Comparison between historical population archives and decentralized databases

Marijn Schraagen and Dionysius Huijsmans
Leiden Institute of Advanced Computer Science (LIACS)
Leiden University, The Netherlands
{schraage,huijsman}@liacs.nl

Abstract

Differences between large-scale historical population archives and small decentralized databases can be used to improve data quality and record connectedness in both types of databases. A parser is developed to account for differences in syntax and data representation models. A matching procedure is described to discover records from different databases referring to the same historical event. The problem of verification without reliable benchmark data is addressed by matching on a subset of record attributes and measuring support for the match using a different subset of attributes. An application of the matching procedure for comparison of family trees is discussed. A visualization tool is described to present an interactive overview of comparison results.

1 Introduction

In the historical demographics and genealogy domain, research data can be collected from centralized databases such as a historical census or civil registry. Alternatively, decentralized data can be collected from, e.g., personal archives or local organizations. Complementary and conflicting information between these sources can be valuable for research if the overlap, i.e., matching records, is correctly identified. This paper describes a method to discover matching records in centralized and decentralized data for the problem of family reconstruction. An overview of related work is presented in Section 2. Section 3 describes the two different data formats. Section 4 provides a mapping procedure between the different data formats. In Section 5 the matching procedure is explained at the conceptual and technical level. Section 6 provides a verification procedure and results for a test database. An application of the matching in a family tree visualization tool is provided in Section 7. A conclusion and directions for future research are provided in Section 8.

The most important concepts used throughout this paper are defined as follows:
Record. Unit of matching and linkage. A record refers to a Genlias certificate (Section 3) or a Gedcom certificate reconstruction (Sections 3 and 4), unless stated otherwise.
Record match. A pair of records that refer to the same event (birth, marriage, or death).
Record link. A pair of records that refer to related events (e.g., birth and death of the same person).
Field similarity measure. Similarity between field values, e.g., number of days between dates.
Record similarity measure. Similarity requirements for selected fields and relations between the requirements.
Edit distance. Minimum number of character insertions, deletions and/or substitutions needed to transform one string into another (Levenshtein distance).
Name sequence. Concatenation of person names from a record.
Person name. Single name, i.e., given name or family name.

2 Related work

Automatic matching and linkage of historical records has been researched for several decades. An early example can be found in (Winchester, 1970), using Soundex as field similarity measure to compare census data. An approach from the French-speaking Canadian province of Quebec, using a custom phonetic code, is described in (Bouchard and Pouyez, 1980). In this approach different types of data are merged together. The researchers state that "the most fundamental rule is that we never try to link individuals, but rather

Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, pages 20–28,
Sofia, Bulgaria, August 8 2013.

pairs of individuals; that is: couples [...] It can be demonstrated easily that individual linkage is liable to result in uncertain, false, or missed links, whereas the result of the linkage of couples is very reliable". The approach is implemented as follows: "we accept as candidates for linkage those pairs which have at least two exactly identical elements". Experience with the dataset used in the current research has resulted in a similar approach (see Section 5). Linkage on the Quebec dataset has been developed by the same authors (Bouchard, 1992). The 1992 paper discusses the use of field values: "the various fields can serve as identifiers (linkage), controls (validation), or variables (analysis)." The notion of internal validation is discussed further in Section 6. Later approaches to record linkage have focussed on scalability of linkage methods for large amounts of data, see, e.g., (Christen and Gayler, 2008).

A detailed overview of elements from genealogical records and the application of each element for linkage is provided in (Wilson, 2011). Besides historical and/or genealogical data, various other types of data have been used for development and testing of algorithms, such as hospital records, phone book records, customer records, etc. However, algorithms generally assume a certain level of uniformity in data representation, both at a technical and at a conceptual level. This means that generally pedigrees are linked to other pedigrees but not to civil certificates, and vice versa. Events and individuals (actors) have been modelled together using NLP techniques (Segers et al., 2011), however these approaches are mostly not applicable for genealogical data both because of the lack of natural language resources to identify and link instances of actors and events, as well as the difference in scope of the model (participants of historically significant events vs. every person that existed during a certain time period). Some attempts have been made to facilitate data exchange and accessability in the genealogical domain, either by presenting a standardized format (the Gedcom standard (GEDCOM Team, 1996) being the most succesful example), by conversion into a standardized format (Kay, 2006; Kay, 2004), by enforcing a Semantic Web ontology (Zandhuis, 2005), or by defining a framework that accepts custom data models as metadata to be provided upon exchange of the genealogical data itself (Woodfield, 2012). Algorithmic solutions for merging of pedigrees have been proposed (Wilson, 2001) that take into account matches between individuals and matching links between individuals. More elaborate linkage of pedigree data is described in (Quass and Starkey, 2003), using feature weights and thresholds to increase linkage performance.

