MASTER

Modelling, Computing, and Visualizing
Uncertainty in Pedigrees from Genealogical Datasets

Feng, Z.

Award date:
2020

Link to publication
Modelling, Computing, and Visualizing

Uncertainty in Pedigrees from Genealogical Datasets

Student Name: Zirui Feng
Student Number: 0947135
Supervisor: Michael Burch

Msc. of Data Science in Engineering
Algorithm and Visualization group

August 18, 2020
Abstract

The document is designed to show the implementation process of the project Modelling, Computing, and Visualizing Uncertainty in Pedigrees from Genealogical Datasets. It is aimed to create a user interactable application for pedigrees data visualization.

Genealogy, the establishment of a pedigree, is the investigation and research of the family, family history, and past footprints. In the past, genealogy often used oral interviews, genetic analyses, historical records, and other queryable information to collect information about the family, and to prove the family lineage and various kinship relationships. Due to the complexity and difficulty of information collection, historically, genealogy most times only served for large or noble families. In contemporary society, genealogy has become more ubiquitous, and ordinary people have begun to research and compile genealogies for their families just like the previous nobles. This is because of the emergence of the Internet, the first-hand resources available to genealogy scholars have increased explosively, which has also led to an explosive increase in the focus of this topic. These media allow curious people to investigate and study their ancestors. This curiosity is especially acute among those whose genealogy is lost, or whose separation of flesh and blood resulted from adoption and bereavement. The Internet is already the main place for genealogy scholars to obtain resources.

The original demographic information datasets are provided by the Dutch organization BHIC (Brabants Historisch Informatie Centrum). The BHIC is the memory of the province of Noord-Brabant. It is the centre of the history records of Noord-Brabant. Demographic and civil registration information such as births, marriages, deaths, and criminal of the local population is the focus of BHIC records. This report describes how to find people with possible kinship relationships from vast and complicated original databases, and to establish the genealogical database. Afterward, through a straightforward graphical user interface and data visualization software, it provides more ordinary users with a way to query and observe their genealogical information.

Keywords — Genealogical data, Pedigrees, Visualization.
Summary

The BHIC maintains many different types of records related to demographic information. In this project, the civil registration information is mainly used. They are civil death registration records, civil marriage records and civil birth registration records. This report describes the design and application of the methods for helping users to query and construct genealogical data and the genealogy from BHIC's existing data information. According to layer-by-layer selection and comparison of persons’ names, birthplaces, birth dates, and essential event registration time, similar people can be found from the vast population database, and possible kinship can be judged. Because most of the data stored by BHIC is recorded independently as events, and most citizens only have very few identifiable information, such as name and gender. Therefore, the most reasonable solution is to introduce the level of CERTAINTY as an important factor when detecting the relative relationship between two citizens through these data resources. Based on the degree of information matching in the comparison process, the credibility of the kinship relationship between any two people can be graded.

Because the amount of data stored in the original database is enormous, then algorithmic and visual support is important in order to help users save time when searching related genealogies. The method adopted in this project is first to query and save all possible kinship relationships from the original data resources. In this way, the user can directly generate the genealogical pedigree through all the storage relationships of the relevant people. The disadvantage of this approach is that the data processing and retrieval for all possible kinship relationships takes a lot of computing time.

Meanwhile, the limitation of the project is that all the family relationship data are excavated from BHIC’s original database. The time period of these data is around 140 years from the 19th century to the early 20th century, and the area is limited to Noord-Brabant. Thus, the genealogical information that can be queried for most families is very limited. If users need to discover a more complete genealogy and learn more about family history, then they need to combine more historical demographic information.
Contents

Abstract 1

summary 3

1 Introduction 2

1.1 Background Introduction 2

1.2 Project Introduction 2

2 Project Analysis 4

2.1 Background Analysis 4

2.1.1 Data Collection and Processing 4

2.1.2 Kinship Relationship Determination 4

2.1.3 Pedigree Visualization and Construction 5

2.2 Research Questions 5

3 Related Work 6

3.1 Genealogy Creating 6

3.2 Genealogy Visualization 7

3.2.1 Tree Structure Visualization 8

3.2.2 Radial Layout 10

3.2.3 Time Concentration Layout 11
4  Solution and Implementation  

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1  Preliminary Data Preparation</td>
<td>14</td>
</tr>
<tr>
<td>4.1.1  Data Overview</td>
<td>14</td>
</tr>
<tr>
<td>4.1.2  Data Model</td>
<td>15</td>
</tr>
<tr>
<td>4.2  Data Selection</td>
<td>21</td>
</tr>
<tr>
<td>4.3  Genealogical Pedigree Construction</td>
<td>22</td>
</tr>
<tr>
<td>4.3.1  Foundation of Genealogical Data Processing</td>
<td>22</td>
</tr>
<tr>
<td>4.3.2  Genealogical Pedigree Construct</td>
<td>24</td>
</tr>
<tr>
<td>4.3.3  Certainty Level Coefficient</td>
<td>25</td>
</tr>
<tr>
<td>4.3.4  Relationship Determination Model</td>
<td>26</td>
</tr>
<tr>
<td>4.4  Genealogical Pedigree Visualization</td>
<td>38</td>
</tr>
<tr>
<td>4.5  User Interaction</td>
<td>39</td>
</tr>
<tr>
<td>4.6  Implementation</td>
<td>40</td>
</tr>
<tr>
<td>4.6.1  Preliminary Data Preparation</td>
<td>41</td>
</tr>
<tr>
<td>4.6.2  Genealogical Data Processing</td>
<td>42</td>
</tr>
<tr>
<td>4.6.3  User Interface With Genealogical Pedigree Visualization</td>
<td>42</td>
</tr>
<tr>
<td>4.6.4  Result Example</td>
<td>43</td>
</tr>
</tbody>
</table>

5  Result Discussion  

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1  Answer of Research Questions</td>
<td>46</td>
</tr>
<tr>
<td>5.2  Limitation</td>
<td>48</td>
</tr>
<tr>
<td>5.3  Future Work</td>
<td>50</td>
</tr>
</tbody>
</table>

6  Conclusion  

Bibliography  

A  Examples of genealogical pedigrees
Chapter 1

Introduction

This chapter mainly introduces the background and basic situation of the project establishment.

1.1 Background Introduction

BHIC (Brabants Historisch Informatie Centrum) is the memory of the province of Noord-Brabant. It is the centre of the history records of Noord-Brabant. They collect huge amounts of demographic information for Noord-Brabant. Along with the rapid development of the Internet and information technology, BHIC chose to provide their data on the Internet as an open resource for people to share. Everyone can download original resources they provide on the Internet, or directly visit their website for some data queries. Many ordinary people hope to find the history of their family or ancestry through BHIC. More and more scholars and researchers hope to obtain more academic information about the historical changes in the population of Noord-Brabant, such as population growth trends, average life expectancy, and male to female ratio, by studying the BHIC datasets. All in all, for contemporary society, the data held by BHIC has an excellent discovery and exploration value in helping to study the development history of Noord-Brabant.

