Temporal group linkage and evolution analysis for census data

Victor Christen
University Leipzig
Germany
christen@informatik.uni-leipzig.de

Anika Groß
University Leipzig
Germany
gross@informatik.uni-leipzig.de

Jeffrey Fisher
Australian National University
Australia
jeffrey.fisher@anu.edu.au

Qing Wang
Australian National University
Australia
qing.wang@anu.edu.au

Peter Christen
Australian National University
Australia
peter.christen@anu.edu.au

Erhard Rahm
University Leipzig
Germany
rahm@informatik.uni-leipzig.de

ABSTRACT

The temporal linkage of census data allows the detailed analysis of population-related changes in an area of interest. It should not only link records about the same person but also support the linkage of groups of related persons such as households. In this paper, we thus propose a new approach to both temporal record and group (household) linkage for census data and study its application for change analysis. The approach utilizes the relationships between individuals to determine the similarity of groups and their members within a graph-based method. The approach is also iterative by first identifying high quality matches that are subsequently extended by matches found with less restrictive similarity criteria. A comprehensive evaluation using historical census data from the UK indicates a high effectiveness of the proposed approach. Furthermore, the linkage enables an insightful analysis of household changes determined by so-called evolution patterns.

1. INTRODUCTION

Census data provides valuable information about individuals and households within cities or regions at a specific point in time [18]. Moreover, the temporal linkage of different census datasets allows analyzing the changes that occur in a population which is of increasing importance for social, demographic, economic and health-related studies [8, 13, 18]. In general, the temporal analysis of changing information about individuals and other entities is seen as a major requirement and challenge for future data analysis [6].

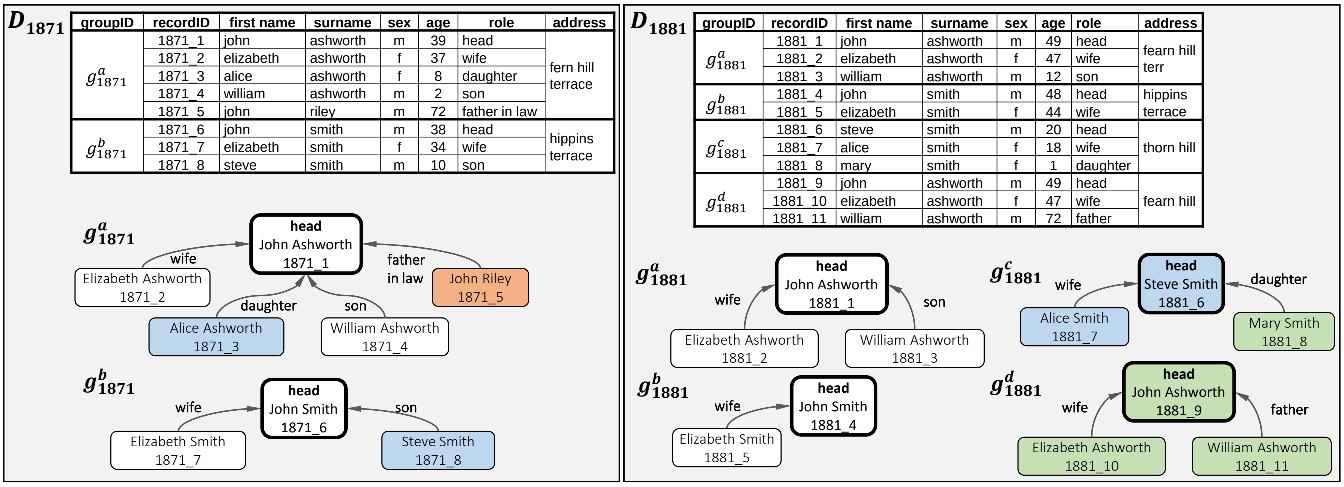
There is a large number of available census datasets for different regions of interest. Normally such census datasets are collected on a regular basis, e.g., every ten years, so that multiple successive versions can be utilized to analyze population- and household-related changes. A key prerequisite for such change studies is the temporal linkage of person records as well as of households, representing a group of individuals living together. There has been a modest amount of previous work on such temporal linkage problems, mainly focusing on temporal record linkage taking into account that linkage-relevant attributes such as surname, address or occupation may change over time [2, 5, 15, 17] (see Section 6). These studies mostly ignore the relationships between individuals, e.g., people living together in a household. Moreover, they do not consider the linkage and evolution of groups of related individuals, such as in a household, which is a main focus of this paper.

Fig. 1 illustrates the problem for two successive historical census datasets from 1871 and 1881. In each dataset, individuals are associated to a single household and have a household-specific relationship or role, such as head of household or daughter (of the head of household). These relationships can be represented in *household graphs* as shown in the lower part of Fig. 1. To understand the changes between the two considered points in time, one has to find matching individuals and their changes which is challenging, in particular due to the occurrence of frequent names (first names like 'John' and 'Elizabeth' or surnames like 'Ashworth' and 'Smith' in our dataset) and attribute changes. Of course, we also need to identify people who occur only in one of the datasets because of deaths, emigration, births and immigration. Obviously, a person in one census dataset should match to at most one person in another census dataset so that temporal linkage aims at a 1:1 mapping between person records. Moreover, we want to identify household-related changes, e.g., to what degree the individuals in a household have stayed together or moved to other households. In this case, we have to identify a many-to-many mapping between households.

In our example in Fig. 1, the daughter of the head of household in g^a_{1871} (*Alice*) married *Steve* from household g^b_{1871} and they both moved into the new household g^c_{1881} as shown in the 1881 census data (see blue nodes in household graphs). *John Riley* died within the considered time

Series ISSN: 2367-2005

10.5441/002/edbt.2017.83



D_{1871}

groupID	recordID	first name	surname	sex	age	role	address
g^a_{1871}	1871_1	john	ashworth	m	39	head	fern hill terrace
	1871_2	elizabeth	ashworth	f	37	wife	
	1871_3	alice	ashworth	f	8	daughter	
	1871_4	william	ashworth	m	2	son	
	1871_5	john	riley	m	72	father in law	
g^b_{1871}	1871_6	john	smith	m	38	head	hippins terrace
	1871_7	elizabeth	smith	f	34	wife	
	1871_8	steve	smith	m	10	son	

D_{1881}

groupID	recordID	first name	surname	sex	age	role	address
g^a_{1881}	1881_1	john	ashworth	m	49	head	fearn hill terr
	1881_2	elizabeth	ashworth	f	47	wife	
	1881_3	william	ashworth	m	12	son	
g^b_{1881}	1881_4	john	smith	m	48	head	hippins terrace
	1881_5	elizabeth	smith	f	44	wife	
g^c_{1881}	1881_6	steve	smith	m	20	head	thorn hill
	1881_7	alice	smith	f	18	wife	
	1881_8	mary	smith	f	1	daughter	
g^d_{1881}	1881_9	john	ashworth	m	49	head	fearn hill
	1881_10	elizabeth	ashworth	f	47	wife	
	1881_11	william	ashworth	m	72	father	

Figure 1: Example census data for two points in time (1871 and 1881). Red / green / blue colored nodes denote individuals who disappear / newly appear / moved to another household.

period (red node for the first census), while the child *Mary Smith* was born (green node for the second census). Furthermore, a new family (household g^d_{1881}) moved into the region. Note that the groups g^a_{1881} and g^d_{1881} have highly similar attribute values, but only g^a_{1871} should be linked to g^a_{1881}. To overcome such ambiguities of person-related attributes, our linkage approach will utilize stable attributes (such as birth year) as well as stable relationships between records, such as family relations or age differences.

In this paper, we propose and evaluate a novel approach for temporal group and record linkage for historical census data that considers the relationships between individuals. Moreover, we use the linked information for an initial change analysis for individuals and households. Specifically, we make the following contributions:

- We propose a new graph-based approach to linking households and person records between successive versions of census data. The approach works in several steps and utilizes an approximate record matching approach to identify pairs of related households. The linkage of households is based on their graph representation, and identifies common subgraphs referring to individuals with stable attributes and relationships. The final record links are derived from the linked subgraphs. The approach is iterative and determines group and record links in multiple rounds with decreasing restrictiveness. In this way we start with finding the best matches and apply less restrictive similarity criteria only for the more difficult to match records and groups.