Using various definitions of *record*, such as a single individual, multiple individuals, families (i.e., multiple individuals in a family relation), or events (i.e., multiple individuals in a certain relation at a specific point in time), most research in record linkage is either directed towards matching of records, i.e., asserting equal reference, or linkage of related (but not equal) records using matching of record elements (e.g., a birth record linked to a marriage record based on a match between the child and the bridegroom). In social networks research a different type of linkage is common, where records are linked but not matched (e.g., two people sharing common interests). Occasionally this type of link is used in historical record linkage as well (Smith and Giraud-Carrier, 2006).

Test corpora have been developed (Schone et al., 2012), (Bouchard, 1992), however these are intrinsically domain- and language-specific. Moreover, these corpora are generally not readily available for research.

3 Data formats

The centralized data used in the experiments is extracted from the Dutch Genlias[1] database. Genlias contains civil certificates (around 15 million in total) from the Netherlands, for the events of birth, marriage and death. Most documents originate from the 19th and early 20th century. A record (see Figure 1 for an example) consists of the type of event, a serial number, date and place, and participant details. The parents are also listed for the main subject(s) of the document, i.e., the newborn child, bride and groom, and deceased person for birth, marriage and death certificates, respectively. The documents do not contain identifiers for individuals. No links are provided between documents or individuals.

The decentralized data is extracted from a family tree database in the Gedcom (Genealogical Data Communication) format. In this format genealogical data is stored based on individuals and nuclear (immediate) families, instead of events as

[1] The Genlias data is currently maintained by WieWasWie, see `http://www.wiewaswie.nl` (in Dutch)

Type: birth certificate
Serial number: 176
Date: 16 - 05 - 1883
Place: Wonseradeel
Child: Sierk Rolsma
Father: Sjoerd Rolsma
Mother: Agnes Weldring

Figure 1: Genlias birth certificate.

in Genlias. Every individual or family in Gedcom is assigned a unique identifier. Records for individuals usually contain personal information like names, birth and death date, etc. The families in which the individual participates, either as child or as parent, are also indicated. A family record lists the individuals and their roles. Marriage information (date, place) can also be present in a family record. Using the record identifiers, a link network between individuals and families can be constructed.

Gedcom is a text-based free entry format. The standard (GEDCOM Team, 1996) states that "A record is represented as a sequence of tagged, variable-length lines, arranged in a hierarchy. A line always contains a hierarchical level number, a tag, and an optional value. A line may also contain a cross-reference identifier or a pointer." (see Figure 2 for an example). The Gedcom standard is used by a wide variety of genealogical applications, ranging from full-featured commercially available software to small scripts. The implementation of the standard can differ between applications, as well as the content format entered by users. The next section describes a parsing procedure designed to process this kind of data.

4 Parsing

Prior to the actual record matching, a mapping between the data formats must be performed. This requires either a reconstruction of events from the Gedcom file, or vice versa a reconstruction of individuals and nuclear families from Genlias. The first option requires links between Gedcom records, for example to construct a birth record from the three individual records of the child and parents using the intermediate family record. The second option requires links between Genlias certificates, for example to construct a family record from the birth certificates of several children. Record links are available in Gedcom only, and therefore reconstruction of events from Gedcom is the preferred option.

```
0 @F294@ FAM
1 HUSB @I840@
1 WIFE @I787@
1 MARR
2 DATE 30 MAY 1874
2 PLAC Wonseradeel
1 CHIL @I847@
1 CHIL @I848@
1 CHIL @I849@
0 @I840@ INDI
1 NAME Sjoerd/Rolsma/
1 BIRT
2 DATE 13 FEB 1849
1 DEAT
2 DATE 17 JAN 1936
1 FAMS @F294@
0 @I787@ INDI
1 NAME Agnes/Welderink/
1 SEX F
1 BIRT
2 DATE ca 1850
1 FAMS @F294@
0 @I849@ INDI
1 NAME Sierk/Rolsma/
1 BIRT
2 DATE 16 MAY 1883
2 PLAC Wonseradeel
2 SOUR
3 REFN 176
1 FAMC @F294@
```

Figure 2: Gedcom database fragment, showing a selection of fields from a FAM record (family) and three INDI records (individual).