1.2 Project Introduction

The genealogical data study is the research of families, family history, and the tracing of their lineages. Typically, lots of records about people with their names, to some extent, their relationships, places they lived, when they are born, when they died, and many more information are stored in certain databases accessible online. As it is mentioned above, the BHIC (Brabants Historisch Informatie Centrum) collects all of the available data of people living in the Noord-Brabant region to provide a way to search for relatives and their simple relationships. However, the database has become that big in these days, reaching more than 18 million records in its
simplest form, that filter options and visualizations are required to find insights in the data, in particular, to create genealogical pedigrees and observe uncertainties in the data.

The Eindhoven University of Technology is always at the forefront of data science development, and is committed to using data science technology to help people solve the difficulties encountered in life. Thus, Michael Burch, the teacher of the TU/e algorithm and data visualization group, initiated this project. At present, they have not done any depth research on the BHIC original data. Meanwhile, for them, the research of genealogy and kinship is also a relatively unfamiliar direction. Therefore, the primary purpose of this master graduation project is to expect the student to understand the BHIC data resource, then model, compute, and visualize possible genealogical data from the resource, as well as indicate the uncertainty level of the relationships between the characters in genealogical pedigrees. The basic requirements of the project are:

- Exploring and understanding the provided data resources from BHIC for building a data model (because the provided dataset is very primitive, which contains a lot of messy and useless parts. It needs to use data mining to get really valuable parts.)
- Providing an overview about the names and relations as one or several pedigrees.
- Finding and computing a suitable model for uncertainty of names, relations, geography, dates etc. (for example, using an algorithm or calculation formula to determine the certainty of the relationship between two or more members in the pedigrees according to existing data.)
- Designing a graphical user interface containing the required views to explore the datasets and pedigrees.
- Interacting with the visualization and linking the views.
- The solution is preferred to be implemented in the programming language Python.
Chapter 2

Project Analysis

This chapter mainly introduces the analysis of the project’s basic situation and the determination of the investigation direction.

2.1 Background Analysis

In general, the problems of the project are mainly composed of three aspects. The first is raw data collection and processing. The second is the search and confirmation of the relatives of the people, finally, the establishment and display of the personal relationship pedigree. Therefore, the main problem of the project can be refined into these three aspects to help confirming the research problem and direction.

2.1.1 Data Collection and Processing

Most data stored in BHIC is classified according to the type of event, such as the birth records, death records, criminal records, and marriage registration of citizens. The original data is recorded and stored by various government administrative departments of Noord-Brabant. The original data was initially recorded on paper, and then BHIC registered it as an electronic version and uploaded it. The data resource download format provided by BHIC is an XML format, and the size of these datasets is very large. Moreover, the data in the XML file is not directly and simply observable. Thus, the direction that needs to be determined first is which types of data belong to the range of genealogical data. Then, the selected dataset ranges can be narrowed depending on the requirements.

2.1.2 Kinship Relationship Determination

Afterward, the problem needs to be confirmed, which is what are the essential roles in a genealogical pedigree and what are the most fundamental relationships between these people.
In addition, after grasping the basic information such as name, gender, and nationality of the two unrelated persons, the question comes up if it is possible to determine whether there is a relative relationship between them? In other words, if we want to find out whether there may be a kinship relationship between two people, what information can provide assistance, then according to the needs to filter out specific information from the datasets. Finally, establishing the calculation model, and the relationship between the people is calculated using specific information as input features. Also, according to the different computing results, the credibility levels of relationships are classified.

2.1.3 Pedigree Visualization and Construction

Moreover, when relationships between persons are confirmed, it is necessary to select an appropriate data structure to store it and display it in the appropriate chart form. Therefore, the clear question is how to select the data structure when creating and storing the genealogical pedigree. Meanwhile, how to make it easier for users to find and observe different roles in the family tree, relationships between characters and the different reliability of the relationships, are also aspects that cannot be ignored. Thus, choosing a suitable data visualization model is also a very important research direction.

2.2 Research Questions

Through the analysis of the necessary background situation, the main research directions can be determined as the selection of related datasets of the genealogy, genealogical information retrieval and data mining, possible kinship searching, and uncertainty measurement. Therefore, the research questions are:

- What kind of data does BHIC provide?
- How to visualize the genealogy?
- What are the necessary information to establish the genealogical pedigrees?
- How to retrieve genealogical data from a huge database?
- How to set up a computing model to measure the uncertainty of the relationships between people based on people’s basic information?
Chapter 3

Related Work

This chapter mainly introduces the investigation of knowledge and resources related to the project.

3.1 Genealogy Creating

According to Philip Kretsedema’s research result [Kre17], there are two different ways of using the term ”genealogy”. One definition refers to the search for the development history of the entire family, which includes the study of all people in the entire family who have blood relations. For this kind of situation, genealogists have their own methods of research. They first start by obtaining information about their immediate family members, which may be easier to obtain before delving into the history of future generations. After that, they rely on the knowledge of immediate family members to discover more family members. Repeating the above steps leads to a completion of the entire genealogy. Another way to describe a genealogy is to process people from a specific ancestor. For example, a genealogy researcher can compile a list of all descendants of his great-grandfather. It will make the list of the genealogy of that particular ancestor.

Based on the introduction of genealogist Ann Lawthers [Law16], the following steps are required to create a genealogy. The first step is to determine what you know. It means that the information you have known will be used as the starting point for creating a genealogy, for example, the personal information of an ancestor, or all the people you know who have kinship relationship with you. The second part is to determine what you want to know, which is to determine the research goal. For example, you need to specify whether you want to make a genealogy for all descendants of a single ancestor, or it is for all people with relatives. Thirdly, identify and find resources that can help you. There are many different ways to find genealogical information such as the Internet, a library, record files saved by your family, and so on. The last step is to investigate, analyze, and summarize the collected data. The establishment of genealogy is to find the relationship between people. The analysis of the collected data is to find
more people who may have kinship relationships with you or relatives you have known. The
genealogy is the combination of all the kinship information and the people in these relationships.

The most critical step in creating a genealogy is to collect all possible kinship information. According to the opinion of Ann Lawthers, the family can be used as the basis to store the kinship relationship data. That is everyone’s immediate family relationship, which are parents with children, brothers, and sisters. After that, all related families can be combined to form a complete genealogy. Collecting immediate family relationships is much simpler than querying overly complicated kinship relationships or people who are many generations away. If there are no record documents of family history, then each person’s direct family relationships can be searched through the following sources.

- Civil records created by the government (i.e. birth certificate, death certificate)
- Records created by religious institutions
- Census
- Immigration records
- Naturalization and citizenship records
- Land and property records
- Probate
- Military service records

In short, any record or file containing personal historical information is a valuable resource. These fragmented event records can be used to concatenate a person’s life experience. In these records, it must contain valuable kinship information, such as a person’s parents, children, spouses, and so on. Using this information can help to generate a person’s immediate family relationship. In addition, some important temporal details can also be used as part of the genealogy. For instance, the same persons are classified according to their birth years. Or in another situation, a person from a family might have been immigrated to a new area at a certain age and reproduced a branch of the family in there, then the genealogy can distinguish the persons according to the year that person left.

