Multi-Source Entity Resolution for Genealogical Data

Julia Efremova[1], Bijan Ranjbar-Sahraei[2], Hossein Rahmani[2],
Frans A. Oliehoek[3,5], Toon Calders[1,4], Karl Tuyls[5], and Gerhard Weiss[2]

Abstract In this chapter we study the application of existing entity resolution (ER) techniques on a real-world multi-source genealogical dataset. Our goal is to identify all persons involved in various notary acts and link them to their birth, marriage and death certificates. We analyze the influence of additional ER features such as name popularity, geographical distance and co-reference information on the overall ER performance. We study two prediction models: regression trees and logistic regression. In order to evaluate the performance of the applied algorithms and to obtain a training set for learning the models we developed an interactive interface for getting feedback from human experts. We perform an empirical evaluation on the manually annotated dataset in terms of precision, recall and F-score. We show that using the name popularity, geographical distance together with co-reference information helps to significantly improve ER results.

1 Introduction

The process of integrating disparate data sources for understanding possible identity matches has been studied extensively in literature and is known under many different names such as Record Linkage [2, 32], the Merge/Purge problem [19], Duplicate Detection [25, 9], Hardening Soft Databases [10], Reference Matching [23], Object identification [9], and Entity Resolution [18, 14].

Gradually, Entity Resolution (ER) has become the first step of data analysis in many application domains, such as digital libraries, medical research and social net-

[1]Eindhoven University of Technology, The Netherlands e-mail: i.efremova@tue.nl ·
[2]Maastricht University, The Netherlands e-mail: {b.ranjbarsahraei, h.rahmani, gerhard.weiss}@maastrichtuniversity.nl ·
[3] University of Amsterdam, The Netherlands e-mail: f.a.oliehoek@uva.nl
[4] Université Libre de Bruxelles, Belgium e-mail: toon.calders@ulb.ac.be
[5] University of Liverpool, United Kingdom e-mail: k.tuyls@liverpool.ac.uk

works. Recently, ER has found its way into the genealogical domain as well [21, 27]. In this domain, a real person entity could be mentioned many times, for instance in civil certificates such as birth, marriage and death or in notary acts such as property transfer records and tax declarations. Usually, no common entity identifiers are available and therefore the real entities have to be identified based on alternative information (e.g., name, place, and date). All information presented in the corpus is distributed over different sources such as civil certificates and notary acts. As an example, consider a person named *Theodor Werners* born in *Erp* on *August 11th, 1861*. He got married to *Maria van der Hagen* in *1888*. *Maria Eugenia Johanna Werners* was their child, born in *Erp* in *October 1894*. Two years after the child's birth, they bought a house in *Breda*. *Theodor* died in *Breda* on *September 1st, 1926*. In our corpus, this information is spread over respectively the birth record of *Theodor*, a marriage certificate of *Theodor* and *Maria*, the birth certificate of their child, a notary act available in full text, and the death certificate for *Theodor*. All these documents do not contain personal identifiers, may contain name variations, or be available in full text only. Applying ER to such a problem poses many challenges such as name alternatives, misspellings, missing data and redundant information.

Genealogical data contains a huge amount of inaccurate information and different types of ambiguities, therefore applying proper ER techniques for cleaning and integrating the reference extracted from different historical resources, has received much attention. Sweet et al. [35] use an enhanced graph, based on genealogical record linkage, in order to decrease the amount of human effort in data enrichment. Schraagen et al. [31] predict record linkage potential in a family reconstruction graph by using the graph topology. Lawson [22] uses a probabilistic record linkage approach for improving performance of information retrieval in genealogical research. Recently Bhattacharya and Getoor [3] propose a collective entity resolution approach where they use the relational information about references and combine it with similarity between common attributes. Christen [9] describes in depth a variety of data matching techniques from a statistical perspective. He addresses main challenges in the overall data matching process including data pre-processing, name variations, indexing, record comparison and classification. The key application of information retrieval is also addressed by the work of Nuanmeesri and Baitiang [26], in which they discussed the design and development of suitable techniques that can improve efficiency of a Genealogical Information Searching System. Singla and Domingos [34] propose an integrated solution to the entity resolution problem based on Markov logic that combines first-order logic and probabilistic graphical models by attaching weights to first-order formulas.

The mentioned work in Genealogical ER mainly focus on linking references with *homogeneous structures* where the number of descriptive features and their types are identical in all references. In this chapter, in contrast, we are interested in applying ER to a real-world dataset with a *heterogeneous structure* where different references come from qualitatively different sources and references no longer have similar descriptive features. We refer to this problem as *ER* on multi-source data.

In particular, we are interested in performing multi-source ER on a database of historical records of a Dutch province called North Brabant. There are two types

of sources in this dataset: “Civil Certificates” and “Notary Acts”. The former type has a structured form and contains three certificate types birth, marriage and death certificates while the other type contains free-text historical documents indicating involvement of references in different formal activities such as property transfers, loans, wills, etc. We give the detailed description of the input source types in Section 2. To integrate these types of sources we, first, identify all the references involved in a given set of notary acts and then link the extracted references to their birth, marriage and death certificates. This process faces many challenges such as ambiguity due to name alternatives, misspellings, missing data or redundant information.

The remainder of this chapter is structured as follows. In Section 2, we describe our real-world collection of historical data. In Section 3, we discuss the general ER approach and its implementation to our data. The reference extraction approach is described in Section 4. The indexing techniques that we use to generate potential candidate record pairs we describe in Section 5. In Section 6, we introduce informative attributes of references, describe a computation of attribute similarities and then present a final classification of reference pairs. In Section 7, we introduce the tools developed for historians to label data and we show the evaluation of the obtained results. In our analysis we study the influence of the individual steps on the overall precision and recall. Section 8 offers a discussion about drawbacks and potential extensions of the proposed approach. Concluding remarks are included in Section 9.

2 Data Description and Problem Formulation

The genealogical data used in this chapter is provided by the Brabants Historisch Informatie Centrum (BHIC)[1]. The data consists of two main sources. The first source, civil certificates, is comprised of the birth, marriage and death certificates belonging to North Brabant, a province of the Netherlands, in the period 1811-1940. The level of detail of each certificate varies very much. Table 1 lists the descriptive features for each certificate type. As shown in Table 1, Birth certificates include three individual references (i.e., child, father and mother). Death certificates include four individual references (i.e., deceased, father, mother and partner of deceased). Finally, Marriage certificates include six references (i.e., groom, bride and parents of each). Each mentioning of a person in each certificate is called a *reference*.

This database consists of around 1,900,000 certificates with around 7,500,000 references in total. The exact number of documents and details about the distribution between the different certificate types are provided in Table 2. Volunteers digitize scans of the original manuscripts and make them available in a database format. At this moment, the digitisation work is the most complete for marriage and death certificates and the database continuously grows.

[1] http://www.bhic.nl/, the website of BHIC is available in Dutch only

Table 1 Available features for each certificate type. PoB and PoD stand for place of birth and place of death respectively.

Birth cert.	FIRSTNAME, LASTNAME, GENDER, BIRTHDATE, POB, FATHERFIRSTNAME, FATHERLASTNAME, MOTHERFIRSTNAME, MOTHERLASTNAME
Death cert.	FIRSTNAME, LASTNAME, GENDER, BIRTHDATE, POB, DEATHDATE, POD, FATHERFIRSTNAME, FATHERLASTNAME, MOTHERFIRSTNAME, MOTHERLASTNAME, PARTNERFIRSTNAME, PARTNERLASTNAME
Marriage cert.	GROOMFIRSTNAME, GROOMLASTNAME, GROOMAGE, BRIDEFIRSTNAME, BRIDELASTNAME, BRIDEAGE, GROOMFATHERFIRSTNAME, GROOMFATHERLASTNAME, GROOMMOTHERFIRSTNAME, GROOMMOTHERLASTNAME, BRIDEFATHERFIRSTNAME, BRIDEFATHERLASTNAME, BRIDEMOTHERFIRSTNAME, BRIDEMOTHERLASTNAME

Table 2 Statistical information of civil certificates.