There are various tools available to perform the required data transformation. Many genealogy programs can export Gedcom data to, e.g., XML or SQL databases which can be queried to construct events. Alternatively, dedicated Gedcom parsers exist for a number of programming languages (such as Perl (Johnson, 2013), C (Verthez, 2004), Python (Ball, 2012), XSLT (Kay, 2004)) that provide data structures to manipulate the Gedcom data from within code. However, the data structures are still centered around individuals and families and the performance of the tools is to a greater or lesser degree sensitive to violations of (some version of) the Gedcom standard. The rest of this section describes a more general parsing algorithm that can be applied to any kind of level-numbered textual data.

The parser (see Figure 3) uses a Prolog DCG-style grammar to specify the elements of target records (see Figure 4 for an example). Tags found in lines from the database file are pushed on a stack one by one. Before a tag is pushed, all cur-

```
S ← ∅
while L ← readline(database) do
    if(L.level = 0) then
        id ← L.value
    while(S.top.level ≥ L.level) do
        S.pop()
    S.push(L.tag)
    foreach terminalList ∈ grammar do
        if(S = terminalList) then
            index(id,terminalList) ← L.value
foreach id ∈ index do
    foreach target ∈ grammar do
        if(pointerList ∈ target) then
            duplicate(target,id,pointerList)
        foreach protoRecord ∈ ({target} ∪ duplicates) do
            foreach terminalList ∈ protoRecord do
                output ← index(id,terminalList)
            output ← record separator
```

Figure 3: Parser algorithm.

```
birthcertificate −−> [@],[fam,chil(+)]:birthbasic,
    [fam,husb]:personname, [fam,wife]:personname.
birthbasic −−> birthdate, birthplace, birthref, personname.
birthdate −−> [indi,birt,date].
birthplace −−> [indi,birt,plac].
birthref −−> [indi,birt,sour,refn].
personname −−> [@],[indi,name].
target −−> birthcertificate.
```

Figure 4: Grammar fragment. Special characters: '@' level 0-value (record id), '+' pointer list, ':' pointer dereference.

rent elements with an equal or higher level number are popped, which makes the stack correspond to the current branch in the database hierarchy. If the stack corresponds to a list of terminal symbols in the grammar, then the current line is indexed for later use by the value at level 0. All grammar rules are expanded to terminal symbols and subsequently dereferenced for each of the index values in the previous step. If an expanded rule contains a pointer list (indicated by a + symbol) then the rule is duplicated for each element of the pointer list associated to the current index value before dereferencing. As an example the algorithm in Figure 3 applied to the database in Figure 2 using the grammar in Figure 4 on the index value @F294@ will produce three duplicate protorecords which can be dereferenced to certificates. Figure 5 provides an example that matches the Genlias certificate in Figure 1. Note that the family name of the mother differs between the databases.

The use of a domain-independent grammar provides a flexible parser for Gedcom or structurally

Protorecord

[@], [fam,chil(2)]:[indi,birt,date],
[fam,chil(2)]:[indi,birt,plac],
[fam,chil(2)]:[indi,birt,sour,refn], [fam,chil(2)]:[@],
[fam,chil(2)]:[indi,name], [fam,husb]:[@],
[fam,husb]:[indi,name], [fam,wife]:[@],
[fam,wife]:[indi,name]

Certificate

@F294@, 16 MAY 1883, Wonseradeel, 176,
@I849@, Sierk/Rolsma/, @I840@, Sjoerd/Rolsma/,
@I787@, Agnes/Welderink/

Figure 5: Parsing example for index value @F294@ using the pointer [@F294@,CHIL(2)], which is @I849@.

similar data formats. Additionally, only information that corresponds to an element of a target record is indexed, resulting in a light-weight procedure. The output of the parser can be directly used for record matching, which is described in the next section.

