Linking Dutch Civil Certificates

Joe Raad, Rick Mourits, Auke Rijpma, Ruben Schalk, Richard Zijdeman, Kees Mandemakers, and Albert Meroño-Peñuela

Abstract. Finding and linking different appearances of the same entity in an open Web setting is one of the primary challenges of the Semantic Web. In social and economic history, *record linkage* has dealt with this problem for a long time, linking historical individual records at a local database level. With the advent of semantic technologies, Knowledge Graphs containing these records have been published, raising the need for large-scale linking techniques that consider the particularities of historical individual linking. In this paper we focus on our current investigation of such techniques to link the Dutch civil certificates in the LINKS/CLARIAH project. We describe the production of the LINKS Knowledge Graph, and we show its potential at answering domain research questions through its large number of `owl:sameAs` links.

Keywords: linked data, digital humanities, civil certificates linking

1 Introduction

Finding and linking equivalent entities (persons, places, events, concepts) on the Web is one of the most important challenges of a Semantic Web of Linked Data. The distributed data publishing paradigm and the scale of the Web exacerbate this problem; various approaches have been proposed to address it, including heuristic-based linking (e.g. string similarity) [12], cluster-similarity linking [11], and deep learning-based knowledge graph completion [14]. The goal is to produce identity links that use the `owl:sameAs` or `skos:exactMatch` predicates so data consumers are aware of identity clusters and classes [3].

Interestingly, the problem has been dealt with in other fields of research; in particular in economic and social history. There, *record linkage* is a challenging and active area of research, as shown in a recent *Historical Methods* special issue on the subject [19]; and is becoming ever more important in economic and social history. Mass digitisation of archival material means that further insight can be obtained by linking individuals and households across different records, especially now that sources with complete population coverage are becoming available. Historical civil certificates are the authoritative sources of birth, marriage and death events in municipality registers, and allow for the reconstruction of lives of the past [4][23]. In the Netherlands, the LINKS project [15] has shown that this reconstruction is, however, often very challenging, as there is generally no ground truth. Individuals are not actively followed over time, but observed during the registration of a vital event. As a result, it is unclear whether, where, and when an individual can be observed. It is not even certain whether follow-up

is available at all, because individuals could migrate out of the region of observation [4]. To complicate matters further, large quantities of historical certificates have been indexed, which gives rise to data entry errors. These spelling mistakes can be hard to deal with, as twins and other multiple births often receive similar names. Furthermore, first names were often reused in families to "replace" earlier-born, deceased siblings. Finally, civil servants were known to indicate non-standard mutations, such as name changes, acknowledgement of children, and divorces as side notes. As a result, very important relational information is often not standardised [23].

In this paper, we summarise our efforts in the LINKS and CLARIAH projects to overcome these challenges, and link the appearance of the same person in 1.5 million birth (1812–1919), marriage (1812–1944) and death (1812–1969) certificates in the Dutch province of Zeeland. Specifically, our contributions are:

- A description of the LINKS knowledge graph production process by using standard semantic technologies (Section 4)
- A highly scalable certificate linking method based on efficient string similarity through Levenshtein automaton (Section 5)
- A preliminary evaluation based on SPARQL queries that use such links (Section 6)

In the next sections, we survey related work (Section 2), describe the original dataset (Section 3), explain our contributions (Sections 4, 5 and 6), and conclude (Section 7).

2 Related Work

Historical record linkage generally requires a previous effort on digitising large amounts of individual-level historical records, a goal shared by projects like NAPP/IPUMS [9], the Balsac Population Database[1], the Utah Population Database[2], Familysearch[3], the Scottish Longitudinal Study[4], Digitising Scotland[5], the Norway Historical Population Register[6], Link-Lives[7], POPLINK/DDB[8], the Scanian Economic Demographic Database (SEDD)[9], the North Orkney Population History Project[13] and Death and Burial Data in Ireland 1864-1922[10]. Linking individuals in the US 1850 and 1860 census is generally considered one of