3.2 Genealogy Visualization

The study of genealogy is because people tend to be interested in their origins to understand the family and their position on a larger scale. Today, with the increase in the amount of available genealogical data and the rapid development of visualization technology, visual information can help people to receive and understand data faster and more directly.
3.2.1 Tree Structure Visualization

A family tree is the most common way to visualize genealogy data. At the same time [BBS13], the tree is also the most classic structure for displaying hierarchical data structures. The tree is widely used in the visualization of different hierarchical datasets because it can fully highlight the three characteristics of a hierarchical dataset: structural information as a hierarchical structure, content information of each section, and connection information of each node. In short, according to Herman and M.S. Marshal [HMM00], the primary tree drawing task is to calculate the position of each node in the set and draw edges to represent the relationship between them. There are many different layouts for tree visualization according to the complexity of the data structure. Sometimes even for the same data, different layout structures can provide different meanings.

The simplest and most classic tree is the rooted tree. It is called a rooted tree because it looks like an upside down tree, which means that it has roots facing up and leaves facing down. In this structure, each node has only a limited number of child nodes or no child nodes at all, each node has only one parent node, and the node without the parent node is the root node [GT09]. A binary tree is a rooted tree data structure in which each node has at most two child nodes, called the left child node and right child node. The branches of a binary tree have a left-right order and cannot be reversed at will. Figures 3.1a and 3.1b show examples of these two types of a tree, respectively. The biggest specificity when using it to represent genealogical data is that the nodes representing people of the same generation are aligned, so it can well separate generations.

![Figure 3.1: The classical tree structure](image)

But its shortcomings are also prominent. Because of the unidirectional structure and the restriction to a child node number when expressing genealogical data, the number of people that can be included in each generation would be limited, for instance, as it shows with Figure 3.2 which is a genealogical tree applying the binary tree layout. It transforms the vertical structure of the binary tree into a horizontal structure, and the progressive direction of time is from leaves to the root. Every two child nodes represent parents, and their root node is the children. The apparent limitation of this structure is that for the same pair of parents, only one of their children can be shown, but no other sisters of brothers. This kind of binary tree genealogy is only suitable for tracing the origin of a specific object’s ancestor. Thus, this type of genealogy visualization structure is also called ancestor tree [MB05a].

Besides, a prevalent structure is the hourglass tree, which was developed from the binary tree.
3.2. Genealogy Visualization

Figure 3.2: An example for using a binary tree for representing genealogical data [pg20]

The characteristic of this structure is that child nodes can be generated from the two opposite directions of the root. As shown in Figure 3.3a, the node $x$ is the root of the tree. According to the hierarchical structure, the child nodes in the upper half of the $x$ node represent his ancestors, and the nodes in the lower half are his descendants. However, according to Michael J. McGu et al. [MB05b], if it is required to show more characters in the family tree, the task is very challenging. For example, in Figure 3.3b, it also displays the descendants of $y$, or worse still, the descendants of each ancestor of $x$, and the ancestors of each descendant of $x$. Because for the hierarchical tree structure, avoiding the intersection of edges and not using curves are important factors to ensure that the entire structure is bright and beautiful. Therefore, if the application of aesthetic rules as a benchmark, then some simple tasks will become complicated when they are executed. Thus, when using an hourglass tree to show a genealogy with a large population, how to reasonably set node position distributions to meet the aesthetic criteria is the most significant difficulty.

Figure 3.3: The hourglass tree structure [MB05b]
3.2.2 Radial Layout

As mentioned above, it is necessary to avoid the intersection of edges in order to meet aesthetic criteria in a tree of vertical or horizontal situation. Although the layout of the classic tree is easy to understand, this layout cannot effectively use the available space because more than 50% of the pixels are used as the background [DR08]. Therefore, the radial layout method can meet the requirements of saving space.

The first and most basic is the radial tree layout. It is an improvement to the traditional tree layout or a variant of the node edge tree representation. Different from the conventional tree layout, the radial layout does not use the vertical construction method, which means the child nodes are not located above or below the parent node but on an outer circular trajectory radiating from the root node. So the nodes in this structure are all located on concentric circles according to their hierarchy. As shown in Figure 3.4a, its most important advantage is the reasonable allocation and utilization of space. For a tree structure, if each node has more than one child node, the number of nodes at each depth increases in proportion to the bottom. That is to say, there will be more nodes at each lower level than the previous level. However, in a radial layout tree, since the perimeter of each concentric circle increases with an increasing radius, there are more locations available for placing nodes. This layout makes the space more efficiently than the classic tree layout. In addition, the use of a radial tree layout is more convenient to express the chronology in the genealogical data.

Keller [KRS10] proposed a method for encoding historical or temporal information in the radial tree. The concentric circle attribute can be set to the date or year, which is increasing from the inside to the outside. Meanwhile, classify people who belong to the same year range into the same concentric circle area. The distance from the root node to any other node is the age difference of the characters represented by the two nodes. Therefore, the greater the distance from the central node to the character node, the more significant the age gap between the character and the central character. At the same time, if a node is on the more peripheral concentric circles than another node, it means that the character is relatively large in the year of birth. This allows users to directly and clearly observe the chronological information of each character.

But whether it is a traditional tree layout or a radial tree layout, there is a serious flaw, the node cannot display more information. Because in the tree structure, if you want to add descriptive
information to the nodes, the size of the nodes will change according to the length of the information content. As mentioned earlier, the rational arrangement of the spatial distribution in the tree structure is an important point. Therefore, when there is uncertainty in the size of the node, it may cause the nodes to overlap or the edges cross. Geoffrey M. Draper [DR08] et al. proposed a radial distribution of interactive fan charts for genealogical data visualization. Compared with the radial structure of the tree, the radial structure of the fan chart is more compact, but the center of the concentric circle is still the key role. Dividing the area between two adjacent concentric circles according to the number of people needed to display can reflect more role information while ensuring a clear structure. Meanwhile, it can ensure the primary feature of the radial structure, the farther away from the center and the greater the age gap between the central characters. But the limitation is that it can only show the parent-child relationship, and cannot include more kinship relationships such as marriage.

### 3.2.3 Time Concentration Layout

Genealogy can be summarized to the study of family history, which not only limits the historical research of the family, but also emphasizes the traceability of the past of every person in the family. For a historical study, time is a feature that cannot be ignored. Each character usually contains many different kinds of temporal information, for example, date of birth, death, marriage, etc. Sometimes, temporal information can be used as a condition for classifying family relationships.

TimeNets is a genealogy visualization tool proposed by Nam Wook Kim et al. [KCH10]. TimeNets combines the timeline with the character object to express the genealogy with time coherence. The foundation of the TimeNets structure is a horizontal timeline, which represents the continuous-time from left to right. Each character object has its own life timeline, which is placed horizontally and parallels to the reference timeline. The left starting point of the lifeline represents the birth date of the person, and the endpoint at the right side represents the time of death. If a character has been alive for the entire reference timeline, then his life timeline will continue until the end of the reference timeline. In addition, the line color is used to distinguish the gender of the person, red and blue represent female and male, respectively. If the total number of characters displayed is under control, then the structure will use thicker lines for the task object and place the person’s name inside the line. If the thickness of the line is not enough to store the title, then the name will be placed on top of the line.