5 Matching

After parsing, both databases are represented in the same data format. This enables a definition of similarity between records based on the values of corresponding fields. In the current experiments a partial similarity measure is used, meaning that any sufficiently large subset of the corresponding fields must be similar whereas the complement set remains unchecked. This approach assumes sparseness of high-dimensional data, which implies that the set of field values of each record is unique and moreover any large subset of field values is also unique. This property can easily be verified on a given database and if it holds, the similarity measure can be simplified accordingly. For the current experiments this allows for name variation in civil certificates which is hard to detect automatically by similarity measures. A certificate, as discussed in Section 3, generally contains at least three individuals, which amounts to six names in total (given names and family names). If one of the names is subject to large variation in two matching records (for example *Elizabeth* vs. *Lisa*), this match might be undetected when using all names in the record comparison. However, by ignoring this field in a partial comparison the match will be discovered.

A partial record similarity measure can be de-

fined by stating similarity requirements for each of the fields used in the measure and relations between the requirements. As an example, consider the matching between marriage certificates based on the year of marriage and the names of the bride and bridegroom (four names in total) which is used in the current experiments, as stated in Figure 6. Note that the first clause in this definition requires an exact match on person names. This has the conceptual advantage that exact matching is more reliable than similarity matching based on, e.g., edit distance. Additionally, exact matching allows for straightforward string indexing and efficient look-up. Memory consumption is less efficient, the example index of two names out of four requires $\binom{4}{2} = 6$ entries per record. Therefore it might be necessary to adjust the similarity measure to meet computational resources.

> At least two out of four names are exactly equal, *and*
>
> > the year of marriage is equal or different by a small margin, *or*
> >
> > the year of marriage is different by a larger margin and the edit distance between name sequences is below a small threshold, *or*
> >
> > the year of marriage in a record is specified as a range and the year of marriage in another record is within this range, and the edit distance between name sequences is below a small threshold.

Figure 6: Record similarity measure for marriage certificates.

6 Results and verification

The record similarity measure in Figure 6 is applied to the Genlias database and a sample Gedcom database containing 1327 individuals and 423 families. As preprocessing, given names are reduced to the first token (for example: *Quentin Jerome Tarantino* → *Quentin Tarantino*). Separate family name prefixes, which are common in Dutch, are stripped using a stop list (for example: *Vincent van Gogh* → *Vincent Gogh*). The edit distance threshold and year margins required by the similarity measure are set according to empirical knowledge of the domain. A subset of the Gedcom records is used to match the timeframe of the Genlias database (1796–1920). Settings and matching results are displayed in Table 1. The matching is performed for the three main types of civil certificates: birth, marriage and death. For birth and death certificates the marriage record similarity measure (Figure 6) is used replacing the roles of bride and bridegroom by mother and father of the child or deceased person for birth and death certificates respectively (i.e., the name of the child or deceased person itself is not used). To avoid confusion with other siblings, the small year margin for birth and death certificates is set to zero. If multiple matching candidates are found using the record similarity measure, the match with the smallest edit distance between name sequences is used. The large amount of missed matches for birth and death certificates is expected, because the Genlias database is still under active development and a significant number of birth and death certificates are not yet digitized. Moreover, Gedcom databases generally contain many peripheral individuals for which no parents are listed (usually inlaws of the family of interest), prohibiting the reconstruction of birth and death certificates.

Edit distance threshold		5
Large year margin		10
Small year margin		
marriage		2
birth, death		0
Marriage	match	153
	no match	23
Birth	match	335
	no match	276
Death	match	100
	no match	239

Table 1: Matching parameters and results.

Verification of record matches should ideally be performed using a test set of known matches (a *gold standard*). However, for this particular combination of databases such a test set is not available. The lack of test sets extends to the majority of decentralized historical data, as well as Genlias itself (which does not have any kind of internal links or verification sets). This is a quite undesirable situation given the large variation in data quality and coverage between databases in the historical domain. Because the characteristics of any

two databases regarding the contents can differ to a large degree, the performance of a matching algorithm obtained on one database is not indicative for other databases. Put differently: every application of a matching algorithm has to perform its own verification, which is difficult in the absence of test sets.

6.1 Internal verification

A possible solution for the verification problem is to re-use the sparseness assumption to obtain a measure of support for a match. The matches returned by the similarity measure are based on a subset of fields. If other field values are equal or similar as well, they provide additional support for the match independent of the similarity measure. Note that this solution is only applicable if there are fields available which are not used in the record similarity measure. Moreover these fields should have a certain discriminative power, which rules out categorical variables like gender or religion. For many linkage tasks extra fields are not available, for example linking a marriage certificate of a person to the marriage certificate of this person's parents, in which case the only available information about the parents are the person names. However, in the current experiments a certificate from one database is being matched to the same certificate in another database, therefore the amount of available information is much larger.