Type	Number of documents
Birth Certificate	345,046
Marriage Certificate	391,273
Death Certificate	1,042,558
Number of references	7,557,051

A sample civil certificate is shown in Table 3. We see that the certificate has a pre-formatted structure. We illustrate the certificate as it is presented in the database. Notice that although this record is structured, there may be inconsistencies in the way the fields have been completed. For instance, the field *gender* is filled as *zoon van*[2] instead of explicitly mentioning being *male* or *female*.

The second source, the dataset of notary acts, consists of around $234,000$ free-text documents of North Brabant before 1920. These free-text documents include information about involvement people in different formal activities such as property transfers, loans, wills, etc. Notary acts are in a free-text format and not all details are mentioned in a structured way. They require additional Natural Language Processing (NLP) techniques to extract information such as person names from the text. According to the type of formal activity, the detailed information mentioned in each notary act varies very much. For instance, an inheritance act many person names and many relationships are mentioned, whereas in the purchase agreements usually only one person name is mentioned.

Table 4 shows statistical information about the dataset of notary acts.

An example of a notary act is shown in Table 5 (the person names are underlined). The notary act also contains a short summary and details provided by volunteers: the date and the place of a document.

[2] 'zoon van' is the Dutch term for 'son of'.

Table 3 An example of civil certificate showing birth data.

Person Name	Teodoor Werners
Gender	zoon van
Place of Birth	Erp
Date of Birth	14-04-1861
Father Name	Peter Werners
Father Profession	shopkeeper
Mother Name	Anna Meij
Mother Profession	-
Certificate ID	6453
Certificate Place	Erp
Certificate Date	16-04-1861

Table 4 Statistical information of notary acts.

Description	Number of acts
Number of acts	234, 259
Number of act types	88
Number of notary acts of type *'property transfer'*	23275
Number of notary acts of type *'sale'*	17016
Number of notary acts of type *'inheritance'*	12335
Number of notary acts of type *'public sale'*	10593
Number of notary acts of type *'obligation'*	9006

Table 5 An example of a notary act.

Theodor Werners, burgemeester van Boekel en Erp, wonend te Boekel bekent schuldig te zijn aan gemeente Erp Fl. 200,–. Waarborg: woonhuis, tuin, erf, bouw- en weiland Dinther en bouw- wei- en hooiland te Boekel. Zijn vader Peeter Werners . . . (*Theodor Werners, mayor of Boekel and Erp, living in Boekel, admits to owe the township of Erp 200 guilders. Security: house, garden, yard, farmland, and pasture Dinther and farmland, pasture, and meadowland in Boekel. His father Peeter Werners . . .*)

TextID	100
Place	Boekel
Date	24-07-1896

To integrate these two heterogeneous types of input sources we, first, extract all the references from the civil certificate. Second, we identify all the references involved in a given set of notary acts. Finally, we link the references mentioned in each notary act to the references extracted from civil certificates. Our main goal is to find all birth, marriage and death certificates for every person mentioned in a notary

act. We formalize the ER problem as follow. Let $\mathcal{R} = \mathcal{R}_N \cup \mathcal{R}_C$ denote the total set of references, where $\mathcal{R}_N = \{r_{n_i}\}_{i=1}^{k}$ and $\mathcal{R}_C = \{r_{c_j}\}_{j=1}^{l}$ are the sets of references extracted from notary acts and civil certificates respectively. Each reference r_{n_i} and r_{c_j} has a value for each attribute in $\mathcal{A} = \{a_i\}_{i=1}^{m}$. We aim to find a set of real world entities $\mathcal{E} = \{e_i\}_{i=1}^{m}$ such that $e_i \subseteq \mathcal{R}$. The set of entities can be represented as a partitioning of the references, in which each partition corresponds to the set of all references that belong to the same entity. Every reference can belong to only one entity: $r \in e_i \wedge r \in e_j \Rightarrow i = j$. Then the ER problem can be defined as: $\forall r_{n_i}, r_{c_j} \in \mathcal{R} : \exists e' \in ER(\mathcal{R}) : r_{n_i} \in e' \wedge r_{c_j} \in e'$ and vice versa. The objective is to determine whether $r_{n_i}, r_{c_j} \in \mathcal{R}$ are the same entity e' in the real world.

3 Entity Resolution for Genealogical Data

To apply ER to the multi-source collection of historical data we use the following steps: *data collection and preparation*, *indexing*, *similarity computation*, *learning algorithm and classification* [9, 25]. We illustrate the overall ER process in Fig. 1.

The first step is data collection and preparation, during which the raw data is collected from various sources, then cleaned and preprocessed. During this step we have to assure that all references have the same format (standardized date, null values, special characters, etc.) and extract all person references from civil certificates and notary acts. As discussed in Section 2, reference extraction from civil certificates requires data cleaning and standardization of null values. The notary acts, however, require more complicated preprocessing techniques to extract person names and other information from them. Dealing with the notary acts we use the natural language processing techniques and named entity recognition approaches [6, 24] which we discuss in Section 4.

The second step of the ER process is data indexing and generation of candidate record pairs for further comparison. In order to avoid having to compare every references in one source with every references in another source, we split the references into different partitions using an indexing technique. This partitioning allows us to reduce computational complexity by reducing the number of candidate record pairs. We discuss the applied indexing algorithm in Section 5.

The next step is the similarity computation step. The similarity score between two attributes, associated with two distinct references, is computed based on their types. We compare two attributes with type $String$ using the *hybrid string similarity measure* described in [12] and trained on the dataset of Dutch names from Meertens Instituut[3]. The hybrid measure combines using logistic regression [29] five string similarity functions: *Soundex* (SN), *Double Metaphone*, (DM), *IBMAlphaCode* (IA), *Levenshtein distance*, *Smith Waterman distance* [15, 37, 25].

For attributes of type $Date$ we calculate the similarity as the date difference in years. For every pair of references we compare essential attributes using an appro-

[3] http://www.meertens.knaw.nl/nvb/

priate similarity measure. We discuss more about the features and methods for the similarity computation in Section 6.

The last step of the overall ER process is learning a model and classification. The score function computes the final similarity score between candidate record pairs using a supervised classification. Then pairs of references are classified into classes *Matched* or *non-Matched*, based on a threshold value of the score function. The classification step is described in Section 6.5.

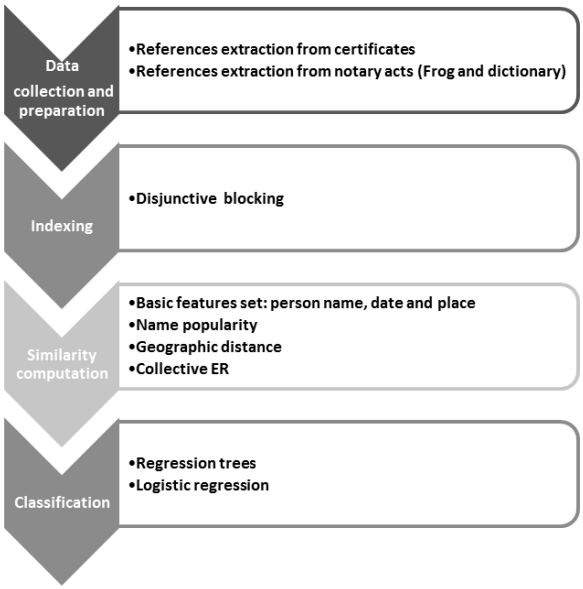


Fig. 1 The overall ER process.

