Matching Techniques for Historical Datasets

by Matthew Currie

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

B.A. COMPUTER SCIENCE HONOURS

in

Irving K. Barber School of Arts and Sciences

(Computer Science)

Supervisor: Dr. Ramon Lawrence

THE UNIVERSITY OF BRITISH COLUMBIA
(Okanagan)

April 2021

Abstract – Sacramental data does not normally contain information on a person's birthplace or birthdate, which are more commonly available in census records. As a result, the probabilistic techniques used in matching census records are not normally applicable to sacramental data. This paper looks to fill a gap in historical record linkage research by linking sacramental records using relational data. The techniques developed rely on the presence of familial relations in two records in order to match. The algorithm's precision was tested on baptismal records, marriage records, and death records. Finally, a visualization of people and their relationships across historical records was explored using network graphs.

Table of Contents

Table of Figures

1. Background

The method presented in this thesis was created to link individuals appearing in sacramental datasets (i.e. baptismal records, marriage records, and death records) contained within the Digital Archive Database Project (DADP), a historical archive that aggregates a variety of Métis historical documents. The fields utilized in matching individuals are first name, last name, location of the event (e.g. the church married/baptized in), approximate date of the event, the person's role in the event, and other people also recorded in the event. Two benefits of linking individuals in the DADP are that it reduces redundant data in searching the database, and, in addition, can provide a way of visualizing a person's relationships across historical data.

The matching techniques developed in this paper came about due to necessity. Statistical methods commonly used in matching census records were attempted but deemed ineffective largely due to sacramental data lacking the stronger identifying data available in census records such as birthplace and birthdate. This led to our approach's novel use of relational data to factor into a set of records matching criteria.

2. Introduction

2.1 The Matching Problem

Historical records such as census data, marriage records, and baptismal records are increasingly becoming digitized. As a result, there is an increasing need to link these historical records in relevant ways. The main difficulty in matching people across historical records comes from the data's lack of unique identifiers such as a Social Insurance Number. This prevents matches from easily being made with a high degree of certainty. Furthermore, this limitation requires record matching methods to rely on imperfect data by necessity. This limitation mostly comes in the form of enumeration errors as first name, last name, reported age, and birthplace are all prone to them. On top of this, situations can occur where two individual records share several commonalities yet refer to different people in actuality. An example of this is illustrated in **Figure 1**.

A record matching algorithm must then aim to strike a balance between two trade-offs: making as few false matches as possible (Type I errors); and making as many true matches as possible (Type II errors). Recent research in the field of historical record matching suggest using a probabilistic approach [1]. This approach centers around the use of field similarity scores and the Expectation-Maximization algorithm to determine a likelihood that two historical records are a match [2]. This likelihood is then used to check whether the pair of records belongs in the set of matches or nonmatches. Existing methods like this normally aim to make studying populations over time more feasible. This ultimately differs from the aims of record matching the DADP because there is no concern about over representing certain portions of the population in matching. Rather linking individuals in the DADP is intended to make it easier to search its archives and find connections between individuals.

Figure 1

First Name	Last Name	Date Recorded	Location	Relations	Role
William	Swan	1888/07/10	Grand Rapids Church	Alexander Swan; Julia Larance; Thomas Anderson	Husband
William	Swan	1888/10/10	Grand Rapids Church	William Swan (son); Mary Legree; Thomas Anderson	Husband's Father

The figure above illustrates two records of William Swan that come from the DADP's marriage records. In the top record, William Swan was the husband, and, in the bottom, William Swan is the husband's father. Even though there is a lot of similarity between the two records, its clear that the two records are not referring to the same person. Instead it seems much more likely that the two records are referring to a father and son as the roles and relations suggest. Moreover, records like these appear with a high enough frequency that matching based on names, locations, and time periods alone would lead to too many false positives.

2.2 Trouble with Probabilistic Matching on the DADP

Probabilistic record matching is not always a viable approach. When many records do not have an age and a location, for instance, probabilistic methods cannot adequately match these records. Consequently, sacramental data is not well suited for a probabilistic approach to record matching as fields like age and birthplace are unlikely to be in the record at all. As a result, the sacramental records within the DADP did not lend themselves to being matched with a probabilistic method.

One thing worth noting is how the DADP's sacramental data differs from census data. Census data often contains fixed information about a person such as their birthdate and birthplace, while sacramental data, on the other hand, contains information about where a person was at a given time. This difference creates some important distinctions for how sacramental data should be matched.