A candidate field for verification is the serial number, which has been recorded since the start of the civil registry in the Netherlands. The numbers are assigned per year by the municipality issuing the certificate, meaning that the combination of municipality, year and serial number uniquely references a certificate (also known as a *persistent identifier* or PID). A shared PID between two records in a match therefore provides strong support for this match. However, in a Gedcom database serial numbers are not necessarily included. The source of the data can be something different than the civil registry, such as church records, or the database author might just have omitted the serial number. Moreover, if the source of the Gedcom record is the civil registry, then the match is not very indicative of the performance of the similarity measure in combining different data sources. Therefore, the serial number is of limited use only for verification purposes. Other candidate fields are dates and toponyms (location names). The year is used in the similarity measure, but the day and month can be used for support. For the current experiments three levels of support are defined: exact date match, a difference of 1 to 7 days, or a difference of more than 7 days.

In case of limited support from the verification fields, edit distance (or any other string similarity measure) can be used as an indication of the correctness of a match.

6.2 Toponym mapping

Toponyms cannot always be compared directly, because of the difference in use between Genlias and most Gedcom databases. In Genlias the toponym that denotes the location of the event is always the municipality that has issued the certificate. In a Gedcom database often the actual location of the event is used, which can be a town that is part of a larger municipality. A comparison between toponyms is therefore more informative after mapping each toponym to the corresponding municipality. In the current experiments a reference database of Dutch toponyms is used to perform the mapping. Because the municipal organization in the Netherlands changes over time, the year of the event is required for a correct mapping. Ambiguity for toponyms (i.e., multiple locations with the same name) can generally be resolved using the province of the event. In case that the province is not recorded the toponym can be disambiguated by choosing the location with the most inhabitants by default.

6.3 Interpretation of support figures

Table 2 shows the results of verification using serial numbers, dates and mapped toponyms as support fields. The support figures should be interpreted with the distribution of data values in mind. The first two rows of Table 2 represent matches with equal serial numbers. Most of these matches have equal PIDs (toponym and year equal as well). Given that each PID is unique these matches are correct. Differences in toponym are usually small for matches with equal serial numbers, therefore a PID match can be assumed (although support is higher for true PID matches). The third row represents matches with the same toponym and date, and also two names equal (by definition of the similarity measure). Note again that the match was selected using the names and the year only, and verified using the toponym and the full date. These matches could be incorrect, because it is

possible that different couples with (partially) the same name got married on the same day in the same place, for example. In the Genlias database this is the case for around 0.3% of all marriage certificates. Therefore, the sparseness assumption largely holds for this set of fields and these matches can also be considered correct. Similarly, other verification field values can be interpreted in terms of confidence in a match (based on the validity of the sparseness assumption) or counterevidence against a match (in case of large differences in field values). For the current experiments, the last row of matches should be considered incorrect. The relatively large number of incorrect matches for birth and death certificates can be attributed to the lack of coverage in Genlias. The best match is returned, however this assumes true matches to be present in the data set. The record similarity match can be adjusted using the verification fields, however it is preferred to keep similarity computation and verification separated.

field				marriage	birth	death
s	*t*	*d*	*e*			
+	+	+		69	170	9
+	-	+		2	30	0
–	+	+		41	20	1
–	+	$\sim$		0	33	6
–	+	–		2	1	0
–	–	+		10	2	7
–	–	$\sim$		2	5	10
–	–	–	≤ 3	11	2	3
–	–	–	> 3	16	72	64
total				153	335	100

Table 2: Verification results. Columns: (s)erial number, (t)oponym, (d)ate, (e)dit distance. Support level: + equal, $\sim$ 1–7 days difference,– not equal (*s,t*) or $>$ 7 days difference (*d*). Edit distance is only used for the matches without support from the verification fields (final two rows).

7 Application

The previous sections have discussed matching records from different databases that refer to the same event. However, most research in historical record linkage is focussed on links between events, such as a birth and the marriage of the parents listed in the birth certificate. These links can be added to a database by manual annotation or using automatic linkage methods. Different databases in the same population domain are likely to contain complementary and conflicting links, which can be used to increase the quality and quantity of links in both databases. To compare links between databases the records need to be matched first, which can be achieved using the record matching method from the current research.

