Intelligent Data Analysis 20 (2016) 455–475
DOI 10.3233/IDA-160814
IOS Press

Entity resolution in disjoint graphs: An application on genealogical data

Hossein Rahmani[a,b,*], Bijan Ranjbar-Sahraei[b], Gerhard Weiss[b] and Karl Tuyls[c]
[a]*School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran*
[b]*Maastricht University, Maastricht, MD, The Netherlands*
[c]*University of Liverpool, Liverpool, UK*

Abstract. Entity Resolution (ER) is the process of identifying references referring to the same entity from one or more data sources. In the ER process, most existing approaches exploit the content information of references, categorized as content-based ER, or additionally consider linkage information among references, categorized as context-based ER. However, in new applications of ER, such as in the genealogical domain, the very limited linkage information among references results in a disjoint graph in which the existing content-/context-based ER techniques have very limited applicability. Therefore, in this paper we propose first, to use the homophily principle for augmentation of the original input graph by connecting the potential similar references, and second, to use a Random Walk based approach to consider contextual information available for each reference in the augmented graph. We evaluate the proposed method by applying it to a large genealogical dataset and we succeed to predict 420,000 reference matches with precision 92% and discover six novel and informative patterns among them which can not be detected in the original disjoint graph.

Keywords: Entity resolution, disjoint graphs, genealogy

1. Introduction

In many real-world applications of data science, the input data contains substantial errors and inconsistencies which make the process of identifying multiple references referring to the same entity as the key requirement in any integration task. In literature, this requirement is introduced in different ways such as Record Linkage [1–6], the Merge/Purge problem [7], Duplicate Detection [8–10], Hardening Soft Databases [11], Reference Matching [12], and Entity Resolution [13–20].

The key idea behind almost all existing ER techniques is to propose a similarity measure among references and use this measure to label the most similar pairs as matches. We classify the previous methods into two categories: 1) Content-based ER and 2) Context-based ER. Content-based methods consider only the content of each reference, for instance by converting it into numerical/string vectors and using the standard similarity measures such as Jaro-Winkler, Levenshtein, TF-IDF, Cosine-Similarity, and Jaccard Coefficient [21–23] to calculate the similarity among reference pairs. Finally, the most similar pairs are predicted as matches.

The main problem of content-based methods is that they neglect the contextual information available for each reference. For example, consider a genealogical dataset as an input of the ER process. The

*Corresponding author: Hossein Rahmani, Maastricht University, PO Box 616, Maastricht 6200 MD, The Netherlands. E-mail: h.rahmani@maastrichtuniversity.nl.

FIRSTNAME and LASTNAME of each reference can form the content information for that reference. This content information is used to measure the similarity between reference pairs. However, one can imagine that by using the contextual information such as similar family members in the genealogical certificates, the accuracy in ER increases. The main goal of context-based methods is to use this kind of additional contextual information to improve the content-based measure.

The prediction results of context-based methods depend strongly on the available contextual information, mostly in the form of linkage information among references, available for each reference. As one possible way of measuring amount of available linkage information in a graph we take the number of connected components into consideration. We believe that there is an inverse relationship between the amount of linkage information and the number of connected components in the graph; this inverse relationship will be described formally in Section 3. As the number of connected components increases, the graph becomes more disjoint and the linkage information to be used for the ER process decreases.

In domains such as the genealogical domain with a disjoint graph consisting of large number of connected components, there is not much contextual information available for each reference such that the most complex context-based methods produce the same matching results as traditional content-based techniques. In this domain, each historical certificate is modeled as a connected component in which each node represents a reference and each edge shows a family relationship between two connected references. There is no linkage information available for any two references belonging to different certificates. In such a domain with a small connectivity ratio (large number of connected components) among references, applying context-based methods will not improve the content-based methods significantly.

In this paper, we improve the context-based ER technique for domains which are initially modeled as a disjoint graph. First we augment the original graph by adding potential links among the references. Second, we apply the Random Walk approach, which has been applied successfully to varied range of domains [24–27], to consider the contextual information available for each reference in the augmented graph. Third, we propose a hybrid similarity measure which considers both content and context information in predicting the matches among references. Fourth, we evaluate the proposed approach from different perspectives and finally, according to the provided empirical studies, we conclude that our hybrid similarity measure outperforms the existing content and context-based similarity measures in disjoint graphs.

The rest of this paper is organized as follows: Section 2 discusses the related work in ER. In Section 3, we formally define the concept of "connected component" and its relation with graph connectivity. Section 4 describes the input data from genealogical domain. In Section 5, we discuss our proposed approach for augmenting the original disjoint graphs. Section 6 considers both content and context information for calculating a hybrid similarity among the candidate reference pairs. In Section 7, we evaluate the proposed method and discuss the discovered patterns. Section 8 groups the predicted matched references into one entity and builds the entity graph from the original reference graph. Section 9 concludes.

2. Related work

The Entity Resolution (ER) problem is the first step in any application which is involved in cleaning and integrating the references extracted from different resources. In literature, the process of identifying multiple references referring to same entity has been addressed in different ways such as Record Linkage [1–6], the Merge/Purge problem [7], Duplicate Detection [8–10,28], Hardening Soft Databases [11], Reference Matching [12] and Entity Resolution [13–20].

We classify the previous studies on ER into two main categories: 1) Content-based ER and 2) Context-based ER. Content-based approaches first convert the available information of each reference into string/numerical vectors and then, they apply standard similarity measures such as Jaro Winkler, Levenshtein, TF-IDF, Cosine-Similarity, Jaccard Coefficient [22,23,29] to calculate the similarity among reference pairs. Finally, the most similar pairs are introduced as matches. Content information is mostly modeled as a string type. So, defining the proper similarity measure for string values has been addressed a lot in the ER literature. Bases on the availability of labeled training data the existing approaches are classified into Unsupervised [30–33] and Supervised [34–37] approaches. The difficulties related to construction of proper training set with enough number of positive and negative cases introduces the new approaches which use active learning to classify ambiguous cases by the learner [37,38]. The number of proposed methods to calculate the similarity among the numeric data types are primitive comparing to string similarity measures. In the simplest way, the numbers are treated as strings (then, we could apply the discussed similarity measures) or simple range queries, which locate numbers within pre-defined boundaries. Koudas et al. [39] proposed as a future direction to consider the distribution and type of the numeric data in the similarity calculation process. Some studies compare the performance of different similarity measures on varied number of datasets [40,41]. The general conclusion is that no single similarity measure is suitable for all data sets [41]. Even measures that demonstrate robust and high performance for some data sets can perform poorly on others.

In addition to measuring attribute similarity, Context-based approaches consider also the contextual information, mostly in the form of linkage information, available for each reference. For example, two partially similar references r_i and r_j should be more likely to match if they both link to third reference r_k. Ananthakrishna et al. [10] propose a similarity metric that considers the "co-occurrence" similarity of two entries in a database in addition to their textual similarity. Singla and Domingos [42] improve duplicate record detection by propagating information from one candidate match to another via the attributes they have in common. Dong et al. [43] resolve entities of multiple types by propagating evidence in a dependency graph. Many recent methods consider relational structure to provide a relatively computationally efficient approach for the ER problem [4,10,17,44–46].

3. Connected components

An undirected graph $G = (V, E)$ is said to be connected if there is a path between any pair of vertices (u, v), such that $u \in V$ and $v \in V$. A connected component, $G_c = (V_c, E_c)$, is a connected subgraph of a graph G. A graph $G = (V, E)$ is said to have multiple connected components $G_1, G_2, \ldots, G_n$ (where $G_1 = (V_1, E_1), G_2 = (V_2, E_2), \ldots, G_n = (V_n, E_n), V = V_1 \cup V_2 \cup V_3 \cdots \cup V_n$, and $E = E_1 \cup E_2 \cup E_3 \cdots \cup E_n$) if, for any pair of such components, say $G_i = (V_i, E_i)$ and $G_j = (V_j, E_j)$, for any node pair (u, v) such that $u \in V_i$ and $v \in V_j$ there is no path from u to v, or vice versa. We believe that a graph becomes more disjoint as the number of connected components increases. We define the connectivity ratio of graph $G(V, E)$ with N_c connected components according to Formula (1).

$$\text{CONNECTIVITY\_RATIO}(G) = \frac{V - N_c}{V - 1} \tag{1}$$

In two extreme cases where $N_c = V$ and $N_c = 1$, Formula (1) returns 0 and 1 as the connectivity ratio of graph G, respectively. In the former case, there is no edge in the graph ($E = \{\}$) and each isolated node represents a connected component. In the latter case, there is only one connected component which contains all the nodes.