2.3 Overview and Purpose of Proposed Method

The method proposed in this paper utilizes the relational data in the DADP to strengthen the certainty of matching two person records with a common name. In essence, this method aggregates person records with a common name then builds and compares a network graph of a person's relations. One beneficial effect of using network graphs in our matching algorithm is that it also provides a useful means for visualizing an individual's relationships. It's important to note that a large part of this matching algorithm's purpose was to identify relationships between people that were not immediately apparent before. And, as such, it is unlikely that this algorithm would be useful in some other contexts such as using record matching to study a population's economic mobility.

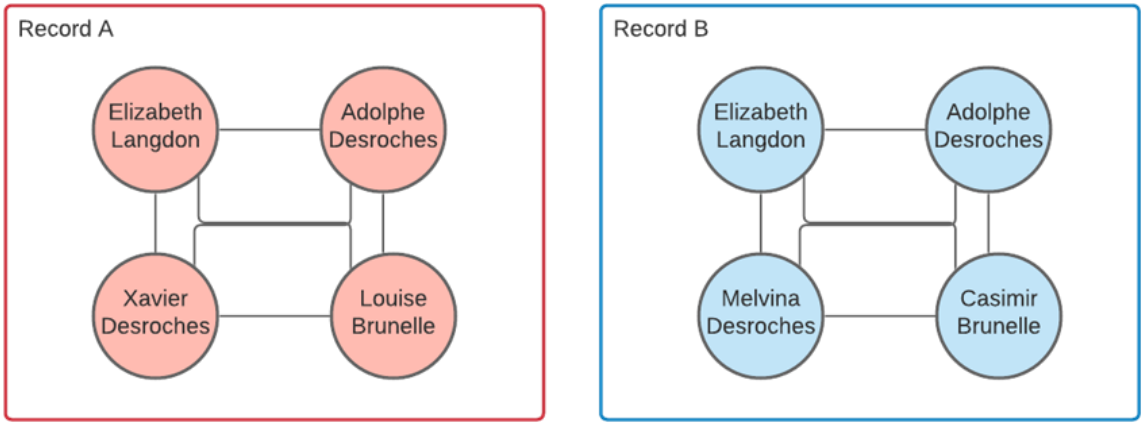
3. Proposed Method

3.1 Assumptions

All record matching algorithms must make certain assumptions about their data in order to differentiate matching records from nonmatching records. In statistical approaches, the core assumption is that matching records and nonmatching records will cluster based on the degree of similarity between their fields [3]. With this data, statistical methods can produce a likelihood that two records are a match, then the researcher can set a threshold for what likelihood is and is not a match [4]. In contrast, the approach outlined in this paper determines that two records are a match if their fields pass all the matching criteria. Although the ability to assign a likelihood to a pair of records is lost, this method is capable of producing quality matches without fields like birthdate and birthplace.

The key assumption this matching technique makes is that relational data is a strong indicator of whether two records are match. The reason why this assumption works to increase the quality of matches is that it is quite common for two family members to appear in sacramental records together. **Figure 2** illustrates an example of this that commonly occurs in marriage records. In short, relational dyads like a husband and wife will commonly be found in multiple sacramental records together. The result of this is that relational data can be used in conjunction with fields like first name and last name to make higher quality matches than before.

Figure 2



The figure above illustrates two simplified marriage records which occur in the DADP. The nodes on the graph represent a person contained within the marriage record while the edges indicate the connection between them. In the two records, Adolphe Desroches and Elizabeth Langdon appear in the marriage records and their roles in both indicate that they are a husband and wife. Because familial dyads like this occur regularly in sacramental data, it can be used as a strong indicator that the two records are in fact a match.

3.2 Choosing Records to Match

The most critical step in a matching algorithm is determining which records are possible duplicates [5]. This is commonly done by indexing the data by a set of fields, normally name and location [6]. The purpose of this is to reduce run time complexity and reduce the number of necessary comparisons. It thus prevents obviously not matching records from being grouped together and making unnecessary comparisons.

When working with more limited data, it is useful to index by a concatenation of first and last name as to ensure the highest degree of similarity in matching possible. One side effect of indexing by first and last name is that it results in people with name changes from not being matched. Someone anglicizing a name like François to Francis, for instance, would prevent their two records from being deemed a match even if all their other features match perfectly. This also commonly occurs with women that adopt their husband's surname. Although similarity scores between names could be calculated to combat part of this, doing so would likely result in an increase in the number of false positive matches being made.