Except for encoding the horizontal axis for the display of temporal information, the TimeNets structure also uses the encoding of the vertical axis to display the relationships between the characters. If the lifelines of two different character objects converge at a certain point, it indicates the marriage relationship of the two characters. The horizontal axis value of the meeting point is the corresponding marriage time. If the two lifelines are separated again then it means divorce. After that, if the two lifelines are close again after separation, it means that the two are close to remarrying. Meanwhile, the TimeNets structure can also describe the existence of multiple marriages. In general, if a person has multiple marriages, the lifelines of all spouses of that person will be arranged vertically and parallel before the date of the first marriage, but
the inevitable problem is that the lifelines will cross. Therefore, the improvement method is to alter
cately arrange the lifeline of the spouses above and below the person in focus. However,
this method will not be able to highlight the marriage order of the intersection character well.

![Image](image1.png)
Figure 3.5: TimeNets structure of genealogy visualization [KCH10].

As same as the tree structure layout, how to use TimeNets to process large genealogical data
is also a challenging problem. As shown in Figure 3.5, this is a simple example of a TimeNets
genealogy data visualization. It is conceivable that when there are many people in the data, the
entire structure will become very complicated. To improve this situation, the degree-of-interest
is added to the TimeNets structure as an essential parameter. Nowadays, it is also called DOI.
The DOI of each person in the genealogical data is calculated independently and compared
with a set threshold. If the DOI of a character is greater than the threshold, the person will be
included in the visualization, otherwise it will not be displayed. In addition, the calculation of
the DOI value is closely related to the central character. The central role has the highest DOI,
and the DOI value of each remaining character decreases linearly with his relationship with the
central character. In other words, the closer the relationship between a person and the central
role, the more likely it will be displayed in the TimeNets visualization.

![Image](image2.png)
Figure 3.6: Genelines structure of all descendants [Gen20]

In addition to TimeNets, Genelines is another software
that provides visual genealogical data with temporal in-
formation. Its primary architectural principles are sim-
lar to TimeNets. The lifelines represent character ob-
jects in the structure. The horizontal axis is used to en-
code time and the vertical axis is used to encode events
between characters. In this visualization, colors are
used to distinguish the genders of characters. Lifeline
with blue and red surface represent male and female,
respectively. The most significant difference between
Genelines and TimeNets is that Genelines chooses to
place the child’s lifeline between the parent’s lifelines
rather than below. As shown in Figure 3.7, the red
lifeline below the key character represents his mother,
and the blue lifeline above represents the father. It is
very important that the Genelines will estimate the uncertain date for people who do not have a definite date of birth or death. Then, in the visualization, the lifeline of the uncertain date part will be displayed without filling the color. Genelines can also help to construct a full descendant pedigree (see Figure 3.6). The purpose of this layout is to display all descendants of the central character. The basic principle is the same as the normal mode. In the DOI result, it shows all selected people, but the difference is that the central role and his spouse need to be placed on the top. Lifelines of the central character and his spouse are green and descendants use different colors. The vertical axis also encodes the relationship between different offspring characters and central characters.

In summary, the time concentration layout is very conducive to combining genealogical data and temporal information. It fits perfectly with the original intention of genealogy, which is the study of family history. But the difficulty of constructing it lies in the relatively high requirements on the original data. It requires that each character in the layout contains some temporal information to meet the classification conditions.
Chapter 4

Solution and Implementation

This chapter mainly introduces how to design and implement an appropriate solution according to the actual situation of the project. The solution and implementation process will be divided into multiple parts to elaborate.

4.1 Preliminary Data Preparation

At the start of the project, there was not much detailed information about the data provided by BHIC. Therefore, the first task is to have a comprehensive understanding of the primary resources and filter out the parts that are truly valuable to the current project. This section introduces all relevant aspects of the primary data resource.

4.1.1 Data Overview

The first job is to understand resources. According to the website description, BHIC has the largest demographic database in Noord-Brabant. These data are derived from many preserved paper documents or registers, such as shown in Figure 4.1. With the help of volunteers, they not only confirmed and indicated many registers, but also scanned many deed records. These data contain more than 18 million individual entries and cover more than four centuries. The data contains records of original statements based on original deeds. Each record includes some important personal data of the person being recorded (e.g. name, gender, and date of birth) as well as the person’s role in the deed (e.g. the deceased,
the bride, or the mother of the children), or some relevant additional information (e.g. military service or prisoner crime). It should be noted that all data records and dataset names are composed in Dutch. Meanwhile, the database is incomplete until now. There are still a lot of documents and registers have not been scanned. Thus, the database is always updated frequently.

The data are classified into the following dataset categories (in the download path provided by BHIC, their names are all in Dutch. Here it is based on Dutch literal translation of English.):

- DTB: Baptismal, Marriage, Burial
- Civil registration: Birth, Marriage, Death
- Population registration
- Militia registration
- Prison registration

It seems that the types of events stored in the first and second datasets appear to be repetitive, and both contain marriage and death event records. The actual difference is that the data source of the first dataset is the ecclesiastical population records. The retrospective record of the civil register is the registers in which the ecclesiastical population records were kept until the introduction of the civil registration [Wik20b]. Dutch genealogists call it the DTB register (register of baptisms, marriages, and burial). In short, before the official civil registration, most of the demographic information was kept by the local ecclesia. Therefore, the recording time of these data is relatively longer. Besides, the content of ecclesiastical records and civil records are also different, which will be introduced in detail below.

### 4.1.2 Data Model

All data is constructed based on the A2A model. It is a generic metadata format that is used to exchange and access various sources of personal historical data. The target group is mainly heritage institutions that want to disclose personal historical data in a standardized way to improve interoperability. The A2A data format is open source code developed by the Internet agency Mindbus and used in the WieWasWie platform [rJM13]. Within the A2A data format, the data on the certificate/register is used. The events described in the contract are particularly relevant to personnel. In addition, the relationship between the person and the event will be described. The A2A data format also includes source data, such as access rights, record numbers, and stock numbers. In detail, the A2A data model consists of five main parts:

1. Person
2. Event
3. Object
4. Source (such as source component involving source data, such as record number, inventory number, etc.)
5. Relationship (the relationship part involves people, events and the mutual relationship between events, such as the relationship between people and the relationship between person and event.)

The elements "Person", "Event", "Object", and "Relation" can appear multiple times in the data model. For various types of behavior, these behaviors may contain multiple events. For example, the baptism certificate may include both the event "baptism" and the event "birth". However, each record in the "Source" section appears only once and is the only required field in A2A. The five main parts mentioned above have a large number of different parts properties. For example, the "Person" elements may contain characteristics of marital status, age, and gender. Also, Persons, Events, and Objects have their own identifiers (Person identifier (pid), Event identifier (eid), and Object identifier (oid)). Based on these identifiers, mutual relationships can be established between Persons, Events, and Objects.

The A2A data model uses an XML format UTF-8 encoding. This XML format is defined using XSD (XML schema definition). In XSD, in addition to conventional data types such as strings (a series of characters) and integers (a series of numeric data), there are two data types that allow you to define fields yourself: SimpleTypes and ComplexTypes.

As shown in Figure 4.2a, SimpleType is a data type that does not contain any other tags. This element contains only the start and end tags. Therefore, it cannot be subdivided into other partial attribute fields. As shown in the example of a simple field Gender, this field has no other subsections. The second type of definition, complexType, as its name implies, has a relatively complex structure. ComplexType is an element that can contain child-elements or subtags. Thus, the field can be subdivided into partial attributes. As shown in Figure 4.2b, ComplexType PersonName consists of subtags PersonNameFirstName, PersonNamePrefixLastName, and PersonNameLastName.