Table 1
Considered features for each certificate type

Birth certificate	FIRSTNAME, LASTNAME, GENDER, BIRTHDATE, BIRTHPLACE, FATHERFIRSTNAME, FATHERLASTNAME, MOTHERFIRSTNAME, MOTHERLASTNAME
Death certificate	FIRSTNAME, LASTNAME, GENDER, BIRTHDATE, BIRTHPLACE, DEATHDATE, DEATHPLACE, FATHERFIRSTNAME, FATHERLASTNAME, MOTHERFIRSTNAME, MOTHERLASTNAME
Marriage certificate	GROOMFIRSTNAME, GROOMLASTNAME, GROOMAGE, BRIDEFIRSTNAME, BRIDELASTNAME, BRIDEAGE, GROOMFATHERFIRSTNAME, GROOMFATHERLASTNAME, GROOMMOTHERFIRSTNAME, GROOMMOTHERLASTNAME, BRIDEFATHERFIRSTNAME, BRIDEFATHERLASTNAME, BRIDEMOTHERFIRSTNAME, BRIDEMOTHERLASTNAME

Table 2
Statistical information of input data

Number of birth certificate	110,000
Number of marriage certificate	350,000
Number of death certificate	710,000
Number of extracted references	5,300,000

Neither $N_c = V$ nor $N_c = 1$ are possible in our historical datasets. However, $N_c = 1$ is definitely possible in other domains. In this case, we only have one connected component which contains all the nodes. As an example, consider Protein-Protein Interaction (PPI) Networks in which there is a biological path between any two protein pairs. $N_c = V$ means that there is no linkage information among the individuals and accordingly our method cannot benefit from the network information (Refer to Sections 4 and 5 for details).

4. Input data

The genealogical dataset used in this paper, consists of three certificate types, namely "Birth", "Death" and "Marriage" certificates, as input to the proposed ER method. Table 1 lists the content features for each certificate type. As shown in Table 1, Birth certificates include three individual references (i.e., child, father and mother). The Death certificates include four individual references (i.e., deceased, father, mother and relative of deceased). Finally, the Marriage certificates include six references (i.e., groom, bride and parents of each).

We use the Relational database model [47] to integrate and maintain the three discussed certificate types considering Entity, Referential and Domain integrity constraints [48]. We choose the Relational database model since this model is widely used, easy to apply and is highly maintainable. Elmasri et al. [48] discuss in details the Relational database model and its advantages over other databases models. The final integrated database consists of 5,300,000 individual references extracted from 1,170,000 certificates (details are provided in Table 2). Considering the non-mentioned parents or relatives in some certificates, 500,000 references do not have any name (i.e., first_name = *null* and last_name = *null*). Therefore, we have 4,800,000 informative references. Among these references we have 170,000 distinct first names and 100,000 distinct last names. The dates mentioned in different certificates span a period of time between 1810 and 1920. The certificates are registered in 200 different municipalities.

5. Augmenting the original disjoint graph

We model a genealogical dataset as an undirected annotated graph $G(R, E, \lambda)$ where R is a set of references, $E \subseteq R \times R$ is a set of family relationships among these references, and λ is so-called

annotation function; for each reference r, λ denotes the additional information we have about r. We also call this additional information $\lambda(r)$ the *content* of reference r.

The original graph representation of genealogical certificates suffers from small amount of linkage information available for each reference. In this section, we augment the original graph representation using the homophily principle in graph [49]. Homophily refers to the fact that in a graph, edges are more likely among similar nodes than between dissimilar nodes. The most direct way of finding similar references in the original graph is to apply pairwise comparison among the references. The computational order of this process is $O(n^2)$ which makes it infeasible in our project with roughly 5,000,000 references.

In order to avoid having to compare all pairs of references, we use a technique known as Blocking in the literature [50–53] to split all references into different blocking partitions. This process reduces the search space and diminishes the number of potential candidate pairs. In this paper, we deal with Dutch reference names and accordingly, the proposed blocking method should be capable of resolving the common errors happen in Dutch names. Details of the implemented blocking method is discussed in the next Section.

5.1. Proposed blocking method for dutch names

There are different writing variations for each (Dutch) name. For example, "Ghendrik", "Haendrik", "Handrikus", "Hanri" and "Hedrik" are all referring to the same entity "Hendrik". The reason behind this name variations could be of typing error or some historical/geoghraphical issues. As a blocking strategy, the previous work have used the standard string encoding systems such as Soundex [54], metaphone [55] and double-metaphone [56]. Soundex, indexes the names based on their pronunciation in English. The main goal in this algorithm is that the letter with similar pronunciations be encoded with same characters so that spelling errors can be resolved. Metaphone is an extension of the Soundex code. Compared to Soundex, this code takes into account more information about variations and inconsistencies in English spelling and pronunciation. Afterwards, Double Metaphone was proposed which takes into account spelling peculiarities of a number of other languages. This indexing algorithm generates up to two codes for each word, that can improve some of the limitations of the original Metaphone for dealing with foreign languages.

Rahmani et al. [57] used the dataset of Meertens Institute [58] to find the relationships among different variations of Dutch names with their standard format. In total, the Meertens database contains $44,000$ distinct first names (18,000 and 26,000 for male and females, respectively) and 120,000 distinct last names. The main attributes of the dataset are *name* and *standard name*. They extracted 12 features F1–F12 from the Meertens dataset:

F1: If first 2 letters of name and standard name are equal.
F2: If first 3 letters of name and standard name are equal.
F3: If last 2 letters of name and standard name are equal.
F4: If last 3 letters of name and standard name are equal.
F5: If size of name and standard name are equal.
F6: Absolute difference of name length and standard length.
F7: Number of longest first equal chars.
F8: Number of longest last equal chars.
F9: If *soundex* code of name and standard name is equal.
F10: If *metaphone* code of name and standard name is equal.

Table 3
Feature analysis of dutch names. Features F6, F7, F8 and F12 are continuous features and the rest of features are all binary

	First name (male)				First name (female)				Last name			
Feature	Min	Mean	s.t.d.	Max	Min	Mean	s.t.d.	Max	Min	Mean	s.t.d.	Max
F1	0	0.71	0.46	1	0	0.70	0.46	1	0	0.79	0.40	1
F2	0	0.52	0.49	1	0	0.50	0.49	1	0	0.60	0.48	1
F3	0	0.36	0.49	1	0	0.42	0.49	1	0	0.54	0.50	1
F4	0	0.27	0.44	1	0	0.30	0.45	1	0	0.45	0.49	1
F5	0	0.35	0.48	1	0	0.34	0.48	1	0	0.43	0.5	1
F6	0	1.15	1.31	15	0	1.10	1.31	13	0	0.77	0.88	10
F7	0	2.90	2.07	13	0	2.90	2.06	11	0	3.57	2.41	16
F8	0	1.57	2.07	13	0	1.77	2.06	11	0	2.59	2.65	16
F9	0	0.50	0.5	1	0	0.47	0.5	1	0	0.58	0.49	1
F10	0	0.31	0.47	1	0	0.29	0.46	1	0	0.42	0.49	1
F11	0	0.39	0.49	1	0	0.37	0.49	1	0	0.49	0.49	1
F12	0	3.90	1.85	14	0	3.98	1.85	12	0	4.82	2.1	17

F11: If *double-metaphone* code of name and standard name is equal.
F12: Longest common chars between name and its standard name.

Table 3 provides detailed information about these 12 very basic and important features of Dutch names. Among all the features, F1 is a very discriminative feature as it is true in more than 70% of the cases (i.e., the first two letters of a name and its standard name are equal in more than 70% of the cases). Among the phonetic-based string similarity measures (F9, F10 and F11) Soundex code has the highest score of being identical between name and its standard form in about 50% of the cases. However, the absolute difference of name length and its standard form length F6 has a maximum of 15, which means some name lengths can deviate very much from length of its standard form.

Considering the informative features for Dutch names, Rahmani et al. [57] proposed a novel blocking key (Formula (2)) which considers both name variations and spelling errors for each reference r_i.

$$\begin{aligned}\text{BLOCKING\_KEY}(r_i) = {} & \text{GENDER}(r_i) + \text{FIRSTNAME}(r_i)[:3] + \text{FIRSTNAME}(r_i)[-2:] \\ & + \text{LASTNAME}(r_i)[:3] + \text{LASTNAME}(r_i)[-2:] \\ & + \text{soundex}(\text{FIRSTNAME}(r_i)) + \text{soundex}(\text{LASTNAME}(r_i)) \end{aligned} \tag{2}$$

where in Formula (2), '+' denotes the concatenation operation on strings, and STRING$[:i]$ and STRING$[-i:]$ refer to the first and last i characters of the STRING, respectively. Formula (2) is, intuitively, in line with our assumption that in the data entry phase, people are more careful about the first i and last j characters and accordingly, the typing errors mostly happen in in middle characters c ($i < c < j$). According to this blocking method, two references with a spelling mistake on the second character of the name would not be grouped in a similar block while a mistake on the middle characters would not be problematic.

Finally, for each reference r in the dataset Formula (2) is used to generate a blocking-key; then all the references with similar blocking key are assumed to be in the same block. This process builds blocks with different sizes (= member count).