Before moving to comparing the records with matching indexes, it is useful to sort them by the date the record was recorded followed by any additional sorting conditions. Sorting like this can be useful in handling edge cases that arise such as comparing a father and son with a shared name.

3.3 Comparing and Matching Records

The basic idea of the matching process is to compare and group all records with a common index into sets of matching records. This is done by comparing a pair of records fields for whether they are referring to the same person, and, if they are, group them into a set of matching records. This set of matching records gets added every time one of the records within the set matches with another record outside of it. One thing of note is how this set can match two records that do not meet all the matching criteria in isolation but do meet all the criteria when considering the other records in the set. What can happen because of this is one record can act as a link between two records that would not have otherwise matched. For example, given three records A, B, and C, such that A matches with B, B matches with C, but C does not match with A, the fact that B matches with both A and C will result in A and C being in the same set of matching records despite the fact that they would not match if B was not present. **Figure 3** provides an example of how this happens with real data and produces a desired result.

Comparing fields is the basis for determining whether two records are a match. As such, it is worth exploring how fields should be compared and what results from the comparisons.

Figure 3

First Name	Last Name	Date Recorded	Location	Relations	Role
Adolphe	Desroches	1891/10/01	Tiny	Elizabeth Langdon (wife)	Father
Adolphe	Desroches	1894/10/10	Tiny	Elizabeth Langdon (wife)	Father
Adolphe	Desroches	1896/02/05	Tiny	Elizabeth Langdon (wife)	Father

The figure above illustrates three records of Adolphe Desroches that all refer to one individual. Given that records are only considered a match if they appear within 5 years of each other, the record on the top row will not match with the record on the bottom row. Meanwhile, the records on the top row and bottom will both match with the middle row since the middle row occurs within the allotted time frame for both records. Because the middle row exists, all three of the records will be added into the same set of matching records. This illustrates how the existence of one record can create a link to several other records.

3.3.1 Comparing Names

As one would expect, names are the most critical components in record matching, especially in sacramental data. This is because a first name and last name combination are the closest thing a person has to a unique identifier. For this reason, the index used in matching the DADP was a concatenation of a person's first name and last name in all lowercase. Although other indexing options could be viable such as using a phonetic algorithm like Soundex to create indexes, the benefit of using a fuzzy matching algorithm did not seem worth the potential errors that could be introduced by such indexing [7]. The reason for this is two records that do not match perfectly on a first name and last name can be considered enough uncertainty to validly waive comparison.

If indexes were not used as the means of comparing two records names, then it is worth considering the ways they could be compared. Requiring a strict equality between the two names, for instance, would create the same result as indexing them based upon their names. To create matches of names with a degree of similarity, a string metric like the Levenshtein edit distance or the Jaro-Winkler distance should be considered [8]. These would require names to meet a minimum similarity to be considered a match.

3.3.2 Comparing Locations

Locations, like names, are prone to some differences in how they are recorded. One church, for instance, can be referred to by different names such as the Rapids Church also being referred to as the Grand Rapids Church. This make comparison between two locations slightly more complicated. One way to combat this is to first check if one string is a substring of another. This would resolve the issue illustrated in Grand Rapids Church

vs Rapids Church example. While a simple check for whether one record is a substring of another can resolve most name situations, it can be worthwhile to use a string metric for comparing location names. The Damerau-Levenshtein distance is especially useful for comparing location names as it tolerates spelling errors that occur in long strings [9].

Since sacramental data normally contains data on where the record was recorded, such as what church a marriage took place in, and not a person's birthplace, it is necessary to only match individuals that have appeared in the same location. The reason why can be seen in the records of John Setter. John Setter appeared in multiple baptismal records at Rapids Church and Beaver Creek. Even though it is possible that all of these records are referring to the same John Setter, it is presumably more likely that a John Setter record from Rapids Church and a John Setter record from Beaver Creek are not a true match than two John Setter records both from the Rapids Church. However, what is there to say that there are not two different John Setters going to Rapids Church? Ultimately, an answer to that cannot be discerned from location comparisons alone and is what requires a comparison in both time and the relations that exist in the record.

