificates, on which the person references are found, consist of birth, marriage, divorce and death certificates. It is expected that, during a person's life, multiple of these events are documented. This implies that different person references can refer to the same **unique historical individual**.

ELO is interested in reconstructing the lives of the documented individuals. This data enables ELO to conduct genealogy research by constructing family trees and analysing the life courses. To reconstruct a person's life, all person references that mention the same individual need to be found. Because these person references are documented during events like birth, marriage, divorce or death, a timeline can be constructed, displaying that person's life course. An example of a reconstructed life course is displayed in Figure 1.

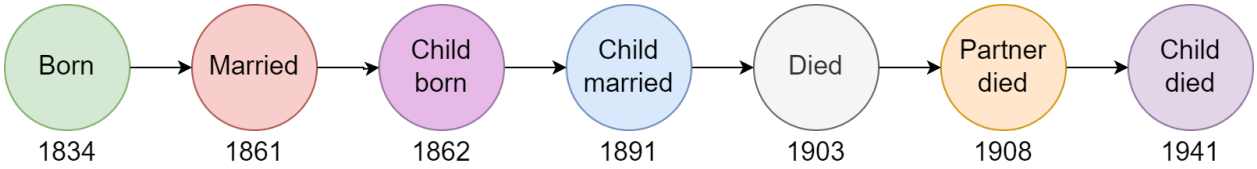


Figure 1: An example of a reconstructed life course. Every type of person reference is included.

In order to find all person references that are likely to mention the same unique historical individual, first, all person references need to be compared. Due to the amount of data, an efficient linking algorithm should be used for this task.

A linking algorithm compares the person references to determine whether or not they mention the same unique historical individual. The names documented on the certificates are compared. However, simply checking if two names are equal will not bring the desired results. Names are complex. The way they are written can change throughout the years. Inconsistent naming can be especially an issue going back hundreds of years to when there were no name standards.

Also, the same name can come in different forms. These variants imply a person's name at birth can differ from other documents. A link between these documents is not found by checking if the names coincide. Another problem can be the data quality. Because the data has been written by hand, archived for decades and digitised by volunteers, it is expected that mistakes happen. These mistakes can be a typo or caused by volunteers operating on different standards. For example, some volunteers always type exactly what is documented on the certificate, whilst others actively correct mistakes they have found. All this can impact the data quality, which affects the number of links found by an algorithm.

In this thesis, the following research question is answered:
How can life courses be reconstructed with a linking algorithm that interconnects data from archived birth, marriage and death certificates after 1810 in the region Leiden?

1.4 Thesis overview

This chapter contains the introduction. Related work is discussed in Section 2. The method and data are described in Section 3. The outcomes of the study are given in Section 4. The discussions and future work are laid out in Section 5. The research question is answered in the concluding Section 6.

The following helpful experts at LIACS and ELO supported the formation of this bachelor thesis;
(LIACS)
Richard van Dijk
Hendrik Jan Hoogeboom

(ELO)
Ellen Gehring
Leida van Hees
Cor de Graaf

2 Related Work

2.1 Record linkage

The foundations of record linkage were laid in 1946 by Halbert L. Dunn [Dun46]. He described the idea and importance of the Book of Life. Every person creates a Book of Life. It starts at birth. It ends at death. Each page contains a record of the **documented events** in life.

With the rise in computing power, research has come out exploring ways of reconstructing the Books of Life with the use of algorithms. M. Schraagen explored the aspects of record linkage [Sch14]. These aspects include the impacts of name variants, the optimum maximum accepted Levenshtein distance and the different string comparison methods. James J. Feigenbaum proposed a machine learning approach to record linkage [Fei16]. An algorithm is trained to replicate how a researcher would find links between children in the 1915 Iowa State Census and adults in the 1940 Federal Census. He concluded that his method can automate record linking with high efficiency and high accuracy. Automated methods are compared by Ran Abramitzky [ABE+21]. In this study, FamilySearch data from 1910 to 1920 is linked by three linking algorithms. The comparison of the automated results to the human generated results showed that the automated methods generated very low false positive rates.

Another approach is LINKS or "linking system for historical family reconstruction". This project is a collaboration between the IISG (International Institute for Social History), LIACS (Leiden Institute of Advanced Computer Science), the P.J. Meertens Institute, VKS (Virtual Knowledge Studies), NIDI (Netherlands Interdisciplinary Demographic Institute) and GENLIAS (now WieWasWie.nl) [MBL+23]. This method was tested and evaluated with marriage certificates from the Dutch province of Zeeland. The results showed that 80% of the matches were exact. LINKS has been integrated with WieWasWie.nl to show linked certificates [Wie].

Especially relevant to this thesis is an approach named the BurgerLinker. The BurgerLinker is a project focused on Dutch civil records, built to replace the existing LINKS project [RMR+20]. This algorithm is developed by CLARIAH, a Dutch organisation that develops and integrates digital tools and services for humanities research [CLAb]. The advantages of using the BurgerLinker over LINKS include:

- being faster and more scalable
- incorporated function for life course reconstruction
- less linking restrictions
- open software

2.2 String comparison

String comparison is an integral part in each of the record linking algorithms. String comparison methods are used to determine whether two certificates mention the same people, by comparing the names on these certificates. A quantifiable metric is assigned to express the similarity between the two names. This metric can be evaluated to label names as matching.

The Jaro-Winkler similarity and the **Levenshtein distance** are string comparison methods used in the linking algorithms described in Section 2.1. The Jaro-Winkler similarity measures the similarities between words, whilst the Levenshtein distance measures the differences between words. The Levenshtein distance calculates the number of edits that need to happen for one word to change into another [L+66]. The Jaro-Winkler similarity returns a value between 0 and 1. This value is dependent on the amount of characters in the words. The value 0 means that the words are exactly the same and 1 represents no similarity between the words.

As a part of this study, a novel linking algorithm is created, named the RecordLinker. The Levenshtein distance is the string comparison method used in the RecordLinker. The reasoning behind this is explained in Section 3.3.3.

3 Method

To find the best way of reconstructing **unique historical individuals**, linking algorithms are compared. The BurgerLinker is an existing linking algorithm, specifically designed for Dutch civil data sets. This method links **certificates** that mention the same persons. Also, as part of this thesis, a new method is created, named the RecordLinker. The RecordLinker algorithm features improved name processing and additional linking modes. This algorithm does not just save the linked certificates, but also the linked **person references**. The resulting links of both methods are converted to fit the unique historical individual data structure outlined in 3.1.3. In this thesis, the quality of the unique historical individuals produced by the BurgerLinker and the RecordLinker are compared.

3.1 Data

The impact of inconsistent naming is reduced by filtering the data. In 1811, Napoleon Bonaparte implemented the civil status in the Netherlands [His11]. This legislation required every person to adopt a family name. The civil status helped Napoleon enforce the obligatory military service and led to a more efficient tax system. This significant event is used as the cutoff point for the data, as before 1811, name inconsistency likely negatively impacts the data quality of the results. Further research can be done to ensure this assumption is correct. The assumption is not expected to affect the linking methods comparison done in this study.

The data used in this study is provided by Erfgoed Leiden en Omstreken and contains certificates from 26 former municipalities in the region of Leiden. Some of these municipalities have merged. Hillegom, Katwijk, Kaag en Braassem, Leiden, Leiderdorp, Lisse, Nieuwkoop, Noordwijk, Oegstgeest, Teylingen and Zoeterwoude are the partnered municipalities currently.

3.1.1 Input data

- **314 628 birth certificates from the period 1811-1950**
  Birth certificates contain three **person references**: a mother, a father and a child. The sex of the child is unknown. The contents of these certificates are displayed in Figure 2.
- **100 915 marriage certificates from the period 1811-1950**
  Marriage certificates contain six person references: a groom, a bride, the parents of the groom and the parents of the bride. The contents of these certificates are displayed in Figure 2.
- **2 190 divorce certificates from the period 1811-1993**
  Divorce certificates are not considered in this thesis, due to the BurgerLinker missing the functionality for linking this type of certificate.
- **338 759 death certificates from the period 1811-1995**
  Death certificates contain four person references: the deceased person, their partner, their father and their mother. The contents of these certificates are displayed in Figure 2.

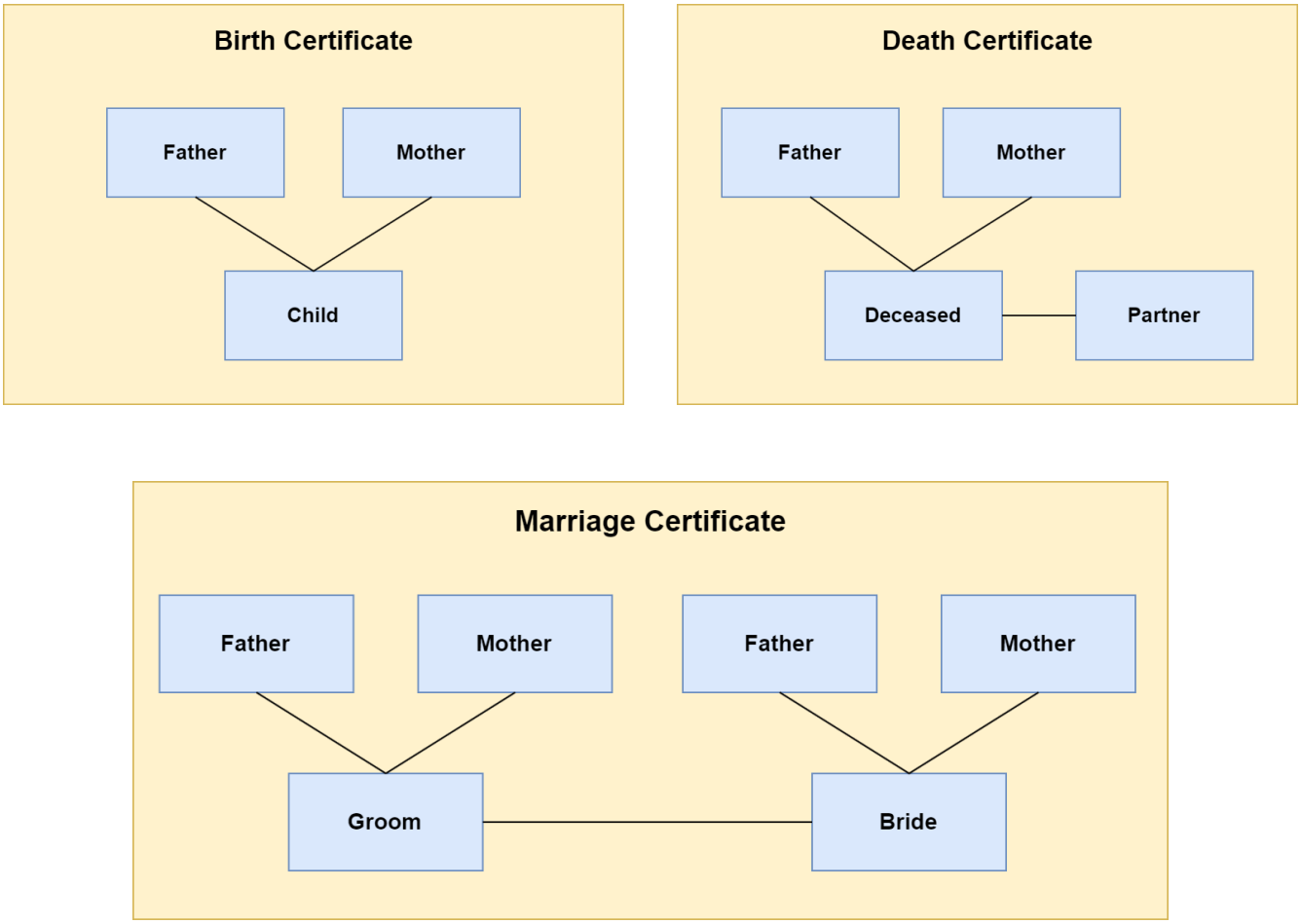


Figure 2: The schematic diagrams of the contents of a birth, marriage and death certificate. Each blue rectangle is a person reference.

Each person reference contains the following fields:

- **uuid** (universally unique identifier)
- first names
- prefix
- family name
- age
- occupation
- place of birth
- date of birth
- place of residence
- comment (for additional information)

Also, each person reference contains information about when the information was added and by whom.

Note that the data is not complete. Some person references are missing. For example, when the father is not known. Additionally, not all the fields of a person reference are filled in. Person references of parents of the married couple are less complete than person references of brides and grooms. For example, the age of a bride is almost always known, but the age of her parents rarely.

3.1.2 Links

Links are described as two uuid's of person references that are found to have a Levenshtein distance smaller or equal to the maximum accepted distance. One of the person references was held as reference and the other was a potential match that is linked. The uuid of the person reference held as reference is the 'reference_uuid'. The linked potential match person reference is the 'link_uuid'. The mode of the linking method and the sex of the individual are also saved. The mode of the linking method entails what type of person references are linked. An example of the data structure can be found in Table 1.