- We utilize the determined record and group links for an initial change analysis based on different evolution patterns, including the splitting and merging of households.

- We apply and evaluate the proposed approaches for six historical UK census datasets. The evaluation shows that the proposed linkage approaches are highly effective and that they allow insightful observations regarding the changes over time.

In the next section, we formalize our problem of temporal record and group linkage. The linkage approach is described in Section 3, while Section 4 discusses the use of evolution patterns for change analysis. In Section 5, we evaluate our temporal linkage approach and analyze the evolution of households for the considered census datasets. We then discuss related work and conclude.

2. PROBLEM DEFINITION

Our approaches to temporal linkage and evolution analysis work on a set of census datasets $\mathbb{D}$ referring to different points in time. Each dataset D_i of time t_i consists of a set of person records R_i and a set of groups G_i representing households. The records in R_i are homogeneously structured and have attributes such as *first name*, *surname*, *age*, *occupation*, and so on. A group $g_i \in G_i$ consists of associated person records (household members) of R_i as well as relationships between them. Each record is part of one group (household) only, i.e., groups are not overlapping.

Groups are represented as (household) *graphs* g_i=(V_i,E_i) where the vertices of V_i correspond to the group members and the edges of E_i represent their relationships. Relationships (edges) have attributes or properties, in particular a relationship type or role, e.g., *daughter*. Such relationships can be part of the input data (as in Fig. 1) or can be derived later, e.g., the age difference between two persons. For our example, we may record in the graph for group g^a_{1871} not only the role *daughter* between *Alice* and her father *John* but also the age difference 31 (39-8). Our algorithm not only determines additional properties such as age differences but also additional relationships among group members, e.g., that *Alice* and $William$ are siblings with an age difference of 6.

Given these datasets and graphs, we want to determine for each pair $D_i = (R_i, G_i)$ and $D_{i+1} = (R_{i+1}, G_{i+1})$ of successive census datasets a so-called record mapping $\mathcal{M}^{i,i+1}_R$ and a group mapping $\mathcal{M}^{i,i+1}_G$. The *record mapping* $\mathcal{M}^{i,i+1}_R$ includes all pairs of records referring to the same real-world person (person links). The mapping is of cardinality 1:1 since each person in R_i can match with at most one person in R_{i+1} and vice versa:

$$\begin{aligned}\mathcal{M}_R^{i,i+1} := \{(r_i, r_{i+1}) | (r_i, r_{i+1}) \in R_i \times R_{i+1} \wedge \\ \exists (r_i, r'_{i+1}) \in \mathcal{M}_R \rightarrow r'_{i+1} = r_{i+1} \wedge \\ \exists (r'_i, r_{i+1}) \in \mathcal{M}_R \rightarrow r'_i = r_i\}\end{aligned} \tag{1}$$

A *group mapping* $\mathcal{M}_G^{i,i+1}$ consists of group pairs where a group g_i of G_i corresponds completely or partially to a group g_{i+1} of G_{i+1} according to the common records:

$$\mathcal{M}_G^{i,i+1} := \{(g_i, g_{i+1}) | (g_i, g_{i+1}) \in G_i \times G_{i+1}\} \tag{2}$$

Group mappings can be of cardinailty many-to-many (N:M) since persons of a household can match persons of several households in a different census.

For our running example of Fig. 1, the record mapping includes seven person links between the white and blue colored graph vertices, e.g. link $(1871\_1, 1888\_1)$ for *John Ashworth* and $(1871\_3, 1888\_7)$ for the link between *Alice Ashworth* and *Alice Smith.* The two groups in the first census dataset are split among two groups each in the second dataset, so that there are four group links including (g^a_{1871}, g^a_{1881}). In our evolution analysis, we will also consider person records and groups that are not reflected in these mappings, e.g. relating to newly occurring or disappeared persons and households.

3. TEMPORAL GROUP LINKAGE

Determining the record and group mappings for the temporal linkage of census datasets is challenging not only due to changing attribute values for the same person (e.g., for surname or occupation) but also due to the high ambiguity and frequent occurrence of certain attribute values, as well as because of data quality issues, e.g., misspelled names, errors for age etc. Group linkage has hardly been studied before [1] and requires a flexible approach to determine many-to-many mappings taking into account that households may split or merge. Similar in spirit to collective entity resolution [1, 20], we determine the similarity between records not only based on attribute values but also considering relationships between records (persons) within a graph-based approach. Furthermore, we not only address record linkage but solve record and group linkage jointly within a combined approach. To better deal with the partially low similarity of matching person records and the need to determine many-to-many group mappings we propose an iterative approach for temporal linkage. We first identify safe matches with a high similarity and then continuously relax the similarity criterion to find additional record and group links.

Algorithm 1 describes our approach for determining a group mapping $\mathcal{M}_G^{i,i+1}$ and a record mapping $\mathcal{M}_R^{i,i+1}$ between two successive census datasets D_i and D_{i+1}. The input of the algorithm includes two similarity functions for record matching and parameters for the iterative adjustment of a similarity threshold δ. We first give a high-level description of the algorithm and its main steps. These steps are then explained in more detail in the four following subsections of this section.

At first, we enrich the graphs for each group (household) in the two input datasets by adding implicit relationships between group members, such as derivable family relations. Moreover, we compute for each relationship between persons the age difference as an additional relationship property for later use in the similarity computations.

The main part of the algorithm is a loop to iteratively identify and extend the group mapping $\mathcal{M}_G^{i,i+1}$ and the record mapping $\mathcal{M}_R^{i,i+1}$. In each iteration, we first apply a similarity function *Sim_func* to determine an initial linking and clustering of person records based on attribute similarities only (pre-matching step). The similarity function *Sim_func* specifies the person attributes, a weighting vector ω, and a similarity threshold δ (i.e., two persons are considered to match if the weighted sum of their attribute similarities exceeds δ). In the first iteration, we apply a high value δ_high for δ to start with identifying safely matching persons as a basis for also finding safe group matches. Group matches are only determined for pairs of groups connected by at least one (initial) person link. For such group pairs, we apply a *subgraph matching* to determine shared subgraphs

Algorithm 1: Iterative record and group linkage

```
Input:
-$D_i$: old census dataset
-$D_{i+1}$: new census dataset
-$Sim\_func$: similarity function for initial record matching
-$\Delta$: delta for relaxing similarity threshold
-$\delta\_high$: upper bound of similarity threshold
-$\delta\_low$: lower bound of similarity threshold
-$Sim\_func_{rem}$: similarity function for remaining records
Output:
-$\mathcal{M}_R^{i,i+1}$: record mapping
-$\mathcal{M}_G^{i,i+1}$: group mapping
// initialization
$\mathcal{M}_R^{i,i+1} \leftarrow \emptyset, \mathcal{M}_G^{i,i+1} \leftarrow \emptyset$
$\mathcal{M}_R^p \leftarrow \emptyset, \mathcal{M}_G^p \leftarrow \emptyset$
$G_i \leftarrow$completeGroups $(G_i)$
$G_{i+1} \leftarrow$completeGroups $(G_{i+1})$
$Sim\_func.\delta \leftarrow \delta\_high$
// iterative subgraph matching
repeat
    // identification of candidates
    $\mathcal{C} \leftarrow$ prematching $(R_i, R_{i+1}, Sim\_func)$
    // subgraph matching and criteria computation
    $Sub_G \leftarrow$subgroups $(\mathcal{C}, G_i, G_{i+1}, Sim\_func)$
    $\mathcal{M}_G^p \leftarrow$selectGroupMatches $(Sub_G)$
    // extend group mapping
    $\mathcal{M}_G^{i,i+1} \leftarrow \mathcal{M}_G^{i,i+1} \cup \mathcal{M}_G^p$
    // extend record mapping
    $\mathcal{M}_R^p \leftarrow$extractRecordMapping $(\mathcal{M}_R^p, Sub_G, R_i, R_{i+1})$
    $\mathcal{M}_R^{i,i+1} \leftarrow \mathcal{M}_R^{i,i+1} \cup \mathcal{M}_R^p$
    // extract unlinked records and records that are
       related to unlinked records
    $R_i \leftarrow$ nonMatchedRecords $(R_i, \mathcal{M}_R^{i,i+1})$
    $R_{i+1} \leftarrow$ nonMatchedRecords $(R_{i+1}, \mathcal{M}_R^{i,i+1})$
    $Sim\_func.\delta \leftarrow Sim\_func.\delta - \Delta$
until $\mathcal{M}_G^p = \emptyset \vee Sim\_func.\delta < \delta\_low$
// match remaining records
$\mathcal{M}_R^p \leftarrow$match $(R_i, R_{i+1}, Sim\_func_{rem})$
$\mathcal{M}_R^{i,i+1} \leftarrow \mathcal{M}_R^{i,i+1} \cup \mathcal{M}_R^p$
$\mathcal{M}_G^{i,i+1} \leftarrow \mathcal{M}_G^{i,i+1} \cup extractGroupLinks(\mathcal{M}_R^p, G_i, G_{i+1})$
return $< \mathcal{M}_R^{i,i+1}, \mathcal{M}_G^{i,i+1} >$
```