As shown in Figure 4.2a, SimpleType is a data type that does not contain any other tags. This element contains only the start and end tags. Therefore, it cannot be subdivided into other partial attribute fields. As shown in the example of a simple field Gender, this field has no other subsections. The second type of definition, complexType, as its name implies, has a relatively complex structure. ComplexType is an element that can contain child-elements or subtags. Thus, the field can be subdivided into partial attributes. As shown in Figure 4.2b, ComplexType PersonName consists of subtags PersonNameFirstName, PersonNamePrefixLastName, and PersonNameLastName.

As shown in Figure 4.2a, SimpleType is a data type that does not contain any other tags. This element contains only the start and end tags. Therefore, it cannot be subdivided into other partial attribute fields. As shown in the example of a simple field Gender, this field has no other subsections. The second type of definition, complexType, as its name implies, has a relatively complex structure. ComplexType is an element that can contain child-elements or subtags. Thus, the field can be subdivided into partial attributes. As shown in Figure 4.2b, ComplexType PersonName consists of subtags PersonNameFirstName, PersonNamePrefixLastName, and PersonNameLastName.

Each record starts with a root tag (see Figure 4.3). The root label A2A has a mandatory version attribute and contains the version number. Each record has its own root tag. The root label A2A has a mandatory version attribute and contains the version number. Therefore, all tags in each record start with the root tag.
4.1. Preliminary Data Preparation

The following part will take the DTB-Burial dataset as an example to introduce how the structure of XSD is represented in the real dataset. Each record in each dataset is composed of the five parts mentioned above, Person, Event, Object, Source, and Relationship, and each part is a tag with a ComplexType structure. As mentioned above, these data mainly store the population registration information held by the ecclesia before the formal population registration form is generated. Specifically, these data are saved by three different datasets, DTB Trouwakten (Marriage certificates), DTB Doopakten (Baptismal certificates), and DTB Begraafakten (Burial Deeds).

The Event element of each record in the Burial dataset contains subtags, EventType, EventDate, EventPlace, and EventReligion. All records’ event types are Begraven (Burial). The EventDate tags are subdivided into subtags Year, Month, and Day. The EventPlace tag reports the location of the event. Because it is the ecclesiastical record file, a subtag EventReligion is used to store the beliefs of the deceased. Figure 4.4 shows an example.

```xml
<za:Event id="Event1">
  <za:EventType>Begraven</za:EventType>
  <za:EventType>Begraven</za:EventType>
  <za:EventDate>
    <za:Year>1771</za:Year>
    <za:Month>12</za:Month>
    <za:Day>14</za:Day>
  </za:EventDate>
  <za:EventPlace>
    <za:Place>Utrecht</za:Place>
  </za:EventPlace>
  <za:EventReligion>
    <za:Religion>Reformert</za:Religion>
  </za:EventReligion>
</za:Event>
```

Figure 4.4: Event tag

As shown in Figure 4.5a, most records contain only one Person tag. The identity attribute pid of each person tag is unique in the entire dataset, and also unique in the entire database. The PersonName and Gender in the person element are mandatory subtags, and PersonRemark is an optional subtag. There are four different optional subtags under the PersonName tag which are PersonNameFirstName, PersonNamePatronym, PersonNamePrefixLastName, and PersonNameLastName. These four subtags will be selected and added according to different situations. The Gender tag contains only four different values that are Man, Vrouw (woman) and Onbekend (Unknown). Some people will have a special subtag PersonRemark, which is used to record some special description information about the character, such as the cause of death. For records with only one Person element, its RelationEP element will be used to express the relationship between the person and the event. The subtags PersonKeyRef and EventKeyRef are used to identify the corresponding event and person, respectively. The subtag RelationType is used to clarify the person’s role in the event, in this dataset the most important role is Overledene (deceased).

Some records will contain two Person elements, one is the primary person and the other is the secondary person. To be precise, one is the deceased and the other is the person who is related to the deceased. The structure of the two Person elements is similar. There will be two RelationEP elements in the record, one is used to store the relationship between the primary person and the event, and the other is used to store the relationship between the secondary person and the primary person. The structure of two RelationEP tags is also similar. The only
Chapter 4. Solution and Implementation

(a) Single person tag

(b) Relationship tag

Figure 4.5: Single person tag with relationship tag

difference point is that there will be two PersonKeyRef subtags in the second RelationEP elements and no EventKeyRef subtags. Meanwhile, the value of the RelationType subtag in the second RelationEP is Relatie (Relationship), but there is no clear relationship name or related information. Sometimes there might be information about the exact relationship in the PersonRemark subtag of the secondary person. Figure 4.6 shows an example.

(a) Double person tag

(b) Double relationship tag

Figure 4.6: Double person tag with double relationship tags

Finally, as shown in Figure 4.7, the data resource information is mainly stored in the Source element. The subtags are used mostly to record the storage location, storage date, data type, and data reference information of the original resource. In general, this part of the data is not particularly related to events and persons, so it will not be the focus of attention in the project.

Figure 4.7: Source tag

Furthermore, because most of the original data saved by BHIC is event registration, the main subjects of these data are events and person. Therefore, the Object element rarely appears in most datasets.
The main content and structure of the remaining datasets are described below.

- **BTD-marriage:**
  The event type is marriage registration. Mandatory information related to the event includes the date and place. There are two person elements in each record. The mandatory subtags for each person are name and gender. Meanwhile, each person's optional information tags are residence and person remark. Each record has two relationship elements, which are used to record the roles of two people in marriage, namely the groom and the bride.

- **BTD-baptismal:**
  The event type is baptismal. Mandatory information of the event is the date, place, and religion. Each record contains at least two person elements. The mandatory subtags for each person are name and gender, and an optional tag is a personnel remark. The role of one person is the child (baptized person) and the others are the parents (father or mother) when there are only two person elements. Most records have five or four people. Except for the roles of parents and baptized persons, the rest roles are witnesses of the event.

- **Civil registration-birth:**
  The event type is the civil registry, birth certificate. Mandatory information related to the event is the date and place of the event. There are three mandatory person elements in each record. There are three mandatory relationships, indicating that the roles of the three people are child, father, and mother, respectively. The child is the person born in the registration event. Therefore, each child has a mandatory birth date subtag in the person. The birth date value is equal to the date value in the event elements. Furthermore, the fixed subtags owned by all three person elements are person name and gender, and the optional tags are personnel remarks.

- **Civil registration-marriage:**
  The event type is the civil registry, birth certificate. The mandatory event information is the date and place of the event. There are five mandatory person elements in each record. Meanwhile, there are five corresponding relationships, indicating that the roles of the five people are groom, bride, groom's father and mother, and the bride's father and mother, respectively. The mandatory information contained in each person element is name and gender. For the groom and bride, the optional subtags include birthday, age, and place of birth in addition to personnel remarks.