5.2. Graph augmentation using blocking

After having a set of potential similar references, now we could augment the original graph by adding the linkage information to the graph. For each high confidence block b_i with $|b_i|$ members, we add a

new node b_i to the original graph representing the block b_i and $|b_i|$ edges connecting all b_i's reference members to node b_i. As the size of one block increases, the confidence for that block decreases. In this paper, we consider all the blocks b_i with size less than or equal to θ ($|b_i| \leqslant \theta$) as high confidence blocks. As a result, there is at least a path with length 2 among all the references belonging to similar high confidence blocks. The augmented graph $G'(R', E', \lambda)$ has $R' = R + |\cup_{|bi|\leqslant\theta} b_i|$ nodes in which R nodes representing the original references and $|\cup_{|bi|\leqslant\theta} b_i|$ nodes representing the union of all high confidence block nodes added to the original graph. The number of edges in the augmented graph equals to number of original family relationships among references in addition to number of edges connecting block nodes to their reference members ($E' = E + \sum_{|b_i|<\theta} |b_i|$). In the next section, we discuss our hybrid similarity measure to gain knowledge from the augmented graph.

6. Hybrid similarity measure for ER

In Section 5, we resolved the limitations of context-based methods on disjoint graphs which have low linkage information among references, by adding new nodes and edges to the original graph. As a result there is more linkage information available in the augmented graph. In this section, we propose a novel ER approach which gets the augmented version of original graph as input and discovers the set of references referring to the same entity as output. For this purpose, our approach should compare all the reference pairs belonging to the same block (co-block references). To calculate the similarity among the co-block reference pairs, we propose the following hybrid similarity measure to consider both content and context information available for each reference.

$$\text{SIM}_{\text{hybrid}}(r_i, r_j) = \alpha * \text{SIM}_{\text{content}}(r_i, r_j) + \beta * \text{SIM}_{\text{context}}(r_i, r_j) \tag{3}$$

In Formula (3), $\text{SIM}_{\text{content}}(r_i, r_j)$ and $\text{SIM}_{\text{context}}(r_i, r_j)$ calculate the similarity between two references r_i and r_j using the content and context information, respectively. α and β represent the weight of content and context similarity scores, respectively and could be determined according to domain experts feedback or learned by machine learning methods. In this paper, our aim is to propose a general similarity measure even for cases where we do not have any access to domain experts feedback. Therefore, we assume $\alpha = \beta = \frac{1}{2}$. This parameters assignment aligns the values of hybrid similarity measure between 0 and 1.

The content-based similarity score can be computed using Formula (4), in which we use Jaro-Winkler algorithm [21] to calculate the string similarity between FIRSTNAME and LASTNAME of two references r_i and r_j.

$$\begin{aligned}\text{SIM}_{\text{content}}(r_i, r_j) = \frac{1}{2}\Big[&\text{JaroWinkler}(\text{FIRSTNAME}(r_i), \text{FIRSTNAME}(r_j))\Big] \\ &+\frac{1}{2}\Big[\text{JaroWinkler}(\text{LASTNAME}(r_i), \text{LASTNAME}(r_j))\Big]\end{aligned} \tag{4}$$

The context-based similarity measure has a rather more complicated way to be computed. As discussed in Section 4, the references that are used in this paper are extracted from three different type of certificates: "Birth", "Death" and "Marriage". Each of these certificate types contain the information of a group of references which should be considered when we compare two references. For example, imagine two arbitrary references r_i and r_j where both belong to the same block b_k (i.e., $r_i \in b_k$ and $r_j \in b_k$). If their partners both belong to another block b_L (i.e., $\text{partner}(r_i) \in b_L$ and $\text{partner}(r_j) \in b_L$)

then the probability that r_i and r_j referring to the same entity should be increased, comparing to the case that their partners belong to different blocks.

In order to measure the contextual similarity between two references r_i and r_j, first, we exclude the node b_k and all its connected edges from the augmented graph. The reason behind this node exclusion is that this part of graph represents the content similarity of node r_i and all its co-block references; we do not consider the content information twice. Second, we use the steady state distribution of a Random Walk with Restarts (RWR) technique [59] to calculate the graph similarity between two references r_i and r_j. We simulate the trajectory of random walker that starts from r_i and moves to its neighbors with the uniform probability. We keep the random walker close to the original node r_i by allowing transition to the original node with probability c as the restart probability. Formally, the RWR technique can be represented by following formula

$$\mathbf{x}_{k+1} \leftarrow (1-c)\mathbf{A}\mathbf{x}_k + c\mathbf{x}_0 \tag{5}$$

where $\mathbf{x}_k$ denotes the proximity vector at iteration t (i.e., a vector which contains the probability of reaching each node from r_i in k steps in the corresponding element). Therefore, $\mathbf{x}_0$ is a vector with all elements being zero except the i^{th} element which is one, and $\mathbf{A}$ is the adjacency matrix. This formula is used iteratively to generate the steady state RWR proximity vector (For more details refer to Can et al. [59]).

7. Empirical studies

In this section, first, we apply the blocking strategy discussed in Section 5.1 to the genealogical dataset and we use the generated blocking keys to augment the original graph with informative contextual information. Second, we compare the three content, context and hybrid similarity measures for all compared pairs to show how discriminative the hybrid measure can be in predicting the matched references. Third, we use manual evaluation to measure the precision of predicted links. Fourth, the major patterns discovered from the predicted matches are discussed and a detailed example of predicted matches is elaborated. Finally, the process of population reconstruction is explored by using the predicted matches, which shows the high potentials of this work for future research.

7.1. Results of proposed blocking technique

In this work, the *original graph* of genealogical data consists of 5,270,000 *reference* nodes, each referring to an individual reference from a Birth, Marriage or Death certificate; in total providing 5,270,000 nodes. In the original graph, nodes are linked together based on family relations of *father-child*, *mother-child* and *husband-wife* (i.e., 3 links for a Birth certificate, 4 links for a Death certificate and 7 links for a marriage certificate). In total this graph has 1,170,000 isolated connected components of size 3, 4 or 6.

We applied the blocking strategy proposed by Rahmani et al. [57] to our dataset. As a result, 690,000 blocks are constructed with different sizes ranging from size 1 to 3,845, where each of the blocks of size 1 contains just one reference and the block with largest size contains 3,845 references; the block key of this largest block is "female_Mar_Jan_ia_en_M600_J525" which turns out to be the most frequent pattern among Dutch references reported between 1890 and 1920. The average block size is 7 and the standard deviation is 29. This shows that not many blocks of very large size exist. Figure 1 shows the block size distribution by focusing on the blocks with size 2 to 50.

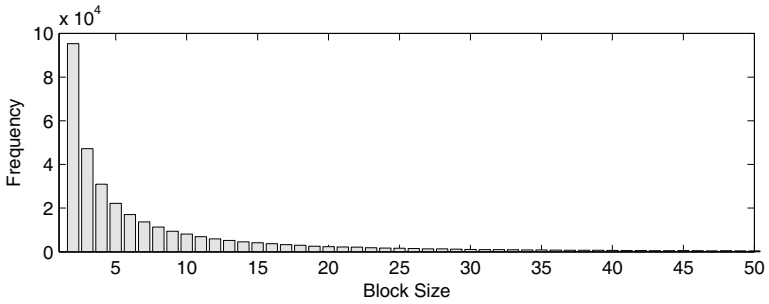

Fig. 1. Distribution of block sizes. The average block size is 7 and the standard deviation is 29. Blocks with size 1 are excluded from the figure as they contain just one reference and are not informative in matching.

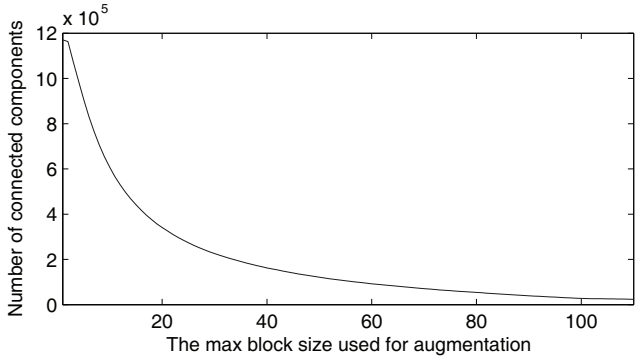

Fig. 2. Comparison of different block sizes chosen for augmentation of the graph. The initial graph has 1,170,000 connected components with maximum size of 6 nodes, and the augmented graph by choosing the blocks of size less than 100 results in 27,000 connected components with maximum size of 5,590,000 nodes.

Given that originally about 25×10^{12} pairwise comparisons (i.e., 5,000,000 $\times$ 5,000,000) were required for accomplishing the task of traditional ER, based on the partitions introduced by the proposed blocking strategy, the average search space is now reduced by 3.5×10^{-5} (i.e., $\frac{690{,}000 \times 7^2}{25 \times 10^{12}}$).