[1] http://balsac.uqac.ca/english/
[2] https://uofuhealth.utah.edu/huntsman/utah-population-database/
[3] https://www.familysearch.org
[4] https://sls.lscs.ac.uk/
[5] https://digitisingscotland.ac.uk/
[6] https://www.rhd.uit.no/nhdc/hpr.html
[7] https://link-lives.dk/
[8] https://www.umu.se/en/centre-for-demographic-and-ageing-research/databases/parish-registers-databases/
[9] https://www.ed.lu.se/databases/sedd
[10] https://www.dbdirl.com/

the earliest efforts [8], and similar approaches for Canada [2] and Sweden [25] have followed. [17] provides a critical review of these and other historical record linkage efforts, with a focus on US data. We share with these efforts a focus on string-based comparison linkage. In other projects (e.g. Digitising Scotland) the goal is to perform group-level linkage as well [1]. Recent machine learning approaches have gotten a lot of traction in the field [6]. For example, recent work on historical US census data uses manually labelled data from `familysearch.com` as training data [18]. In the Netherlands, earlier work on Dutch civil certificates focuses on methodological aspects of record linkage [21]. The work done in the LINKS project [15] constitutes a basis for our contribution.

3 Dataset

The digitised civil registry consists at the moment of 27.5 million certificates. In total, there are 10.3 million birth certificates, 4.4 marriage certificates, and 12.7 million death certificates in the digitised registry at the International Institute of Social History[11]. The number of available birth and death certificates differs strongly as, due to privacy laws, only death certificates that are more than 50 years old are available for research. Birth certificates become available with a 100-year delay, marriage certificates with a 75-year delay, and death certificates with a 50-year delay. For the moment, the experiments in this paper are restricted to the civil registries produced in the Zeeland region. This dataset of Zeeland civil registries, known as *LINKS_Zeeland_cleaned_2016_01* [16], consists of 1.5 million certificates, which represents ∼5.5% of the total certificates. Specifically, there are 698,285 birth certificates (6.7% of the total birth certificates), 193,921 marriage certificates (4.4%), and 665,999 death certificates (5.2%). This dataset is cleaned, standardised and distributed in a restricted manner [15] in the form of three CSV files:

1. **Locations:** containing the locations that show up in the civil certificates, describing the municipality, province, region and the country of a location. This file consists of 6 columns and 2,456 rows.
2. **Registrations:** containing general data from a certificate registration which exceed the individual level, such as the date and place of birth, marriage or death. This file consists of 10 columns and 1,558,205 rows, with each row representing a single registration in the Zeeland province.
3. **Persons:** containing all appearances of persons. In general every birth certificate generates records for three persons (newborn child, mother and father), a marriage certificate generates minimally six person records (bride with her parents and the groom with his parents) and a death certificate generates three or four person records (deceased, father, mother and possibly a spouse). This file consists of 33 columns and 5,526,393 rows.

[11] `https://iisg.amsterdam/en`

4 LINKS Knowledge Graph

The process of converting the CSV files of the LINKS dataset into a Knowledge Graph consists of three steps. Firstly, we manually design a model for describing and enriching the civil registries data, following Linked Data best practices. Secondly, we transpose the CSV data into an RDF Knowledge Graph, according to our designed model. Finally, we make the graph available for browsing and querying in an efficient manner.

4.1 Designing the civil registries schema

For modelling the civil registries data, we designed a new simple model that reuses, whenever possible, existing vocabularies. This model is presented in Figure 1, and has four main components:

- **Civil Registrations.** The first component (concepts coloured in brown) describes each civil registration (birth, marriage, or death certificate), listing its identifier, its sequential number, the location, and date of the registration.
- **Life Events.** The second component (in green) describes the actual life events (birth, marriage, or death event), listing the main individuals involved in this event, the location and the date of this event. In this model, a distinction is made between the civil registration and their associated life events, as certain civil registrations can be produced in different dates and locations from where the life event actually happened.
- **Individuals.** The third component (in blue) describes each individual involved in these life events, listing their names, sex, civil status, and birth dates.
- **Locations.** The final component (in orange) describes the location where each life event has happened and the location where it was registered. In this component, information regarding the municipality, the province, the region, and the country can be available.