4 Data Collection and Preparation

We pre-process the data in order to extract references and other information from various sources. Civil certificates require a cleaning phase. There are many situations when person name in civil certificates is unknown. It happens, for instance, when child died at birth and did not receive a name. Then in birth and death certificates his name may be filled as: *onbekend*[4], *niet vermeld*[5], etc. We replace the common terms by *null* values.

[4] 'onbekend' is the Dutch term for 'unknown'

[5] 'niet vermeld' is the Dutch term for 'not mentioned'

Other fields in civil certificates also require standardization. We generalize the date of a document to its year, because it is specified in different formats: as a year, as an exact date or as text, for instance *[1861] Augustus*. We use regular expressions and consider the first four digits in the date field as the year of the document. The gender of a person is not always clearly mentioned. Instead of the direct specification of male or female, the gender may be given is textual format, for instance using terms such as: *zoon van*, *zoontje van* [6], etc. We standardize those values to an appropriate format.

After the cleaning phase civil certificates are ready for the reference extraction. Table 6 shows three sample references which are extracted from the civil certificate of Table 3.

Table 6 The references extracted from the sample civil certificate in Table 3.

ref_ID	Person Name	Place	Date	Cert_ID
124358	Teodoor Werners	Erp	14-04-1861	6453
124359	Peter Werners	-	-	6453
124360	Anna Meij	-	-	6453

To extract references from notary acts we apply the NLP tool Frog [5] which is a Dutch morpho-syntactic analyzer and dependency parser. The Frog tool extracts most of the names from notary acts, although some names are missed. To check the recall of name extraction we manually extracted names from randomly selected notary acts and compared to Frog results. Frog failed to identify 41 out of 166 manually extracted names. These missed references have a huge influence on the overall performance of our ER task, as there is no way to compensate for these missed references later on in the chain. Therefore, in addition to Frog name extraction we designed our own special-purpose name NLP rules.

The process of gathering the names from the notary acts hence proceeds in two steps. In the first step, **preprocessing**, some basic text polishing algorithms are run on the text, such as removing extra spaces and wrongly encoded symbols. Also punctuation is detected and checked, mistakes are corrected or reported for a manual inspection. In the second stage, which is **word labeling**, punctuation, the position of a word in the sentence, and dictionary information extracted from the structured data are used to label the words in the text as *person name*, *person name prefix*, *location name*, *location prefix*, *number*, *relation indicator*, *conjunction*, and *preposition*. This approach is iterative: for instance detecting a *location prefix* such as "te" (variant of "in" in Dutch used with locations) helps recognizing a following name as a location. Having labeled the words in the text, in the third stage, named **person name resolution**, the person references are extracted by considering every connected set of words labeled as "person name" and possibly a person name pre-

[6] 'zoon van' and 'zoontje van' are Dutch terms for 'son of'.

fix in word sequence. Location entities usually follow a location prefix and contain a set of location names. The total number of extracted references and the general statistics about name extraction from notary acts is presented in Table 7.

Table 7 Statistical information of reference extraction notary acts.

Total number of extracted references	1,155,400
Minimum number of references in a notary act	1
Maximum number of references in a notary act	214
Average number of references in a notary act	5.7

As we see from Table 7 every notary act contains at least one reference. However, the number of person references per document varies a lot from only 1 to 214 references per document.

Returning to our example, using the NLP techniques described above, a sample person reference extracted form the notary act of Table 5 is shown in Table 8. The date and the place of the document are available in a short human-annotated summary of a notary acts and do not require an NLP extraction.

Table 8 The references extracted from the sample notary act in Table 5.

ref_ID	Person Name	Place	Date	TextID
94254	Theodor Werners	Boekel	24-07-1896	100
94255	Peeter Werners	Boekel	24-07-1896	100

The data extracted from a notary act has only few features as compared to the structured data shown in Table 3.

5 Candidate Generation

It is computationally very expensive to compare every reference extracted from a notary act with every reference occurring in the civil certificates. Therefore, we use *indexing* to reduce the total number of potential candidate pairs, as this would require comparing $|\mathcal{R}_N| \times |\mathcal{R}_C|$ pairs. We do not compare every reference from a notary act with every reference from a certificate, but instead divide the references into buckets based on some basic characteristics, such as for instance the first four letters of the last name. Only references that fall into the same bucket will be compared. Obviously, the smaller the buckets, the faster we will be able to carry out

all comparisons, but on the other hand, we may lose some pairs of references that refer to the same entity, because they accidentally get assigned to different buckets. To reduce this risk, we need to carefully select the characteristics on which we will decide the division into buckets, in order to optimize this trade-off.

In this work we apply an adaptive blocking algorithm proposed by Bilenko et al. in [4] which is based on learning an optimal set of disjunctions of blocking functions based on a labeled training set. To construct the set of predicates we use heads and tails of phonetic functions with variable size: 2, 3 or 4 characters for the heads, and 3 or 4 characters for the tails. There is a variety of phonetic functions. They use several rules to transform a name to a phonetic encoding. Some algorithms ignore all vowels and group the consonants, other algorithms analyze consonant combinations. For the experiments in this chapter, we construct an *indexing* using specified head and tales of the four phonetic functions: *Soundex*, *Double Metaphone*, *IBMAlphaCode* and *New York State Identification and Intelligence System* [7, 12]. Table 9 shows an example of applied phonetic to encode imprecise names.

Name	SN	DM	IA	NYSIIS
Theodoor	T600	TTR	0114	TADAR
Theodor	T600	TTR	0114	TADAR
Theodorus	T620	TTRS	0114	TADAR

Table 9 An example of phonetic keys.

We analyze the performance of phonetic keys on the dataset of Dutch names as is described in [12]. To index our data we use disjunctions of the following: $Head(Soundex, length = 4)$, $Head(DM, length = 4)$, $Head(NY, length = 4)$, $Tail(IA, length = 4)$. We apply the resulting formula to index first and last names in historical documents. That is, two references r_{n_i} and r_{c_j} will be compared if and only if they agree on at least one of these functions, hence the name "disjunctive blocking". In this way we can significantly reduce the number of candidates to be checked without losing too many true matches. Using different disjunctions of phonetic predicates helps us to reduce the number of candidate pairs to compare however some name variations can still occur in different partitions. In Section 7, we show that maximum achieved recall is above 92% which is relatively high. The missed 8% is partly because of the selected indexing approach and partly because of the name extraction phase. The first four letters of phonetic keys (e.g., first four letter of Soundex) are commonly used in literature for indexing purposes [9]. It is possible to use a less restrictive indexing strategy: only first letter of person names. However this leads to a significant increase in the number of potential candidate pairs.

6 Feature Similarity Computation and Classification

One of the main challenges in multi-source ER is the lack of available information. It is virtually impossible to decide whether or not the person mentioned in a notary act is the same person as a person in a specific civil certificate, if there are more than 1000 other civil certificates that belong to persons with the same name in the same time period. For instance, it is much easier to find civil certificates that belong to *Bernardus Wijngaarden* whose name appears only few times in historical documents, than to find civil certificates that belong to *Theodor Werners* whose name appears much more often in the database. Therefore, in this section we, first, describe in detail informative features that we use to compare references, then we show how to compute a similarity for every feature and to classify a reference pair into *Matched* and *Non-Matched*.