To demonstrate the application of the method, a comparison is performed on links between marriage certificates in Genlias and corresponding links in the sample Gedcom database used in the matching experiments. A marriage certificate contains the marriage couple and the parents of both bride and bridegroom. A link can be defined between a marriage and the marriage of one of the parent couples (see Figure 7). For the Genlias database links have been constructed by selecting all record pairs with a maximum Levenshtein edit distance of 3 between name sequences. Additional record links are computed by converting each name to a base form and selecting record pairs with matching base name sequences. The details of the link computation are beyond the scope of this paper, for the current experiments

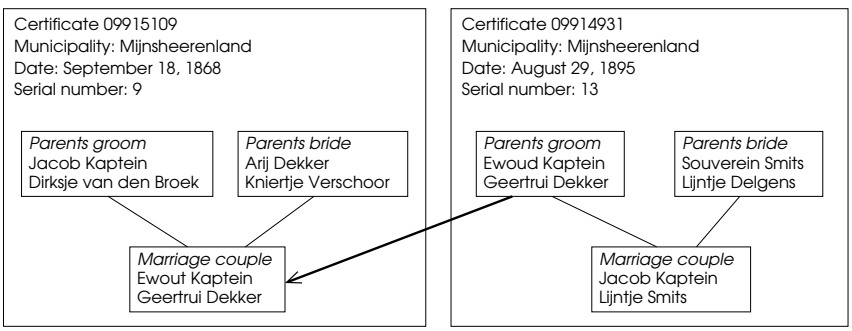

Figure 7: Example of a link between two Genlias marriage certificates, containing a small spelling variation: *Ewout* vs. *Ewoud*.

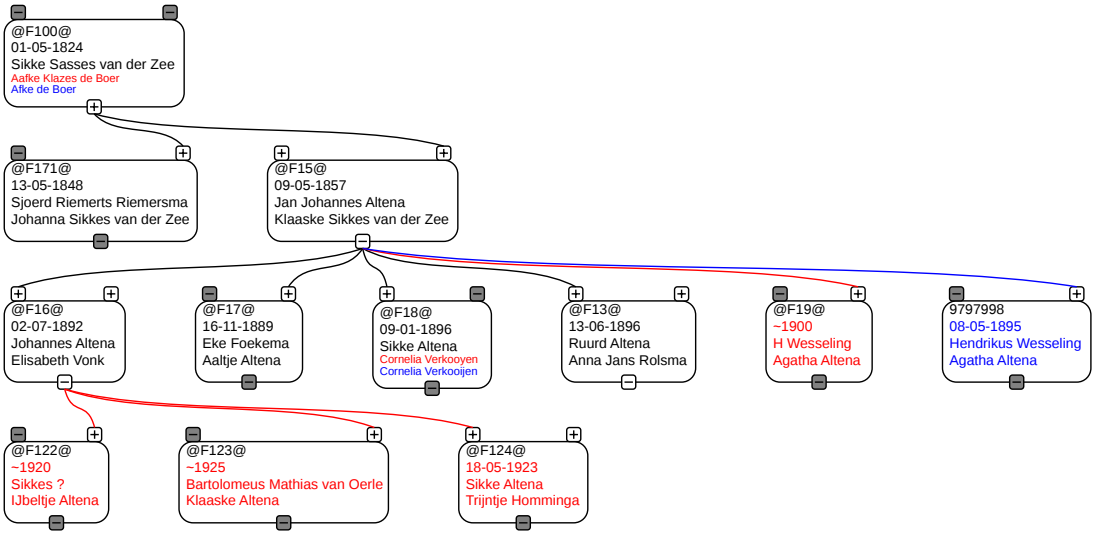

Figure 8: Visualization of link comparison.

only the resulting set of links between Genlias marriage certificates is of interest. In the Gedcom database links between marriages are already present. The link comparison procedure is as follows: first, marriage certificates are matched using the method described in Section 5. For every matched certificate the marriages of the children are identified using the links from Genlias and the Gedcom database (cf. Figure 7). These two sets of marriages are aligned using a slightly more strict version of the record similarity measure in Figure 6, to accomodate for the inherent similarity in names and timeframe of sibling marriages. Using the alignment, the links can be divided into three categories: present in both databases, present in the Gedcom database only, or present in Genlias only. A visualization tool is developed that shows the results of the comparison in a link tree (see Figure 8), which can be browsed by expanding or collapsing record links. Colours indicate differences between databases (red and blue for the Gedcom database and Genlias, respectively). Records @F19@ and 9797998 are an example of a false negative. The lower row is found in the Gedcom database only because these records are outside of the Genlias timeframe. The tool enables users to provide their own Gedcom database and identify differences with the nation-wide Genlias database. Due to data licensing issues the tool has not yet been released, however it could be integrated in the Genlias website in the future.