4.2 Transposing the data to RDF

For converting the Zeeland dataset to a Knowledge Graph, we use the tool CoW (CSV on the Web converter)[12]. This batch tool, developed within the CLARIAH project [10], allows the conversion of datasets expressed in CSV. It uses a JSON schema expressed using an extended version of the CSVW standard, to convert CSV files to RDF in scalable fashion. In the case of the Zeeland dataset, we run the conversion process separately for the three CSV files, by manually designing

[12] `https://csvw-converter.readthedocs.io/`

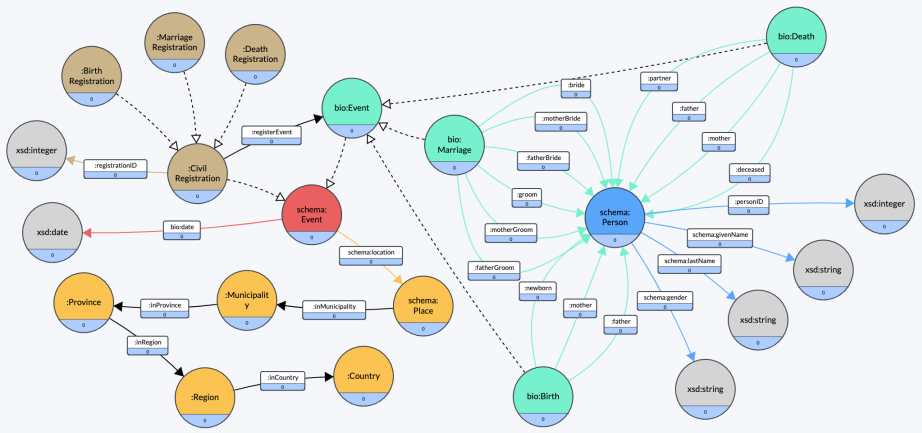

Fig. 1: Schema of the LINKS Knowledge Graph. For a higher resolution figure, we refer the reader to https://github.com/CLARIAH/wp4-civreg/blob/master/schema/LINKS-schema.png

the JSON schema for *Locations*[13], *Registrations*[14], and *Persons*[15]. This manual design of these JSON files allows us to transpose the data according to the model presented in Figure 1. After designing a JSON file for each of the three CSV files, we use the command line to convert each of these files separately. For instance, having both the *Locations.csv* file with its associated JSON schema *Locations.csv-metadata.json* in the same directory, the following command is sufficient to convert the data to RDF, creating the RDF file *Locations.nq* encoded in the *N-Quads* format.

```
$ cow_tool convert Locations.csv
```

The conversion process takes 30 seconds for the file *Locations*, 100 minutes for *Registrations*, and around 5 hours for *Persons* on a SSD disk, with 64GB of memory.

4.3 Accessing the RDF knowledge graph

Combining the three resulted N-Quads files results in the LINKS knowledge graph, composed of 58,513,388 triples. This knowledge graph can be accessed online through Druid[16], the CLARIAH instance of the TriplyDB triple store[17].

[13] `https://raw.githubusercontent.com/CLARIAH/wp4-civreg/master/json/locations.csv-metadata.json`
[14] `https://github.com/CLARIAH/wp4-civreg/blob/master/json/registrations.csv-metadata.json`
[15] `https://github.com/CLARIAH/wp4-civreg/blob/master/json/persons.csv-metadata.json`
[16] `https://druid.datalegend.net/`
[17] `https://triply.cc/`

Druid allows the storage of knowledge graphs, and provides tools to browse, query and visualise our data. For privacy reasons, the LINKS knowledge graph is uploaded as a private dataset on Druid, restricting its access[18] to members of the LINKS organisation[19] on Druid. We provide publicly accessible links to resources of the knowledge graph when possible.

In addition to accessing the LINKS knowledge graph through the Druid Web hub, authorised users of the LINKS knowledge graph can also access this dataset locally. For enabling easy and efficient access on a normal local machine, we convert the LINKS knowledge graph from N-Quads to HDT (Header, Dictionary, Triples) [7]. This compact data structure and binary serialisation format for RDF keeps big datasets compressed to save space while maintaining search and browse operations without prior decompression. Converting the LINKS knowledge graph into HDT consists of two simple steps: (i) merge the three RDF N-Quads files into one larger N-Quads file, (ii) convert the resulting merged file to HDT using the rdfhdt library[20].

5 Certificate Linkage

For linking Dutch civil registries, we heavily rely on the string similarity between individuals' names. This is motivated by the high quality of the registered names in most civil certificates, and the limited spelling variation between different civil certificates for the same individual. An example of such quality maintenance can be observed in marriage registrations, where both the bride and the groom are required to bring their own birth certificates when registering their marriage. Moreover, married women in the Netherlands keep their own family name in the civil certificates, which highly facilitates the problem at hand. In the case of death registrations, they are generally registered by next of kin —parents, spouses, children, or siblings— which also highly limits variations in name spelling [23].