[1] We are only aware of one approach for group-based linkage of census data [8] that is non-iterative and less sophisticated regarding the use of relationships. In our evaluation in Section 5, we will compare the results for this scheme with our approach.

with both matching persons and matching relationships. In general, a group of the first census dataset has several candidate group matches in the second dataset so that we select the best group matches considering multiple criteria such as the degree of record and relationship similarity. The matching subgraphs of linked groups are then used to extract the matching records for inclusion into the record mapping (line 10 of Algorithm 1).

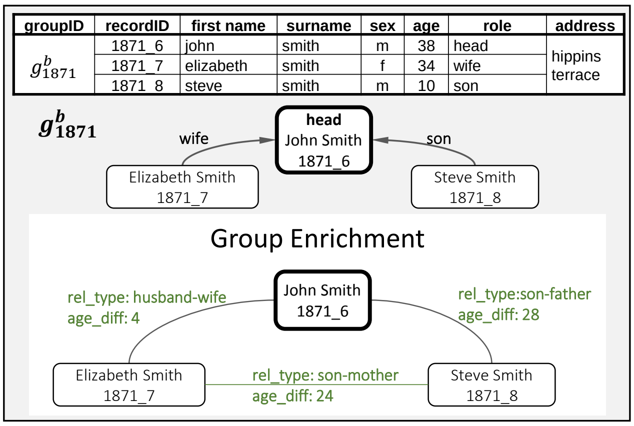
Further iterations only process records not yet included in the record mapping determined so far. We continuously relax the similarity threshold by a decrement Δ until a minimal similarity threshold δ_low is reached (or no further group links are identified). Using such relaxed similarity thresholds aims at finding additional matches between records and groups even in the presence of erroneous or changed attribute values.

After all iterations are performed we have finished subgraph -based group linkage. For the remaining records not yet associated within matching subgraphs, we apply a second attribute-based similarity function Sim_func_{rem} to identify further person links for inclusion into the record mapping (line 17). Moreover, we extend the group mapping by adding the group pairs that are now linked by the newly found record links $\mathcal{M}_G^{i,i+1}$ (line 19).

In the following subsections, we describe the discussed steps in more detail. We start with explaining the preprocessing step to enrich the existing household graphs by implicit relationships and additional relationship properties (Subsection 3.1). In Subsection 3.2, we describe the prematching step of records. In Subsection 3.3, we outline our subgraph matching approach to identify common subgraphs. We then introduce the criteria and algorithm used to select the group matches (Subsection 3.4).

3.1 Group Enrichment

In the initialization phase, we enrich each household group by adding implicit relationships and stable properties such as age differences between persons. In our case, each individual of a household is given a role related to the head of household (which is a special role). This role may not be preserved in future census datasets since individuals may become members of a different household and the head of household may change as well. Hence, comparing households based on these relations only is insufficient in the presence of household changes. We therefore enrich the household graphs by implicit relationships for each record pair of the original group and replace the head-dependent relationship types by a unified type. To increase the semantics of a relationship, we further add the age difference between two household members as a time-independent relationship property. Fig. 2 shows an example of the group enrichment phase for group g^b_{1871}. The relationship between *Elizabeth Smith* and *Steve Smith* is added. Moreover, the age differences age_diff between persons as well as the relationship types rel_type are added to the relationships.



groupID	recordID	first name	surname	sex	age	role	address
g^b_{1871}	1871_6	john	smith	m	38	head	hippins terrace
	1871_7	elizabeth	smith	f	34	wife	
	1871_8	steve	smith	m	10	son	

Figure 2: Example of the group enrichment phase for group g^b_{1871}.

3.2 Pre-Matching

Pre-matching clusters similar records in the census datasets based on their attribute similarity and assigns a cluster label to each record. These labels are utilized to simplify subgraph matching since the labels identify similar records without further similarity computation.

Pre-matching first applies similarity function Sim_func to compare each record of R_i with each record of R_{i+1}. The similarity function specifies the attributes to be compared as well as the attribute-specific similarity function, e.g., q-gram string matching [4]. Furthermore, it uses a weighting vector ω and a required minimum similarity δ. Applying the attribute-specific similarity functions to a pair of records r_i and r_{i+1} results is a similarity vector $\vec{sim}_{(r_i,r_{i+1})}$. Using ω we determine an aggregated similarity $agg\_sim_{(r_i,r_{i+1})}$ by calculating a weighted sum of the attribute similarities:

$$agg\_sim_{(r_i,r_{i+1})} = \omega \cdot \vec{sim}_{(r_i,r_{i+1})} \qquad (3)$$

We then keep only the record pairs whose similarity is above the specified threshold δ as potential record matches. Furthermore, we determine the transitive closure or connected components of these match pairs (record links) to cluster together all directly and indirectly matching records. We

Cluster label	recordID	first name	surname
A	1871_1	john	ashworth
	1881_1	john	ashworth
	1881_9	john	ashworth
B	1871_2	elizabeth	ashworth
	1881_2	elizabeth	ashworth
	1881_10	elizabeth	ashworth
C	1871_4	william	ashworth
	1881_3	william	ashworth
	1881_11	william	ashworth
D	1871_6	john	smith
	1881_4	john	smith
E	1871_7	elizabeth	smith
	1881_5	elizabeth	smith
F	1871_8	steve	smith
	1881_6	steve	smith
G	1881_8	mary	smith
H	1871_5	john	riley
I	1871_3	alice	ashworth
K	1881_7	alice	smith

Figure 3: Pre-matching result for running example. Records with the same cluster label represent similar records.

assign to each record of a cluster a unique label, so that records of the same cluster have the same label.

Fig.3 shows the resulting clusters for the running example by using the attributes *first name* and *surname*, $\omega = (0.5, 0.5)$ and similarity threshold 1. Pre-matching results in the shown ten clusters where all records of a cluster share the same first name and surname. We then assign the cluster labels A, B etc. to the respective records of the clusters.

3.3 Subgraph Matching

Subgraph matching looks for common subgraphs in each pair of groups g_i and g_{i+1} of $G_i \times G_{i+1}$ to determine likely group links. To avoid the computation of the cross product between G_i and G_{i+1}, subgraph matching is only applied for pairs of groups sharing at least one similar record, i.e., having the same cluster label.

The subgraph g_{sub} between two groups g_i and g_{i+1} (represented by their enriched graphs with g_i=(V_i,E_i) and g_{i+1}= (V_{i+1},E_{i+1}) consists of a set of vertices R_{sub} and a set of edges E_{sub}. Each vertex in R_{sub} represents a pair of equally labeled (i.e., similar) records v_i from V_i and v_{i+1} from V_{i+1}. Two vertices $(v1_i, v1_{i+1})$ and $(v2_i, v2_{i+1})$ of R_{sub} are connected by an edge of E_{sub} if both the old records $v1_i$, $v2_i$ and the new records $v1_{i+1}, v2_{i+1}$ of these vertices are connected within their enriched graphs of g_i and g_{i+1}, respectively. Furthermore, we require that these edges must have the same relationship type and highly similar relationship properties, in our case regarding the age differences.