The average number of needed comparisons (690,000 $\times$ 7^2) is still high. Therefore, we decided to focus on high confidence blocks. We assume that when the size of a block increases, its confidence value decreases. Previous studies have followed this assumption by starting their resolution process with high confidence (less frequent) reference pairs [60]. Figure 2 shows that as the size of considered blocks increases the number of connected components decreases and accordingly the graph connectivity increases. Section 5.2 discusses the detailed description of augmenting the original graph using the blocks. In Fig. 2, as size of blocks increases, there is no major change in number of connected components after the block size 100. Therefore, focusing on blocks with maximum size 100 not only reduces the search space significantly but also augments the graph connectivity considerably. In the following sections, we consider blocks with $size \leqslant 100$ as high confidence blocks.

7.2. Augmenting the original graph

In order to establish connections between the immense number of isolated connected components of the original graph, we introduce the Augmented Graph by addition of 680,000 new block nodes

Table 4
The structural properties of the *Original Graph* and *Augmented Graph* of genealogical data. The number of nodes and links in the biggest component has immensely changed from former graph to the latter one

Features	Original graph	Augmented graph
Number of nodes	5,270,000	5,850,000
Number of links	5,620,000	9,070,000
Number of connected components	1,170,000	26,000
Number of nodes in biggest component	6	5,600,000
Number of links in biggest component	7	8,700,000
Connectivity ratio (Formula (1))	0.778	0.995

corresponding to the blocking keys with block size less than $\theta = 100$ (100 is chosen to assure acceptable confidence in block, and provide sufficient increase in connectivity of the graph as discussed in previous subsection). Each block node is connected to the reference nodes that share the same blocking key. This results in addition of 3,450,000 new links. Therefore, in total the augmented graph has 5,850,000 nodes and 9,070,000 links. The augmented graph has a very huge connected component consisting of 5,590,000 nodes and 8,700,000 links (i.e., 96% of the nodes in augmented graph). In addition to this huge component, the augmented graph consists of 26,300 very small connected components of average size 6.8. Table 4 compares the structural properties of the original and augmented graphs.

7.3. Applying the random walk approach

Using the described RWR technique in Section 6 and the steady-state RWR proximity computation introduced in Formula (5), the following steps are carried out for extracting the appropriate matches for a candidate reference r_i.

1. Node r_i in the augmented graph is considered as an starting point for RWR.
2. Graph G_B is constructed using the local neighborhood of r_i considering nodes with maximum distance of 5 from r_i.
3. If reference $r_i \in b_i$, then block node b_i connected to r_i is removed from the G_B (Section 6 discusses the reason behind this node removal).
4. The proximity vector $\mathbf{x}_k$ is generated for G_B using Formula (5), $c = 0.3$, and k is chosen large enough such that the vector converges.
5. The l non-zero elements of $\mathbf{x}_k$ corresponding to reference nodes

$$v_{E_{R1}}, v_{E_{R2}}, \ldots, v_{E_{Rl}}$$

which have the same blocking key as r_i are extracted as $E_{R1}, E_{R2}, \ldots, E_{Rl}$.
6. The normalized proximity of nodes $v_{E_{R1}}, v_{E_{R2}}, \ldots, v_{E_{Rl}}$ are reported as

$$\text{SIM}_{\text{context}}(r_i, v_{E_{R1}}), \text{SIM}_{\text{context}}(r_i, v_{E_{R2}}), \ldots, \text{SIM}_{\text{context}}(r_i, v_{E_{Rk}}),$$

respectively.

It should be mentioned that for each reference individual r_i, the RWR is restricted to its neighborhood of distance 5. This is done to decrease the computational complexity of the approach, while existence of a real match for a reference of a distance further than 5 in the augmented graph is not expected (for any two matched references, at least one family member of each reference should share a blocking key).

Table 5
A comparison between 3 main similarity measures on 42,300,000 compared pairs. "#Zero Scores" and "#One Scores" counts the number of the cases with 0 or 1 for a specific similarity score.

Sim. score	Min	Max	Avg	#Zeros	#Ones
Content-based	0.52	1.0	0.946	0	27,583,526 (65%)
Context-based	0.0	1.0	0.471	41,880,000 (99%)	242,202 (0.61%)
Hybrid measure	0.26	1.0	0.003	0	218,570 (0.51%)

Table 6
Required sample size, given a finite population [62]

Population size	Sample size
50	44
100	80
500	217
1,000	278
1,500	306
3,000	341
5,000	357
10,000	375
50,000	381
100,000	384
254,000	384
10,000,000	384

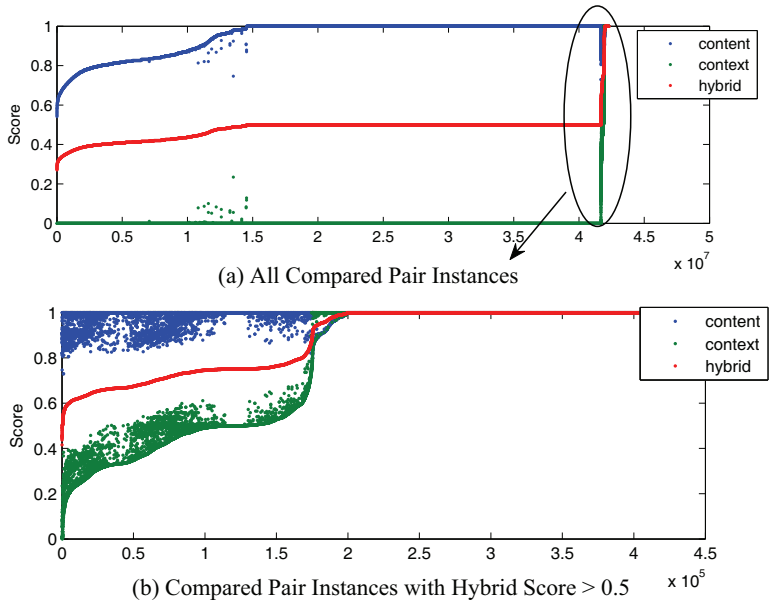

Fig. 3. Comparison of content-based, context-based and hybrid measures for compared pairs in the blocks of size less than 100. (a) for 99% of the compared pairs the context-based similarity has a zero score and the hybrid score is less than or equal to 0.5. (b) for the remaining 1% of the compared pairs the hybrid score is larger than 0.5; clearly the hybrid score can provide a reasonable discrimination between the compared pairs. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/IDA-160814)

7.4. Statistical comparison of similarity measures

Here, we provide a statistical comparison between the three main similarity measures proposed in Section 6: Content-based, Context-based and Hybrid measures.

Figure 3 compares these measures according to the similarity scores calculated for 42,300,000 reference pairs (i.e., all possible pairs from blocks of size less than 100). In this figure, the x-axis shows the compared pair instances and the y-axis shows the similarity score; the compared pairs are sorted based on the hybrid measure. As can be seen from the Fig. 3(a), the hybrid measure is less than or equal to 0.5 for a major part of the comparisons. However, there is a sudden rise seen on the right side of this figure. Figure 3(b) provides a focused view on the last 420,000 compared pairs which have a hybrid score more than 0.5. In this subfigure, 218,570 out of 420,000 (52%) of the compared pairs have a hybrid score of 1, which discovers the matches with no content error. In these cases, the context score of 1 shows that not more than two references are found for a single entity. In the remaining 48% of the cases with hybrid scores of less than 1 and larger than 0.5, the compared pairs have either content error, or more than two references referring to the same entity, or both.

Table 5 shows briefly the general statistical information of each similarity measure. According to this table, the content-based measure is discovered to be not an informative discriminative method for the ER process, as the average similarity score in this measure is 0.94 and 65% of the pairs have a similarity score equal to 1. This shows only a minor improvement over the blocking strategy, since the blocking strategy is already designed based on the content information. However, by looking at the context-based measure 99% of the compared pairs have zero score, and just 1% of the data has non-zero score. This clearly shows a distinctive measure to reveal the matches in the data. In next subsection, further studies on the 1% compared pairs with non-zero context-based score confirms the high precision of this measure in predicting the correct matches between references.

7.5. Manual evaluation of matching results

Manual evaluation of predicted matches is a very time-consuming process for domain experts. Clearly it is not possible for domain experts to evaluate all the 420,000, reasonably similar reference pairs, in an acceptable time. The simple solution is to approximate the precision of recommended matches according to evaluation of N randomly selected reference pairs. In this section, we apply simple random sampling [61] to select $N = 1200$ matched pairs from the 420,000 compared pairs with hybrid score larger than 0.5. Krejcie et al. [62] proposed an efficient table (Table 6) for determining the sample size that is representative of a given population. According to Table 6, 1,200 samples sufficiently represent the population with size 420,000.