Similarity between two names can be measured in several ways, such as calculating the Levenshtein, Jaccard, or Jaro-Winkler distances. In this work, we take the Levenshtein distance as a basis for matching individuals in civil certificates. This distance measures the number of single character edits (insertions, deletions or substitutions) required to change one name into the other. The standard algorithm for calculating the Levenshtein distance between two names was proposed by Wagner and Fisher [24], but can lead to a quadratic time complexity. In this work, as we aim to match individuals from a list of millions of certificates to individuals in another large list of certificates, the standard approach (or its variants) of calculating the Levenshtein distance by comparing each pair of certificates is not feasible, as the time complexity of the approach can grow exponentially with the size of the given lists. Therefore, we adopt the approach and the library proposed by Dylon Devo[21], based largely on the work of Schulz

[18] `https://druid.datalegend.net/LINKS/links-zeeland/`
[19] `https://druid.datalegend.net/LINKS`
[20] `http://www.rdfhdt.org/manual-of-the-java-hdt-library/`
[21] `https://github.com/universal-automata/liblevenshtein-java/`

and Mihov [22], for the fast selection of candidate individuals within a certain Levenshtein distance. In this approach, the list of target individuals are indexed as a Minimal Acyclic Finite-State Automata (MA-FSA), where a Levenshtein transducer is initialised according to a maximum distance specified by the user. When a name is given as a source query with a maximum accepted Levenshtein distance, the states of the Levenshtein automaton corresponding to that name are constructed on-demand as the automaton is evaluated. According to its author, this approach allows to find for a given name n all candidate names in a list M in linear time on the length of n, and not on the size M. In the following, we describe how we deploy this approach for matching newborns registered in birth certificates to their marriage certificates. The general process remains unchanged for other types of linkage, where only the roles of the considered individuals and the link's timeline consistency are adapted accordingly.

5.1 Approach

Finding the marriage certificate of a certain newborn, when applicable, requires matching three individuals: (i) the newborn in the birth certificate with the bride or groom of a certain marriage certificate, (ii) the newborn's mother with the bride's or groom's mother, (iii) the newborn's father with the bride's or groom's father. Once a match, according to a maximum Levenshtein distance, between the three individuals of a birth certificate and a marriage certificate is found, we check whether the logical timeline is respected. Only when a match between two certificates based on the three individuals is found, with a correct logical timeline, a match between the three individuals is registered in the Knowledge Graph. Specifically, our approach for matching newborns to a marriage certificate can be divided into 5 main steps:

1. Create six indices, with each index representing a MA-FSA containing the list of all full names of a certain role in marriage certificates. For instance, the index of the role "bride" contains the full names (first name + last name) of all women individuals that got married (i.e. role of bride). For each of these indices, a Levenshtein transducer is initialised according to a maximum Levenshtein distance, given by the user.
2. Create six Key-Value databases, with each database covering a single role r in the marriage certificate. A key in a database represents a full name fn, and the value represents a list of marriage certificate identifiers that have for the role r an individual with the name fn. For instance, the entry "Anna Aartsen" $\rightarrow$ {123323,232344} indicates that both these certificates have a bride registered with the full name "Anna Aartsen". While such information can be directly queried from the Knowledge Graph, Key-Value databases are a better mean for frequent read requests. In particular, we rely on the RocksDB[22] disk-based Key-Value database.