6.1 ER Basic Features

We define a basic feature set $\mathcal{F}$= $\{f_1, \ldots, f_n\}$, which are used to compare pairs of potential candidate matches. These features can be obtained directly from one notary act and one civil certificate and do not require additional information. To construct a basic feature set $\mathcal{F}$ we use person *FullName*, *Date* (in years) and *Place*. Those attributes can all be extracted from a notary act. We use NLP techniques to extract person names as described in Section 4. Date and place of the document are specified by volunteers in a summary of the notary act. We compare the attribute $FullName$ by a hybrid string similarity function [12], the similarity between dates as the difference in years and the similarity between places as a Boolean value which is $true$ when the two places in the pair of references have exactly the same name, and otherwise $false$. During the next step we extend the basic set of features and experiments by introducing additional attributes.

6.2 Considering Name Popularity

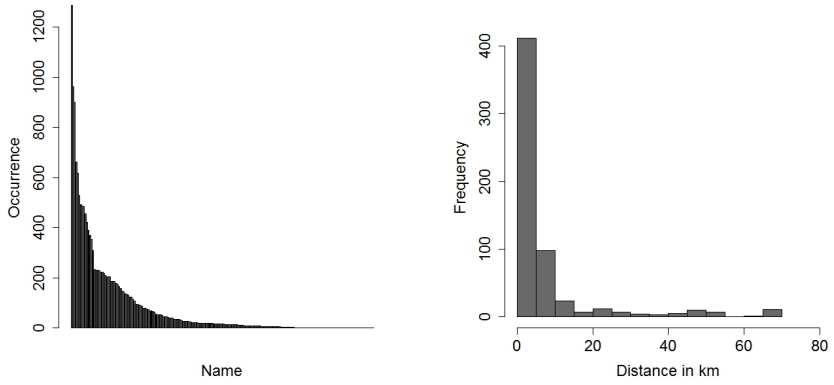
The person name is an important attribute in genealogical ER, however it is more difficult to find a certain match for a very common name than for an uncommon one. We think that the uncertainty caused by popular names is inevitable, and therefore aim at designing an algorithm to consider this important feature.

To compute the popularity of each name in the database, we make a list of full names using information from death certificates. We use only death certificates because they are more prevalent than the other types of certificates (i.e., birth and marriage). Under $FullName$ we consider the combination of $FirstName$ and $LastName$ of each person. We did not consider documents where first or last name were not filled. In the next step, for every full name we estimate its popularity as the

fraction of name occurrence in death registers to the total number of death registers. In this way we assign the lowest score to uncommon names and the highest score to the most popular ones.

We assign a name popularity value to a full name of a reference in a civil certificate. We do not compute name popularity of references extracted from notary acts because the name extraction using NER techniques is not always accurate. For instance, the name can be extracted with an extra symbol or an extra word like *'Theodor Werners.'* or *'Theodor Werners te Erp'* which will not appear in the list. In the first case the name is extracted with an extra dot at the end and in the second case with the location prefix *te Erp*. If the name does not exist in the list, we assign popularity value 0. We extend the basic feature set $\mathcal{F}$ by adding name popularity as an extra feature: $\mathcal{F} \leftarrow \mathcal{F} \cup \{f_{popular}\}$.

We explore manual matches by humans on a manually annotated dataset described in Section 6.5. We are interested to see how often a match is assigned to popular names. Fig. 2a shows the occurrence of every name matched manually in the overall collection of civil certificates. The highest values on the diagram belong to names such as: *Maria Janssen* (occurs *1,242* times in civil certificates), *Martinus Heijden* (962 times), *Johanna Martens* (900 times). It means that humans during the manual annotation identified only few matches that belong to very common names and most of manually annotated matches belong to relatively uncommon names. Name popularity information helps to improve the ER results compared to the basic set of features as discussed in Section 7.



(a) Occurrence of person names that were manually matched.

(b) Geographic distance between pairs of manually labelled references in km.

Fig. 2 Distributions of manual matched references.

6.3 Considering Geographical Distance

Although the historical documents belong only to North Brabant, which is relatively small, it is more likely to find a match between people from the same place than from different places in Noord Brabant that are farther apart. Therefore, we consider the geographical distance. We define the following three main groups based on geographical distances.

- intra city distance (from 0 to 5 km)
- inter villages distance (from 5 to 20 km)
- inter cities distance (more than 20 km)

For each place mentioned in the documents we define a spatial component: longitude and latitude (α, δ). We use the database of places provided by The Historical Sample of the Netherlands (HSN)[7]. This database contains 7925 names of places in the Netherlands and their geographical coordinates. More details about the database of places can be found in [20]. Another way to retrieve geographical coordinates is to use the Google Geocoding API[8] with geo lookup functionality. However, the tool often confuses places that existed in the past with recent different more recent locations that have the same name. We calculate the geographical distance in kilometers for each pair of potential matches using the coordinates of the two places: (α_1, δ_1) and (α_2, δ_2) using Equation 1 obtained from [28]:

$$distance = 2\mathcal{R} \cdot arctan\left(\frac{\sqrt{hav(\theta)}}{\sqrt{1 - hav(\theta)}}\right), \tag{1}$$

where $hav(\theta) = sin^2(\frac{\delta_1 - \delta_2}{2}) + cos(\delta_1) \cdot cos(\delta_2) \cdot sin^2(\frac{\alpha_1 - \alpha_2}{2})$ and $\mathcal{R}$ is the Earth radius.

We compute the geographic distance between two references for every candidate pair. To analyze how often humans are able to find a match between references from different places we made a distribution of geographical distances between two references in the manually annotated dataset as presented in Fig. 2b. Human annotators mainly find links between references that are from places not far apart. However there are some references that were identified where the distance was up to 80 km within North Brabant. On the next step we convert geographic distances to defined groups and add this feature to the feature set: $\mathcal{F} \leftarrow \mathcal{F} \cup \{f_{migration}\}$. Adding the geographical distance helps slightly to improve results as it described in Section 7.

[7] http://www.iisg.nl/hsn/data/place-names.html

[8] https://developers.google.com/maps/documentation/geocoding/

6.4 Collective ER with Co-Occurrences of References

We carry out experiments with collective entity resolution [36, 3] and take into account entity co-occurrence across the documents. All references within the same document are related to each other by a co-occurrence relationship. The co-occurrence relationship of references is widely used in ER and information retrieval. The idea behind it is that if entities often occur together, they are probably related to each other. We deal with the co-occurrence information by treating it as an additional feature for ER. For each pair of references (r_{n_i}, r_{c_j}) we construct the neighborhood sets $Nbr(r_{n_i})$ and $Nbr(r_{c_j})$ which include all co-occurred references of r_{n_i} and r_{c_j} respectively. We perform pairwise $FullName$ comparisons between all possible pairs of co-occurred references generated from $Nbr(r_{n_i})$ and $Nbr(r_{c_j})$.

Returning to our example, the neighborhood of the reference *'Theodor Werners'* extracted from a notary act contains one name $Nbr(r_{n_i})$ = *(Peeter Werners)* and the neighborhood of the reference *'Teodoor Werners'* extracted from a civil certificate has two names $Nbr(r_{c_j})$=*(Peter Werners, Anna Meij)*. We see that two neighborhoods have one similar name in common which has to be taken into account during the comparison of the references.

To compare $FullName$ attributes of co-occurred references we use again the hybrid string similarity function described in Section 3. Then we assign the final similarity score as the highest similarity score between all possible pairwise comparisons.