Fig. 4 illustrates subgraph matching for group g^a_{1871} from the first census dataset and the two groups g^a_{1881} and g^d_{1881} from the second dataset. For the group pair (g^a_{1871}, g^a_{1881}) we have three matching vertices with labels A, B and C. The three edges have the same relationship types and the same or very similar age differences. The second group pair (g^a_{1871}, g^d_{1881}) also shares three vertices with labels A, B and C but only one of the edges has the same relationship type and similar age difference. Hence the common subgraph is reduced to the one shown in the bottom right of Fig.4.

3.4 Selection of Group Links

Subgraph matching generates candidates for group linkage based on common subgraphs for different group pairs. There may be several linkage candidates per group in G_i and in G_{i+1} so that we have to find the best matching group pairs. The necessary selection should especially guarantee that each record of a group is only linked to one record of another group (This is not the case for the example in Fig.4 where we have two linkage candidates for members of group g^a_{1871}). However, a group can link to more than one group if their subgroups are disjoint.

To select for a certain group g_i the best-matching groups in G_{i+1} we consider all subgraphs g_{sub}=(R_{sub},E_{sub}) involving g_i and apply an aggregated similarity measure. This measure combines three scores capturing the record similarity (Eq. 5), edge similarity (Eq. 6) and the uniqueness (Eq. 7) of a subgroup g_{sub}. The results of the similarity functions are aggregated according to Eq. 4 whereby α determines the influence of record similarity and β represents the weight of edge similarity.

$$g\_sim = \alpha \cdot avg\_sim + \beta \cdot e\_sim + (1 - \alpha - \beta) \cdot unique \tag{4}$$

- *Average Record Similarity*

For this score we determine the average of the aggregated similarities *agg_sim* for the record pairs of R_{sub}. These aggregated similarities are already determined during pre-matching for each record pair (see section 3.2) and can be obtained from the respective clusters in $\mathcal{C}$.

$$avg\_sim(g_i, g_{i+1}, g_{sub}) = \frac{\sum\limits_{(r_i, r_{i+1}) \in R_{sub}} agg\_sim_{(r_i, r_{i+1})}}{|R_{sub}|} \tag{5}$$

- *Edge Similarity*

The edge similarity *e_sim* evaluates the similarity of the relationship properties *rp_sim* in the edges in a subgraph, for example the similarity of the age differences between two individuals in the older group g_i vs. the age difference in the newer group g_{i+1}. Furthermore, we apply an aggregation measure similar to the Dice-Coefficient to relate the edge similarities to the total number of relationships of the considered groups g_i and g_{i+1} thereby giving higher weight to those subgraphs covering a large portion of their relationships.

$$e\_sim(g_i, g_{i+1}, g_{sub}) = 2 \cdot \frac{\sum\limits_{e \in E_{sub}} rp\_sim(oldEdge(e), newEdge(e))}{|E_i| + |E_{i+1}|} \tag{6}$$

- *Uniqueness*

If two group pairs are similar w.r.t both the average record similarity as well as the edge similarity, we like to prefer the group link between the two groups containing records that are less ambiguous than the records of other group pairs. Therefore, we define the uniqueness for a group pair based on the number of vertices of R_{sub} of g_{sub} and the aggregated number of records that are assigned to the same label like the records of R_{sub}. The uniqueness is defined as follows:

$$unique(g_i, g_{i+1}, g_{sub}) = 2 \cdot \frac{|R_{sub}|}{\sum\limits_{r_i \in R_{sub}} |label(r_i)|} \tag{7}$$

The uniqueness of a group pair g_i and g_{i+1} is 1, if the labels are only assigned to the common records of g_i and g_{i+1} and there exists no other record of R_i or R_{i+1} that has the same label.

For the example of Fig. 4, we obtain the following similarity values for the group pairs (g^a_{1871}, g^a_{1881}) and (g^a_{1871}, g^d_{1881}):

$$\begin{aligned} avg\_sim(g^a_{1871}, g^a_{1881}, g_{sub}) &= \frac{1+1+1}{3} = 1 \\ e\_sim(g^a_{1871}, g^a_{1881}, g_{sub}) &= 2 \cdot \frac{1+1+1}{10+3} = 0.46 \\ unique(g^a_{1871}, g^a_{1881}, g_{sub}) &= 2 \cdot \frac{3}{3+3+3} = 0.66 \end{aligned} \tag{8}$$

$$\begin{aligned} avg\_sim(g^a_{1871}, g^d_{1881}, g_{sub}) &= \frac{1+1}{2} = 1 \\ e\_sim(g^a_{1871}, g^d_{1881}, g_{sub}) &= 2 \cdot \frac{1}{10+3} = 0.15 \\ unique(g^a_{1871}, g^d_{1881}, g_{sub}) &= 2 \cdot \frac{2}{3+3} = 0.66 \end{aligned}$$

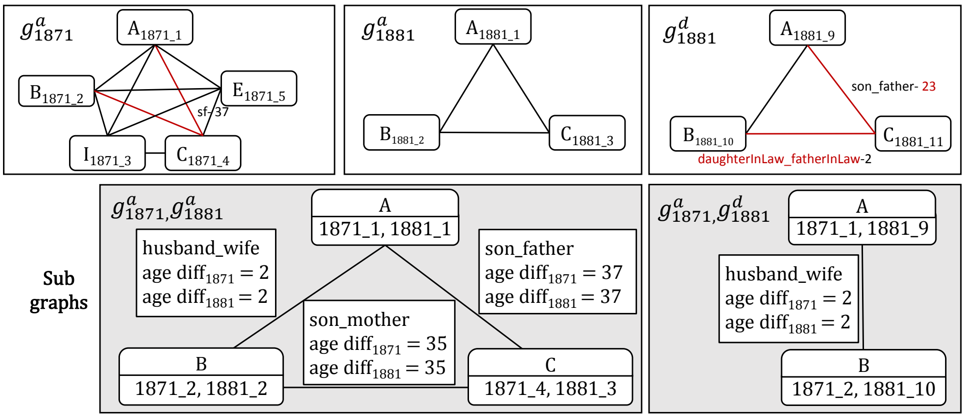


Figure 4: Subgraphs for group pairs (g^a_{1871}, g^b_{1881}) and (g^a_{1871}, g^d_{1881}) of the running example. For (g^a_{1871}, g^d_{1881}), the red-coloured edges are not matched due to a different relationship type or non-similar age difference.

Algorithm 2: Selection of group links

```
Input:
-$Sub_G$: set of quadruples of $<g_i, g_{i+1}, g_{sub}, g\_sim>$
Output:
-$\mathcal{M}^p_G$: partial group mapping
1  $\mathcal{M}^p_G \leftarrow \emptyset$
2  $lookup \leftarrow \emptyset$
   // initialize priority queue ordered by $g\_sim$
3  for $(g_i, g_{i+1}, g_{sub}, g\_sim) \in Sub_G$ do
4      $pq \leftarrow pq.insert(g_i, g_{i+1}, g_{sub}, g\_sim)$
5  while $pq \neq \emptyset$ do
6      $< g_i, g_{i+1}, g_{sub}, g\_sim > \leftarrow pq.max()$
7      $pq \leftarrow pq.remove()$
       // sets of linked records of $g_i$ and $g_{i+1}$
8      $linked\_R_i \leftarrow lookup.get(g_i)$
9      $linked\_R_{i+1} \leftarrow lookup.get(g_{i+1})$
       // records of $g_i$ and $g_{i+1}$ contained in $g_{sub}$
10     $R^i_{sub} \leftarrow getOldRecords(g_{sub})$
11     $R^{i+1}_{sub} \leftarrow getNewRecords(g_{sub})$
12     if $linked\_R_i \cap R^i_{sub} = \emptyset \wedge linked\_R_{i+1} \cap R^{i+1}_{sub} = \emptyset$ then
13         $\mathcal{M}^p_G \leftarrow \mathcal{M}^p_G \cup \{(g_i, g_{i+1})\}$
14         $linked\_R_i \leftarrow linked\_R_i \cup R^i_{sub}$
15         $linked\_R_{i+1} \leftarrow linked\_R_{i+1} \cup R^{i+1}_{sub}$
16         $lookup \leftarrow lookup.update(g_i, processed\_R_i)$
17         $lookup \leftarrow lookup.update(g_{i+1}, processed\_R_{i+1})$
18 return $\mathcal{M}^p_G$
```

The aggregated similarity of these values reaches a higher value for group pair (g^a_{1871}, g^a_{1881}) than for (g^a_{1871}, g^d_{1881}) due to the higher edge similarity of the former pair. As a result, we would only include group pair (g^a_{1871}, g^a_{1881}) in the group mapping and derive the record mapping only for the common subgraph of this pair.