- **Civil registration-death:**
  The event type is the civil registry, death certificate. The mandatory event information is the date and place of the event. Normally, there are three person elements with three relationship elements, indicating that the roles of people are deceased, deceased father and mother, respectively. In other cases, each record has four person and four relationship elements. An extra person is a person who has a relationship with the deceased, but no clear relationship information is indicated. The mandatory information contained in each person is name and gender, and optional information is personnel remark.
• Militia registration:
The event type is the civil militia register. Each record only contains one person element and one relationship. The role of the person in the event is the serving citizen. The event elements do not include the date and location of the registered service. The source elements have the date and address where the file was recorded, but this is not completely equal to the place where the citizen serves. The mandatory information in the person is the person's name, gender, date of birth, and place of birth. There is not any optional subtag.

• Prison registration:
The event type is the civil prison register. Each record only contains one person and one relationship. The person's role in the event is the criminal registered in prison. Only the type of event is included in the event elements and no date and place information of registration. The source elements have the date and address where the file was recorded, but this is not completely equal to the prison location. The mandatory information in the person tag is the person's name, gender, and place of birth. The optional subtags are birth date and age.

• Population registration:
The event type is the civil population register. Each record only contains one person element and one relationship. The role of the person is a registered citizen. The type and date of the event are included in the event elements. The person tag contains the person's name, gender, and resident place as mandatory information. The optional subtags are birth date and age.

It is imperative to note that there is no relation between each record, whether in the same dataset or between different datasets. In short, each person tag and its relationship tags only appear in its own record, and never appear in other records. In other words, the BHIC data processing and storage process does not check whether each person in each event has been repeatedly stored in the database, so there might be cases of person duplication. Even if there is a duplicate person, it will still use an independent person tag with the unique identification attribute in different records. It leads to the fact that each person's accurate kinship resources are very limited. For a person recorded in the BTD or civil registration datasets, the kinship types that can be 100% confirmed can only be the parent, children, or spouse. And the people in the remaining datasets do not contain any measurable kinship information.

Finally, it also should be mentioned that since the information stored in each dataset is different, then the size of the dataset is also uneven. For example, currently, the size of the BTD-marriage dataset is around 5.5GB, but the BTD-baptismal and BTD-burial are both approximately 1.3GB. The reason is that each record in the BTD-marriage contains more relationship information. Furthermore, the size of each type of civil registration datasets is around 6GB. The reason is similar, because each civil registration record includes at least three pieces of kinship information. Therefore, it should be noted that this is a tremendous database and using all the data will significantly exceed the workload that can be carried by the current
project. Thus, it is more feasible to filter some of the data as the original resources of the project.

### 4.2 Data Selection

As mentioned in previous sections, there is a wide variety of demographic information that the BHIC database can provide. Still, the current project needs to select a part of it to help establishing relevant genealogical pedigrees. Thus, it is indispensable to determine which types of data need to be chosen as the resource data. Through related work research, genealogical data can be classified into the following parts:

- The people
- The kinship relationships between people
- The important events
- The temporal information of important events
- The reliability of the information.

According to the above requirements, the selection range can be determined from the BHIC database. From the previous section introduction, it can be realized that the dataset types BTD and civil registration are mainly containing kinship information data. The remaining datasets contain only information about the event and main person, and it is complicated to judge the relative relationship of characters through unrelated different event records. Therefore, the main focus of this project is BTD and civil registration data. Due to the different formats and sizes of the three datasets in the BTD category, there is a big difference in the kinship information that can be collected. For instance, in the DTB-burial dataset, only the information of a dead person and a person related to the deceased can be obtained. Still, the precise kinship relationship cannot be determined. Then the kinship data that can be collected in the BTD-marriage dataset includes the spouse and the parents of both spouses, which is a very complete family relationship information. Besides, the original resource of the BTD dataset was recorded by the local church. Its scope has certain limitations and cannot cover the local population on a large scale. For the above reasons, the civil registration datasets are the most appropriate option.

Firstly, the original data of civil registration was recorded and kept by the local government and has uniform specifications. Thus, it covers a wide range of information and has high credibility. Secondly, each of its records is composed of important events and people related to the event, and also contains the temporal information of the critical events. Finally, each record can provide at least one set of parent and child relationship information. These characteristics fully meet the requirements of genealogical data. Therefore, civil registration data is adopted as the primary current source of the project.
Because BHIC has been updating the data, so far they have not processed all the Noord-Brabant historical civil registration events. In the current project, the last downloaded data was on May 2, 2020. The earliest record of civil registration events was in 1800, and the latest was in 1940. All remaining events were recorded in the 19th and early 20th centuries.

### 4.3 Genealogical Pedigree Construction

After having an in-depth understanding and filtering of the original data resources, we need to convert the essential information into the format of genealogical data and construct visualization structure. In the following we will describe and discuss several issues around genealogical pedigrees.

#### 4.3.1 Foundation of Genealogical Data Processing

The ultimate purpose of this project is to help users query and display their genealogical data, and users should get feedback in the shortest time after providing query requirements. BHIC used the XML format for data storage in the selected original datasets, which are the civil registration records, event information, character information, and character relationship information are mixed. Therefore, it is necessary to recombine and process the information resource and construct a reasonable genealogical pedigree data structure.

As mentioned in the previous article, the civil registration data was divided into three datasets according to the type of event: birth, marriage, and death. Each record contains information about the person, town, and date of the event. The relationship between the characters can be summarized into three categories: father-child, mother-child, and spouse. According to this situation, the first decision is to split each original XML file into two sub genealogical datasets and store them in a CSV format. As shown in Figure 4.8, one is the Person dataset and the other is the Relationship dataset. For different event types, the formats of the datasets are slightly different.

The Person table is mainly used to store the necessary information for each person. The P_id is the primary key of the table, which uses the pid attribute of the person tag in the source file directly. The Name is the combination of all the name information of each person tag in the source file. The essential situation is the first name and last name, other cases may include patronym or prefix last name between first name and last name. If the gender of the person in the source record is determined, then the data will be directly inherited in the new table. If the gender of the person is unknown, the gender can be judged according to the person’s role in the original event record. For example, if the person is the father of a deceased, then his gender is male. But if the person’s gender cannot be determined eventually, the gender value will be uncertain in the table. The Role represents the person’s role in the kinship in the original recorded event. There are three possibilities for the person’s role in birth and marriage events, father, mother, and child. The death event has an extra role possibility which is relatives. The
4.3. Genealogical Pedigree Construction

(a) Civil Registration-Birth

(b) Civil Registration-Death

(c) Civil Registration-Marriage

Figure 4.8: Original genealogical data

*Sum of relation* is the total number of kinship data of the person. For each person in the birth event, there is only one kinship data instance, but there might be two instances for the person in the marriage and death records. Moreover, because the person tag in the marriage registration event may contain additional information, except for the person remark, there are two optional indexes, which are birthday and hometown.