Considering only the highest similarity score between $Nbr(r_{n_i})$ and $Nbr(r_{c_j})$ makes an algorithm to disregard that compared references may have more than one co-reference. However finding at least one co-reference already helps us to improve the results significantly compared to the previous set of features as discussed in Section 7. Algorithm 1 demonstrates this approach.

Algorithm 1 Computation of reference co-occurrence

Input: A pair of references (r_{n_i}, r_{c_j}), a set of co-references $Nbr(r_{n_i})$ to r_{n_i}, a set of co-references $Nbr(r_{c_j})$ to r_{c_j}
Output: Computed co-occurrence information $f_{collective}(r_{n_i}, r_{c_j})$

$\mathcal{C} \leftarrow \emptyset$
for each co-reference m in $Nbr(r_{n_i})$ **do**
  for each co-reference n in $Nbr(r_{c_j})$ **do**
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{ComputeSim(m, n)\}$
  end for
end for
$f_{collective}(r_{n_i}, r_{c_j}) \leftarrow max(\mathcal{C})$
return $f_{collective}(r_{n_i}, r_{c_j})$

We add a collective feature based on the co-occurrence of references to the feature set: $\mathcal{F} \leftarrow \mathcal{F} \cup \{f_{collective}\}$.

6.5 Classification

The last step of the overall ER process is classification. Earlier in this section we described informative attributes of reference pairs and appropriate attribute similarity metrics to compare them. However, to compute the overall similarity score of every reference pair we need to assign an appropriate weight to each attribute. This approach allows to estimate the final probability of each match using a score function. The score function computes the final similarity score between two references based on the results of single attribute comparisons. We learn the score function on a training dataset that we will discuss in detail in Section 7. After that, pairs of references are classified into *Matched* or *non-Matched* based on a threshold value.

There exists a variety of techniques from statistics, modeling, machine learning and data mining [16, 8] for designing a score function that combines individual similarity scores. We apply two predictive models. First, we use logistic regression [29] and calculate the score function as follows:

$$\text{Score}(r_{n_i}, r_{c_j}) = \frac{1}{1 + e^{w_0 + \sum_{l=1}^{k} w_l \cdot \text{sim}(r_{n_i}.a_l, r_{c_j}.a_l)}} \tag{2}$$

where parameters w_0 to w_k are learned in the training phase.

The function $\text{sim}(r_{n_i}.a_l, r_{c_j}.a_l)$ represents similarity measures of the attribute a_l between two arbitrary references r_{n_i} and r_{c_j}, while reference r_{n_i} and r_{c_j} have k attributes in common.

Additionally we apply *Regression Trees* [29]. The leaves of a tree represent class labels ($Matched$ or $Non - Matched$) whereas its nodes represent conjunctions of the features values.

7 Experiments and Results

The application of the multi-source ER approach and its evaluation on real-world data requires additional steps. The first step is the process of gathering expert opinions. This is a crucial requirement for the evaluation. Therefore in this section first we present an interactive web-based interface which is used for getting input from humans. Then we elaborate on the application and the evaluation of the model.

We have two sets of experiments. **Experiment 1** is to obtain the performance results of ER algorithms on the manually annotated dataset. After the first experiment we select all *false-positive* (FP) matches that correspond to the maximal *F-score* value in order to evaluate to what extent they are really incorrect links or rather concern omissions in the human labeling. Given the extraneous nature of the labeling tasks it is indeed conceivable that human annotators may have missed a significant part of the links. Hence, in **Experiment 2** we evaluate new precision value after a manual review of false positive matches according to the prediction. In order to

assess the performance of our results we apply the 10-fold cross-validation method on the entire ER approach.

7.1 Manual Labeling Phase

In order to generate adequate training/test set for the classification process, a web-based interactive tool was developed [13] which allows historians to navigate through the structured and unstructured data, and label the matches they find between various references. This tool uses various programming tools for storage, exploration and refinement of available data; it benefits from an intelligent searching engine, developed based on the Solr[9] enterprise search platform, with which historians can easily search through the dataset. Basically, the required data can be found via person name, location, date and relationship types.

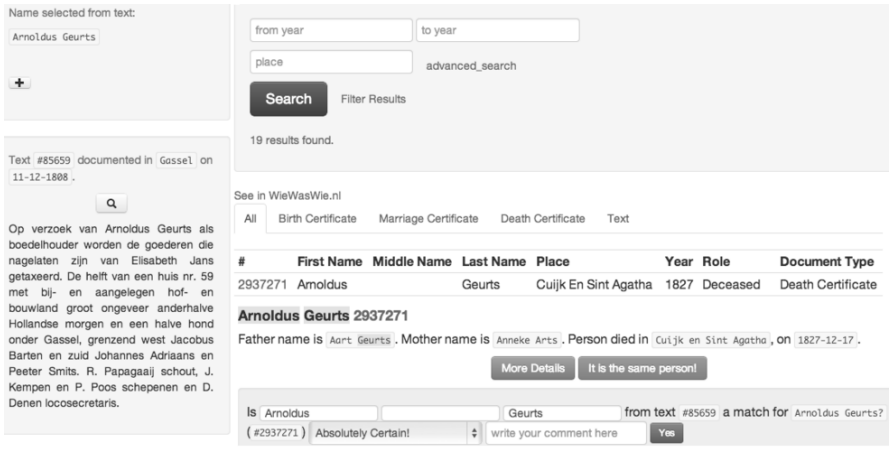


Fig. 3 The developed web-based labeling tool for generating the required training/test dataset.

The developed Labeling tool, shown in Fig. 3, is very powerful and easy to use, which assists historians to link name-references mentioned in notary acts to name-references mentioned in civil certificates.

The time required to report a correct match between two name-references varies from a few seconds to probably hours of time, depending on how similar two references are (e.g., whether places, dates, ages, professions and relatives match or not), and how easy it is to compare those two references. Consequently, the level of confidence in reporting a match varies. Therefore, the actions that historians take (e.g., which keywords they take and how fast they can recognize a match), and their level

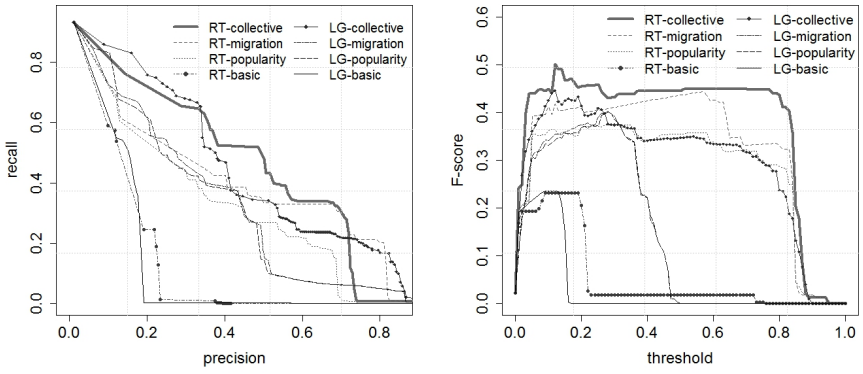
[9] http://lucene.apache.org/solr/

of confidence in reporting the match are all stored in the database. As a result, a rich benchmark is generated that includes the list of matches, the level of confidence and the list of actions that historians search for before reporting the match.

We consider each pair of references labeled by a historian as an example of a positive match between two different sources of data. Due to insufficient information in a notary act, incomplete civil certificates or a very frequent person name, no matches might be found for some references. We assign a zero-matched status to such references. Using the developed tool we manually annotated 643 entity resolution decisions (matches between notary acts and civil certificates) from 82 randomly selected notary acts.