After the determination of the introduced similarity values per subgroup, we apply Algorithm 2 for the selection of the best-matching group pairs. The algorithm follows a greedy strategy by considering subgraphs in the order of their aggregated similarity score. It also considers the disjointness of subgraphs and can determine group mappings of cardinality N:M.

In each iteration, we select the group pair with the highest group similarity from a priority queue *pq*. The selected pair (g_i, g_{i+1}) is added to the group mapping $\mathcal{M}^p_G$ if the overlap between the already linked records of g_i as well as g_{i+1} and the records of the record pairs of g_{sub} is empty (line 12). Thus, we ensure that a record is linked at most to one record. The linked records are represented by $linked\_R_i$ resp. $linked\_R_{i+1}$. Moreover, the records of g_i and g_{i+1} that correspond to a record pair of R_{sub} of g_{sub} are represented by the sets R^i_{sub} and R^{i+1}_{sub}. These sets are returned by *getOldRecords* and *getNewRecords* respectively for a certain subgroup g_{sub}. If a group link is added, we update sets of linked records $linked\_R_i$ and $linked\_R_{i+1}$ for g_i resp. g_{i+1} (line 14 to 17).

Based on the selected group matches, we are able to identify the record matches contained in the corresponding subgraph g_{sub}. The record links are included in each vertex of g_{sub} since R_{sub} is defined as a set of pairs r_i and r_{i+1}. These pairs are the most appropriate links since the related groups are linked.

4. EVOLUTION ANALYSIS

We will now use the results of the temporal record and group linkage to detect changes between different census datasets in order to support the comprehensive evolution analysis of temporal census data. Such a change analysis should not be restricted to a low-level evaluation of individual links but should be realized at a higher, application-specific level to generate relevant and expressive change patterns. We will also include disappearing as well as newly appearing records and groups that are not reflected in the identified mappings but appear only in one of the census datasets. The analysis should further not be limited to two datasets but involve a series of successive census datasets covering longer periods of time.

In this initial study, we use the given census datasets and the determined linkage results to identify a set of basic and more complex changes for records and groups of records that can be identified with the help of so-called evolution patterns (Subsection 4.1). Furthermore, we propose the use of a so-called evolution graph (Subsection 4.2) to provide an aggregated change representation that is extensible to more than two census datasets. Such an evolution graph is a promising basis for advanced graph mining techniques, e.g., to determine frequent or unusual change scenarios.

(a)

Record evolution patterns
$preserve_R(1871\_1, 1881\_1)$,
$preserve_R(1871\_2, 1881\_2)$,
$preserve_R(1871\_3, 1881\_7)$,
$preserve_R(1871\_4, 1881\_3)$,
$preserve_R(1871\_6, 1881\_4)$,
$preserve_R(1871\_7, 1881\_5)$,
$preserve_R(1871\_8, 1881\_6)$,
$add_R(1881\_8), add_R(1881\_9),$
$add_R(1881\_10), add_R(1881\_11),$
$remove_R(1871\_5)$
Group evolution patterns
$preserve_G(g^a_{1871}, g^a_{1881})$,
$preserve_G(g^b_{1871}, g^b_{1881})$,
$add_G(g^c_{1881}), add_G(g^d_{1881})$,
$move(1871\_3, g^a_{1871}, 1881\_7, g^c_{1881})$,
$move(1871\_8, g^b_{1871}, 1881\_6, g^c_{1881})$

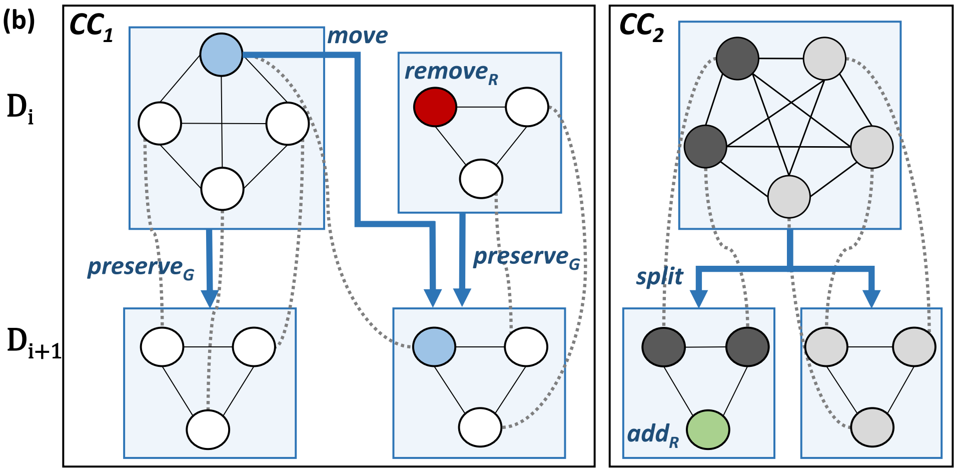


Figure 5: (a) Record and group evolution patterns for the running example. (b) Evolution graph and patterns for two successive census datasets D_i and D_{i+1}. Gray dotted lines represent record links, blue arrows indicate evolution patterns between related households.

4.1 Evolution Patterns

We define evolution patterns on individual records and on groups of records. There are three *record evolution patterns* called $preserve_R$, $remove_R$ and add_R. We identify these patterns by utilizing the record mapping $M_R^{i,i+1}$ as well as record sets R_i and R_{i+1} for two successive census datasets D_i and D_{i+1} as follows:

- $preserve_R$ is a record pair representing one individual linked between R_i and R_{i+1}.
  $\forall r_i, r_{i+1} \in R_i \times R_{i+1}$:
  $preserve_R(r_i, r_{i+1}) \leftrightarrow \exists(r_i, r_{i+1}) \in \mathcal{M}_R^{i,i+1}$

- add_R denotes an individual $r_{i+1} \in R_{i+1}$ that is not linked to any record of R_i.
  $\forall r_{i+1} \in R_{i+1} : add_R(r_{i+1}) \leftrightarrow \nexists(r_i, r_{i+1}) \in \mathcal{M}_R^{i,i+1}$

- $remove_R$ denotes an individual $r_i \in D_i$ that is not linked to any record of D_{i+1}.
  $\forall r_i \in R_i : remove_R(r_i) \leftrightarrow \nexists(r_i, r_{i+1}) \in \mathcal{M}_R^{i,i+1}$

To analyze the dynamics of groups, we further define *group evolution patterns* based on changes within groups. These patterns are add_G and $remove_G$ as well as the more complex patterns $preserve_G$, *move*, *split* and *merge*. The patterns $preserve_G$ and *move* both relate to pairs of linked groups but differ on whether the linked groups contain at least two preserved members ($preserve_G$) or only one (*move*). Each pattern is identified by utilizing the census datasets, the group mapping $\mathcal{M}_G^{i,i+1}$ and the record mapping $\mathcal{M}_R^{i,i+1}$:

- add_G denotes a new group $g_{i+1} \in G_{i+1}$ that did not exist in D_i. Thus, the group mapping $\mathcal{M}_G^{i,i+1}$ does not contain any link with g_{i+1}.

- Similarly, $remove_G$ contains a group of $g_i \in G_i$ that does not exist in G_{i+1} anymore.

- $preserve_G$ is a group pair connected by a 1:1 link. Moreover, each group consists of at least 2 individuals satisfying the $preserved_R$ pattern. This condition allows us to identify preserving households across censuses. The requirement that a 'preserved' household should have at least two remaining members is influenced by real-world situations such as households where only the parents remain after their children have moved to another household.