Table 1: Example of the data structure of the generated linked person references. For each linked person reference, the linking mode, person reference uuid's and sex are saved.

mode	reference_uuid	link_uuid	sex
1	0485e974	e7b998a2	m
2	bf31a4e2	7b185532	v

Table 2: Example of the data structure of the generated unique historical individuals. The person references and their corresponding unique person id are saved.

uuid	unique_person_id
0485e974	1
e7b998a2	1
bf31a4e2	2
7b185532	2

3.1.3 Unique historical individuals

Unique historical individuals are described as the uuid of a person reference and the unique integer of the unique historical individual it is linked to. A unique historical individual can contain many person references. The example links in Table 1 are processed into the unique historical individuals in Table 2.

3.2 BurgerLinker

The BurgerLinker is a method for linking Dutch civil records [RMR+20]. The steps to apply this method and to process the results to unique historical individuals are laid out.

1. Formatting data in "format.py"
   The input data, provided by ELO, is transformed into a format that the BurgerLinker accepts.
2. Generating linked certificates with the BurgerLinker
   The BurgerLinker is run with the formatted data to retrieve the linked certificates.
3. Processing linked certificates to linked persons in "processing.py"
   A Python script determines for each linked certificate what persons were linked.
4. Processing linked persons to unique historical individuals in "individuals.py"
   The linked person references are grouped to find all person references that refer to the same person.

Step 4 is described in Section 3.4. This step is the same for both the RecordLinker and the BurgerLinker.

3.2.1 Linking

The input data is formatted into a .csv file containing all certificates and a .csv file containing all person references. Utilising a script, available on the BurgerLinker GitHub page, these .csv files are converted and combined into a RDF file [CLAa]. This file is converted by the BurgerLinker to create a HDT file. The HDT file is used for linking the certificates. The output of the BurgerLinker requires processing before the links are turned into unique historical individuals, because the links are on certificate level, but person links are needed. A Python script is developed to convert the certificate links to person links. This is done by determining what persons were compared on each linked certificate. The resulting person links are used for unique historical individual reconstruction.

Table 3: Table of the linking modes of the BurgerLinker. Each linking mode links a select type of person references.

Mode	Mode name	Reference pairs	Potential link pairs
1	Within_B_M	Newborns	Brides/grooms
2	Between_B_M	Parents of newborns	Brides & grooms
3	Between_D_M	Parents of deceased	Brides & grooms
4	Between_M_M	Parents of brides/grooms	Brides & grooms
5	Within_B_D	Newborns	Deceased
6	Between_B_D	Parents of newborns	Deceased & partners

3.2.2 Modes

The BurgerLinker has six linking modes. These modes dictate what type of person references are used for linking. For example, the BurgerLinker in mode 2 links parents of newborns to brides and grooms. All modes are described in Table 3.

3.3 RecordLinker

RecordLinker is a novel record linking method. The basic steps to generating unique historical individuals are described in this section. The process is described in detail in the subsections.

1. Preprocessing data in "pairs.py"
   The input data provided by ELO is processed into pairs of person references.

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 "individuals.py"
   The linked person references are grouped to find all person references that refer to the same person.

Step 3 is described in Section 3.4. This step is the same for both the RecordLinker and the BurgerLinker.

3.3.1 Pairs

Each certificate mentions people. These people can be divided into **pairs**. A pair describes a man and a woman. The pairs are compared to find similar pairs. Table 4 contains the different types of pairs used for linking. Also, for some types of pairs, a child is known. When a link is found between two pairs, the children are compared as well. Children are incorporated into the pairs to find the birth certificates of unique historical individuals. By not comparing a combination of names when linking children, the likelihood of incorrectly linking two person references increases. The fields for each pair are presented in Table 5.

Table 4: Table of the types of pairs used for the RecordLinker

Type of pair	Person 1	Person 2	Child	Name
1	father	mother	newborn	Parents of newborns
2	groom	bride	-	Married couples
3	father of groom	mother of groom	groom	Parents of married couples
4	father of bride	mother of bride	bride	Parents of married couples
5	deceased	partner	-	Deceased and partners
6	father of deceased	mother of deceased	deceased	Parents of deceased

Table 5: Table of the fields for each pair

Field	Description
year	The year the event took place
first_letters	First letter of the last name of the man of the woman
pair	Type of pair (see Table 4)
man	The name of the man
woman	The name of the woman
child	The name of the child
age	Either the age of the bride or groom, depending on the type of pair
uuid	The uuid of the certificate
man_uuid	The uuid of the person reference of the man
woman_uuid	The uuid of the person reference of the woman
child_uuid	The uuid of the person reference of the child

3.3.2 Name processing

The input data contains fields for first name, prefix and last name. 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 is cleaned up individually
3. First names are ordered alphabetically and the last name is appended without spaces

The cleaning up removes capital letters, removes accents and other diacritics, only keeps symbols in the alphabet and replaces:

- ch for g
- c for k
- z for s
- ph for f
- ij for y

These letters are replaced to help standardise the names. These letters are historically used inconsistently and can therefore lead to a larger Levenshtein distance [Blo14]. More links can possibly be found by standardising the letters in this way.

Example: Mariä Anña van 't Schip → annamariasgip

3.3.3 Linking

The persons, described by the pairs, are compared to find similar pairs. A list of pairs of the selected type is initiated by the RecordLinker. These are the reference pairs. For each reference pair, a selection of potentially linking pairs is made before linking. By restricting the number of potential links, the algorithm compares fewer pairs, leading to a reduced run time. This method can also lead to fewer mistakes, as pairs should not be linked that fall in the wrong period of time. For example, a person cannot marry before being born. Disregarding pairs that break this rule, prevents the algorithm from incorrectly linking them. The downside, however, is the possibility of missing links that would have been found, had some pairs not been filtered out. The selection of potential links is done in the three steps described below.

1. Select all pairs with the corresponding type of potential link pair
2. Filter out all pairs that do not have matching first letters as the reference pair
3. Pairs are filtered out that are documented outside of the determined period

Table 6: Table of the linking modes of the RecordLinker. Each linking mode links a select type of person references.

Mode	Reference pairs	Potential link pairs
1	Parents of newborns	Parents of newborns
2	Parents of newborns	Parents of deceased
3	Married couples	Parents of newborns
4	Married couples	Parents of married couples
5	Married couples	Parents of deceased
6	Parents of married couples	Parents of newborns
7	Parents of married couples	Parents of married couples
8	Parents of married couples	Parents of deceased
9	Deceased and partners	Parents of newborns
10	Deceased and partners	Married couples
11	Deceased and partners	Parents of married couples
12	Deceased and partners	Deceased and partners
13	Deceased and partners	Parents of deceased
14	Parents of deceased	Parents of deceased

The RecordLinker has 14 different linking modes. Each linking mode links different types of pairs. Table 6 shows for each mode what types of pairs are linked. For each pair comparison, the names of the reference pair are compared to the names of the potential link pair.

To help address the data quality problem, a string comparison method is used to compare the names instead of simply checking if they are equal [Blo14]. The possible approaches are outlined in Section 2.2, namely the Levenshtein distance and the Jaro-Winkler similarity. The Levenshtein distance is an absolute measure, whilst the Jaro-Winkler similarity is a relative measure. The relative measure incorporates the length of the words. With a relative string comparison measurement, longer names would be allowed more differences. This is a problem when linking persons with long last names and short first names, as family members can be incorrectly linked. An absolute measure is more fitting for the purpose of this thesis. This is why the Levenshtein distance is used in the RecordLinker.

The Levenshtein distance finds the number of edits that need to happen for one name to change into another. The usage of this method entails that, even with a mistake in names, a link can still be found. If the distance is smaller or equal to the maximum accepted distance, the pairs are deemed similar and are linked. The maximum accepted distance is explored in Section 3.6. In the case of a link, the children are also compared. If deemed similar, the person references of the children are linked. This is done to find the birth certificates of unique historical individuals.

3.3.4 Algorithm

The logic of the RecordLinker algorithm is laid out in Algorithm 1. The input of the algorithm are the pairs of person references, as described in Table 5. The output are linked person references, as described in Section 3.1.2.

Algorithm 1 The RecordLinker Algorithm

```
Require: 0 < mode ≤ 14
  reference_pairs ← ReferencePairs(pairs, mode)
  potential_links ← PotentialLinkPairs(pairs, mode)

  for reference_pair in reference_pairs do
      potential_links ← FilterPotentialLinks(reference_pair.firstletters, potential_links)
      potential_links ← FilterPotentialLinks(reference_pair.period, potential_links)

      for potential_link in potential_links do                    ▷ Compare the pairs
          distance ← Levenshtein(reference_pair.names, potential_link.names)
          if distance ≤ MAX_DISTANCE then
              SaveLink(reference_pair.person1, potential_link.person1)
              SaveLink(reference_pair.person2, potential_link.person2)

              distance ← Levenshtein(reference_pair.child, potential_link.child)
              if distance ≤ MAX_DISTANCE then                     ▷ Compare the children
                  SaveLink(reference_pair.child, potential_link.child)
              end if
          end if
      end for
  end for
```

3.4 Life Course reconstruction

The links generated by the BurgerLinker and the RecordLinker need to be processed to create unique historical individuals. In essence, each cluster of links is saved as a group of uuid's that mention the same person. A group of uuid's defines the unique historical individual. Each uuid refers to an event. These events can be displayed on a timeline to create a life course. The technology used for processing the links into groups is the Union Find algorithm, also referred to as the disjoint-set algorithm. This method produces a data structure from which groups of connected nodes can be derived. These groups are linked uuid's that are given a unique person identifier, as explained in Section 3.1.3. The Union Find algorithm is explained in more detail in Section 3.4.1.

3.4.1 Union Find algorithm

The Union Find algorithm is a method for finding groups of connected nodes in an undirected graph. This method can be applied to the links, generated by the BurgerLinker or the RecordLinker, by structuring them in a graph. Each person reference is a node, represented by the uuid. Each link between person references is an edge in the undirected graph.

The Union Find algorithm works by assigning a root node to each node [GI91]. In the beginning, each node is its own root node. When looping through each edge, the root nodes are adjusted so that each node in a group of linked nodes has the same root node. If two nodes have the same root node, they are connected. The Union Find algorithm consists of two functions; a 'union' function and a 'find' function. The 'find' function recursively finds the root node of a node. The 'union' function reassigns the root node of a node in an edge, essentially merging the groups.

This algorithm is time efficient, so it is ideal for processing the large networks created by the linking methods. Each group of connected person references is saved to a file in the data structure explained in Section 3.1.3.

3.5 Comparing methods

A few metrics are evaluated to compare the quality of the unique historical individuals produced by the BurgerLinker or the RecordLinker. These metrics are group size, life course duration, life course status and coverage. These quality measurements are explained in the following subsections.

3.5.1 Group size

First, the amount of linked uuid's per unique historical individual is evaluated. The size of these groups of uuid's can say something about the quality of the groups. Large groups can be an indication of mistakes. Mistakes combine groups, causing an increase in group size. There is a practical limit for group size, as a person can have a limited amount of children. Certificates involving events related to children make up a big portion of the larger groups. This is because birth, marriage and death take up at most four linked person references, assuming a person does not marry more than once 1. This makes it extremely unlikely that a group size of 100 is correct. Past a certain group size, it becomes more unlikely a group is correct as group size increases.

3.5.2 Life course duration

The time between the first event and the last event in a life course can be evaluated to measure quality. If this life course duration does not match expectations, it can be an indication of mistakes in the links. A person can not live forever. If there's a link between two certificates separated by 150 years, it's probable that the link is incorrect. Some types of person references are not considered when calculating the life course duration, as these can be registered long after the person has died. For example, the parents of deceased are not added to the duration calculations, as these persons are likely already deceased. The duration of a life course is not equal to the age of the person. It is the documented portion of their life. If the only linked references are a person marrying at age 25 and dying at age 75, their life course duration is 50 years.

3.5.3 Life course status

Additionally, each life course is given a classification. The classification is based on the number of births and deaths the life course contains. Every unique historical individual is born once and has died once. Life courses that follow this principle, are referred to as 'complete'. Complete in quotes, because it is impossible to know if a life course is truly complete. There can always be another child born or married that was not found. Birth and death are the only events to have happened to every unique historical individual exactly once.

However, not every life course is expected to be 'complete'. The life courses are based on data from a certain time period. This means that, for some life courses, the birth was before the start of this period. For others, the death was after the end of the period. This should be reflected in the results by producing a peak in life courses without a birth at the beginning of the time period and a peak in life courses without a death at the end of that period. Also, the data only contains certificates from the region Leiden. It is possible some people are born outside this region, resulting in life courses without a birth event.