7.2 Experiment 1: ER before Manual Match Review

We evaluate the performance of the applied algorithms using the standard metrics precision, recall, and F-score. We compute the sets of True Positives (TP), False Positives (FP) and False Negatives (FN) as the correctly identified, incorrectly identified and incorrectly rejected matches, respectively. In Fig. 4a, we show the achieved precision and recall values for different sets of features and for the two prediction modes: regression trees (RT) and logistic regression (LR). Fig. 4b presents the evaluation of results in terms of F-score and threshold values. Table 11 shows the maximum F-score value and corresponding precision and recall.



(a) Precision vs. recall

(b) F-score vs. threshold

Fig. 4 Evaluation of ER quality using different feature sets. The label names: *basic*, *popularity*, *migration* and *collective* correspond to the respective set of features described in Sections 6.

As Fig. 4 and Table 11 show, the results improve significantly by adding the additional information. The basic set of features is clearly not sufficient for obtaining an appropriate performance level. Adding name popularity to the basic set of features almost doubles the maximum F-score. This can be explained as follows: it is a very difficult task to be certain in assigning a proper match among a huge amount of similar documents that belong to persons with the same name, so the final decision requires additional information and the overall score for matches of documents with popular names should be lowered. Adding a geographical distance to the feature set yields also a minor improvement (7.0% for the RT and 0.2% for LR). The last analyzed feature which improves the results significantly, is co-occurrence information. It increases the max F-score by 5.8% and 4.3% for RT and LR respectively. To understand which features are more important we show the coefficients of the logistic regression in Table 10. These coefficients are applied to calculate the final similarity score using the function described in equation 2.

Table 10 The coefficients of the logistic regression

(Intercept)	Name similarity	Place	Date	Name popularity	Geographical distance	Co-reference
-6.39	6.30	0.93	-0.01	-21.11	-1.45	2.93

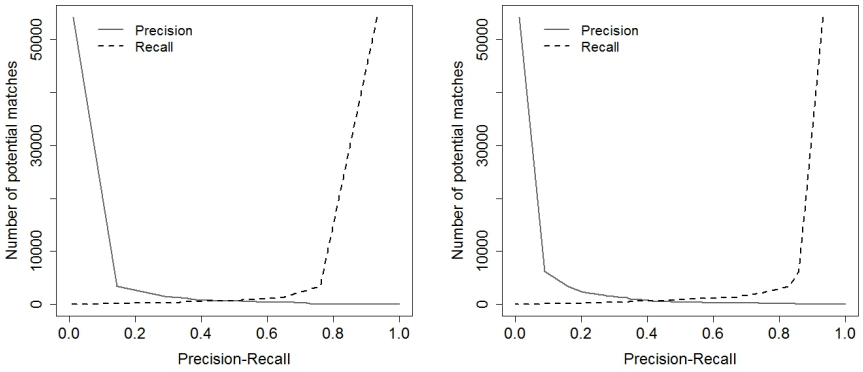
Overall, we compared the results of the two applied regression models. The highest F-score that we achieved is 0.502 by using the RT.

Table 11 The maximum F-score with corresponding precision and recall of different feature sets

	Logistic Regression			Regression Trees		
Features	Precision	Recall	max F	Precision	Recall	max F
basic	0.161	0.445	0.236	0.218	0.246	0.231
basic, name popularity	0.430	0.374	0.400	0.448	0.320	0.374
basic, name popularity, geo distance	0.434	0.375	0.402	0.679	0.330	0.444
basic, name popularity, geo distance, co-occur.	0.338	0.653	0.445	0.486	0.518	**0.502**

To show a computational complexity of applied overall ER approach we analyze a number of comparisons (candidate pairs) for every achieved level of precision and recall. The results are presented in Fig. 5 separately for LR and RT predictive models. The $\mathcal{Y}$ axes on the graph shows the total number of candidate pairs that need to be compared after applying an indexing technique described in Section 5. We see that to identify 643 manually annotated matches for references extracted from notary acts within a large collection of civil certificates we analyze more than 54000 candidate pairs. This is much less than comparing each reference from notary

act with every reference from civil certificates but this is still much larger than a number of true positive matches. The applied indexing strategy is not restrictive and generates among the true-positive matches a lot of extra candidate pairs to compare. Considering such a large amount of pairs we have recall value above 92%. Then we use robust classifiers and the extended feature set that leads to promising results in distinguishing *Matched* pairs from a large amount of *Non-Matched* ones.



(a) using regression tree

(b) using logistic regression

Fig. 5 Distributions of the number of potential candidate matches, and corresponding precision/recall values for two applied predictive models

7.3 Experiment 2: ER after Manual Match Review

In this second experiment, we present the increase in precision after the manual cross-check of false positive matches which corresponds to the situation with the maximum *F-score* in Experiment 1. Experts manually review matches from the false positives, generated by the two prediction models (LR and RT). Table 12 presents recalculated precision results for each set of features using the logistic regression. We show the previous recall and optimal *F-score* values from Experiment 1 and compare two corresponding precision values: before and after manual matches review. The table shows that the initial accuracy has been greatly underestimated. After an additional review of matches that are positive according to the classifier, volunteers found that they missed 89 matches during the initial data annotation. To avoid boosting the recall artificially, we do not run a full set of experiments similar to the experiments described in subsection 7.2. The cross-check of the false positive matches affects only the precision. Matches which were incorrectly rejected can not

be identified during the manual review of the FN set. As we see from Table 12, for each set of features the precision is underestimated by 7% on average.

Table 12 The improved precision in the Experiment 2 using the *Logistic Regression*

Features	$maxF_{exp1}$	$Prec_{exp1}$	$Prec_{exp2}$	Δ_{prec}
basic	0.236	0.161	0.218	0.075
basic, name popularity	0.400	0.430	0.498	0.068
basic, name popularity, geo distance	0.402	0.434	0.501	0.067
basic, name popularity, geo distance, co-occur.	0.445	0.338	0.413	0.075

Table 13 presents the results obtained using the Regression Trees. The precision is maximally improved by 14%. The largest improvement corresponds to the extended set of features which includes the basic features and additional features such as name popularity, migration information and reference co-occurrence. We see from the table that for each feature set the precision is increased after the manual review of FP matches. An additional review of the FP matches improves the precision evaluation. Nevertheless, the estimation of the precision value is very important for genealogical and population research. Therefore, we emphasize the precision calculation in this experiment.

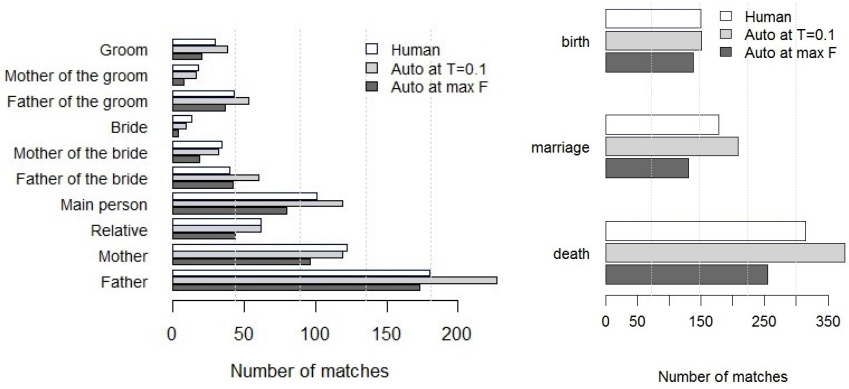
Table 13 The improved precision in the Experiment 2 using the *Regression Tree*