- *move* identifies pairs of linked groups with only one member in common (determined by the $preserve_R$ pattern) that has moved from the old to the new group (household).

- *split* identifies a change situation between a group $g_i \in D_i$ from the old dataset and a set of groups g^a_{i+1}, $g^b_{i+1}, ..., g^k_{i+1} \in G_{i+1}$ in the new dataset, where at least two individuals of g_i must overlap with each of the groups from G_{i+1}. Note, that each individual record can only be contained in one group, i.e., $g^a_{i+1}, g^b_{i+1}, ...,$ g^k_{i+1} are disjoint.

- *merge* covers the opposite situation between a set of groups g^a_i, $g^b_i, ..., g^k_i \in G_i$ from the old dataset and one group $g_{i+1} \in G_{i+1}$ from the new dataset, where at least two individuals from groups in G_i must overlap with the merged group g_{i+1}. Each individual record can only be contained in one group, i.e., $g^a_i, g^b_i, ..., g^k_i$ are disjoint.

Fig. 5(a) shows the corresponding record and group evolution patterns for our running example from Fig. 1. Seven records have been preserved from D_{1871} to D_{1881}. Moreover, there are 4 record additions and one removal. According to the defined group evolution patterns, two groups have been preserved (g^a and g^b), two groups newly appeared in 1881 (add_G for g^c and g^d) and two persons, Alice (1871_3) and Steve (1871_8), moved from their parents' households (g^a_{1871} and g^b_{1871}) to their own new household g^c_{1881}.

4.2 Evolution Graph

Based on the evolution patterns we want to realize further comprehensive evolution analyses for dynamically changing family structures and individual person histories. We propose the use of a so-called *evolution graph* reflecting the history of households across two or more successive census datasets. The graph $\mathcal{G}\_Evolution$ captures both the records and groups per census dataset as vertices and interconnects them across successive datasets by edges that are typed according to the identified evolution patterns (change types). Fig. 5(b) shows a sample evolution graph and evolution patterns for two successive versions D_i and D_{i+1}. Blue boxes represent group vertices and blue arrows represent group evolution patterns, i.e., the changes between households. Two groups have been preserved and are linked via the group pattern $preserve_G$ and one household has been split into two households. One individual moved between two households that are thus connected in the evolution graph. The figure also shows the mapping between individual records (gray dotted lines) as well as a new (add_R) and a removed ($remove_R$) record without incoming/outgoing edges.

The evolution graph enables the application of several graph mining approaches such as cluster analysis, pattern matching or finding frequent subgraphs. One analysis might be to identify households that are preserved across several census periods. A second use case is to identify clusters of related households that can be used for studies of genetic diseases. In Fig. 5(b), a simple computation of connected components on the exemplary evolution graph for two points in time leads to two components consisting of 4 (CC_1) and 3 (CC_2) households, respectively. Running such a computation for larger households graphs for many successive versions can produce longer chains of connected households, e.g., indicating relationships between many generations of families.

5. EVALUATION

In this section, we evaluate the introduced approaches for temporal record and group linkage for different historical census datasets from the UK that have also been used in a previous study [8]. We first describe these datasets and the evaluation setup in Subsection 5.1. We then evaluate the linkage quality of the new approaches for different configurations (Subsection 5.2). In Subsection 5.3 we compare our approach with the results of the previous study [8] as well as with the collective record linkage approach [14]. Finally, we discuss results of an initial evolution analysis for the considered census datasets.

5.1 Datasets and Setup

In our evaluation, we use six census datasets collected from 1851 to 1901 in ten-year intervals from the district of Rawtenstall in North-East Lancashire in the United Kingdom. Table 1 shows an overview of these datasets according to the number of records and households for the different time periods. The table also shows the number of unique value combinations of the first name and surname attributes to illustrate the degree of ambiguity for these attributes. Furthermore, we report the ratio of missing attribute values. The table shows that the number of households and persons has almost doubled within the 50 years period indicating a substantial population growth. There is a high degree of name ambiguity since each combination of first name and surname is far from unique but has an average frequency of up to 2.23 (for 1851) with a highly skewed frequency distribution due to the presence of frequent surnames such as *Ashworth* and *Smith*. Up to 6.5% of the attribute values are missing, which leads to in additional difficulties for finding correct temporal links.

t_i	1851	1861	1871	1881	1891	1901
$\lvert R_{t_i} \rvert$	17033	22429	26229	29051	30087	31059
$\lvert G_{t_i} \rvert$	3298	4570	5576	6025	6378	6842
$\lvert fn + sn \rvert$	7652	10198	13198	15505	17130	19910
$ratio_{mv}$	4.67%	4.19%	3.03%	4.09%	6.33%	6.51%

Table 1: Overview of the census datasets according to the number of records, households, unique combinations of first name and surname $|fn+sn|$ and the ratio of missing values $ratio_{mv}$.

To evaluate the quality of the group and record mappings in terms of precision, recall and F-measure [4], we use the reference mapping determined in [8]. It covers a subset of 1250 matching households from the 1871 and 1881 datasets that consist of 6864 and 6851 members resp. These household were manually linked by experts by focusing on person records found in both datasets.

In our evaluation, we compare different settings for the similarity function considering the string similarity for five attributes and different weight vectors ω_1 and ω_2 as shown in Table 2. We also evaluate different similarity thresholds for pre-matching as well as different weights for determining the aggregated group similarity for selecting group links.

Attribute	Matching method	ω_1	ω_2
First name	q-gram	0.2	0.4
Sex	exact	0.2	0.2
Surname	q-gram	0.2	0.2
Address	q-gram	0.2	0.1
Occupation	q-gram	0.2	0.1

Table 2: Compared set of attributes and the corresponding weighting vector ω to identify the set of blocks $\mathcal{B}$ that are used for the subgraph matching.

5.2 Linkage Evaluation

We first analyze the influence of different similarity functions during pre-matching and then discuss the impact of different similarity functions for selecting matching group pairs. Afterwards we study the effectiveness of incremental linkage.

5.2.1 *Influence of pre-matching configuration*

The proposed linkage approach builds on the initial record matching and clustering performed in the pre-matching step. We thus start our analysis by comparing the results for determining the attribute similarities based on the two weighting schemes ω_1 and ω_2 (Table 2) and different lower similarity threshold bounds δ_low. For iterative matching we use a start value $\delta\_high = 0.7$ for the similarity threshold δ and $\Delta = 0.05$ for decrementing the threshold until the minimal value δ_low is reached.

Table 3 shows the resulting group and record mapping quality in terms of precision, recall and F-measure for the two weighting schemes and four values of δ_low ranging from 0.4 to 0.55. We observe for all configurations high F-Measure results between 94% and 96% for both the determined record mappings and the group mappings, indicating a very high effectiveness of the proposed approach. The best

parameter	ω	ω_1				ω_2			
	δ_low	0.4	0.45	0.5	0.55	0.4	0.45	0.5	0.55
group mapping	Precision (%)	96.1	96.5	96.7	97.0	97.1	97.1	**97.3**	**97.3**
	Recall (%)	92.2	92.2	92.0	91.7	94.8	**94.8**	**94.8**	94.6
	F-measure (%)	94.1	94.3	94.3	94.2	96.0	96.0	**96.0**	95.9
record mapping	Precision (%)	96.6	96.8	96.8	96.8	97.5	97.5	**97.5**	**97.5**
	Recall (%)	91.9	91.9	91.9	91.8	93.7	93.7	**93.7**	93.7
	F-Measure (%)	94.2	94.3	94.3	94.3	95.6	95.6	**95.6**	95.5

Table 3: Quality of group and record mappings for different weighting vectors ω and lower bounds δ_low.

parameter	(α, β)	(1.0,0.0)	(0.0,1.0)	(0.5,0.5)	(0.33,0.33)	(0.2,0.7)
group mapping	Precision (%)	92.3	96.7	96.6	96.7	**97.3**
	Recall (%)	89.1	94.1	94.3	94.4	**94.8**
	F-Measure (%)	90.7	95.4	95.5	96.0	**96.0**
record mapping	Precision (%)	96.2	97.4	97.3	97.3	**97.5**
	Recall (%)	89.8	93.4	93.4	93.4	**93.7**
	F-Measure (%)	92.9	95.4	95.3	95.3	**95.6**

Table 4: Quality of the group and record mappings for different weights α and β to select matching groups.