Life courses can also be classified as incorrect. An incorrect life course is defined as a life course containing more than one birth or more than one death. This is a strong indicator of mistakes in the links, because people can not be born twice or die twice. The incorrectness score, which can be derived from the classifications, is a measure of link quality. The incorrectness score is the portion of life courses that are classified as incorrect. The calculation is described in Equation 1.

$$Q_{incorrect} = \frac{L_{incorrect}}{L} \tag{1}$$

$Q_{incorrect}$ is the incorrectness score
$L_{incorrect}$ is the number of incorrect life courses
L is the number of total life courses

3.5.4 Coverage

Coverage is evaluated to compare the completeness of the life course reconstruction. The term coverage refers to how many of the total person references were linked to unique historical individuals. The more coverage, the more representative the results are for the entire population. More importantly, the higher the coverage, the more likely a certificate in a life course is missing, because the data is missing and not the link. Coverage is not a direct measure of quality, but it gives context to the other measurements. The coverage is calculated per type of person reference to get an overview of the results. The calculation for coverage is described in Equation 2.

$$C = \frac{P_{linked}}{P} \tag{2}$$

C is the coverage
P_{linked} is the number of person references that are linked
P is the number of total person references in the data

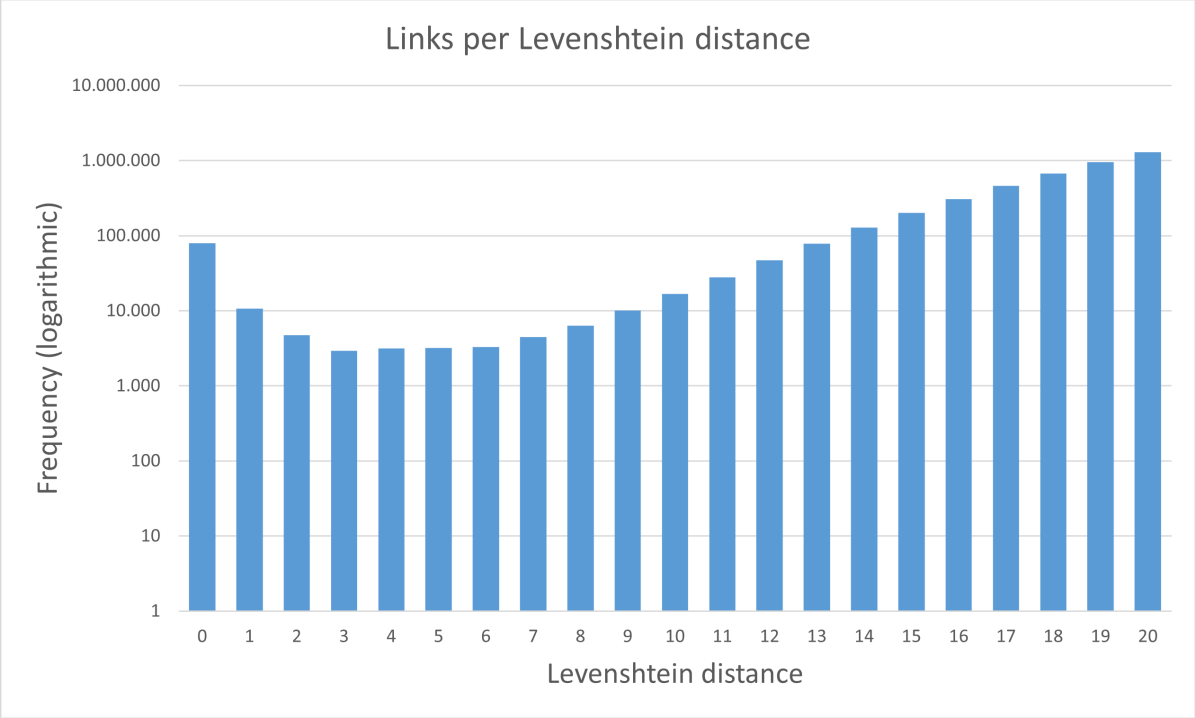


Figure 3: The frequency of links found by the RecordLinker (mode 4) per Levenshtein distance. The frequency is displayed on a logarithmic scale.

3.6 Maximum Levenshtein distance

To determine what maximum Levenshtein distance is most suitable for this thesis, the ratio of correct links versus incorrect links is estimated for each value of maximum Levenshtein distance. To estimate this ratio, the RecordLinker is run in mode 4 (linking 'married couples' to 'parents of married couples') with a maximum Levenshtein distance of 20. This value produces many incorrect links, but allows for the creation of a histogram displaying the frequency of links for each Levenshtein distance in the range of 0-20. This histogram is shown in Figure 3 on a logarithmic scale.

The y-axis of this histogram has a logarithmic scale. Links with a Levenshtein distance of 0 have identical names. It is expected that these links have a high ratio of correct links. On the other hand, links with a Levenshtein distance of 20 have a low ratio of correct links, because the names are not similar. By assuming links with a Levenshtein distance greater than 6 have an insignificant amount of correct links, an estimated amount of incorrect links can be calculated. A clear exponential trend is found at a Levenshtein distance between 7 and 20. By applying curve fitting, an equation is calculated that estimates the number of incorrect links at the lower Levenshtein distances. This relies on the assumption that the exponential trend is consistent. The equation for estimating the number of incorrect links is based on the least squares method [Bjö90]. The Python package 'NumPy' and its function 'polyfit' are used for constructing Equation 3.

$$N_{incorrect} = 189.18e^{0.4544x} \tag{3}$$

$N_{incorrect}$ is the estimated number of incorrect links
x is the Levenshtein distance

The number of correct links can be estimated for each Levenshtein distance by subtracting the estimated number of incorrect links from the total number of links. This is displayed in Equation 4.

$$N_{correct} = N - N_{incorrect} \tag{4}$$

$N_{correct}$ is the estimated number of correct links
N is the total number of links
$N_{incorrect}$ is the estimated number of incorrect links

Figure 4 shows the estimated number of incorrect and correct links per Levenshtein distance on a logarithmic scale. This can be deceiving, as the perception is warped. By calculating the ratio of correct links, the estimated quality of links per distance can be better visualised. This is done in Figure 5. Note that the assumption was made that links with a Levenshtein distance greater than 6 have a correctness ratio of 0.

Figure 5 shows that the highest Levenshtein distance that results in more correct links than incorrect links is 4. However, this does not mean that this value is most suitable for this research. The negative effects of incorrect links need to be considered. An incorrect link has a negative impact on the quality of unique historical individuals, as it can combine two correct groups together, making both groups incorrect. This is why the assumption of incorrect and correct links having equal weight on the quality of unique historical individuals is incorrect.

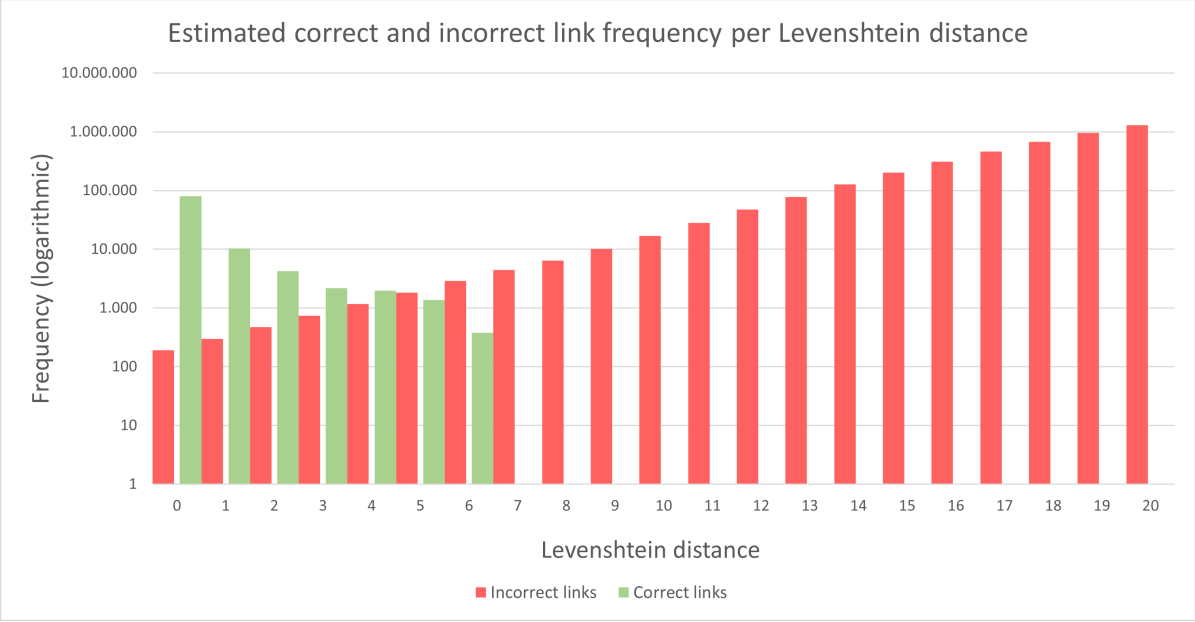


Figure 4: The estimated frequency of correct and incorrect links per Levenshtein distance. The frequency is displayed on a logarithmic scale.

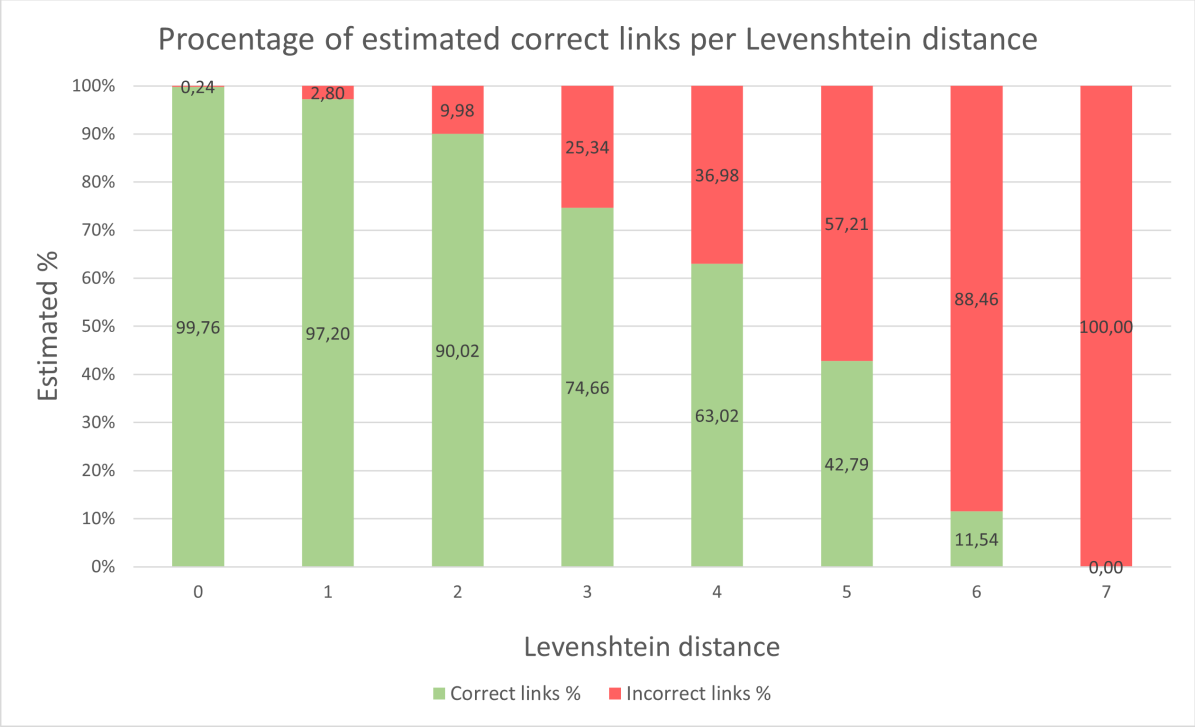


Figure 5: The estimated ratio of correct and incorrect links per Levenshtein distance.

In this thesis, the chosen maximum accepted Levenshtein distance is 3. This value is based on the assumption that the number of correct links gets smaller as the Levenshtein distance increases. Figure 4 shows that after a distance of 3s the number of links found gets larger. This violates the previously mentioned assumption and is the reason the maximum accepted Levenshtein distance of 3 is chosen. Additionally, this value corresponds to a high ratio of estimated correct links, as shown in Figure 5. There is precedent for using a Levenshtein distance of 3, as this value was chosen by M. Schraagen when he explored the optimum maximum accepted Levenshtein distance [Sch14].

4 Results

In this section, the statistics and graphs about the generated unique historical individuals are laid out. The green graphs show the RecordLinker results whilst the blue graphs show the BurgerLinker results. Also, the results are compared based on the methods explained in Section 3.5.