3.3.3 Comparing Time

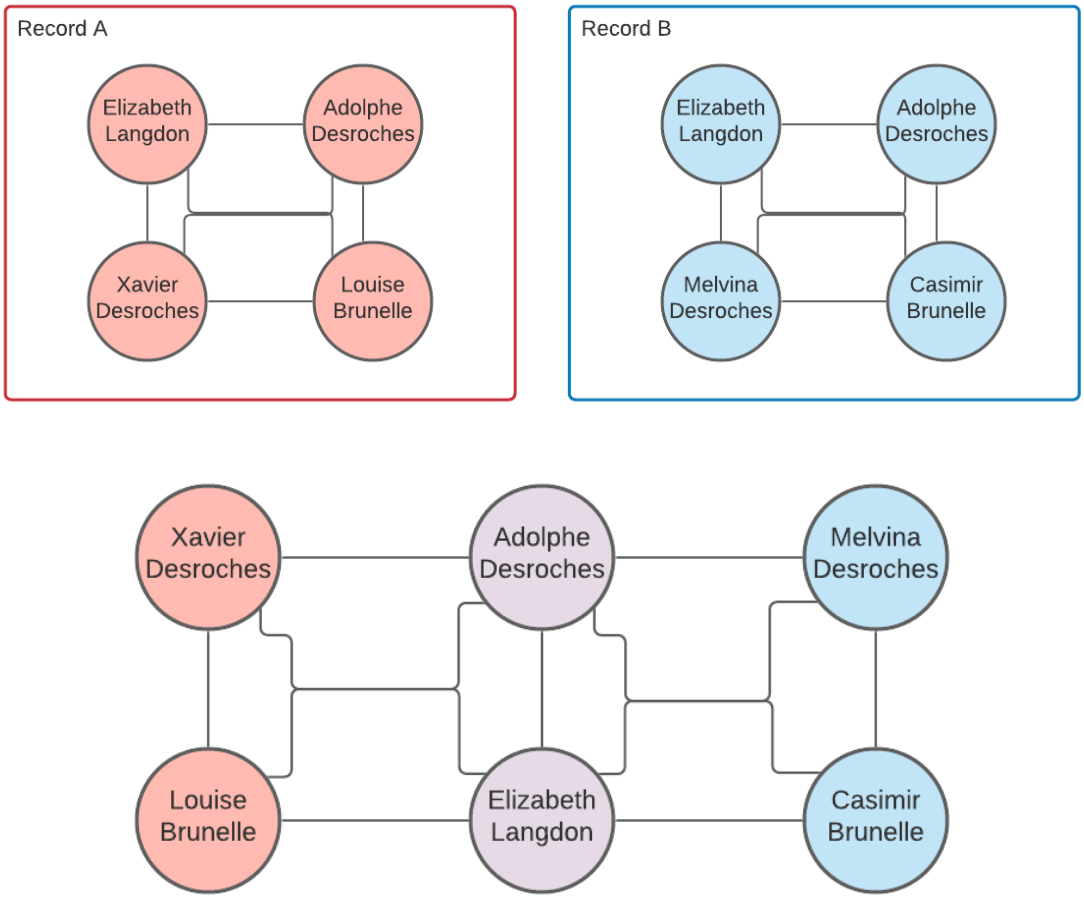
Although age or birthdate are not commonly available in sacramental data, time still factors into whether a comparison between two records is or is not a match. Let's return to the Rapids Church's records of John Setter to illustrate how time is relevant. Given that John Setter occurs in several Rapids Church baptismal records, the dates the records were recorded can act as an indication for whether two records are matching. Since the records occur in 1847, 1848, and 1870, it is presumably more likely that the two records that are recorded within a one-year time span of one another are a match than a record recorded twenty years later. Because of this, one must decide how many years difference there can be between two records to still be considered a match. In matching the sacramental data in the DADP, two records meet the matching criteria if they occur within a five-year time span of one another. This number was chosen through testing different values to get what worked best in yielding the most quality matches. A lower number will likely lead to less matches being made. It's also worth returning to **Figure 3** as it illustrates how two records that do not meet the time span matching criteria in isolation can still be placed into the same set of matches through one or more linking records.

3.3.4 Comparing Relational Data

Utilizing any relational data available in the records is key to improving the quality of matches for sacramental data. In the DADP, familial dyads like a husband and wife frequently occur together across several historical records and are an important way that people are distinguishable from one another. The easiest way to compare relations is to treat all the people that occur within a record as nodes on a graph. A set intersection on the nodes of each graph reveals if the two records contain more than one probable match. Since false matches do not normally share more than one node on a graph, it is safe to assume that a familial relationship is present when a set intersection yields two or more commonalities. **Figure 4** provides an illustration why a match may be made because of this.