For each match a domain expert, familiar with genealogical data, used the provided evidence (names of references, family relationships, place and date of issue, blocking key and block confidence) to evaluate the match by choosing either a *True Positive* or a *False Positive* category.[1] This evaluation approach is similar to the approach described in [57,63].

The statistical study of 1,200 manual matches shows a very high precision of 92% in revealing the matches. According to the available evidence 1,100 matched pairs are considered valid (i.e., true positives) and 100 matched pairs show errors in matching (i.e., false positives). By further exploration of the false positives, a dominant pattern includes the father-child matching where the first name of father and child are the same (or at least very similar).

7.6. Discovered patterns in extracted matches

Based on the results of previous subsection, the compared reference pairs with hybrid measure larger than 0.5 have a high precision in predicting the matches. Therefore, here we explore the common patterns in these matchings. Table 7 shows six major patterns observed from the matched pairs. The first pattern P1 represents all matches between the parents mentioned in marriage certificates. These matches can be very helpful in revealing the new *Sibling* relations which are not explicitly available in database.

In the second pattern P2, matched pairs refer to the brides and grooms in marriage certificates. This pattern can be used to calculate the age of individuals.

Third pattern P3 provides matchings between the parents who are mentioned in marriage certificates and the parents who are mentioned in death certificates. This pattern can be used for either finding the sibling relations as in P1 or calculating the age of deceased person as in P2.

[1]Please note that due to missing data, typing errors, redundancies and lack of extra evidence confirming that two references point to the same real entity is impossible in many of the cases. Therefore, in evaluations of this paper, we stick to the evidence at hand and assume that a reasonable similarity between two references, similar family members, and feasible date and similar places can suggest a true positive match, otherwise it will be considered as a false positive match.

Table 7
Discovered informative patterns among 420,000 matches

#	Matching type	Avg. date Diff.	Freq.
P1	Parents of Bride/Groom *(to)* Parents of Bride/Groom *(reveals the siblings and also record duplicates)*	8.5 years	146,000
P2	Bride/Groom *(to)* Deceased	33 years	92,280
P3	Parents of Bride/Groom *(to)* Parents of deceased	21.4 years	76,090
P4	Bride/Groom *(to)* Parents of Bride/Groom	19 years	58,580
P5	Bride/Groom *(to)* Bride/Groom *(due to multiple marriage or other unknown reasons)*	11 years	56,450
P6	Deceased *(to)* Deceased *(Mostly due to early child death and record duplicates)*	0.2 years	39,420

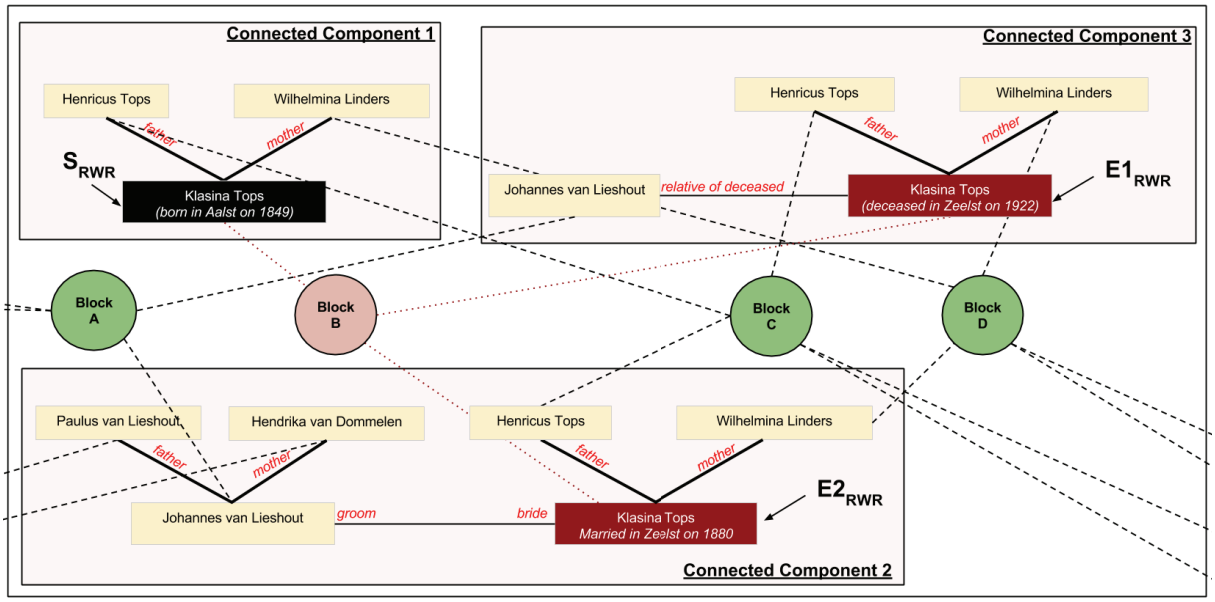

Fig. 4. A sample RWR-based matching in the genealogical graph. Components 1, 2 and 3 represent birth, marriage and death certificates of an entity called *Klasina Tops*. Some of the nodes from other certificates which are connected to these components via Block nodes are removed to improve the visibility of the graph. $\mathbf{S_{RWR}}$ denotes the starting point of the RWR and $\mathbf{E1_{RWR}}$ and $\mathbf{E2_{RWR}}$ denote the acceptable end points for RWR. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/IDA-160814)

The parents mentioned in marriage certificates might have their own marriage recorded in other certificates in a previous time; these matches are revealed in P4. Such matching pairs connect three generations of the population to each other.

The matches provided in P5 refer to brides or grooms that are reported in two marriage certificates. These matches are very probable to reveal multiple marriages of brides or grooms.

The last major pattern is P6 which reveals matches between deceased individuals with an average of 0.2 years in time of death. This can be mainly because of the early child death, where the child's first name is used for naming the next child; following by the death of second child.

Please note that since the dataset of Birth certificates used in this project is not a complete dataset (see Table 2 for detailed comparison), Birth certificates do not appear in any of the major patterns P1–P6.

7.7. Exploring a sample RWR process

Figure 4 illustrates parts of the augmented graph, containing three connected components representing the Birth, Marriage and Death of an individual. Component 1 provides the birth information of a person named *Klasina Tops* born on 1849. Component 2 provides the marriage information of a person with the same name, married 31 years later, and finally Component 3 provides the death information of a deceased with the same name who died 42 years after the marriage in Component 2. Nodes corresponding to Blocks A, B, C and D augment the graph by establishing connections between reference nodes with similar blocking keys.

The $\mathbf{S_{RWR}}$ in Component 1 shows the starting point of the RWR process. According to the proposed RWR process the node corresponding to Block B is excluded from the graph during implementation of RWR. Among the reachable nodes the only acceptable reference nodes in the neighborhood of starting point are the end points $\mathbf{E1_{RWR}}$ and $\mathbf{E2_{RWR}}$.

According to the results of the sample RWR in Fig. 4, new knowledge is discovered about the subject person *Klasina Tops*. For example, now we know that she was born in *Aalst, The Netherlands*[2] on 1849. She was married at the age 31 in *Zeelst, The Netherlands*,[3] and she died at the age of 73 in *Zeelst*.

8. Toward population reconstruction

The prerequisite of applying any data analysis in genealogical domain is building the Entity graph, in which each node represents a real entity (corresponds to set of matched references in the original Reference graph) and each edge shows the family relationship between two entities. This is almost impossible to discover even the basic statistical patterns such as "number of males and females in the population", "average number of children per entity", "average age of marriage for both males and females" without building this Entity graph.

Therefore, to use the predicted matches for further genealogical study, first the Graph_Conversion procedure is introduced which converts the Reference graph to an Entity graph. Second, the new information produced by using the Entity graph is discussed.

8.1. Graph_Conversion Procedure

The Graph_Conversion procedure takes a Reference graph as its input, merges the corresponding matched references in order to construct the real entities and generates an Entity graph as final output.

8.1.1. Input: Reference graph

Consider the Reference graph (i.e., the input of Graph_Conversion procedure) as $RG = (V_1, E_1)$ where each node $r_i \in V_1$ is a reference and each edge $e_i \in E_1$ shows a family relationship between two references. Each reference r_i is described by 9 features $\langle M_1, M_2, \ldots, M_9 \rangle$, where these features are *first name*, *last name prefix*, *last name*, *date* and *place of birth*, *date* and *place of marriage*, and *date* and *place of death*. Depending on the certificate type, some of these features have null value. Figure 5(a) shows a subset of the Reference graph, generated from three Marriage certificates and one Death certificate.

[2] http://en.wikipedia.org/wiki/Aalst,_North_Brabant.
[3] http://en.wikipedia.org/wiki/Zeelst.