The *Relationship* table is used to store the kinship data. All the relationships in the original source data can be summarized into three types, parents and children, spouses, and relatives (death registration only). Therefore, there are four different examples of recording relationship data instances according to the person’s role:

- Father or Mother: \(P_{id1}, Spouse, P_{id2}, Children, P_{id3}, Event date, Place, Certainty level\)
- Child: \(P_{id1}, Father, P_{id2}, Mother, P_{id3}, Event date, Place, Certainty level\)
• Child (marriage only): $P_id1, Spouse, P_id2, NULL, NULL, Event date, Place, Certainty level$

• Relatives (death only): $P_id1, Relative, P_id2, NULL, NULL, Event date, Place, Certainty level$

The first $P_id$ value in each relationship instance is used as the primary key, and also the foreign key connecting the Person table. In the birth event, each person has only one relationship data, the $p_id$ can be used directly. However, in marriage and death events, if the Sum of relation of a person instance is 2, then the first relationship data instance will directly use the corresponding person’s $p_id$. At the same time, the second relationship’s primary key is $p_id - 1$, which is combined by the relevant person’s $p_id$ and a suffix $-1$. The Relation name index stores the the type of the kinship, and the following index save the pid of the corresponding person. If the role of the primary person is the father, then the first Relation name index is the spouse, and the second is the child. If it is the relationship instance for the child, the two Relation name indexes should be father and mother, respectively. The kinship relationship of each person has been saved independently according to their role. Although the father, mother and child in one family actually recorded the corresponding roles in the same relationship, but they all have their own independent instances of kinship data. In addition, in the original marriage registration records, the characters included are the groom, the bride, and the parents of the bride and groom. Based on this situation, when the groom or bride stores as the role of a child in the Person dataset, besides the relationship data with their parents, they also have spouse relation data. In this case, its relationship data instance uses only one set of Relation name and $p_id$ indexes to store the kinship type and the corresponding personal identification. The values of the second set of Relation name and $p_id$ are null. Furthermore, for the deceased in some death registration events, the registration information includes another relative in addition to parents. Thus, when the deceased person is stored as a child with two relationship data instances, the Relation name value of the second instances is Relative. The Event date and Place values of each relationship instance are the date and city of the event record to which the primary person belongs. At last, the Certainty level represents the reliability of the kinship information, which will be fully introduced in the next chapter. Because the data resource in these tables is processed from the original files provided by BHIC, their Certainty level is the highest A level.

### 4.3.2 Genealogical Pedigree Construct

According to the basic situation of the current project data resource, the genealogy that can be created is to display all the related people instead of researching the descendants or ancestors of a particular person. A complicated genealogy may contain several or even dozens of generations, and at most hundreds of people. The relationship between the characters is also diverse. There are close relationships such as parents, sisters, brothers, grandparents, etc. There are also distant relatives, such as an uncle, aunt, great-grandfather, etc. If it is for a single genealogical pedigree, then all the information can be stored in detail, but the current project needs to
solve the problem of querying different genealogies. Thus, it is not a reasonable way to save all the relatives in detail. Then the best approach is to simplify the complicated structure into many single connections. It means each person only stores the information of the immediate family relationships, that is, parents, spouses, and children, such as the format in the previous section description. In this way, after having a person’s kinship data, we are able to obtain the identification of his father, mother, and children. After that, the kinship data belonging to these people can be determined through personal identification. After multiple iterations, all people who have an indirect relationship with the original person can be identified. In general, the complicated pedigree chart is composed of many sub-pedigree graphs. This approach can help to construct the entire genealogical pedigree from one person. The Figure 4.9 below is a simple example of a pedigree construction.

![Figure 4.9: Pedigree construction](image)

### 4.3.3 Certainty Level Coefficient

However, the current situation is that kinship relationships for each person determined from their original civil registration event are sporadic, which means a person has at most two relationship data instances. These data can only form the most basic sub-pedigree structure, which is not enough to establish a high quality genealogy. As we all know, a simple structure with only a few persons connected by one or two relationships cannot be called a pedigree chart. Therefore, it is necessary to determine the kinship between more people and according to the construction approach to form a network-like relationship pedigree.

In this project, the enumeration method is used, which means a comprehensive screening of all the person data instances in the datasets. Pair every two persons in the dataset, and then judge whether there is a possible connection between the two people based on the known information.

There are some situations that can be used to help discover the correlation between different relationship data instances. The first problem comes into play, if there are two relationship data instances with the same name of father and mother, but containing different child names. At the same time, the event dates of the two cases are close, and the event locations are the
same. Then it can be roughly determined that the parents in the two different relationship data instances are the same pair of parents. After that, it is possible to save the kinship data for corresponding persons, and only keep the personal information of one pair of the parent in the person dataset, and store the relationship data of different children for father and mother in separate instances. Also, the two children need to have their own relationship data instance for recording parent information. The second problem occurs, if there are two persons with the same name and gender in two different relationship instances but have different roles in the kinship relationship, for example, one of them is a child and the other is a father. At the same time, the time difference between the two instances is within a specific range, and also the same location. Then it can be considered that the two persons might be the same one. The person whose role is a child must have kinship information about the parent. Thus, it is able to add a relationship instance that used to record the parent information for the person whose role is the father. However, the critical issue is that there is no actual reliable evidence to prove whether these discovered kinship or personal connections are 100% authentic. In addition, in some cases, there are differences in the conditions that help to determine the connection between the persons. For instance, only one parent has the same name, the event locations are different, or the event date interval is too large. When these situations arise, it is necessary to confirm whether the correlation between the corresponding characters can be reliably determined. Therefore, the certainty level coefficient was introduced in this project to measure the reliability of each kinship relationship. Certainty and uncertainty are a set of opposite words. In the current project, certainty level coefficients are used to reflect the uncertainty of the data. In simple terms, the reliability of each relationship data is proportional to the value of the certainty level coefficient. When the value of the certainty level coefficient is higher, the reliability of the relationship data is more elevated. For example, all the kinship information from the initial datasets is the data recorded in the civil registration documents, so these kinship data have the highest reliability and certainty level value. There are four levels of certainty in the current project setting:

- **A: High certainty** - There is an excellent possibility that this kinship relationship exists in reality
- **B: Normal certainty** - There is a possibility that this kinship relationship exists in reality.
- **C: Low certainty** - There is a low possibility that this kinship relationship exists in reality.
- **D: Very low certainty** - The possibility of this kind of kinship existing in reality is extremely low.

4.3.4 Relationship Determination Model

The principle that needs to be followed in the process of determining the possible kinship between people is that a particular person’s birth-marriage-childbirth-death process must occur
4.3. Genealogical Pedigree Construction

in chronological order. That is to say, the events marriage and childbirth must happen in sometime after the birth date, and also before the death date. Meanwhile, it is impossible for a person to marry and have children at any age. Most of the cases are only possible after his adulthood. Therefore, in this project, we chose to use the data of people and relationships in birth and marriage events to find all possible kinship data. Use the data from the death registration events to test the found kinship data. The relationship determination model with certainty level coefficients consists of the following parts.

The whole model is mainly divided into three parts. The goal is to integrate all kinship information contained in three different event datasets into one data set. The first part is to use the datasets of the birth event to determine the possible kinship. Firstly, for each data instance in the Birth − person dataset, compare it with all the remaining person instances in the dataset on by one. Assuming that the person currently selected is A, during the comparison, if it is found that the similarity between the names of A and another person B reaches or exceeds 95%, it should determine whether A and B are the same people in reality. The reason why the name similarity requirement is 95% rather than 100% is that the civil registration data provided by BHIC is basically from the documents of the 19th and early 20th centuries, so it might be some errors or omissions of text during the process of making it into digital data. Therefore, if there is a slight deviation in the text information during the comparison, it will still be regarded as the same name.