Once it is determined that there are at least two probable matches within the pair of records, it is important to ensure that the relations present are valid. Something that occurred frequently in the DADP's marriage and baptismal records was a father and son with a common name get incorrectly matched due to their common relations. These erroneous matches were avoided by comparing the roles of a person in one record to another. Someone with the role of a husband record, for instance, should never match with someone that has a father role in a record that appears just a few years later. The same can be applied for death records as someone buried in 1870 should never match with anyone after 1870. Each database will likely have its own peculiarities and need to be resolved in their own way.

Figure 4



The above figure illustrates how the two records illustrated in **Figure 2** are matched based on their common nodes. When a set intersection of Record A and Record B is performed, Adolphe Desroches and his wife, Elizabeth Langdon, are shown to appear in the two records together. Performing a set intersection on the indices of people's records will often yield person records that have a kind of familial relation. In the case illustrated with Adolphe and Elizabeth, this relation is a husband and wife.

3.3.5 Grouping Matching Records into Sets

As mentioned before, once a record meets all the criteria described above, the pair of records should be stored into a set together. Whenever a new record matches with either of the records within this set, this new record is added to this set. The strictness of the matching criteria works to prevent one set from erroneously accumulating false matches.

3.4 Strengths, Limitations, and Potential Hurdles

The strengths of this approach to record matching is that it is highly flexible and can be made to work with a wide variety of data. Slight altercations can be easily appended to the function used for checking two records validity. This allows additional fields like name frequency to be used in record comparisons.

However, there are some limitations to this approach. In cases where there are lots of people in a community that share a common name, it can be useful to add even stricter measures as to what can be considered a match. To illustrate one potential hurdle, consider the problem of matching a father with their son when they share a name. For example, three records of a Joseph Allery all originate from the same marriage record but each with different roles: first witness, husband's father, and husband. Some measure must be taken in order to prevent matching the father and witness with the Joseph Allery being married, as they are clearly not referring to the same person despite having all of the common details. One suggestion for this scenario is to create a hierarchy of roles that are most likely to least likely to have true matches in the data. In addition, the matching algorithm for marriage records should have additional checks being made such that someone who is getting married is never matched with someone who is a father or mother. Ultimately, these issues will depend on the kinds of data that is being matched and solutions will have to be tailored where appropriate.

4. Pseudocode Implementation

Pseudocode with a python-like syntax illustrates the basic implementation of the matching process. The main process outlines the general structure of the matching algorithm. The comparison and grouping function illustrates how the comparisons of records within an index group can be structured.

4.1 Main Process

The main process for matching runs a data pipeline where data is first preprocessed and has unmatchable data filtered out. Next records are grouped by their index and sorted. This is followed by the comparison and grouping of each of these index groups into sets of matching records. Each set of records is deemed to refer to one person.

```
# Preprocessing Step
# Load a person and standardize names, dates, and other fields
# Filter out records that lack enough data to be matched
person_records = load_and_process(data)
# Indexing Records for Comparison
record_groups = defaultdict(list)
for record in person_records:
    index = concatenate(record.first_name, record.last_name)
    record_groups[index].append(record)
for index in record_groups:
    # Sort groups by date and other relevant data like role
    record_groups[index] = sort(record_groups, by=sorting_condition)
# Comparing and Grouping
matching_record_sets = []
for index in record_groups:
   matching_record_sets.extend(compare_and_group(record_groups[index])
```

4.2 Comparison and Grouping Step

The comparison and grouping step iterates through all of the records grouped by a common index. It adds records that meet the matching criteria into a set of records which are assumed to refer to one individual.

```
def compare_and_group(records):
    # Check if there are no records to match
    if len(records) == 1:
        return [set(records[0])]
    # list to hold sets of matching records
    matching_record_sets = []
    for i, record in enumerate(records):
     # i=index of the record in list, record=person record
     # Check to make sure the record has not already been matched
     if not is_matched(record, matching_record_sets):
         # Create a new set for unmatched record
         current_set = set(record)
         for other_record in records[i:records.length]:
             # compare fields, then group if match
             if not is_matched(other_record, matching_record_sets)
             and is_valid_match(current_set, other_record):
                 current_set.add(other_record)
         # Append set of matching records to
         matching_record_sets.append(current_set)
    # Return list of matching record sets
    return matching_record_sets
```