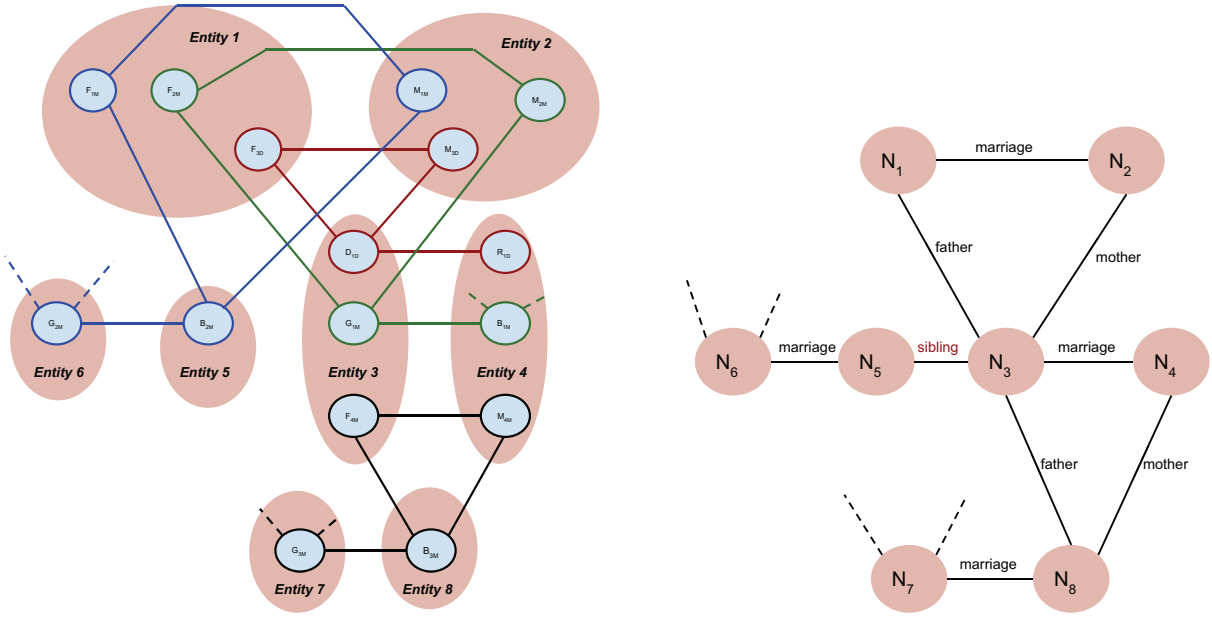

Fig. 5. Converting the reference graph to entity graph, using the predicted matches. In (a) $F_{.M}$, $M_{.M}$, $G_{.M}$ and $B_{.M}$ denote the father, mother, groom and bride in a marriage certificate. The $F_{.D}$, $M_{.D}$, $D_{.D}$ and $R_{.D}$ denote the father, mother, deceased and his/her relative in a death certificate. In (b), $N_1, N_2, \ldots, N_8$ represent the entities. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/IDA-160814)

8.1.2. Conversion method

Using the predicted matches, we introduce a set of matched references in form of $\mathsf{MR}(N_i) = \{r_{i1}, r_{i2}, \ldots, r_{im}\}$ where every two references r_{ij} and r_{ik} for $j, k = 1, 2, \ldots, m$ are predicated to be a match. In order to convert the Reference graph to Entity graph, every set of references $\mathsf{MR}(N_i)$ are replaced with the node N_i which represents the i^{th} entity. Two entities N_i and N_j are connected with an edge either if there exists $r_{ix} \in \mathsf{MR}(N_i)$ and$r_{jy} \in \mathsf{MR}(N_j)$ where$(r_{ix}, r_{jy}) \in E_1$ or if a sibling relation is detected between N_i and N_j.

For instance, the three matched references shown by $F_{1\ M}$, $F_{2\ M}$ and F_{3D} in Fig. 5(a) with the role of Father from two Marriage and one Death certificates belong to the set $\mathsf{MR}(N_1)$ and collapse into a single entity node N_1 as shown in Fig. 5(b).

Furthermore, the features $\langle M_1, M_2, \ldots, M_9 \rangle$ are updated based on following two rules.

R1: In case of conflict among the first three features (i.e., *first name*, *last name prefix* and *last name*) the value with longest length is chosen as the final value.

R2: In case of conflict among the last six features (i.e., *date* and *place* of *birth*, *marriage* and *death*) one of the existing values is chosen randomly.

In practice, R1 compensates for the typos in form of missing letters and words in first and last names. In most of the cases there is no need to use R2 as the references which should be merged have complementary information in form of date and place of birth, marriage and death. In specific cases, for instance merging two born children from two different birth certificates, one of the birth dates and places is chosen randomly and a notification is generated for experts to manually check the conflict.

Table 8
Comparison of basic statistics between the Reference and Entity graphs. The number of nodes, links and connected components in Entity graph are less than the Reference graph. The number of nodes and links in the biggest component of Entity graph are 11 and 17 times larger than the Reference graph, respectively

	Reference graph	Entity graph
Number of nodes	5,270,000	4,950,000
Number of links	5,620,000	5,466,000
Number of connected components	1,170,000	995,000
Number of nodes in biggest component	6	66
Number of links in biggest component	7	118
Connectivity ratio (Formula (1))	0.778	0.799
Number of males	2,636,000	2,476,000
Number of females	2,634,000	2,474,000
Sibling relations	0	146,000

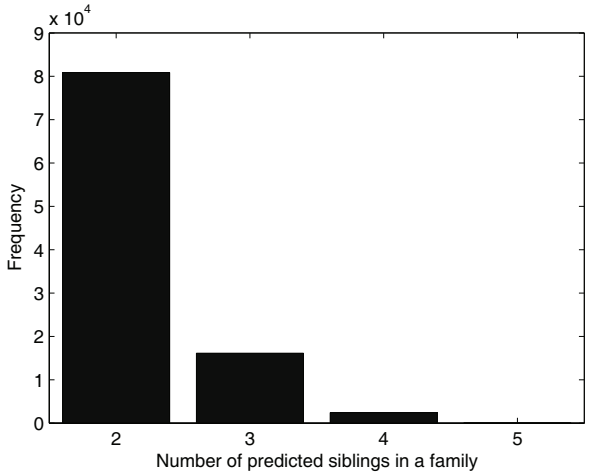

Fig. 6. Number of siblings in a family are extracted by using the matches revealed as pattern P1. In general 146,000 sibling relations are extracted. 80,000 of these relations reveal the families with two children, 54,000 of them reveal the families with three children (i.e., 18,000 families) and 12,000 of these relations reveal the families with four children (i.e., 2,000 families). The families with more children are not revealed, due to the low matching scores produced by the proposed RWR technique.

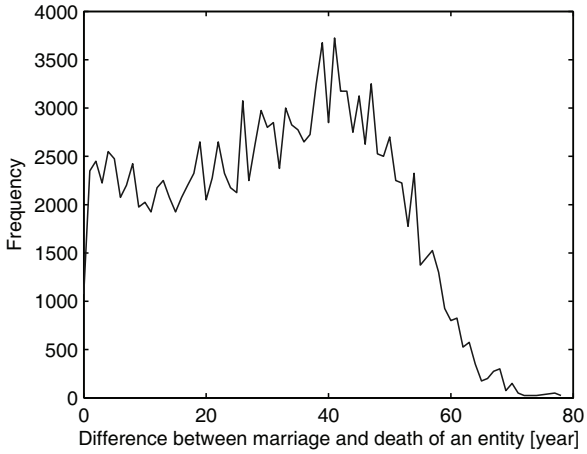

Fig. 7. The year difference between the marriage of an entity and his/her death is extracted using the patterns P2 and P3. The difference between marriage and death of entities is between 0 and 80 years. In average each death happens 29 years after marriage.

8.1.3. Output: Entity graph

Consider the Entity graph (i.e., the ouptut of Graph_Conversion procedure) as $EG = (V_2, E_2)$ where each node $N_i \in V_2$ is an entity, integrating the information of multiple references $r_{i1}, r_{i2}, \ldots, r_{im}$.

Figure 5(b) shows the outcome of the Graph_Conversion procedure applied on all the matched references in Fig. 5(a).

8.2. Studying the entity graph

After building the entity graph from original reference graph, now, we could analyze our data in more concrete way. Table 8 gives a comparison of basic statistics between the Reference and Entity graphs.

The Entity graph contains a new family relationship called "sibling", which hasn't been available in the original reference graph. Therefore, having this new relationship we can study the distribution of number of siblings in the families. Figure 6 shows the distribution of families in which 2, 3, 4 or 5 siblings exists. As a result it can be seen that 80,000 parents are detected who have 2 children, 18,000 of parents have 3 children and 2,000 parents have 4 children.[4] This information can be very helpful for fertility analysis, which is very important for genealogists, though very difficult by just using the raw data.

Furthermore, merging the reference nodes of different certificates generates other new information about the entities. For instance, merging a married groom with a deceased person, immediately provides the year difference between marriage and death. Using this new information, Fig. 7 shows the distribution of difference between death and marriage in the Entity graph. As a result it can be seen that the bride/groom, in average, dies 29 years after his/her marriage; this difference has been mostly reported to be 44 years (3700 times). This type of analysis can be used to compute the marriage age (by merging the born child, with his/her marriage) and also death age (by merging the born child with a deceased), which is beyond the main scope of this paper.