[22] `http://rocksdb.org`

3. Find marriage certificate candidate(s) for each birth certificate. For this, we firstly search for the full name of the newborn in the index of the bride or the groom. Considering that the newborn is a girl, this step retrieves a list of candidate names $C_{newborn}$ from the bride index, representing a spelling variation within the maximum Levenshtein distance specified by the user. If $C_{newborn}$ is not empty, we retrieve from the bride's Key-Value database the list of candidate certificates $E_{newborn}$ that contain this candidate's name. In the case where $C_{newborn}$ contains several candidates, the result will be the union of all returned $E_{newborn}$ for each candidate. The same process is applied when searching for the full name of the newborn's mother and father in the bride's mother and father indices, respectively returning a list of candidate certificates E_{mother} and E_{father}.
4. Filter resulting candidates. Since in the majority of cases, a newborn is expected to have the same registered parents during marriage, we require the match between the birth and the marriage certificates to be based on the three individuals. Therefore, the preliminary marriage candidates consists of the intersection of $E_{newborn}$, E_{mother} and E_{father}. Finally, out of these preliminary candidates only those that respect the logical timeline are considered. In this case consisting of matching a newborn to a bride or a groom, we expect that the marriage certificate is registered at least 14 years, and at most 70 years, after its matched birth registration.
5. Save links, in two formats for respecting the preferences of most researchers: (a) CSV file consisting of the birth certificate identifier, the matched marriage certificate identifier, with the link metadata consisting mainly of the Levenshtein distance between each matched individual in these certificates, and time difference between both registrations, (b) N-Quads file consisting of `owl:sameAs` links between each matched individual, with each link being asserted in a different named graph for describing its context. For instance, the statement ⟨ `iisg:newbornURI`, `owl:sameAs`, `iisg:brideURI`, `iisg:graph/birthToMarriage/0-2-1` ⟩ indicates that the identity link between these two individuals was detected based on a Levenshtein distance of 0 between the newborn and bride's name, a Levenshtein of 2 between their mothers' names, and 1 for the fathers' names.

5.2 Experiments

For testing the scalability of our approach, we evaluated our matching approach on the Zeeland dataset described in Section 3. We firstly evaluated the process of matching newborns in marriage certificates to brides/grooms in marriage certificates, and then evaluated the process of matching parents of brides/grooms in marriage certificates to their own marriage certificate. Table 1 shows that matching civil registries of a Dutch province takes no more than a few minutes[23], with the runtime increasing as the maximum Levenshtein distance per individual increases. It also shows that even with a maximum Levenshtein distance of 1, there

[23] Experiments conducted on MacBook Pro, with SSD disk and 16GB of memory

	NewbornToPartner		**PartnerParentsToCouple**	
Maximum Levenshtein per Individual	Number of Links	Runtime (in mins)	Number of Links	Runtime (in mins)
1	271,230	5	205,477	2
2	289,937	18	224,785	8
3	310,232	74	244,343	25

Table 1: Results of matching newborns in marriage certificates to brides/grooms (newbornToPartner), and matching parents of brides/grooms in marriage certificates to their own marriage certificate (partnerParentsToCouple).

```
SELECT ?year (avg(?samePlace) as ?shareSamePlace) WHERE {
  GRAPH ?g {
    ?fatherBride owl:sameAs ?fatherBride_asGroom .
    ?fatherGroom owl:sameAs ?fatherGroom_asGroom .
  }
  ?mar1 iisgv:fatherBride ?fatherBride ;
        iisgv:fatherGroom ?fatherGroom ;
        schema:location ?loc1 .
  ?mar2 iisgv:groom ?fatherBride_asGroom ;
        bio:date ?date ;
        schema:location ?loc2 .
  ?mar3 iisgv:groom ?fatherGroom_asGroom ;
        schema:location ?loc3 .
  FILTER(?date > "1840-01-01"^^xsd:date && ?date < "1910-01-01"^^xsd:date)
  BIND(if(?loc1 = ?loc2 || ?loc1 = ?loc3, 1, 0) AS ?samePlace) .
  BIND(year(?date) as ?year) .
}
GROUP BY ?year
ORDER BY ?year
```

Listing 1.1: SPARQL query for identifying migrants and non-migrants in LINKS.

is a significant overlinking, since the number of detected links (271,230) is larger than the number of marriage certificates in this dataset (193,921). Therefore, indicating that a number of marriage certificates were matched to multiple birth certificates. The source code of this approach is publicly available[24].

6 Preliminary Evaluation (Use Cases)

While we expect that a dataset containing information on every Dutch person born in the period 1812–1919 and their family relations will be useful to many researchers, it will be especially valuable to demographic, social, and economic historians working with individual-level data. One issue in particular that it can address, is bias in results due to migration.

The key issue there is that, currently, many analyses are based on records from one locality (a village, town, or province). In other words, out-migrants are left out of the data. This is a problem because migrants are different from the rest of the population. For example, according to Ruggles [20] they had different ages

[24] `https://github.com/CLARIAH/wp4-links`

Fig. 2: Share of marriages of father and bride in same location, 1840–1910.

at marriage and life expectancies. A comparison of the civil registry of Zeeland and a smaller population register data set (HSN) that follows individuals as they move, has also shown that the differences between such datasets can be explained by the exclusion of migrants out of Zeeland [5].