F-measure results are generally achieved for $\delta\_low = 0.5$, although the differences are small for the other choices. The simple weighting scheme ω_1 giving equal weight to each of the five considered attributes is consistently outperformed by the alternate approach giving higher weight to attribute *first name* and only reduced weight for the less stable attributes *address* and *occupation*. Pre-matching with weight vector ω_2 thus improves F-measure by around 1.7% for the group mapping and up to around 1.3% for the record mapping.

Of course, there are many more possibilities to define the similarity function and we could also apply learning-based methods to find a near-optimal weight vector [4]. Still our results show that using the similarity function with weight vector ω_2 and $\delta\_low = 0.5$ achieve good and stable results making it an effective default configuration.

5.2.2 Similarity weights for selecting matching groups

We now evaluate the influence of the different weights α and β for determining the aggregated group similarity $g\_sim = \alpha \cdot avg\_sim + \beta \cdot e\_sim + (1 - \alpha - \beta) \cdot rel$ driving the selection of matching groups. Table 4 shows the results of the different weights. The quality of the group mapping highly depends on the edge similarity underlining the importance of considering the structural similarity within our household graphs. Without considering the edge similarity ($\beta = 0$), the F-measure for the group mapping drops to 90.7%, i.e. around 5.3% less than for the best configuration ($\alpha = 0.2, \beta = 0.7$) and also far less than when ignoring the record similarity ($\alpha = 0$). The uniqueness score can also improve the overall F-measure. For ($\alpha = 0.2, \beta = 0.7$) its weight is 0.1 which helped to achieve an improved F-measure compared to the three configurations where it is ignored (when the sum of α and β equals already 1). The best record mapping is also achieved for ($\alpha = 0.2, \beta = 0.7$) making it a good default configuration for our datasets.

5.2.3 Iterative vs non-iterative linkage

We now want to analyze to what degree the iterative group and record linkage with decreasing similarity thresholds is really helpful compared to a non-iterative, one-shot approach applying only a fixed minimal similarity threshold. To evaluate such a non-iterative approach we apply similarity functions with ω_2, $\delta\_high = 0.5$ and $\delta\_low = 0.5$ resulting in only one iteration. The results are shown in Table 5. We observe that the iterative approach indeed outperforms the non-iterative approach with an F-Measure improvement of $\approx 2.2\%$ for the group mapping and 3.1% for the record mapping. The improved quality mainly results from a substantially higher precision of more than 97% for both the group and record mapping. This is achieved because the iterative approach finds high-quality matches for the more restrictive thresholds while the more relaxed similarity threshold, with an increased risk of finding wrong matches, is limited to a subset of the records.

method		non-iterative	iterative
group mapping	Precision (%)	94.5	**97.3**
	Recall (%)	93.1	**94.8**
	F-measure (%)	93.8	**96.0**
record mapping	Precision (%)	91.8	**97.5**
	Recall (%)	93.1	**93.7**
	F-measure (%)	92.5	**95.6**

Table 5: Quality of the group mapping and record mapping by using the iterative vs. non-iterative approach.

5.3 Comparison with Existing Approaches

We compare our approach with two previously proposed methods: the collective entity resolution approach of [14] to determine a record mapping as well as the previous group linkage approach [8] for census data.

In [14], the authors propose a collective approach that is a specialization of [1]. It initially determines seed record links by applying a high record similarity. The seed links are used to incrementally identify additional links from the neighborhood of the linked records based on their attribute similarity and relational similarity. The overall algorithm follows a greedy strategy that selects in each iteration the record pair with the highest similarity. The related records update their similarities according to the selected record pair. In our implementation, we use the same similarity function as

method	CL	iter-sub
Precision (%)	93.5	**97.5**
Recall (%)	81.2	**93.7**
F-measure (%)	86.9	**95.6**

Table 6: Comparison of our approach with the collective linkage approach of [14] (CL) to determine a record mapping.

method	GraphSim	iter-sub
Precision (%)	**97.6**	97.3
Recall (%)	90.1	**94.8**
F-measure (%)	93.7	**96.0**

Table 7: Comparison of our approach with the household linkage approach of [8] (GraphSim).

in our approach (Table 2). Moreover, we filter all record pairs where the normalized age difference is more than 3 years[2]. To generate the seed link, we select the record links with a minimal similarity of 0.9. Table 6 shows the results of the record mapping obtained by collective linking. Our approach outperforms the collective approach w.r.t the record mapping quality by 8.6% for F-measure. The difference between our approach and the collective approach is that we can better link moved records with changed attribute values since we do not only link highly similar records (which is not sufficient for temporal linkage). Furthermore, our subgraph matching utilizes different relationships more comprehensively and benefits from incremental linkage.

The previous group linkage approach of [8] initially generates a highly selective record mapping consisting of 1:1 correspondences only. Based on this record mapping, the method calculates an average record similarity and an edge similarity between each group pair. Contrary to our approach, they calculate the similarities based on the initial 1:1 mapping. If correct record pairs are filtered out due to the 1:1 constraint, the approach is not able to identify these links. Hence, this filter step influences the average record similarity as well as the edge similarity, so that correct group links are not identified. Table 7 shows the results of the quality of the group mappings. Our approach achieves a significantly better F-measure for the group mapping compared to [8] (≈3.7%). This improvement is mainly because of a much higher recall that is limited in the previous approach mainly because of the use of the initial 1:1 mapping.

5.4 Analysis of Household Dynamics

Finally, we analyze the evolution of households from 1851 to 1901. For this purpose, we determine the evolution patterns for each successive census dataset pair based on the identified group and record mapping with the best parameter setting. Fig. 6 shows the frequency of each group evolution pattern for each pair of census datasets. In general, we observe an increasing number of households since the number of add_G patterns is higher than the number of $remove_G$ patterns for each new census. Moreover, we observe an increasing number of $preserve_G$ patterns due to the general increase in the number of households over time. From 1891 to 1901, there is also a high number of $remove_G$ patterns (up to ≈ 2200) indicating that many households may have moved to a new region. The complex patterns such as *split* and *merge* occur only rarely with an average occurrence of ≈ 100 for *split* and ≈ 70 while the *move* patterns are more frequent (≈ 1600 on average).

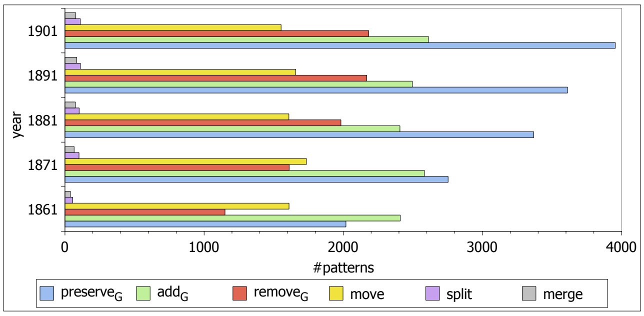


Figure 6: Quantitative Analysis of evolution patterns for census datasets from 1851 to 1901.

time interval	$\lvert preserve_G \rvert$
10	15705
20	7731
30	3322
40	1116
50	260

Table 8: Number of preserving households $\lvert preserve_G \rvert$ according to different time intervals (in years) from 1851 to 1901.

To analyze dependencies between households for the whole time period, we exploit the evolution graph and determine the largest connected component representing all households from 1851 to 1901 that are connected by group patterns. We identified the largest connected component with 17150 households over the complete interval from 1851 to 1901 thereby covering ≈52% of all households. Furthermore, we identify the number of preserved households according to different time intervals for the whole time period from 1851 to 1901. For instance, if we like to identify households that are preserved for 20 years, we define a graph pattern that consists of 2 edges with the pattern type $preserve_G$ since the difference between two census datasets is 10 years. Table 8 shows the number of preserved households for the different time intervals. The number of preserving households for all 10 year intervals (1851-61, 1861-71, 1871-81 etc.) represents the overall number of $preserve_G$ patterns of the quantitative analysis. Moreover, 260 household are preserved over the whole time period from 1851 to 1901.