9. Conclusions and future work

The reliability of any data analysis method strongly depends on the quality of the input data. Considering the domain of genealogical research, a huge amount of inaccurate information and different types of ambiguities can be seen in available datasets. Therefore, as the first step toward any data analysis approach, effective ER techniques are required for enrichment and integration of the references extracted from different historical documents.

Traditional ER methods match references with reasonable precision by considering only the content information (FIRSTNAME and LASTNAME in our case) of each reference. More recent context-based ER techniques improve the traditional content-based ER techniques by taking contextual information, mainly in the form of linkage information among references, into account. However, in domains such as genealogy with very limited linkage information, which results in a disjoint graph, context-based approaches produce the same result as traditional content-based methods. To overcome this limitation, we augment the original graph by adding new nodes and edges to the original graph, and then we apply a random walk based method to consider new contextual information available for each reference.

To augment the original graph, we follow the graph homophily principle which assumes edges are more likely among similar nodes than between dissimilar nodes. In order to avoid having to compare all pairs of nodes (references), we apply the blocking strategy proposed by Rahmani et al. [57] on the same genealogical dataset. This leads to a reduction in search space by 3.5×10^{-5}. After partitioning the potential similar nodes into blocks, for each high confidence block we add a new node to the original graph. Then, we connect the new node to all its reference members. Following this process, 680,000 new nodes and 3,450,000 new edges are added to the graph. The number of disjoint connected components decreased from 1,170,000 in the original graph to 26,000 in the augmented graph and the number of nodes in the largest connected component increased from 6 in the original graph to 5,600,000 in the augmented graph. As a result, according to Formula (1), the connectivity ratio increases from 0.77 in the original graph to 0.99 in the augmented graph.

[4] The proposed RWR technique assigns low confidence scores to siblings in the families with more than 4 children. Therefore, these sibling relations are not taken into account in this phase of research.

Therefore, by using the huge amount of contextual information available in the augmented graph, we applied our proposed method and succeeded to discover 420,000 reference matches among 5,300,000 references. Domain experts evaluated the 1200 randomly selected predicted reference matches manually and they reported the precision of 92% for our approach. The main discovered pattern in 8% False Positive predictions was the false matching between father and child references with exact similar names! We detected six major patterns in the 420,000 predicted reference matches. Among the discovered patterns, a pattern which predicts matching among parent references from marriage certificates turns out to be particularly prominent. This pattern is seen 146,000 times in the database, the matched pairs refer to certificates issued in within 8.5 years, in average. These matches can be very helpful in discovering the new *Sibling* relations which are not explicitly available in database. Section 7.6 discussed all the six patterns in detail.

Regarding future research induced by our work, we see four particularly important directions for refinement and extension of our approach. First, further exploration of possibilities for extensive validation of the achieved results. This is challenging because we typically do not have grounded truth against which the results can be directly compared. We have already discussed and validated our results in Section 7 by human domain experts. However, it would be very useful and considerably more efficient to have a way of (at least partially) evaluating the results automatically or at least semi-automatically by simulating domain-expert behavior. Second, we could augment the original graph even more by analyzing the discovered matches/patterns in a closed loop process where outcomes of the previous iterations used as an input feedback in the new iteration. For instance, we could introduce new types of family relationships such as "sibling relationship" among the references that their parents were matched in the previous iterations. Third, we could study common characteristics of specific groups of entities in entity graph to unravel previously unknown information and connections within the groups [64]. Fourth, our proposed Entity Resolution approach could be applied in other research areas as well, such as life sciences. Elloumi et al. [65] consider entity identification and entity resolution as the first and still very important step in integrating biological databases. Additionally, our proposed hybrid similarity measure could be applied to biological networks, such as Protein-Protein Interaction Networks (PPI) [66], to discover the similar protein pairs in the networks. Then, biologists could evaluate the most similar proteins pairs with respect to sharing the similar biological characteristics [67–69].

Acknowledgments

This research has been supported under the NWO CATCH program in the MISS project (project no. 640.005.003). The authors are grateful to the BHIC center for the support in data gathering and direction.

References

[1] H.B. Newcombe, J.M. Kennedy, S.J. Axford and A.P. James, Automatic linkage of vital records, *Science* **130** (1959), 954–959.

[2] I.P. Fellegi and A.B. Sunter, A theory for record linkage, *Journal of the American Statistical Association* **64** (1969), 1183–1210.

[3] W. Winkler, The state of record linkage and current research problems, 1999, URL: http://www.census.gov/srd/papers/pdf/rr99-04.pdf.

[4] I. Bhattacharya and L. Getoor, 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, ACM, New York, NY, USA, (2004), 11–18. URL: http://doi.acm.org/10.1145/1008694.1008697. doi: 101145/1008694.1008697.

[5] P. Ravikumar and W.W. Cohen, A hierarchical graphical model for record linkage, in: *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, AUAI Press, Arlington, Virginia, United States, (2004), 454–461. http://dlacm.org/citation.cfm?id=1036843.1036898.
[6] W.E. Winkler, Methods for record linkage and bayesian networks, Technical Report, Statistical Research Division, U.S. Census Bureau, 1994.
[7] M.A. Hernández and S.J. Stolfo, The merge/purge problem for large databases, *SIGMOD Rec* **24** (1995), 127–138.
[8] A.E. Monge and C. Elkan, An efficient domain-independent algorithm for detecting approximately duplicate database records, in: *DMKD*, (1997). URL: http://dblp.uni-trier.de/db/conf/dmkd/dmkd97.html#MongeE97.
[9] S. Sarawagi and A. Bhamidipaty, Interactive deduplication using active learning, in: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, ACM, New York, NY, USA, (2002), 269–278. URL: http://doi.acm.org/10.1145/775047.775087. doi: 101145/775047.775087.
[10] R. Ananthakrishna, S. Chaudhuri and V. Ganti, Eliminating fuzzy duplicates in data warehouses, in: *Proceedings of the 28th International Conference on Very Large Data Bases*, VLDB '02, VLDB Endowment, (2002), 586–597. URL: http://dlacm.org/citation.cfm?id=1287369.1287420.
[11] W.W. Cohen, H.A. Kautz and D.A. McAllester, Hardening soft information sources, in: *KDD*, R. Ramakrishnan, S.J. Stolfo, R.J. Bayardo and I. Parsa, eds, ACM, (2000), 255–259. URL: http://dblp.uni-trier.de/db/conf/kdd/kdd2000.html#CohenKM00.
[12] A. McCallum, K. Nigam and L.H. Ungar, 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*, ACM, (2000), 169–178.
[13] P. Singla and P. Domingos, Entity resolution with markov logic, in: *In ICDM*, IEEE Computer Society Press, (2006), 572–582.
[14] O. Benjelloun, H. Garcia-Molina, D. Menestrina, Q. Su, S.E. Whang and J. Widom, Swoosh: A generic approach to entity resolution, *VLDB J* **18** (2009), 255–276.
[15] L. Getoor and A. Machanavajjhala, Entity resolution for big data, in: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, (2013), 1527–1527.
[16] J. Efremova, B. Ranjbar-Sahraei, F.A. Oliehoek, T. Calders and K. Tuyls, A baseline method for genealogical entity resolution, in: *Workshop on Population Reconstruction*, (2014).
[17] I. Bhattacharya and L. Getoor, Entity resolutions in graphs, in: *Mining Graph Data*, D. Cook and L. Holder, eds, Wiley, 2006.
[18] L. Kolb and E. Rahm, Parallel entity resolution with dedoop, *Datenbank-Spektrum* **13** (2013), 23–32.
[19] S.E. Whang, D. Menestrina, G. Koutrika, M. Theobald and H. Garcia-molina, Entity resolution with iterative blocking, Technical Report, 2008.
[20] S. Whang and H. Garcia-Molina, Entity resolution with evolving rules, *PVLDB* **3** (2010), 1326–1337.
[21] W.E. Winkler, String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage, 1990.
[22] G. Navarro, A guided tour to approximate string matching, *ACM Computing Surveys (CSUR)* **33** (2001), 31–88.
[23] J. Efremova, B. Ranjbar-Sahraei and T. Calders, A hybrid disambiguation measure for inaccurate cultural heritage data, in: *The 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, (2014).
[24] P. Bogdanov and A.K. Singh, Function prediction using neighborhood patterns, in: *BioKDD Workshop*, (2008).
[25] J. He, M. Li, H.-J. Zhang, H. Tong and C. Zhang, Manifold-ranking based image retrieval, in: *Proceedings of the 12th Annual ACM International Conference on Multimedia*, MULTIMEDIA '04, ACM, New York, NY, USA, (2004), 9–16. URL: http://doi.acm.org/10.1145/1027527.1027531. doi: 101145/1027527.1027531.
[26] G. Jeh and J. Widom, Simrank: A measure of structural-context similarity, in: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, ACM, New York, NY, USA, (2002), 538–543. URL: http://doi.acm.org/10.1145/775047.775126. doi: 101145/775047.775126.
[27] L. Page, S. Brin, R. Motwani and T. Winograd, The pagerank citation ranking: Bringing order to the web, in: *Proceedings of the 7th International World Wide Web Conference*, Brisbane, Australia, (1998), 161–172. URL: citeseer.nj.nec.com/page98pagerankhtml.
[28] A.E. Monge and C. Elkan, An efficient domain-independent algorithm for detecting approximately duplicate database records, in: *DMKD*, (1997).
[29] W.E. Winkler, Matching and record linkage, *Business Survey Methods* **1** (1995), 355–384.
[30] A. Monge and C. Elkan, The field matching problem: Algorithms and applications, in: *In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, (1996), 267–270.
[31] G. Navarro, A guided tour to approximate string matching, *ACM Comput Surv* **33** (2001), 31–88.
[32] W.W. Cohen, P. Ravikumar and S.E. Fienberg, in: *Proceedings of IJCAI-03 Workshop on Information Integration*.
[33] S. Chaudhuri, K. Ganjam, V. Ganti and R. Motwani, Robust and efficient fuzzy match for online data cleaning, in: *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, SIGMOD '03, ACM, New York, NY, USA, (2003), 313–324. URL: http://doi.acm.org/10.1145/872757.872796. doi: 101145/872757.872796.