Features	$maxF_{exp1}$	$Prec_{exp1}$	$Prec_{exp2}$	Δ_{prec}
basic	0.231	0.218	0.289	0.071
basic, name popularity	0.374	0.448	0.520	0.072
basic, name popularity, geo distance	0.444	0.679	0.760	0.081
basic, name popularity, geo distance, co-occur.	0.502	0.486	0.626	**0.140**

7.4 Alternative Analysis

Since it is very hard to get the ground truth for our dataset we also run some alternative validations based on common sense. For instance, independently if we know which match is correct or not, when a person is matched to two birth certificates, one of the matches has to be wrong. In this subsection we make an effort to evaluate our results using such common sense arguments. In Fig. 6 we show a detailed comparative analysis of the number of matches identified by humans and by the RT

with the extended feature set (the best studied automatic ER method) with two selected threshold levels of score function at max F-Score and at the threshold level $T = 0.1$. We compare the number of matches for each type of certificate: birth, marriage and death and also for different role of people mentioned there. We see that the maximum number of matches are identified for death certificates by humans as well as by the automatic approach. This can be explained by the fact that the collection of death certificates is the most complete. We also see that for males (fathers or grooms) matches are found more often than for females. One reason for that is that males are mentioned more often than females in legal acts. However the numbers of identified matches by humans and by the automatic approach is relatively similar.



(a) for each person role in certificates

(b) for each certificate type

Fig. 6 The comparison of number of matches according to humans and automatic approaches for two threshold levels of RT score function.

8 Discussion

As can be seen from Section 7, the direct application of standard ER solutions to real-world multi-source genealogical dataset brings good results even though some space for the improvements is left. There are quite some differences between the civil certificates and the notary acts. On the one hand, some information which is available in the certificates, such as names of parents, is not always available in the notary acts. Furthermore, the available information in the notary acts is not fixed at all; depending on the type of the act there might be information about husband-wife relation or other family relations, while in other acts no family relations may have been mentioned. When evaluating the precision and recall of our approach we do

not take into account what information may or may not be presented, but only assess the following criteria for each name that occurs in the notary act:

- Which links to certificates humans find for a name, where the algorithm has not reported them (i.e., recall)
- Which links to certificates humans do not find for a name, where the algorithm has reported (i.e., precision)

Since the labeling was not complete (due to the strenuous nature of this task) we additionally checked the top-links found by the humans in order to get an idea to what extent the accuracy figures were biased by the incomplete labeling.

Non-structural differences such as missing information may cause biases in the evaluation because the task becomes more difficult both for humans and computers. By the nature of our evaluation strategy, however, we try to counter this effect as much as possible.

Another challenge that we deal with is the lack of ground truth which makes it difficult to get reliable and high-quality evaluation. This problem is very common when dealing with real-world data [1, 11].

In Fig. 7, using a Venn diagram, we demonstrate all possible intersections when a match is positive according to the absolute ground truth, the human judgment, and the baseline approach. Each circle in the diagram represents positive matches according to absolute ground truth, human judgment and the baseline approach. The closer human judgment agrees with the absolute ground truth, the more accurate is our evaluation.

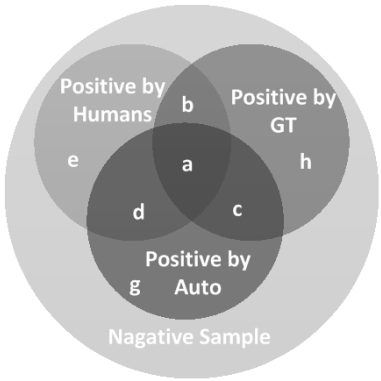


Fig. 7 A diagram of possible intersections between the ground truth, human judgment and the automatic ER approach. GT stands for a ground truth.

In most machine learning approaches there is an implicit assumption that in the test data the absolute ground truth is known. In our diagram this would correspond to the cells labeled e, c, d, and h being empty, and hence the human judgment (green circle; i.e., the labels to which we have access) coincides exactly with the inaccessible ground truth (red circle). Given the nature of our problem, however,

this is not at all true. On the one hand we calculate the perceived precision and recall as:

- perceived precision = $(a+d)/N$
- perceived recall = $(a+d)/(a+b+d+e)$, where $N=(a+c+d+g)$ represents the known number of positives by our classifier,

versus the real precision and recall:

- real precision = $(a+c)/N$
- real recall = $(a+c)/(a+b+c+h)$.

Depending on the size of c, d, e, and h the differences may be significant. Therefore we will now systematically analyze these 4 quantities and see how we can reduce their risk.

On the one hand we can be reasonable certain that the links labeled by humans are correct and hence cells e and d are probably small. On the other hand, however, cells c and h are likely very large given the arduousness of the task of labeling *all* matches. c (number of correct machine matches not found by humans) we control by running all seemingly false positives again by humans as explained in section 7.3. There indeed we detected that there were several matches (7% of the matched found by computer) not found by humans. In this way we could reduce c and hence get accurate numbers for precision. Controlling h, on the other hand is much more difficult, as this concerns true matches not found by humans, nor by the machine. Even though we tried to reduce h as much as possible by reducing the number of notary acts, and requesting the human annotators to find all possible links for this reduced set of certificates, it is inevitable that a large part of true links go by unnoticed. This problem is to a large extent unsolvable and we tried to tackle it by the indirect, common-sense based evaluation in section 7.4.

9 Conclusion

In this chapter we studied the concept of ER in genealogical data research, where the data was provided from sources of different structure. We investigated the application of a number of existing ER techniques. Considering the multi-source characteristic of the data, classical ER techniques are difficult to apply due to the diverse types of data attributes and the lack of sufficient information. We focused our study on the extension of feature sets and on the analysis of the influence of name popularity, migration groups and co-reference information on the overall ER process. We showed that having inferred the name popularity, geographical distance together with the co-reference information helps to significantly improve ER results.

In order to assess the effectiveness of the applied ER approach and also to obtain a training data set, an interactive web-based labeling tool was developed with which the human experts helped to manually identify the matches from an adequate sample of the whole data. The manually labeled matching was used for two purposes:

obtaining training data and computing the evaluation metrics: precision, recall and F-score. We cross-validated the overall ER process. Working with real-world data we had to deal with the lack of ground truth, which makes it difficult to get a reliable and high-quality evaluation. In the second experiment, we showed that experts missed a lot of true positives during the manual data annotation, therefore the precision in our results is underestimated.

The designed ER algorithm has some limitations. One of them is selected indexing strategy to generate candidate pairs. The disjunctions of partial phonetic keys helps us to achieve relatively high recall value, however as a part of our future work we want to exclude the blocking phase completely. One of a potential extensions is implementing a fuzzy name matching by using the *bit vectors* technique [30] or Levenshtein automata [33]. In this case we do not need to apply any data partitioning.

Another extension of the applied approach is to use more information from notary acts. It requires more advanced text processing techniques. For instance, inheritance notary acts contain information about many family relationships (parents, siblings, nephews, etc.) which should be taken into account during the ER.

We also work on more advanced ER techniques and want to improve the ER process by applying collective relational entity resolution [17] where co-reference information is not processed as an additional attribute. Instead we want to apply more advanced graph-based techniques, taking into account that there may be multiple persons in the different acts and certificates that are co-referenced.

Another improvement concerns the applied predictive models. Instead of using logistic regression or regression trees that were proposed in the chapter we want to use Probabilistic Relational techniques, thus applying Probabilist Graphical Models to solve ER problem can be an appropriate next step.

In short, we proposed an ER approach which was capable of extracting various entities from multiple sources of information, some structured and some unstructured; the efficiency of the proposed approach was first improved by means of the labeled data provided by human experts, and afterward was evaluated in detail by human experts. The thorough evaluations of the work, showed good precision and recall, which is sufficient for some prosopographical and demographical researches, yet allows for various extensions. Thus, there is a potential for future work.