6. RELATED WORK

Record linkage or entity resolution has been intensively studied in the past (see [4, 7, 12] for overviews). While the majority of approaches focus on evaluating the similarity of record attributes only, collective or context-based approaches additionally consider the similarity of relationships between entities for improved linkage decisions (e.g. [1, 8, 11, 14, 20, 23]). This idea has also been utilized in our approach but in a tailored way for use within groups such as households. Our approach is especially powerful as it considers different kinds of semantic relationships as well as the

[2] In our approach, subgraph matching ensures that such age differences are not accepted.

similarity of relationship attributes. Previous collective approaches have also not addressed temporal record linkage in contrast to our scheme.

Relatively few studies have investigated temporal record linkage (e.g., [2, 15, 17]) to link records within dynamically changing data. Existing approaches explicitly consider changing attribute values when matching individual records over time, e.g., by computing value transition probabilities [15]. Temporal clustering approaches as proposed in [3] group temporal records that belong to the same entity to reflect the entity history. Temporal record linkage approaches typically focus on matching individual person records while we also match groups of individuals and identify a record as well as group mapping to interconnect temporal records from census data.

Most closely related to our work is the group-based approach of [8] for matching households in historical census datasets. Our evaluation in Subsection 5.3 has shown that this previous scheme is outperformed by our approach due to its novel features such as an iterative group linkage and subgraph matching based on different semantic relationships. Richards and colleagues investigate in [21] the use of learning-based methods to optimize the use of attribute similarities for temporal record linkage (not group linkage) for census datasets. The observations of this study are complementary to ours and could be used for choosing alternate similarity functions for record matching.

Our work is further related to research on time and evolution-based analysis that is gaining increasing interest. For instance, there are studies analyzing historical web contents to find interesting patterns and trends [25], analyzing person histories on Twitter [16], or collecting and analyzing temporal knowledge from Wikipedia [24]. Our definition of change patterns is further related to previous work in the domain of ontology evolution [10, 22], in particular regarding change detection and diff computation (e.g. [9, 19]). These approaches typically identify basic and complex change operations between different ontology versions. We used this idea to identify time dependent patterns between groups of records to represent the semantics of changes in households over time. Based on the change patterns we are able to realize more comprehensive analysis, e.g., on complex evolution graphs.

7. CONCLUSIONS

We outlined and evaluated a new approach for temporal record and group linkage for the analysis of census data. The approach follows an iterative linkage strategy that first identifies high quality links thereby limiting the more error-prone identification of links between less similar records and groups to subsets of the input data. Group linkage is based on the identification of common subgraphs between groups such as households where we utilize the semantic relationships within groups and relationship properties such as the age differences between individuals. The evaluation showed the high effectiveness of the proposed approach that also outperforms a previous approach for linking census data.

We showed that the linkage results support a detailed evolution analysis of census data at both the level of individuals and groups. We proposed several evolution patterns to identify relevant changes including different kinds of group changes such as splits, merges and the movement of individuals from one group to another. All changes can be maintained within an evolution graph that can be used for a wide spectrum of change analysis, e.g., to identify frequent change patterns or to find connected groups over several census periods.

In future work, we plan to extend the change analysis of census data using the evolution graph and graph mining techniques. We also aim to apply and evaluate the proposed approach on larger census datasets. Furthermore, we want to study additional applications for group linkage, e.g., to analyze the changes in research teams or groups of coauthors over time.

8. ACKNOWLEDGEMENTS

This work was partially funded by the Australian Research Council (ARC) under Discovery Project DP160101934, and Universities Australia and the German Academic Exchange Service (DAAD).

References

[1] I. Bhattacharya and L. Getoor. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):5, 2007.

[2] Y.-H. Chiang, A. Doan, and J. F. Naughton. Modeling entity evolution for temporal record matching. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1175–1186. ACM, 2014.

[3] Y.-H. Chiang, A. Doan, and J. F. Naughton. Tracking entities in the dynamic world: A fast algorithm for matching temporal records. *Proceedings of the VLDB Endowment*, 7(6):469–480, 2014.

[4] P. Christen. *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection.* Springer Science & Business Media, 2012.

[5] P. Christen and R. W. Gayler. Adaptive temporal entity resolution on dynamic databases. In *Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference PAKDD*, pages 558–569, 2013.

[6] X. L. Dong, A. Kementsietsidis, and W.-C. Tan. A time machine for information: Looking back to look forward. *ACM SIGMOD Record*, 45(2):23–32, 2016.

[7] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE TKDE*, 19(1):1–16, 2007.

[8] Z. Fu, P. Christen, and J. Zhou. A graph matching method for historical census household linkage. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 485–496. Springer, 2014.

[9] M. Hartung, A. Groß, and E. Rahm. Conto-diff: generation of complex evolution mappings for life science ontologies. *Journal of Biomedical Informatics*, 46:15–32, 2013.

[10] M. Hartung, J. F. Terwilliger, and E. Rahm. Recent advances in schema and ontology evolution. In *Schema Matching and Mapping*, pages 149–190. 2011.

[11] D. V. Kalashnikov and S. Mehrotra. Domain-independent data cleaning via analysis of entity-relationship graph. *ACM Transactions on Database Systems (TODS)*, 31(2):716–767, 2006.

[12] H. Köpcke and E. Rahm. Frameworks for entity matching: A comparison. *Data & Knowledge Engineering*, 69(2):197 – 210, 2010.

[13] H. C. Kum, A. Krishnamurthy, A. Machanavajjhala, and S. Ahalt. Social genome: Putting big data to work for population informatics. *Computer*, 47(1):56–63, 2014.

[14] S. Lacoste-Julien, K. Palla, A. Davies, G. Kasneci, T. Graepel, and Z. Ghahramani. Sigma: Simple greedy matching for aligning large knowledge bases. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 572–580. ACM, 2013.

[15] F. Li, M. L. Lee, W. Hsu, and W.-C. Tan. Linking temporal records for profiling entities. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, pages 593–605, New York, NY, USA, 2015. ACM.

[16] J. Li and C. Cardie. Timeline generation: Tracking individuals on twitter. In *Proceedings of the 23rd international conference on World wide web*, pages 643–652. ACM, 2014.

[17] P. Li, X. Dong, A. Maurino, and D. Srivastava. Linking temporal records. *Proceedings of the VLDB Endowment*, 4(11):956–967, 2011.

[18] V. M. Moceri, W. A. Kukull, I. Emanual, G. van Belle, J. R. Starr, G. D. Schellenberg, W. C. McCormick, J. D. Bowen, L. Teri, and E. B. Larson. Using census data and birth certificates to reconstruct the early-life socioeconomic environment and the relation to the development of alzheimer's disease. *Epidemiology*, 12(4):383–389, 2001.

[19] N. F. Noy and M. A. Musen. PromptDiff: A fixed-point algorithm for comparing ontology versions. *AAAI/IAAI*, 2002:744–750, 2002.

[20] V. Rastogi, N. Dalvi, and M. Garofalakis. Large-scale collective entity matching. *Proceedings of the VLDB Endowment*, 4(4):208–218, 2011.

[21] L. Richards, L. Antonie, S. Areibi, G. W. Grewal, K. Inwood, and J. A. Ross. Comparing classifiers in historical census linkage. In *Proc. ICDM Workshops*, pages 1086–1094, 2014.

[22] L. Stojanovic, A. Maedche, B. Motik, and N. Stojanovic. *User-Driven Ontology Evolution Management*, pages 285–300. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[23] A. Thor and E. Rahm. Moma-a mapping-based object matching system. In *CIDR*, pages 247–258, 2007.

[24] Y. Wang, M. Zhu, L. Qu, M. Spaniol, and G. Weikum. Timely yago: harvesting, querying, and visualizing temporal knowledge from wikipedia. In *Proceedings of the 13th International Conference on Extending Database Technology*, pages 697–700. ACM, 2010.

[25] G. Weikum, N. Ntarmos, M. Spaniol, P. Triantafillou, A. A. Benczúr, S. Kirkpatrick, P. Rigaux, and M. Williamson. Longitudinal analytics on web archive data: it's about time! In *CIDR*, pages 199–202, 2011.