5. Results

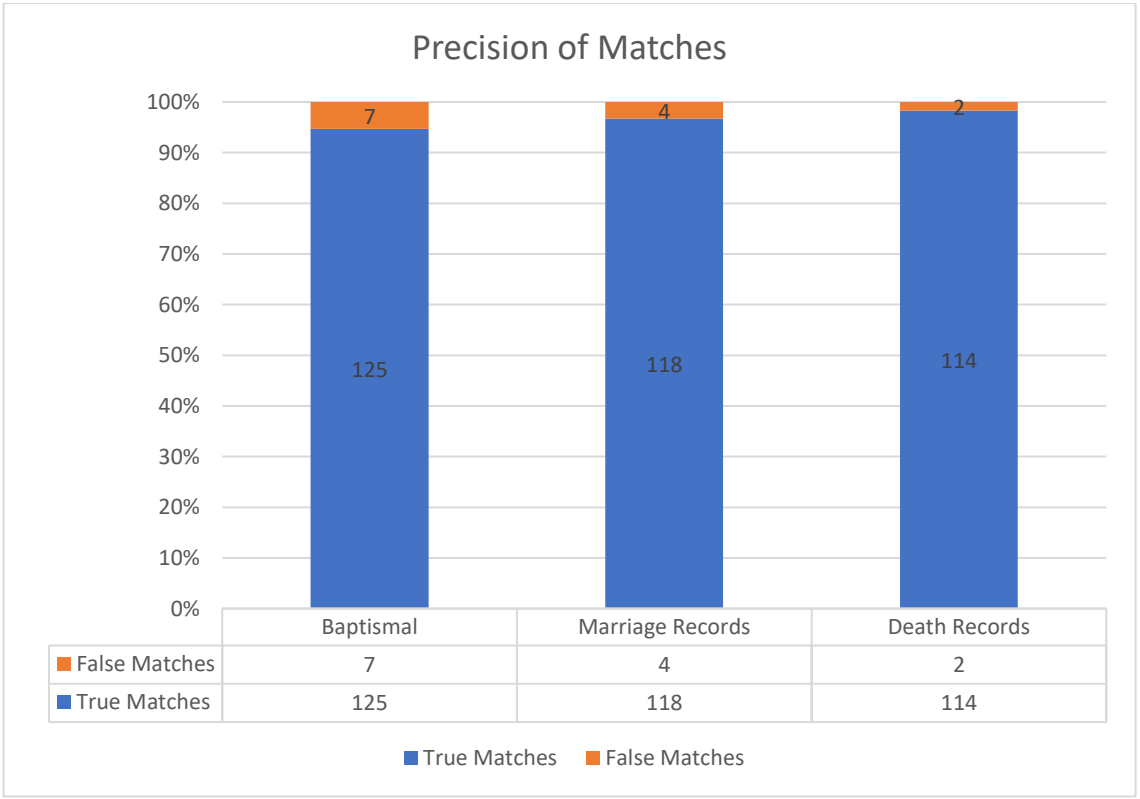
The algorithm was tested on three tables in the sacramental database of the DADP. Due to the sets of true matches and true nonmatches being an unknown, using common metrics for measuring accuracy such as F-score and Recall were not possible. Therefore, it was necessary to use alternative methods for determining the effectiveness of the algorithm. A portion of the results for each table was checked by hand for whether a set of records were a true match or a false match. See **Figure 5** for more details.

It is important to note that slight adjustments to the algorithm were made for matching each individual table, but the overall structure of the algorithm remained the same. These adjustments were, in a sense, tuning the algorithm to avoid table specific pitfalls.

The most common error made by the algorithm in testing was erroneously matching family members, specifically fathers and their sons given that they share a name. In order to avoid this kind of false positive from occurring regularly, measures were put in place to minimize the chances of this happening.

To note how much relational data can affect the number of matching sets, the algorithm was ran with and without the comparisons of relational data. In the case of baptismal records, the number of matching record sets created without the relational data was 6,191, while there the number of matching record sets created with the comparison was 3,837. Although the precision of comparing without relational data is not known, comparing the relational data presumably led to less but higher quality matches.

Figure 5



The above figure illustrates the precision of the algorithm for each type of data the algorithm was tested on. The records were checked by hand to identify true matches vs false matches.