Acknowledgements

The authors are grateful to the BHIC Center for the support in data gathering, data analysis and direction. In particular, we would like to thank Rien Wols and Anton Schuttelaars whose efforts were instrumental to this research and their patience and support appeared infinite. This research has been carried under Mining Social Structures from Genealogical Data (project no. 640.005.003) project, part of the CATCH program funded by the Netherlands Organization for Scientific Research (NWO).

References

1. Alsaleh, M., van Oorschot, P.C.: Evaluation in the absence of absolute ground truth: toward reliable evaluation methodology for scan detectors. Int. J. Inf. Sec. **12**(2), 97–110 (2013)
2. Bhattacharya, I., Getoor, L.: Iterative record linkage for cleaning and integration. In: Proceedings of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD '04, pp. 11–18. ACM, USA (2004)
3. Bhattacharya, I., Getoor, L.: Collective entity resolution in relational data. ACM Trans. Knowl. Discov. Data **1**(1) (2007)
4. Bilenko, M.: Adaptive blocking: Learning to scale up record linkage. In: In Proceedings of the 6th IEEE International Conference on Data Mining (ICDM-2006, pp. 87–96 (2006)
5. Van den Bosch, A., Busser, B., Canisius, S., Daelemans, W.: An efficient memory-based morphosyntactic tagger and parser for Dutch. In: Computational Linguistics in the Netherlands: Selected Papers from the Seventeenth CLIN Meeting, pp. 99–114 (2007)
6. Chowdhury, G.G.: Natural language processing. Annual Review of Information Science and Technology **37**(1), 51–89 (2003)
7. Christen, P.: A comparison of personal name matching: techniques and practical issues. In: Proceedings of the Workshop on Mining Complex Data (MCD06), held at IEEE ICDM06, pp. 290–294 (2006)
8. Christen, P.: Automatic record linkage using seeded nearest neighbour and support vector machine classification. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '08, pp. 151–159. ACM, USA (2008)
9. Christen, P.: Data Matching. Springer Publishing Company, Incorporated (2012)
10. Cohen, W.W., Kautz, H.A., McAllester, D.A.: Hardening soft information sources. In: R. Ramakrishnan, S.J. Stolfo, R.J. Bayardo, I. Parsa (eds.) KDD, pp. 255–259. ACM (2000)
11. Efremova, J., Montes García, A., Calders, T.: Classification of historical notary acts with noisy labels. In: In Proceedings of the 37th European Conference on Information Retrieval, ECIR'15. Springer, Vienna, Austria (2015)
12. Efremova, J., Ranjbar-Sahraei, B., Calders, T.: A hybrid disambiguation measure for inaccurate cultural heritage data. In: the 8th Workshop on LaTeCH, pp. 47–55 (2014)
13. Efremova, J., Ranjbar-Sahraei, B., Oliehoek, F.A., Calders, T., Tuyls, K.: An interactive, web-based tool for genealogical entity resolution. In: 25th Benelux Conference on Artificial Intelligence(BNAIC'13). The Netherlands (2013)
14. Efremova, J., Ranjbar-Sahraei, B., Oliehoek, F.A., Calders, T., Tuyls, K.: A baseline method for genealogical entity resolution. In: Proceedings of the Workshop on Population Reconstruction, organized in the framework of the LINKS project (2014)
15. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate record detection: A survey. Knowledge and Data Engineering, IEEE Transactions on **19**(1), 1–16 (2007)
16. Florian, R., Ittycheriah, A., Jing, H., Zhang, T.: Named entity recognition through classifier combination. In: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4, CONLL '03, pp. 168–171. Association for Computational Linguistics, USA (2003)
17. Getoor, L., Machanavajjhala, A.: Entity resolution: Theory, practice & open challenges. In: International Conference on Very Large Data Bases (2012)
18. Getoor, L., Machanavajjhala, A.: Entity resolution for big data. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1527–1527. ACM (2013)
19. Hernández, M.A., Stolfo, S.J.: The merge/purge problem for large databases. SIGMOD Rec. **24**(2), 127–138 (1995)
20. Huijsmans, D.: Dataset historische Nederlandse toponiemen spatio-temporeel 1812-2012. In: IISG-LINKS (2013)
21. Ivie, S., Henry, G., Gatrell, H., Giraud-Carrier, C.: A metricbased machine learning approach to genealogical record linkage. In: In Proceedings of the 7th Annual Workshop on Technology for Family History and Genealogical Research (2007)

22. Lawson, J.S.: Record linkage techniques for improving online genealogical research using census index records. In: Proceeding of the Section on Survey Research Methods (2006)
23. McCallum, A., Nigam, K., Ungar, L.H.: Efficient clustering of high-dimensional data sets with application to reference matching. In: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 169–178. ACM (2000)
24. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Linguisticae Investigationes **30**(1), 3–26 (2007). Publisher: John Benjamins Publishing Company
25. Naumann, F., Herschel, M.: An Introduction to Duplicate Detection. Morgan and Claypool Publishers (2010)
26. Nuanmeesri, S., Baitiang, C.: Genealogical information searching system. In: Management of Innovation and Technology, 2008. ICMIT 2008. 4th IEEE International Conference on, pp. 1255–1259. IEEE (2008)
27. Rahmani, H., Ranjbar-Sahraei, B., Weiss, G., Tuyls, K.: Contextual entity resolution approach for genealogical data. In: Workshop on Knowledge Discovery, Data Mining and Machine Learning (2014)
28. Ramachandran, S., Deshpande, O., Roseman, C.C., Rosenberg, N.A., Feldman, M.W., Cavalli-Sforza, L.L.: Support from the relationship of genetic and geographic distance in human populations for a serial founder effect originating in Africa. Proceedings of the National Academy of Sciences of the United States of America **102**(44), 15,942–15,947 (2005)
29. Sammut Claude; Webb, G.I.: Encyclopedia of Machine Learning. Springer, Berlin Heidelberg (2010)
30. Schraagen, M.: Complete coverage for approximate string matching in record linkage using Bit vectors. In: ICTAI'11, pp. 740–747 (2011)
31. Schraagen, M., Hoogeboom, H.J.: Predicting record linkage potential in a family reconstruction graph. In: 23th Benelux Conference on Artificial Intelligence(BNAIC'11). Belgium (2011)
32. Schraagen, M., Kosters, W.: Record linkage using graph consistency. In: Machine Learning and Data Mining in Pattern Recognition, Lecture Notes in Computer Science, pp. 471–483. Springer International Publishing (2014)
33. Schulz, K.U., Mihov, S.: Fast string correction with levenshtein automata. IJDAR **5**(1), 67–85 (2002)
34. Singla, P., Domingos, P.: Entity resolution with markov logic. In: Proceedings of the Sixth International Conference on Data Mining, ICDM '06, pp. 572–582. IEEE Computer Society, USA (2006)
35. Sweet, C., Özyer, T., Alhajj, R.: Enhanced graph based genealogical record linkage. In: Proceedings of the 3rd international conference on Advanced Data Mining and Applications, ADMA '07, pp. 476–487. Springer-Verlag, Berlin, Heidelberg (2007)
36. Štajner, T., Mladenić, D.: Entity Resolution in Texts Using Statistical Learning and Ontologies. In: Proceedings of the 4th Asian Conference on The Semantic Web, ASWC '09, pp. 91–104. Springer-Verlag, Berlin, Heidelberg (2009)
37. Winkler, W.E.: Matching and record linkage. In: Business Survey Methods, pp. 355–384. Wiley (1995)