General information about the generated unique historical individuals can be found in Table 7. The RecordLinker has generated more unique historical individuals, as well as more linked person references in total. This doesn't provide extensive insight into the result's quality, but it can influence the decision-making process when selecting the more suitable linking method for the use case of ELO. The average group size is smaller for the RecordLinker results. This can be an indication of quality, as explained in Section 3.5.1. The results of group size are explored further in Section 4.1.

The groups of linked person references of the BurgerLinker and the RecordLinker are compared to find common groups. 72 869 common groups are found. These groups are exactly the same. This is 20.68% of the total unique historical individuals found by the RecordLinker and 36.07% of the total unique historical individuals found by the BurgerLinker. The Venn Diagram of this is displayed to scale in Figure 6.

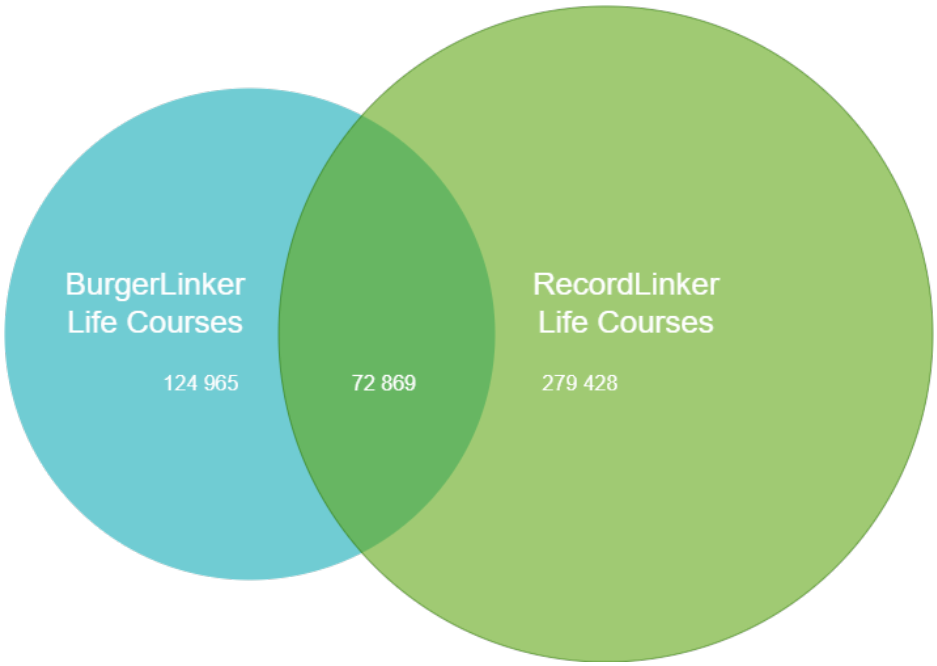


Figure 6: The Venn Diagram of the life courses for the BurgerLinker and the RecordLinker to scale.

Table 7: Table of general information about the results of the generated unique historical individuals per linking method.

	RecordLinker	BurgerLinker
Linked person references	2 119 436	1 382 585
Unique Historical Individuals	352 297	197 834
Average group size	6.02	6.99

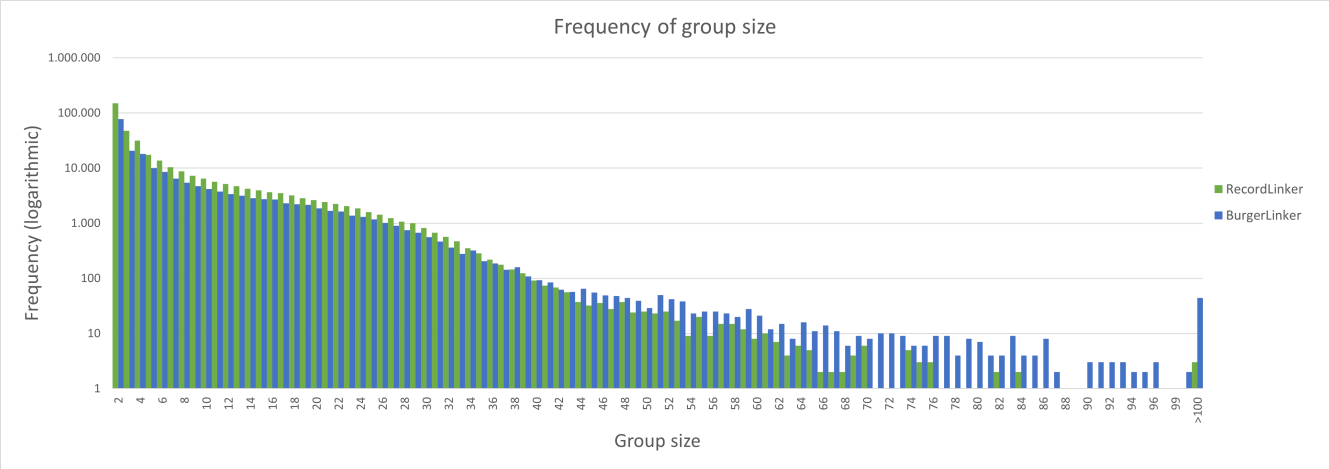


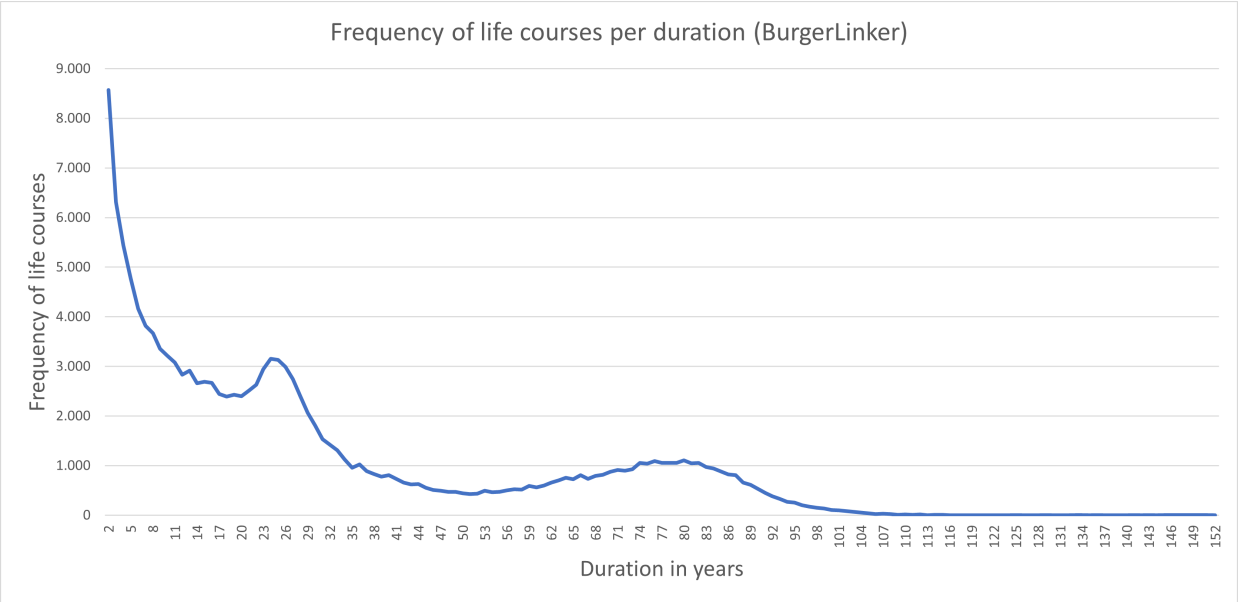
Figure 7: The frequency of group size for the RecordLinker and the BurgerLinker life courses. The frequency is displayed on a logarithmic scale.

4.1 Group size

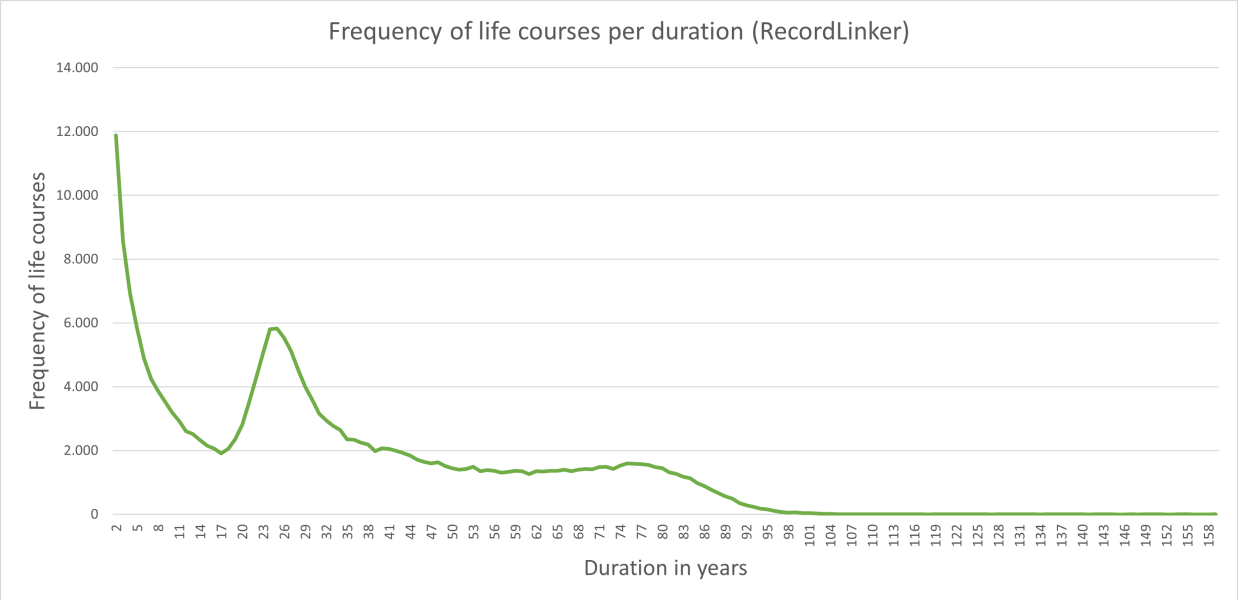
As mentioned in Section 4, the BurgerLinker results have a higher average group size than the RecordLinker results. A more detailed view of the group sizes can be found in Figure 7. This graph displays the histograms of the group sizes for each linker on a logarithmic scale. The shapes of the graphs are similar, however, the RecordLinker generates less large groups and more smaller groups. This is an indicator of better quality life courses.

4.2 Life course duration

By visualising the duration of the life courses, mistakes can become visible. The average duration is 22.6 years for the BurgerLinker life courses and 20.6 years for the RecordLinker life courses. The frequencies of the life course durations are plotted in Figure 8. The duration of many life courses is just a few years. This skews the average. For both linking methods, a spike is seen around 25 years and the frequency starts tapering off around 80 years. This spike can be attributed to missing death certificates. The combination of a birth certificate and a marriage certificate gives a duration of approximately 25 years. This metric does not show a clear difference in quality between the two methods.



(a) BurgerLinker



(b) RecordLinker

Figure 8: The duration in years per life course

4.3 Life course status

The number of life courses is plotted per year and per linking method in Figure 9. This is calculated by determining, for each life course, the first and last year a person reference was found. A life course is included in the sum of life courses for each year in the period between the first and last person reference. However, not every type of person reference is included for determining the first and last person reference. Some types of person references can be registered long after the person has died. These types of references are 'child died', 'child married' and 'partner died'. Observation shows that, although the shapes of the graphs are similar, the RecordLinker method resulted in more persons alive for each period.

As discussed in Section 3.5.3, each life course is also given a status based on the number of linked birth and death certificates. The following life courses statuses are possible:

- **complete**: 1 birth and 1 death
- **only birth**: 1 birth and 0 deaths
- **only death**: 0 births and 1 death
- **no birth/death**: 0 births and 0 deaths
- **incorrect**: >1 births or >1 deaths

Note that a life course always consists of two or more person references, because there is at least one link. This implies that 'only birth' life courses include more than just a birth event; for instance, they might also involve a marriage event, albeit with a missing death certificate.

The life course status results are displayed in Figure 10. The RecordLinker produced more 'complete' life courses and fewer incorrect life courses. Remarkable is how the BurgerLinker only produced 459 life courses with one death certificate and no birth certificates.

The status results are also displayed per year. These results are visualised in Figure 11. Figure 12 displays the values stacked. Observation shows a peak in 'only birth' life courses for both methods as expected. 'Only death' life courses peak close to the starting year cutoff for the RecordLinker as expected. The BurgerLinker results do not show this. Note that the area of these graphs is not solely related to the number of life courses, but also their duration.

A life course can not contain more than one birth or more than one death. This rule is described in Section 3.5.3. In case this rule is broken, the life course is labelled as incorrect. For the BurgerLinker, 49 414 life courses are incorrect. For the RecordLinker, this number is 44 769. Considering the RecordLinker generated 74.41% more total life courses, the proportion of life courses labelled as incorrect is significantly lower for the RecordLinker. 24.46% of the total life courses are incorrect for the BurgerLinker, whilst this is 12.71% for the RecordLinker.