8 Conclusion and future work

In this paper a method is described to compare a dataset based on events (Genlias) to a dataset based on individuals (the Gedcom model). This method is complementary to most current approaches in record linkage, in which only datasets with the same conceptual structure are compared. The parser (Section 4) facilitates the transformation of data formats. A combination of multiple string indexing and field similarity measures provides a computationally efficient and flexible record matching method, as described in Section 5. The problem of verification without gold standard test data is addressed in Section 6. An application of the method in a visualization tool is presented in Section 7.

In future research, other Gedcom databases can be presented to the matching procedure. A crowdsourcing set-up can be envisioned to perform large-scale data collection and evaluation of the approach. The matching procedure itself can be refined by improving the record similarity measure or by incorporating a network approach in which record links can contribute to matching. Finally, functionality can be added to the visualization tool, preferably resulting in a public release.

Acknowledgment

This work is part of the research programme LINKS, which is financed by the Netherlands Organisation for Scientific Research (NWO), grant 640.004.804. The authors would like to thank Tom Altena for the use of his Gedcom database.

References

Madeleine Ball. 2012. python-gedcom: Python module for parsing, analyzing, and manipulating GEDCOM files. `https://github.com/madprime/python-gedcom/`.

Gérard Bouchard and Christian Pouyez. 1980. Name variations and computerized record linkage. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, 13(2):119–125.

Gérard Bouchard. 1992. Current issues and new prospects for computerized record linkage in the province of Québec. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, 25(2):67–73.

Peter Christen and Ross Gayler. 2008. Towards scalable real-time entity resolution using a similarity-aware inverted index approach. In *Seventh Australasian Data Mining Conference (AusDM 2008)*, volume 87, pages 51–60. ACS.

GEDCOM Team. 1996. The GEDCOM standard release 5.5. Technical report, Family and Church History Department, The Church of Jesus Christ of Latter-day Saints, Salt Lake City.

Paul Johnson. 2013. Gedcom — a module to manipulate Gedcom genealogy files. `http://search.cpan.org/~pjcj/Gedcom-1.18/`.

Michael Kay. 2004. Up-conversion using XSLT 2.0. In *Proceedings of XML: From Syntax to Solutions*. IDEAlliance.

Michael H. Kay. 2006. Positional grouping in XQuery. In *Proceedings of the 3rd International Workshop on XQuery Implementation, Experience and Perspectives (XIME-P)*.

Dallan Quass and Paul Starkey. 2003. Record linkage for genealogical databases. In *KDD-2003 Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, pages 40–42.

Patrick Schone, Chris Cummings, Stuart Davey, Michael Jones, Barry Nay, and Mark Ward. 2012. Comprehensive evaluation of name matching across historic and linguistic boundaries. In *Proceedings of the 12th Annual Family History Technology Workshop*. FamilySearch.

Roxane Segers, Marieke van Erp, and Lourens van der Meij. 2011. Hacking history via event extraction. In *Proceedings of the sixth international conference on Knowledge capture*, K-CAP '11, pages 161–162. ACM.

Matthew Smith and Christophe Giraud-Carrier. 2006. Genealogical implicit affinity network. In *Proceedings of the 6th Annual Family History Technology Workshop*. FamilySearch.

Peter Verthez. 2004. The Gedcom parser library. `http://gedcom-parse.sourceforge.net/`.

D. Randall Wilson. 2001. Graph-based remerging of genealogical databases. In *Proceedings of the 1st Annual Family History Technology Workshop*. FamilySearch.

D. Randall Wilson. 2011. Genealogical record linkage: Features for automated person matching. In *Proceedings of RootsTech 2011*, pages 331–340. FamilySearch.

Ian Winchester. 1970. The linkage of historical records by man and computer: Techniques and problems. *The Journal of Interdisciplinary History*, 1(1):107–124.

Scott Woodfield. 2012. Effective sharing of family history information. In *Proceedings of the 12th Annual Family History Technology Workshop*. FamilySearch.

Ivo Zandhuis. 2005. Towards a genealogical ontology for the semantic web. In *Humanities, Computers and Cultural Heritage: Proceedings of the XVI international conference of the Association for History and Computing*, pages 296–300.