The new data created here can take substantial steps to resolve this issue. Only international out-migrants can now go missing, which is a far smaller share of the data. The query[25] in Listing 1.1 shows how migrants and non-migrants are easily identified in the data. To do this, we compare the location of a marriage with that of both the bride's and the groom's parents' marriage. The results of this query (figure 2) show that the share of non-migrants between 1840 and 1910 falls from 80 to 72 percent, which means that by the start of the twentieth century, nearly a fourth of the couples moved between their marriage and that of their child. Linking a civil registry for the entire Netherlands as done here allows us to include this large group in future analyses.

7 Conclusion

In this work, we described our production process of the LINKS Knowledge Graph, containing civil certificates of the Dutch province of Zeeland. We presented our approach for linking all these certificates, within 5 minutes on a regular laptop, and showed how such links can be exploited for conducting demographic analyses using SPARQL. This work is in the process of being extended

[25] `https://github.com/CLARIAH/wp4-queries-links/blob/master/marriage-locations.rq`

to cover all certificates of the Netherlands, enabling larger and more valuable demographic, social and economic analyses.

References

1. Akgün, , Dearle, A., Kirby, G., Garrett, E., Dalton, T., Christen, P., Dibben, C., Williamson, L.: Linking Scottish vital event records using family groups. Historical Methods: A Journal of Quantitative and Interdisciplinary History **0**(0), 1–17 (Mar 2019). https://doi.org/10.1080/01615440.2019.1571466, https://doi.org/10.1080/01615440.2019.1571466
2. Antonie, L., Inwood, K., Lizotte, D.J., Andrew Ross, J.: Tracking people over time in 19th century Canada for longitudinal analysis. Machine Learning **95**(1), 129–146 (Apr 2014). https://doi.org/10.1007/s10994-013-5421-0, https://link.springer.com/article/10.1007/s10994-013-5421-0
3. Beek, W., Raad, J., Wielemaker, J., van Harmelen, F.: sameas. cc: The closure of 500m owl: sameas statements. In: European semantic web conference. pp. 65–80. Springer (2018)
4. Van den Berg, N., Van Dijk, I.K., Mourits, R.J., Slagboom, P.E., Janssens, A.A.P.O., Mandemakers, K.: Families in comparison: An individual-level comparison of life-course and family reconstructions between population and vital event registers. Population Studies: A Journal of Demography pp. 1–20 (2020). https://doi.org/10.1080/00324728.2020.1718186
5. Berg, N.v.d., Dijk, I.K.v., Mourits, R.J., Slagboom, P.E., Janssens, A.A.P.O., Mandemakers, K.: Families in comparison: An individual-level comparison of life-course and family reconstructions between population and vital event registers. Population Studies **0**(0), 1–20 (Feb 2020). https://doi.org/10.1080/00324728.2020.1718186, https://doi.org/10.1080/00324728.2020.1718186, publisher: Routledge _eprint: https://doi.org/10.1080/00324728.2020.1718186
6. Feigenbaum, J.J.: Multiple Measures of Historical Intergenerational Mobility: Iowa 1915 to 1940. The Economic Journal **128**(612), F446–F481 (Jul 2018). https://doi.org/10.1111/ecoj.12525, https://onlinelibrary.wiley.com/doi/abs/10.1111/ecoj.12525
7. Fernández, J.D., Martínez-Prieto, M.A., Gutiérrez, C., Polleres, A., Arias, M.: Binary rdf representation for publication and exchange (hdt). Journal of Web Semantics **19**, 22–41 (2013)
8. Ferrie, J.P.: A New Sample of Males Linked from the Public Use Microdata Sample of the 1850 U.S. Federal Census of Population to the 1860 U.S. Federal Census Manuscript Schedules. Historical Methods: A Journal of Quantitative and Interdisciplinary History **29**(4), 141–156 (Oct 1996). https://doi.org/10.1080/01615440.1996.10112735, https://doi.org/10.1080/01615440.1996.10112735
9. Goeken, R., Huynh, L., Lynch, T.A., Vick, R.: New Methods of Census Record Linking. Historical Methods: A Journal of Quantitative and Interdisciplinary History **44**(1), 7–14 (Jan 2011). https://doi.org/10.1080/01615440.2010.517152, https://doi.org/10.1080/01615440.2010.517152
10. Hoekstra, R., Meroño-Peñuela, A., Rijpma, A., Zijdeman, R., Ashkpour, A., Dentler, K., Zandhuis, I., Rietveld, L.: The datalegend ecosystem for historical statistics. Journal of Web Semantics **50**, 49–61 (2018)