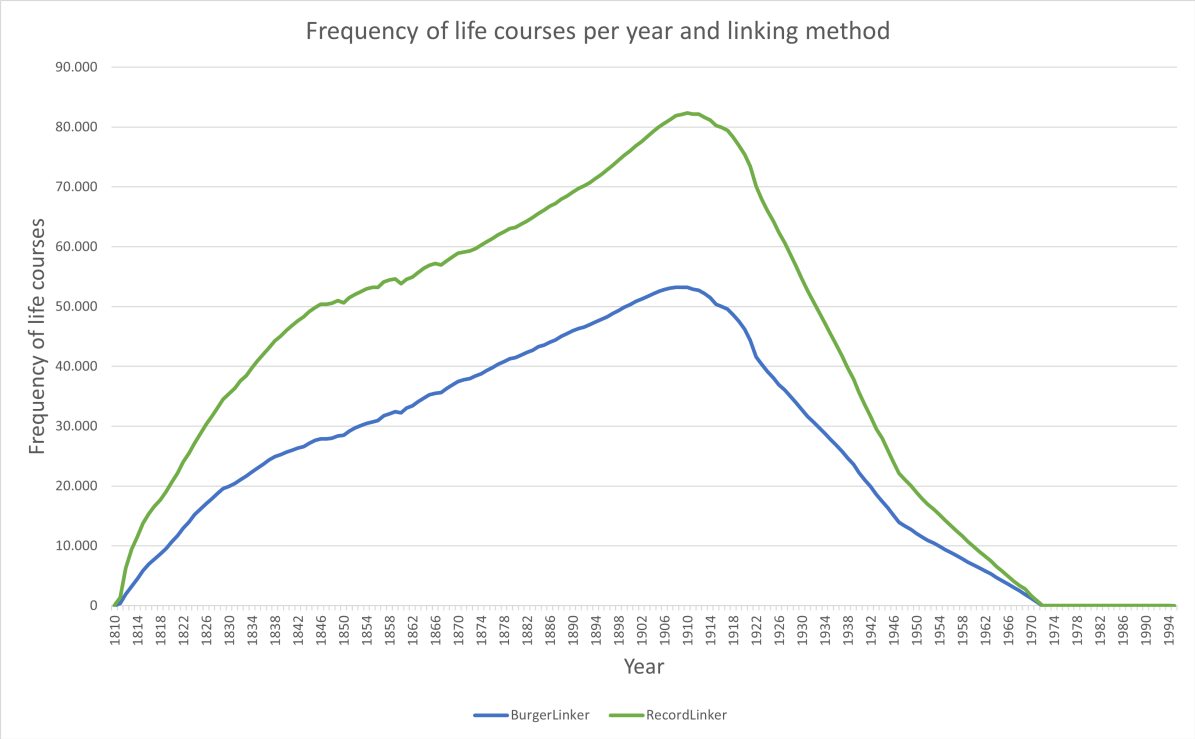


Figure 9: The amount of life courses alive over time for each linking method.

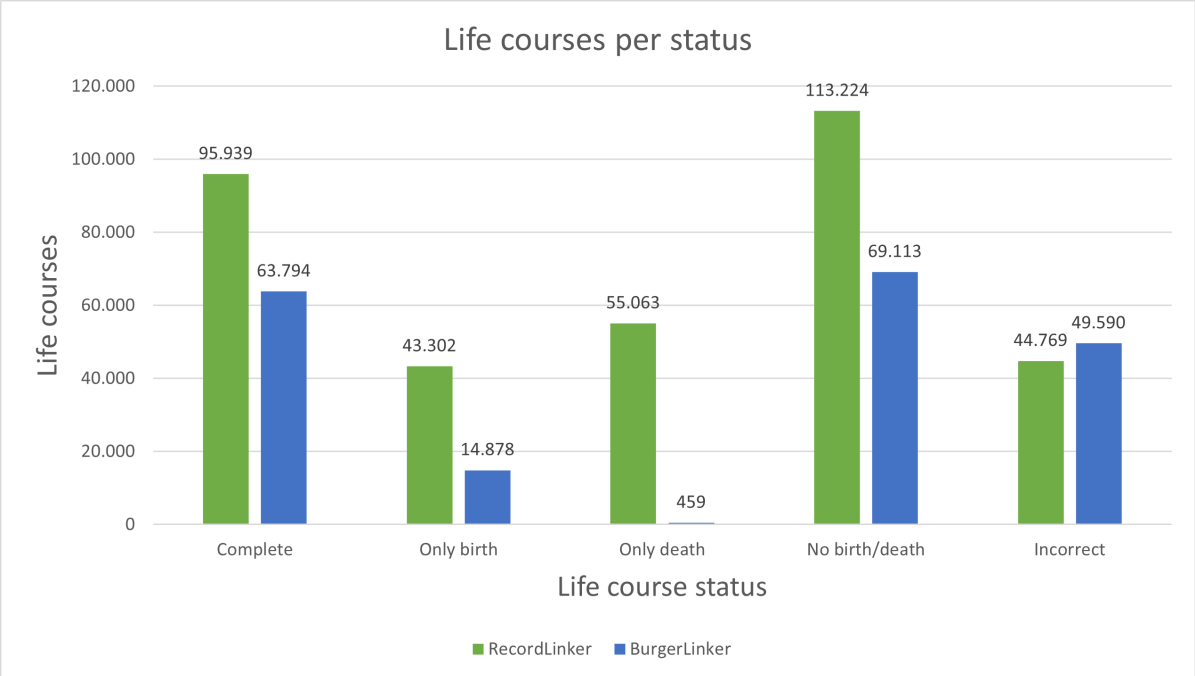
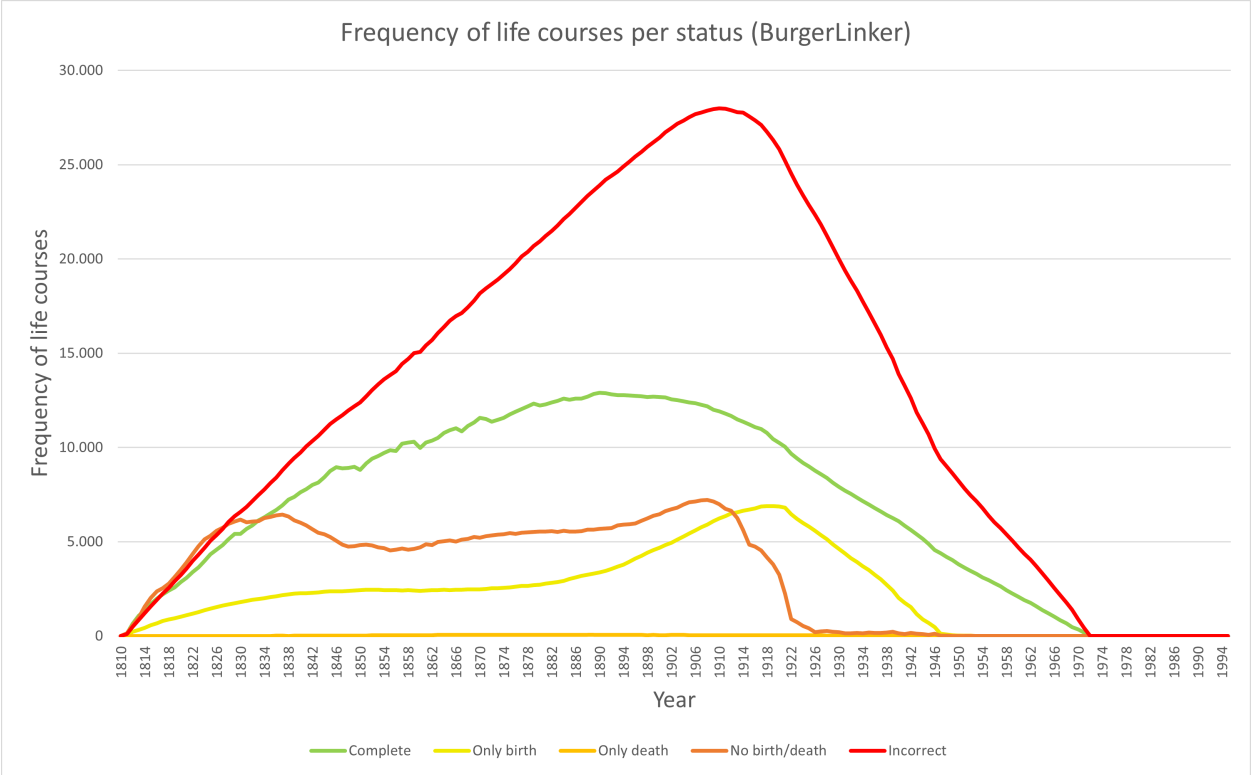
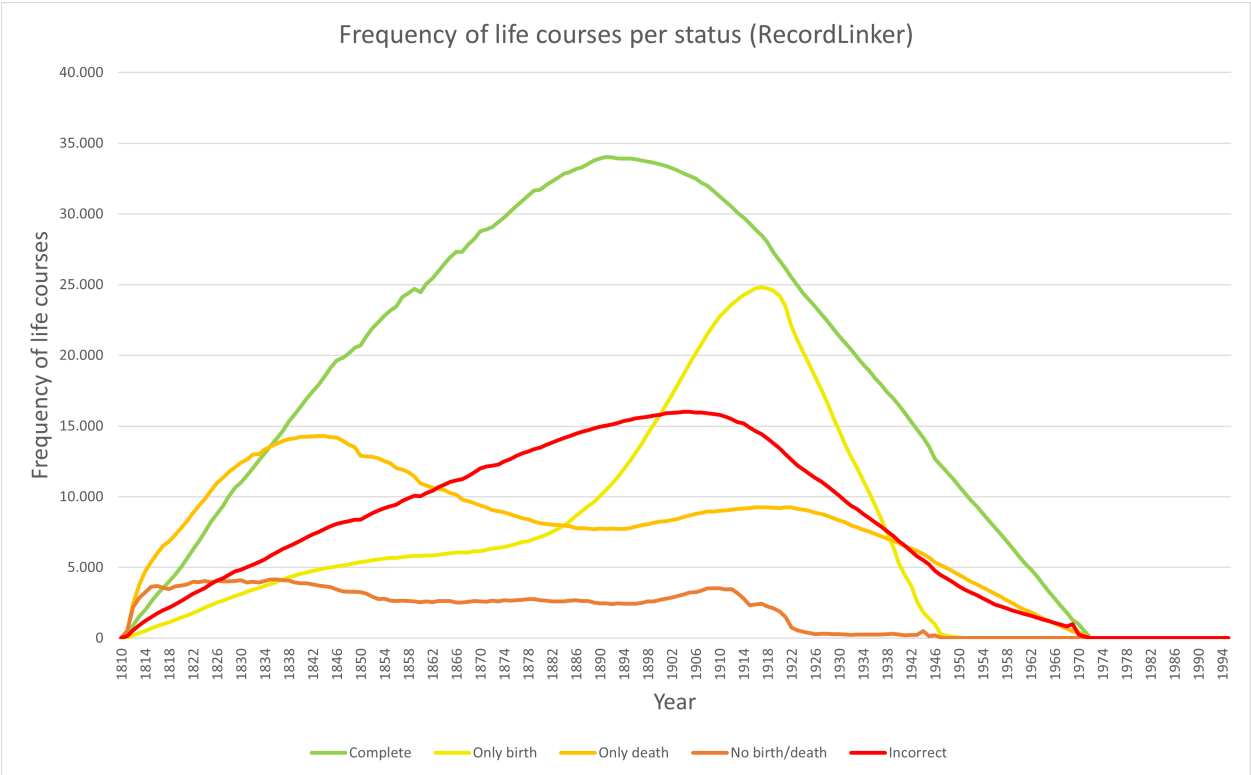


Figure 10: The amount of life courses per status for each linking method.