[34] E.S. Ristad, P.N. Yianilos and S. Member, Learning string edit distance, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20** (1998), 522–532.
[35] M. Bilenko and R.J. Mooney, Adaptive duplicate detection using learnable string similarity measures, in: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, ACM, New York, NY, USA, (2003), 39–48. URL: http://doi.acm.org/10.1145/956750.956759. doi: 101145/956750.956759.
[36] W.W. Cohen and J. Richman, Learning to match and cluster large high-dimensional data sets for data integration, in: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, ACM, New York, NY, USA, (2002), 475–480. URL: http://doi.acm.org/10.1145/775047.775116. doi: 10.1145/775047.775116.
[37] S. Tejada, C.A. Knoblock and S. Minton, Learning object identification rules for information integration, *Information Systems* **26** (2001), 2001.
[38] S. Sarawagi and A. Bhamidipaty, Interactive deduplication using active learning, in: *The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002)*, (2002), 269–278.
[39] N. Koudas, A. Marathe and D. Srivastava, Flexible string matching against large databases in practice, in: *Proceedings of the Thirtieth International Conference on Very Large Data Bases – Volume 30*, VLDB '04, VLDB Endowment, (2004), 1078–1086. URL: http://dlacm.org/citation.cfm?id=1316689.1316782.
[40] W.E. Yancey and W.E. Yancey, Evaluating string comparator performance for record linkage, Technical Report, Bureau of the Census, 2005.
[41] M. Bilenko, R. Mooney, W. Cohen, P. Ravikumar and S. Fienberg, Adaptive name matching in information integration, *IEEE Intelligent Systems* **18** (2003), 16–23.
[42] P. Singla and P. Domingos, Multi-relational record linkage, in: *KDD-2004 Workshop on Multi-Relational Data Mining*, (2004), 31–48.
[43] X. Dong, A. Halevy and J. Madhavan, Reference reconciliation in complex information spaces, in: *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, SIGMOD '05, ACM, New York, NY, USA, (2005), 85–96. URL: http://doi.acm.org/10.1145/1066157.1066168. doi: 101145/1066157.1066168.
[44] D.V. Kalashnikov and S. Mehrotra, Domain-independent data cleaning via analysis of entity-relationship graph, *ACM Trans Database Syst* **31** (2006), 716–767.
[45] A. Doan, Y. Lu, Y. Lee and J. Han, Object matching for information integration: A profiler-based approach, in: *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web*, Acapulco, Mexico, (2003), 53–58.
[46] J. Neville, M. Adler and D. Jensen, Clustering relational data using attribute and link information, in: *In Proceedings of the Text Mining and Link Analysis Workshop, 18th International Joint Conference on Artificial Intelligence*, (2003), 9–15.
[47] E.F. Codd, *The Relational Model for Database Management: Version 2*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.
[48] R.A. Elmasri and S.B. Navathe, *Fundamentals of Database Systems*, 3rd edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
[49] M. McPherson, L. Smith-Lovin and J.M. Cook, Birds of a feather: Homophily in social networks, *Annual Review of Sociology* **27** (2001), 415–444.
[50] A. McCallum, K. Nigam and L.H. Ungar, 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*, KDD '00, ACM, New York, NY, USA, (2000), 169–178. URL: http://doi.acm.org/10.1145/347090.347123. doi: 101145/347090.347123.
[51] R. Baxter, P. Christen and T. Churches, A comparison of fast blocking methods for record linkage, in: *ACM SIGKDD '03 Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, (2003), 25–27.
[52] M. Bilenko, B. Kamath and R.J. Mooney, Adaptive blocking: Learning to scale up record linkage, in: *ICDM*, IEEE Computer Society, (2006), 87–96. URL: http://dblp.uni-trier.de/db/conf/icdm/icdm2006.html#BilenkoKM06.
[53] L.O. Evangelista, E. Cortez, A.S. da Silva and W. Meira, Jr, Adaptive and flexible blocking for record linkage tasks, *JIDM* **1** (2010), 167–182.
[54] D. Knuth, The art of computer programming 1: Fundamental algorithms 2: Seminumerical algorithms 3: Sorting and searching, 1968.
[55] L. Philips, Hanging on the metaphone, *Computer Language* **7** (1990).
[56] L. Philips, The double metaphone search algorithm, *C/C++ Users Journal* **18** (2000), 38–43.
[57] H. Rahmani, B. Ranjbar-Sahraei, G. Weiss and K. Tuyls, Contextual entity resolution approach for genealogical data, in: *Workshop on Knowledge Discovery, Data Mining and Machine Learning*, (2014).
[58] M. Institute, Meertens institute databases, 2014. URL: http://www.meertens.knaw.nl/cms/en/collections/databases.
[59] T. Can, O. Çamoğlu and A.K. Singh, Analysis of protein-protein interaction networks using random walks, in: *Proceedings of the 5th International Workshop on Bioinformatics*, ACM, (2005), 61–68.

[60] I. Bhattacharya and L. Getoor, Collective entity resolution in relational data, *ACM Transactions on Knowledge Discovery From Data (TKDD)* **1** (2007), 5.
[61] D.S. Moore, G.P. McCabe and B.A. Craig, *Introduction to the Practice of Statistics: Extended*, W.H. Freeman, New York, 2009. URL: http://opac.inria.fr/record=b1131072.
[62] R.V. Krejcie and D.W. Morgan, Determining sample size for research activities, *Educational and Psychological Measurement* **30** (1970), 607–610.
[63] J. Efremova, B. Ranjbar-Sahraei, F.A. Oliehoek, T. Calders and K. Tuyls, An interactive, web-based tool for genealogical entity resolution, in: *25th Benelux Conference on Artificial Intelligence*, (2013), 376–377.
[64] V. Koenraad, C. Myriam and D. Jan, A short manual to the art of prosopography, in: *Prosopography Approaches and Applications A Handbook*, K.-R. K.S.B., ed., Unit for Prosopographical Research (Linacre College), Oxford, (2007), 35–69.
[65] M. Elloumi and A.Y. Zomaya, *Biological Knowledge Discovery Handbook: Preprocessing, Mining and Postprocessing of Biological Data*, 1st edition, Wiley Publishing, 2013.
[66] U. Stelzl, U. Worm, M. Lalowski, C. Haenig, F.H. Brembeck, H. Goehler, M. Stroedicke, M. Zenkner, A. Schoenherr, S. Koeppen, J. Timm, S. Mintzlaff, C. Abraham, N. Bock, S. Kietzmann, A. Goedde, E. ToksÃűz, A. Droege, S. Krobitsch, B. Korn, W. Birchmeier, H. Lehrach and E.E. Wanker, A human protein-protein interaction network: A resource for annotating the proteome, *Cell* **122** (2005), 957–968.
[67] T. Milenkovi and N. Prulj, Uncovering biological network function via graphlet degree signatures, *Cancer Informatics* **6** (2008), 257–273.
[68] Z. Dezso, Y. Nikolsky, T. Nikolskaya, J. Miller, D. Cherba, C. Webb and A. Bugrim, Identifying disease-specific genes based on their topological significance in protein networks, *BMC Systems Biology* **3** (2009), 36+.
[69] H. Ruffner, A. Bauer and T. Bouwmeester, Human protein-protein interaction networks and the value for drug discovery, *Drug Discovery Today* **12** (2007), 709–716.