11. Idrissou, A.K., Hoekstra, R., Van Harmelen, F., Khalili, A., Van den Besselaar, P.: Is my: sameas the same as your: sameas? lenticular lenses for context-specific identity. In: Proceedings of the Knowledge Capture Conference. pp. 1–8 (2017)
12. Isele, R., Jentzsch, A., Bizer, C.: Silk server-adding missing links while consuming linked data. In: Proceedings of the First International Conference on Consuming Linked Data-Volume 665. pp. 85–96. CEUR-WS. org (2010)
13. Jenning, J.A., Sparks, C.A., Murtha, T.: Interdisciplinary approach to spatiotemporal population dynamics:the north orkney population history project. Historical Life Course Studies pp. 27–51 (2019), http://hdl.handle.net/10622/23526343-2019-0002?locatt=view:master
14. Lisena, P., Meroño-Peñuela, A., Troncy, R.: MIDI2vec: Learning MIDI Embeddings for Reliable Prediction of Symbolic Music Metadata. Transactions of the International Society for Music Information Retrieval (2020), under review
15. Mandemakers, K., Laan, F.: LINKS dataset Genes Germs and Resources, WieWasWie Zeeland, Civil Certificates, version 2017.01. International Institute of Social History, Amsterdam
16. Mandemakers, K., Laan, F.: LINKS-Zeeland challenge, WieWasWie Zeeland, Civil Certificates, version 2016. International Institute of Social History, Amsterdam
17. Massey, C.G.: Playing with matches: An assessment of accuracy in linked historical data. Historical Methods: A Journal of Quantitative and Interdisciplinary History **0**(0), 1–15 (Mar 2017). https://doi.org/10.1080/01615440.2017.1288598, http://dx.doi.org/10.1080/01615440.2017.1288598
18. Price, J., Buckles, K., Van Leeuwen, J., Riley, I.: Combining Family History and Machine Learning to Link Historical Records. Tech. Rep. w26227, National Bureau of Economic Research, Cambridge, MA (Sep 2019). https://doi.org/10.3386/w26227, http://www.nber.org/papers/w26227.pdf
19. Rijpma, A., Cilliers, J., Fourie, J.: Record linkage in the Cape of Good Hope Panel. Historical Methods: A Journal of Quantitative and Interdisciplinary History **0**(0), 1–16 (Feb 2019). https://doi.org/10.1080/01615440.2018.1517030, https://doi.org/10.1080/01615440.2018.1517030
20. Ruggles, S.: Migration, Marriage, and Mortality: Correcting Sources of Bias in English Family Reconstitutions. Population Studies **46**(3), 507–522 (Nov 1992). https://doi.org/10.1080/0032472031000146486, https://doi.org/10.1080/0032472031000146486
21. Schraagen, M.P., others: Aspects of record linkage. Ph.D. thesis, Leiden Institute of Advanced Computer Science (LIACS), Faculty of Science, Leiden University (2014), https://openaccess.leidenuniv.nl/handle/1887/29716
22. Schulz, K.U., Mihov, S.: Fast string correction with levenshtein automata. International Journal on Document Analysis and Recognition **5**(1), 67–85 (2002)
23. Vulsma, R.F.: Burgerlijke stand en bevolkingsregister. Centraal Bureau voor Genealogie, 's-Gravenhage
24. Wagner, R.A., Fischer, M.J.: The string-to-string correction problem. Journal of the ACM (JACM) **21**(1), 168–173 (1974)
25. Wisselgren, M.J., Edvinsson, S., Berggren, M., Larsson, M.: Testing Methods of Record Linkage on Swedish Censuses. Historical Methods: A Journal of Quantitative and Interdisciplinary History **47**(3), 138–151 (Jul 2014). https://doi.org/10.1080/01615440.2014.913967, https://doi.org/10.1080/01615440.2014.913967