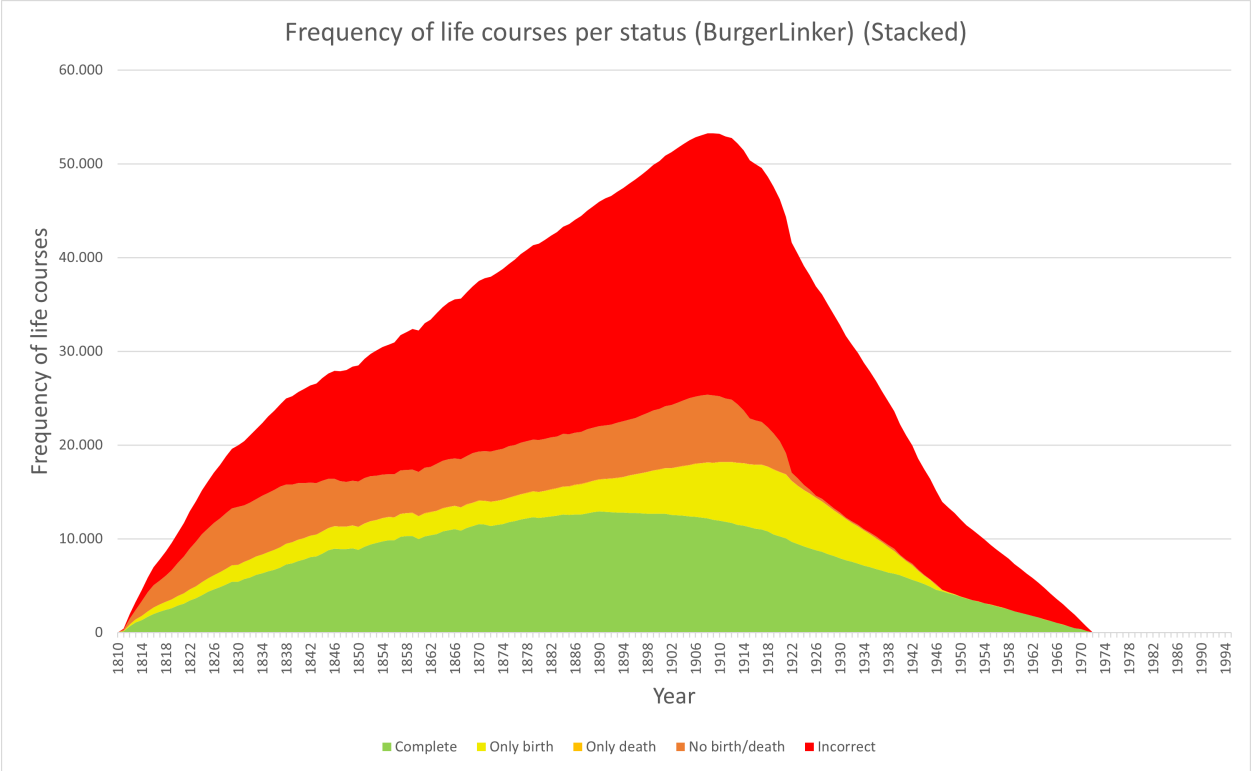


(a) BurgerLinker

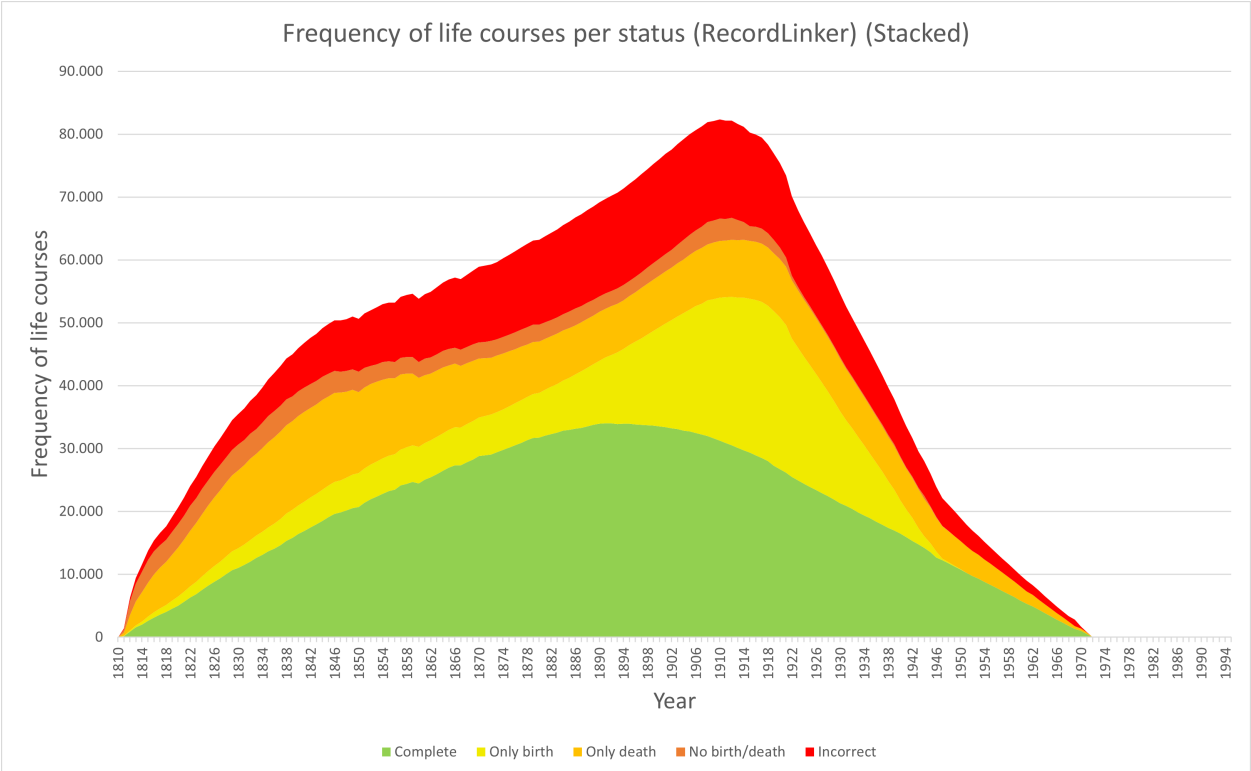


(b) RecordLinker

Figure 11: The amount of life courses over time per status.



(a) BurgerLinker



(b) RecordLinker

Figure 12: The amount of life courses over time per status. The values are stacked.

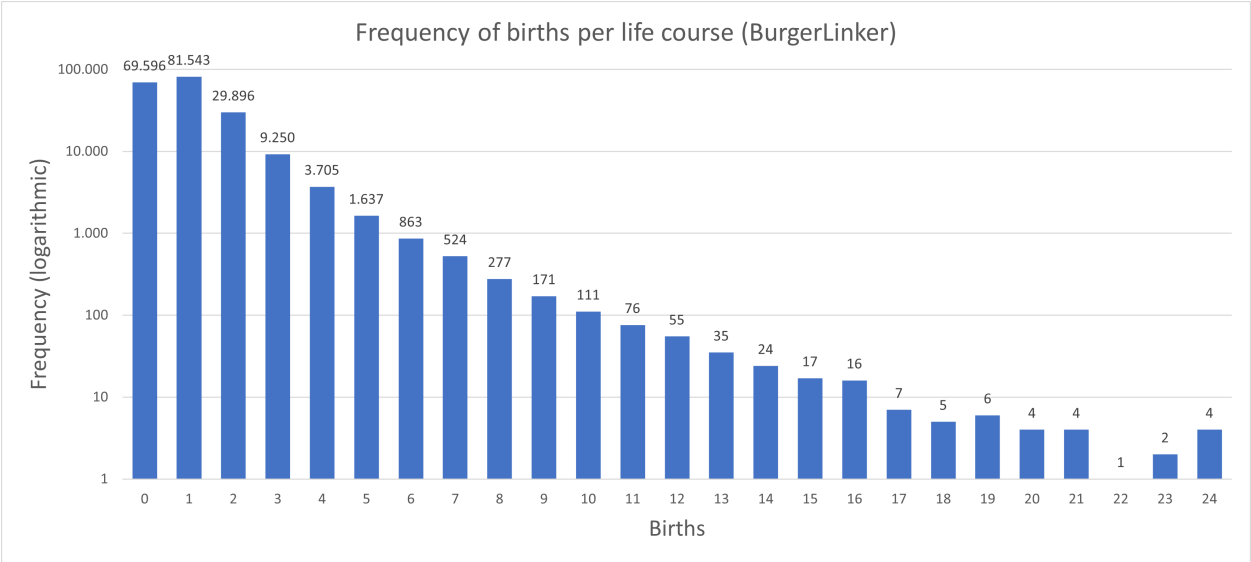
Histograms of the number of births and deaths per life course are displayed in Figure 13 and Figure 16. The frequencies are displayed on a logarithmic scale. The graphs do not just show that the BurgerLinker results contain more life courses with multiple births or deaths. They also show that the life courses are more incorrect. The maximum number of births is 57 for the BurgerLinker whilst it is 12 for the RecordLinker.

The averages of births and deaths reflect how the BurgerLinker life courses are more incorrect. These numbers are displayed in Table 8. Remarkable is that the average of births per life course is greater than one for the BurgerLinker life courses. The average number of births and deaths per incorrect life course is also provided. These values are the averages excluding the values 0 and 1.

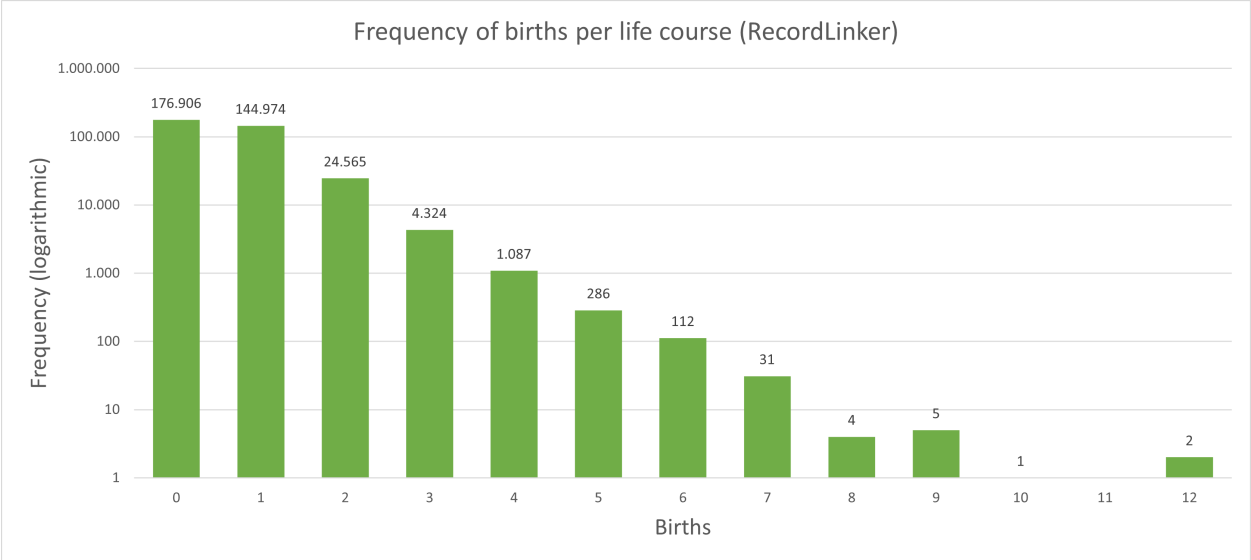
Additionally, life courses with exactly one birth are tested to see if birth is the first event in their timeline. This should always be the case. If not, there has been a mistake. 144 974 of the RecordLinker life courses contain exactly one birth. 1 968 of these life courses do not have a birth as the first event. This corresponds to 1.357% of the life courses with one birth. For BurgerLinker, this is 0.015% out of 81 543 life courses with one birth. This can be an indicator that the BurgerLinker results are of higher quality. Though, the RecordLinker produced more total life courses with one birth.

Table 8: Table of the average number of births and deaths per life course.