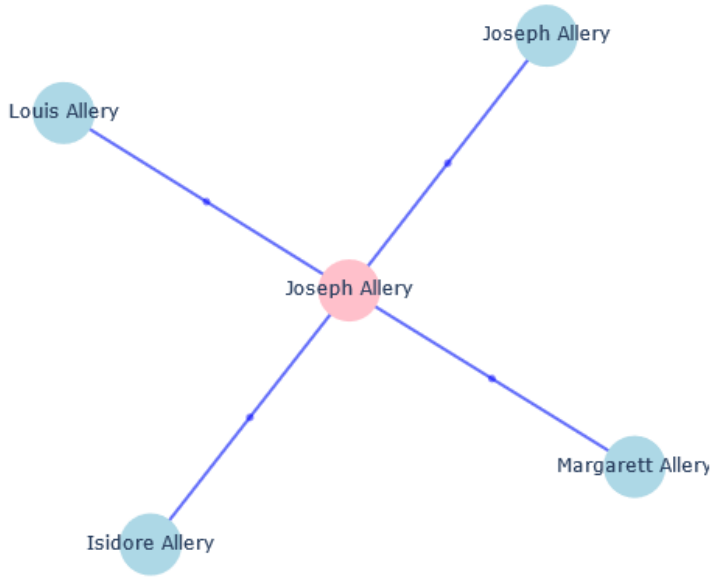
6. Graphing Relationships of Matching Records

One benefit of using network graphs in the construction of a match is that it provides an easy way to visualize a person's relationships across historical records. Since Joseph Allery appears as the father of multiple people in census records, it is easy to create a plot of all the people he is linked to.

These kinds of graphs are useful for discovering links between historical figures that might not have been immediately evident otherwise. As illustrated in **Figure 6**, the relationships between members of the Allery family are evident in the visualization but may not be physically linked in the data.

This kind of interactive graph enables users of the DADP's archives to explore the records within it and find information that they might not have been aware of. Users can click on a node on the graph and be taken to its corresponding page on the DADP. Users can also see additional information by hovering over a node or edge.

Figure 6



The above figure illustrates a graph (created using plotly.js) of the Allery family from census data. Joseph Allery (in red) is connected to each of the other nodes because he is listed as their father in the census record. Even though the relationships between Joseph Allery's children do not have physical links in the database (illustrated by their lack of edges connecting them), the visualization would allow a viewer to infer or make inquiries about their relations.

7. Conclusion

This paper explored how the relational information contained within sacramental data of the Digital Archive Database Project could be used to link individuals. The algorithm created resulted in high quality matches being made within each type of sacramental data tested: baptismal records, marriage records, and death records. Since the DADP does not have a known set of true matches, common statistics in data matching were not applicable. As a result, the precision of the matches made was the only statistic that could be used in assessing the algorithm. Once the records were matched, a network graph visualization was made to illustrate how links in historical data could be visualized.

One area of future research is to explore how relational data can be used to match records across different contexts. Because matching algorithm was only tested with how effectively it could match records of the same kind, it is unknown how applicable it is to linking records from say a baptismal record to a marriage record. Two things that would be important to consider in such matching is how location and time should be compared.

References

[1] S. Ruggles, C. A. Fitch and E. Roberts. "Historical Census Record Linkage," *Annual Review of Sociology* Vol. 44:19-37, July 2018.

[2] W. E. Winkler. "Matching and Record Linkage," *Wiley Interdisciplinary Review: Computational Statistics* 6 no. 5, 2014: pp. 313-325.

[3] R. Abramitzky, R. Mill and S. Perez. "Linking Individuals Across Historical Sources: a Fully Automated Approach," February 2018.

[4] R. Abramitzky, R. Mill and S. Perez. "Linking Individuals Across Historical Sources: a Fully Automated Approach," February 2018.

[5] P. Christen. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*, Singer Science & Business Media, 4 July 2012.

[6] Y. Sunitha and K. Lakshamaiah. "Investigation of Techniques For Efficient & Accurate Indexing for Scalable Record Linkage & Deduplication," 2012.

[7] H. David and M. C. McCabe. "Improving precision and recall for soundex retrieval," *Proceedings. International Conference on Information Technology: Coding and Computing*, IEEE April 2012: pp. 22-26.

[8] P. Christen. "A Comparison of Personal Names Matching: Techniques and Practical Issues" December 2006.

[9] F. Damerau. "A technique for computing detection and correction of spelling errors" 1964.