	RecordLinker	BurgerLinker
Average births per life course	0.61	1.04
Average deaths per life course	0.66	0.76
Average births per incorrect life course	2.26	2.74
Average deaths per incorrect life course	2.24	2.66

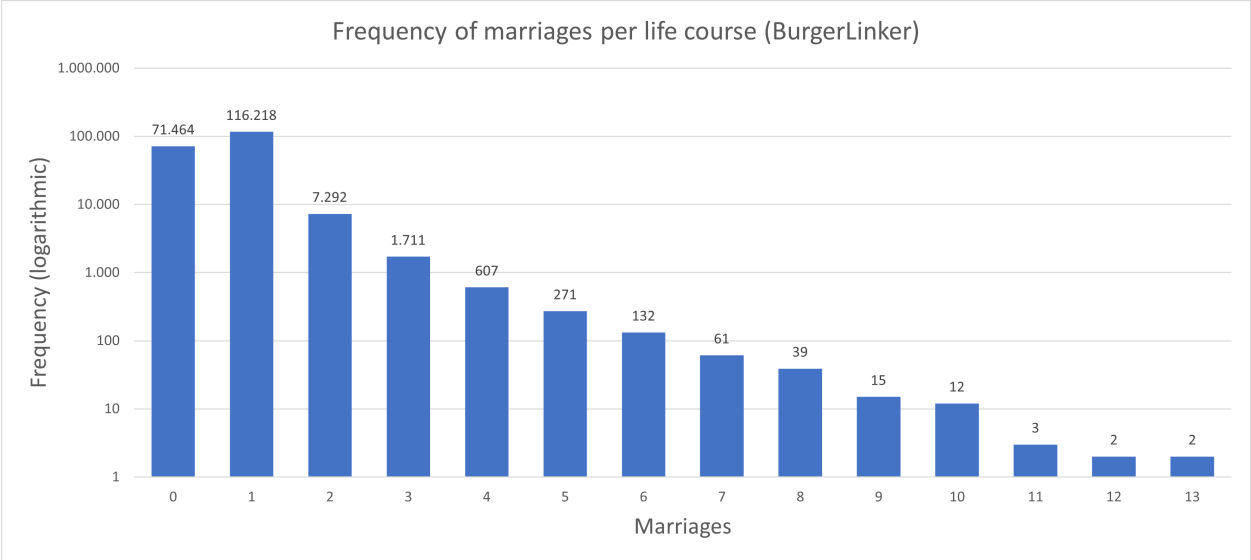


(a) BurgerLinker

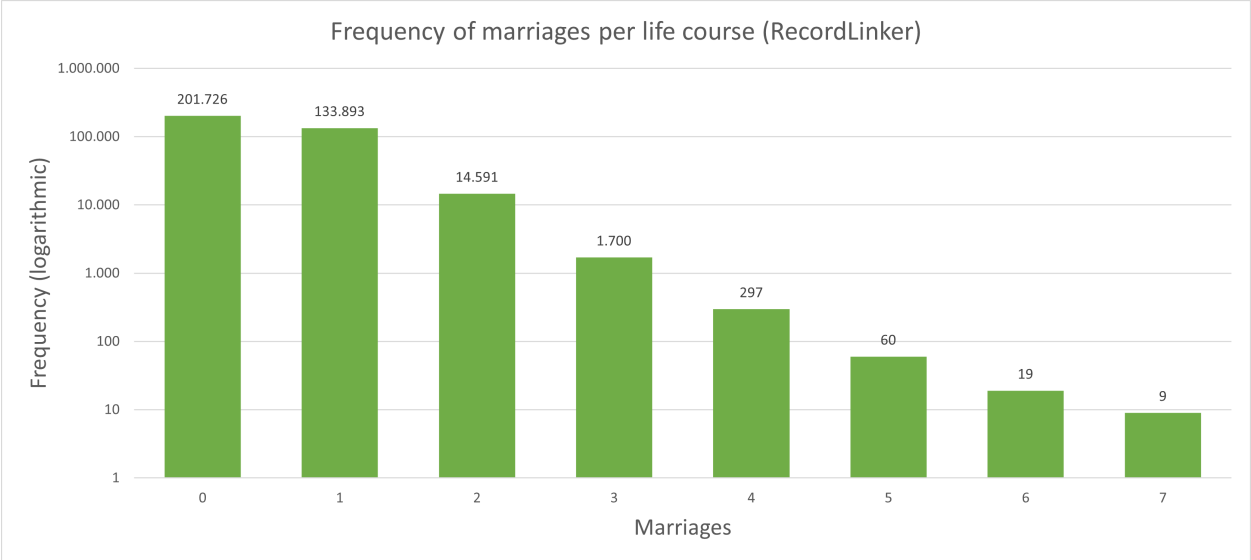


(b) RecordLinker

Figure 13: Histogram for the number of births per life course.

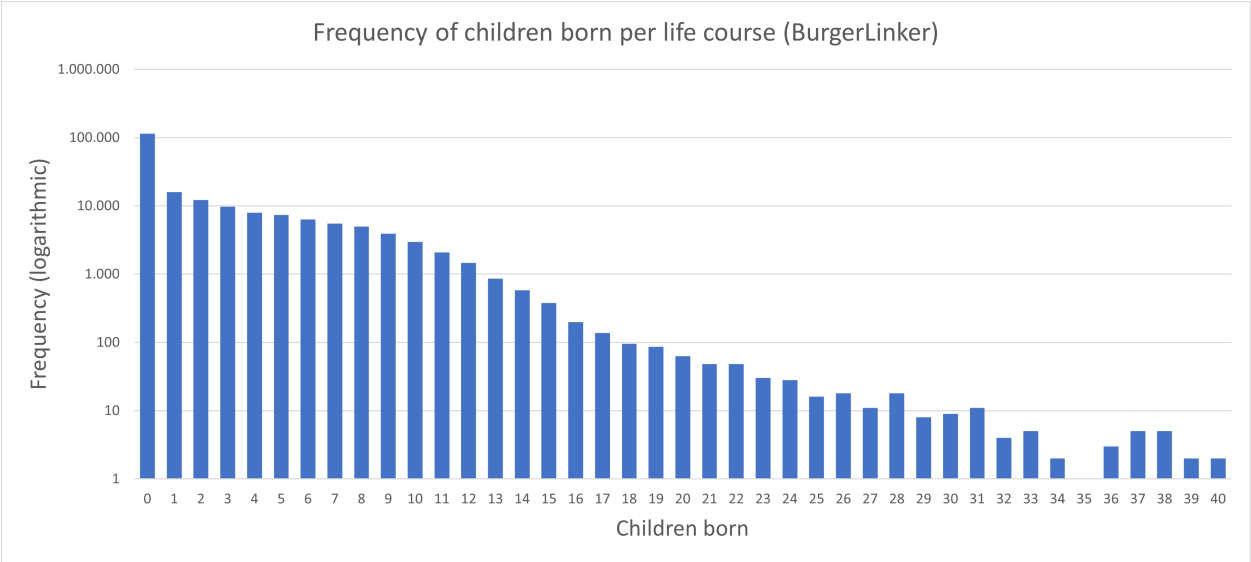


(a) BurgerLinker

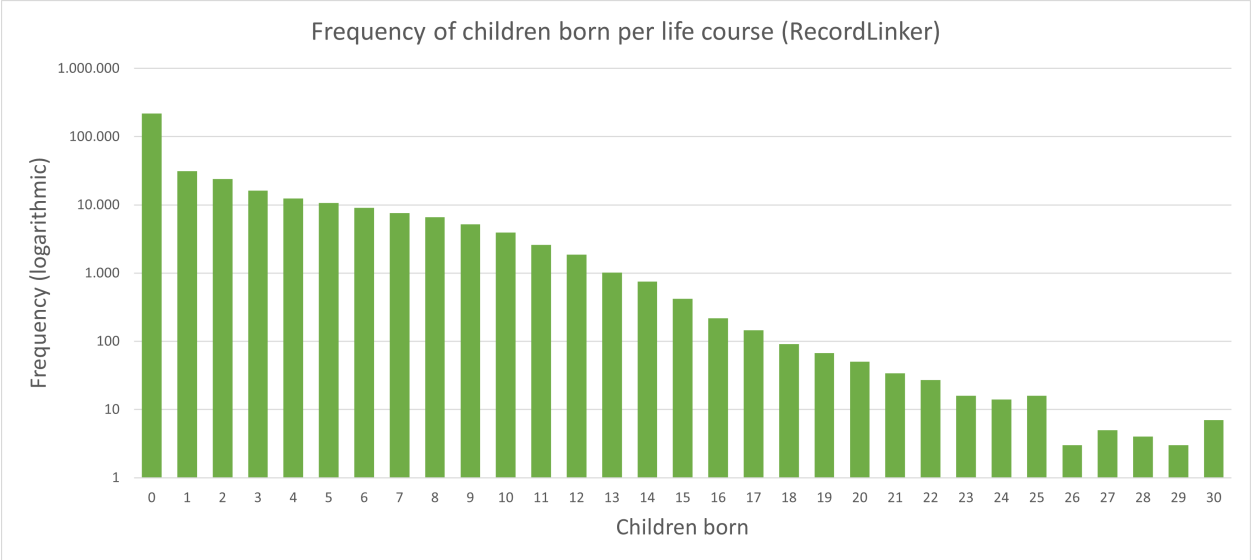


(b) RecordLinker

Figure 14: Histogram for the number of marriages per life course.

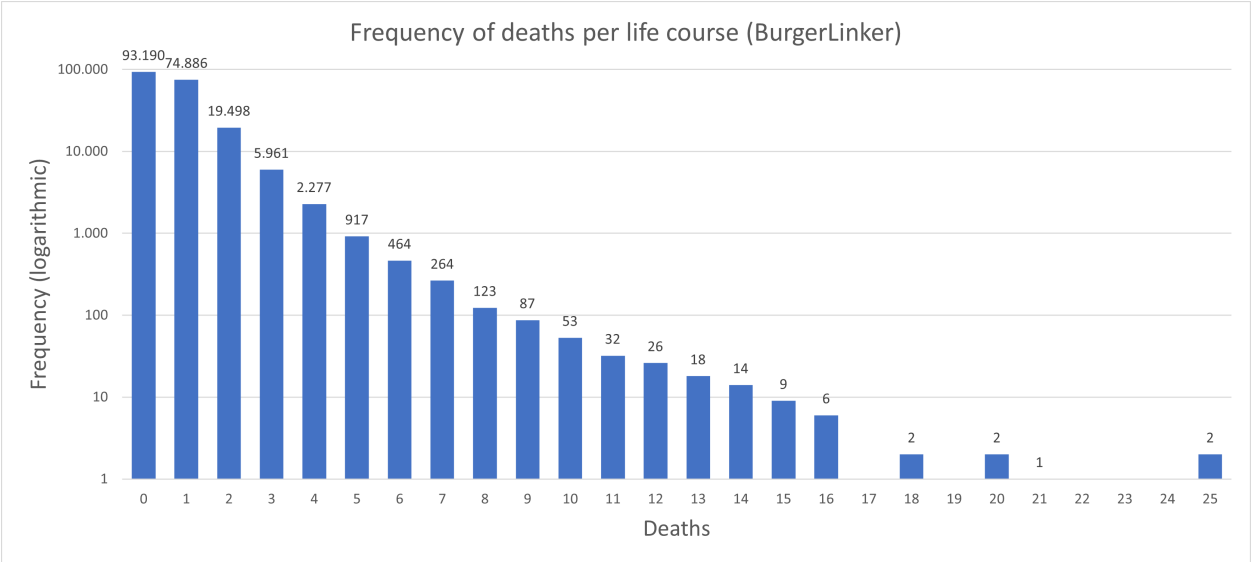


(a) BurgerLinker

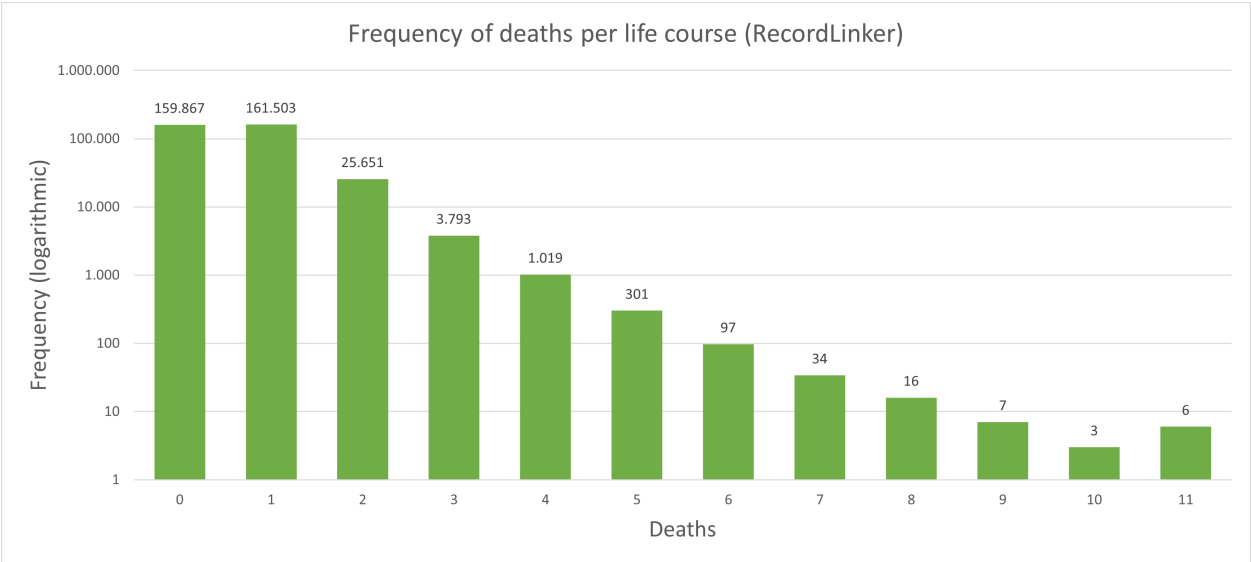


(b) RecordLinker

Figure 15: Histogram for the number of children per life course.



(a) BurgerLinker



(b) RecordLinker

Figure 16: Histogram for the number of deaths per life course.

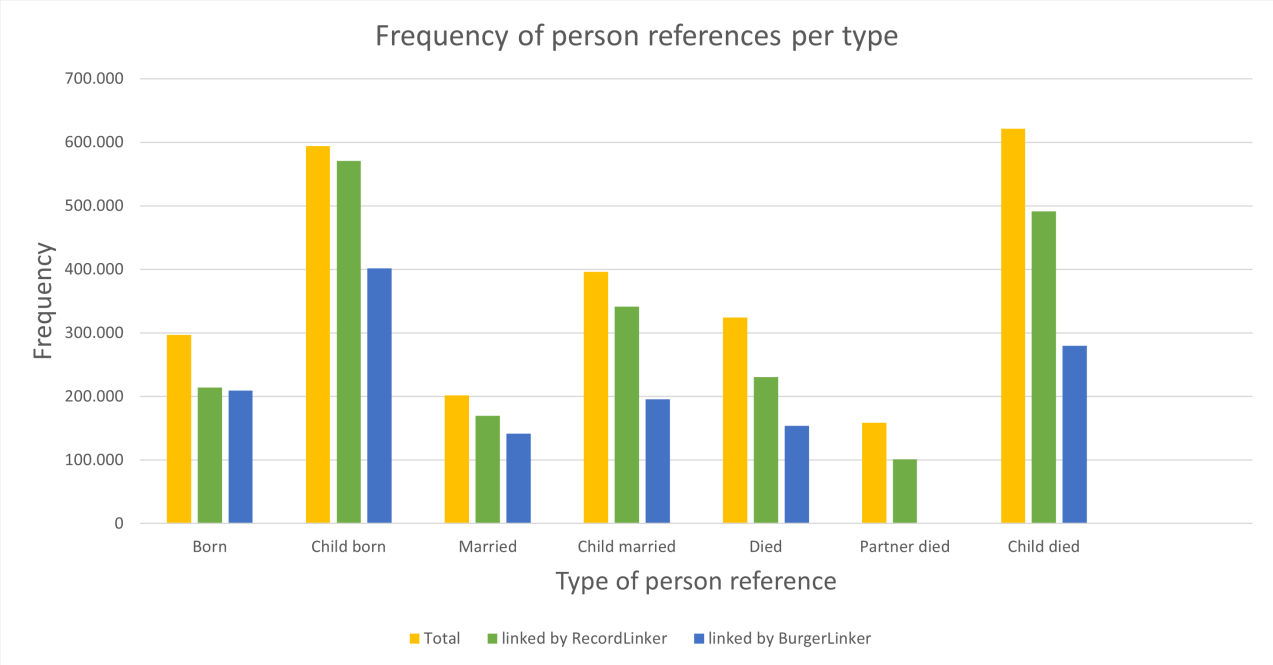


Figure 17: The amount of person references linked by each linking method per type of reference. The total number of person references in the input data is displayed also.

4.4 Coverage

The results showed that the RecordLinker linked more person references. Figure 17 visualises this finding per type of reference. Table 9 shows the same data, but in greater detail. The RecordLinker linked more person references for each type of reference. In other words, the RecordLinker has better coverage than the BurgerLinker. This means that for the RecordLinker, a missing certificate in a life course is more likely to be explained by missing data, rather than a missing link. Note that the BurgerLinker linked 0 died partners due to a linking mode not working properly. This is explained in more detail in Section 5.

The Venn Diagram of the coverage is shown in Figure 18. 25 399 person references are linked by the BurgerLinker that are not linked by the RecordLinker. The RecordLinker found links to 762 250 person references that the BurgerLinker did not find.

Table 9: Table of the coverage for each linking method per type of person reference

		RecordLinker		BurgerLinker	
Status	Total	Linked	%	Linked	%
Born	296 969	213 854	72.0	209 439	70.5
Married	201 768	169 868	84.2	141 684	70.2
Child born	593 972	570 334	96.0	401 522	67.6
Child married	395 988	341 622	86.3	195 970	49.5
Died	324 339	230 872	71.2	153 932	47.5
Partner died	158 526	101 328	63.9	0	0.0
Child died	621 454	491 558	79.1	280 038	47.1
Total	2 593 016	2 119 436	81.7	1 382 585	53.3

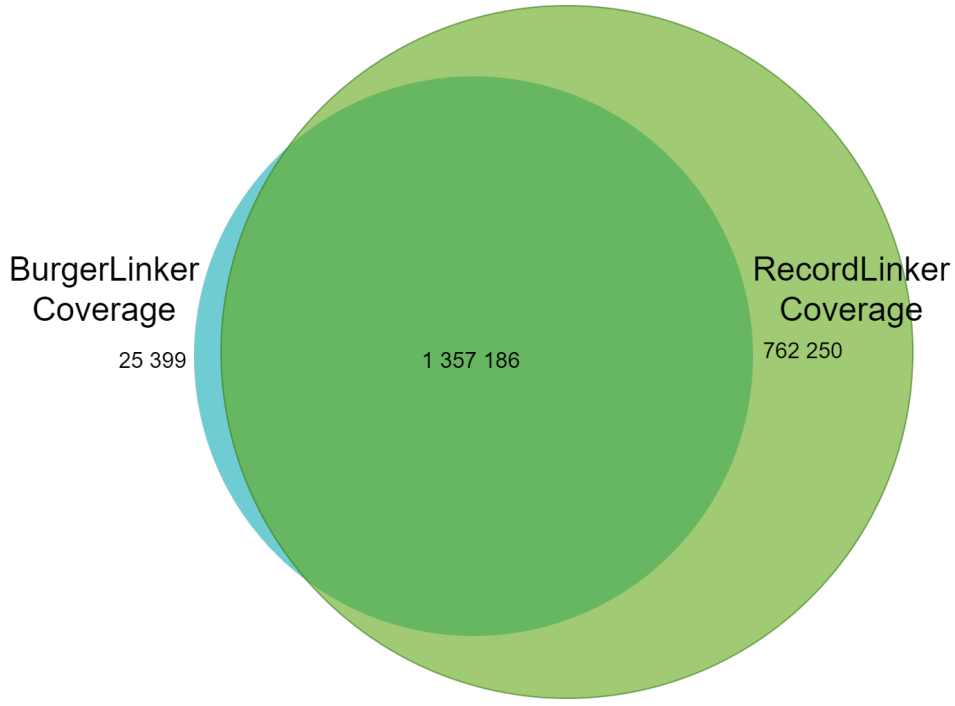


Figure 18: The Venn Diagram of the person references linked by the BurgerLinker and the RecordLinker to scale

5 Discussion and Further Research

The differences in results between the BurgerLinker and the RecordLinker are discussed in Section 4. The biggest differences are the number of links and the incorrectness score. The RecordLinker found links to approximately 750 000 person references that the BurgerLinker did not find. There are differences between the BurgerLinker algorithm and the RecordLinker algorithm that can be a source of the difference in results. The RecordLinker features 14 linking modes, whilst the BurgerLinker features 6 linking modes. More linking modes imply that more person references are compared, increasing the likelihood that a link is found, correct or not. Additionally, the RecordLinker utilises different name processing. It is unclear how much each different aspect impacts the outcome, as only the final outcome is measured.

A different way to measure the quality of the results was not applied. However, this method can possibly provide valuable insight. Erfgoed Leiden en Omstreken has constructed a data set describing families and deceased from 1810-1865. This data set is reviewed and can be used to validate the results of the linking algorithms. The quality measures applied in this study do not explicitly say what method produces results that are more complete and correct. They are merely indicators that show how incorrect the results can be. Validating the results with this data set can provide a meaningful conclusion. However, there are problems with applying this data set. The data structures are not the same and it is not clear what the best way for validating the results is. Using the families and deceased data set from ELO to validate the results of linking algorithms can be interesting for future research.

The BurgerLinker linking mode for linking 'parents of newborns' to 'deceased and partners' did not return any results. It is not known what caused this. This mistake is the explanation for no linked 'died partners' showcased in Figure 17. This linking mode, 'Between_B_D', is the only linking mode that links the partners of the deceased. This does not affect the quality indicators, except for group size and coverage. The group size is expected to be higher for the BurgerLinker, had this linking mode produced results.

In this study, it was decided not to apply documented name variants for processing names into stemmed names. Conversion tables of these name variants exist and they possibly help improve the quality of the links. The reason this information is not applied in this study, is that many stemmed names in the conversion table originated from a different time period. This could result in different names being wrongly assigned to the same stem. This is likely to impact the link quality. Also, when a name variant cannot be assigned to a stemmed name, that person reference is likely not linked, as the other names are stemmed. However, this option should be reevaluated when a name variant/stem conversion table is available that is better suited for this purpose.

The life courses produced by the linking methods can be the basis for future genealogy research. Life courses and family trees provide new information that can lead to new discoveries. Additionally, further research can be done to validate the assumptions made in this study. These assumptions include the chosen time period of the data and the trend-line for determining the maximum accepted Levenshtein distance.

6 Conclusions

The following research question was stated in Section 1.3:
How can life courses be reconstructed with a linking algorithm that interconnects data from archived birth, marriage and death certificates after 1810 in the region Leiden?

The answer to this research question is a combination of name processing, string comparison and unique historical individual reconstruction. Name processing plays a role in how names can be standardised to limit the effect of inconsistent naming. By utilising methods like the Levenshtein distance, person references can be linked, even if the names are slightly different. The generated links can be processed into life courses by applying the Union Find algorithm.

The results of the BurgerLinker and the RecordLinker are evaluated based on quality metrics. These quality metrics are group size, life course duration, life course status and coverage. Group size analysis indicates that the RecordLinker life courses are of higher quality, due to there being fewer large groups. The measurements of life course duration do not show a clear difference in quality between the methods. On the other hand, life course status shows that the BurgerLinker life courses are more often labelled as incorrect. The coverage metrics indicate that the RecordLinker results are more complete, in the sense that significantly more person references are linked. From these quality metrics, it can be concluded that the life courses generated by the RecordLinker are of higher quality than the BurgerLinker results. Additionally, the RecordLinker results are more in line with the expectations, like the expectation of the spikes in 'only birth' and 'only death' life courses.

The BurgerLinker algorithm is more time efficient. However, the process of converting and formatting the input and output can make applying the linking method cumbersome. These extra steps can make the linking process more prone to mistakes as well.

References

[ABE+21] Ran Abramitzky, Leah Boustan, Katherine Eriksson, James Feigenbaum, and Santiago Pérez. Automated linking of historical data. *Journal of Economic Literature*, 59(3):865–918, 2021.

[Bjö90] Åke Björck. Least squares methods. *Handbook of numerical analysis*, 1:465–652, 1990.

[Blo14] Gerrit Bloothooft. Namen: varianten en fouten. *Gen.magazine*, pages 44–46, 2014.

[CLAa] CLARIAH. burgerlinker - civil registries linking tool - github. https://github.com/CLARIAH/burgerLinker. Accessed: 2023-07-10.

[CLAb] CLARIAH. Over clariah. https://www.clariah.nl/nl/over-clariah. Accessed: 2023-04-28.

[Dun46] Halbert L Dunn. Record linkage. *American Journal of Public Health and the Nations Health*, 36(12):1412–1416, 1946.

[Fei16] James J Feigenbaum. A machine learning approach to census record linking. *Retrieved March*, Mar 2016.

[GI91] Zvi Galil and Giuseppe F Italiano. Data structures and algorithms for disjoint set union problems. *ACM Computing Surveys (CSUR)*, 23(3):319–344, 1991.

[His11] Historiek. 200 jaar burgelijke stand. https://historiek.net/200-jaar-burgerlijke-stand/12955/, Sep 2011.

[L+66] Vladimir I. Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.

[Leia] Ergoed Leiden. Beleid. https://www.erfgoedleiden.nl/werkgebied/organisatie/erfgoedvisie. Accessed: 2023-04-21.

[Leib] Ergoed Leiden. Depots. https://www.erfgoedleiden.nl/werkgebied/organisatie/depots. Accessed: 2023-04-21.

[Leic] Ergoed Leiden. Ons verhaal. https://www.erfgoedleiden.nl/werkgebied/organisatie/onze-organisatie. Accessed: 2023-04-21.

[MBL+23] Kees Mandemakers, Gerrit Bloothooft, Fons Laan, Joe Raad, Rick J Mourits, Richard L Zijdeman, et al. Links. a system for historical family reconstruction in the netherlands. *Historical Life Course Studies*, 13:148–185, 2023.

[RMR+20] Joe Raad, Rick Mourits, Auke Rijpma, Ruben Schalk, Richard Zijdeman, Kees Mandemakers, and Albert Merono-Penuela. Linking dutch civil certificates. In *Third Workshop on Humanities in the Semantic Web (WHiSe 2020)*, pages 47–58. CEUR-WS, 2020.

[Sch14] Marijn Paul Schraagen. *Aspects of record linkage*. PhD thesis, Leiden University, 2014.

[Wie] WieWasWie. LINKS. https://www.wiewaswie.nl/nl/bronnen/LINKS. Accessed: 2023-03-15.