Name Similarity Comparison

The persons’ names similarity comparison is dependent on the Levenshtein distance [ZS18] algorithm, also called the Edit Distance algorithm. In simple terms, it refers to the minimum number of editing operations required to convert from one string to another. The permitted editing operations include replacing one character with another character, inserting one character, and deleting one character. In general, the smaller the editing distance, the higher the similarity of the two strings. The algorithm implementation principle is as follows:

<table>
<thead>
<tr>
<th>Table 4.1: Levenshtein distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>abc</td>
</tr>
<tr>
<td>abe</td>
</tr>
<tr>
<td>a</td>
</tr>
<tr>
<td>b</td>
</tr>
<tr>
<td>e</td>
</tr>
<tr>
<td>(c)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>abc</td>
</tr>
<tr>
<td>abe</td>
</tr>
<tr>
<td>a</td>
</tr>
<tr>
<td>b</td>
</tr>
<tr>
<td>e</td>
</tr>
</tbody>
</table>
Assume that there are two simple strings \( abc \) and \( abe \). Use these two strings to form a matrix, as shown in Table 4.1a. Then the value of the index \( x \) in the table depends on the index on its left, the index on its top, and the index on its top left. According to the logic of the Levenshtein distance algorithm, the values of the left and right index need to be incremented by 1. That means the left value is \( 1 + 1 = 2 \), and the right value is also \( 1 + 1 = 2 \). If the characters in the row and column to which \( x \) belongs are the same, then it can directly use the value in the upper left index. In the current situation, since the characters are both \( a \) the value 0 can be used without any change. The value of index \( x \) is equal to the smallest value between 2, 2, and 0, which should be 0. In Table 4.1b, the value of index \( y \) is computed to be the same as index \( x \). However, the characters of the column and row are \( a \) and \( b \), respectively. They are not equal. Then the value of the upper left index of \( y \) is \( 1 + 1 = 2 \). Therefore, the value of index \( y \) is equal to the smallest value between 2, \( 0 + 1 = 1 \) and \( 2 + 1 = 3 \), which should be 1. Furthermore, the index value of \( z \) is 2. The value of each index means the number of steps to make the corresponding strings in columns and rows become the same. For instance, the corresponding column string of the index \( z \) is \( a \), and the row string is \( abc \). To make the string \( a \) become \( abc \) required at least two steps, which are adding the character \( b \) then adding \( c \). Based on the above method, after completing the entire matrix, the last index value in the matrix is the minimum steps to change the two strings to be the same, which is called the Levenshtein value. In the current case, the Levenshtein value is 1. The similarity between two strings is computed by \( 1 - \left( \frac{\text{Levenshtein}}{\text{maxLen}} \right) \), and the \( \text{maxLen} \) is the max length between two strings. The similarity of strings \( abc \) and \( abe \) is \( 1 - (1/3) = 0.666 \). In general, using the Levenshtein distance algorithm to compare any two strings, we can get a float value in the range of \([0,1]\).

The Similar Person Searching Inside Birth Event Datasets

If the name similarity between person \( A \) and \( B \) up to 95%, then it is worth to determine whether \( A \) and \( B \) are the same people, and the process needs to use their existing kinship data. Still, only the relationship data instances with the certainty level \( A \) are used as the comparison parameter. It means only highly reliable data can be used as the basis for judgment. If we use unreliable data for comparison, results can only be the kinship information with an even lower level of certainty. Too much unreliable data will cause a more crowded data structure and affect subsequent data observations. There are two main cases in the birth event datasets:

- **Case 1:** If the roles of person \( A \) and person \( B \) are both Father or Mother:
  The current goal is to find out if \( A \) has other spouses or children. Thus, only their spouse-child kinship data instances will be used, regardless of whether \( A \) and \( B \) have relationship data about their parents. Therefore, the core data of each relationship instances can be summarized as [spouse, child, childbirth date, childbirth place]. In this case, if \( A \) and \( B \) can be determined to be the same person, then the child information in \( B \)'s kinship would be stored in \( A \)'s relationship data. The basic principle of judgment is that in their relationship data instances \( A \) and \( B \) have the same spouse and their child was born in the same place. Meanwhile, the most important point is that the difference between their child’s birth date is within a specific range.
A simple model is designed to compute the new relationship data for the person $A$. It means each person is one object with 3 attributes, which can be written as $A(x_{A1}, x_{A2}, x_{A3})$ and $B(x_{B1}, x_{B2}, x_{B3})$. $x_1$ is the birth year of the children in their respective relationship data instance, $x_2$ is the name of the spouse, and $x_3$ is the birth place of the child.

$$D_p(A, B) = d(Y_1, Y_2, Y_3)$$

$$f(d) = \begin{cases} 
0, & \text{if } Y_1 = 0 \\
\sum_{i=1}^{3} Y_i, & \text{otherwise} 
\end{cases}$$

$$New\text{\_kinship} = \begin{cases} 
[A, spouse_A, child_B, birth\_date_B, birth\_place_B, c.l : A] & \text{if } f(d) = 3 \\
[A, spouse_A, child_B, birth\_date_B, birth\_place_B, c.l : B] & \text{if } f(d) = 2.5 \\
[A, spouse_A, child_B, birth\_date_B, birth\_place_B, c.l : B] & \text{if } f(d) = 2 \\
& \text{and } Y_2 = 1 \\
[A, spouse_B, child_B, birth\_date_B, birth\_place_B, c.l : B] & \text{if } f(d) = 2 \\
& \text{and } Y_2 = 0 \\
[A, spouse_B, child_B, birth\_date_B, birth\_place_B, c.l : C] & \text{if } f(d) = 1.5 \\
& \text{and } Y_2 = 1 \\
[A, spouse_B, child_B, birth\_date_B, birth\_place_B, c.l : C] & \text{if } f(d) = 1.5 \\
& \text{and } Y_2 = 0 \\
[A, spouse_B, child_B, birth\_date_B, birth\_place_B, c.l : D] & \text{if } f(d) = 1 \\
& \text{or } f(d) = 0.5 \\
Empty & \text{if } f(d) = 0
\end{cases}$$

The $c.l$ means the certainty level. The higher the matching result of the personal attributes, the higher the similarity between the two persons, and the greater the probability that the two persons are the same person in reality. Same as mentioned above, the difference between the birth years of their children has a decisive effect on the matching result. Because a person’s reproductive age range is limited, if the birth date interval of two children is too large, such as 50 years, then it is obvious that the two people cannot be the same person. Or the birth interval of the two children is too short, such as less than one year, this is also generally impossible. Thus, we set a specific date range for the birth date interval of their children. Physiologically speaking, a woman takes ten months from pregnancy to delivery, and it is difficult to get pregnant again within a few months after delivery. So for the same person, the birth interval of their two children is at least
two years. Afterward, according to the research of Frans van Poppel [PR03] every Dutch
person had an average of 4 to 5 children in the mid to late 19th century, and an average
of 3 to 4 children in the early 20th century. Therefore, in most Dutch families in the 19th
and early 20th centuries, the difference in the birth years of the two children was between
2 and 12 years. There might be some cases where the difference between the birth year of
children exceeds 12, but in reality, the rare situation is that the age difference of brother
or sister exceeds 20 years. Under these situations, the result is still acceptable, but it
will affect the level of certainty of the final kinship data. Besides, when it is found that
the spouse of A and B are different, then it should take care when adding new kinship
data for A. The spouse name of the new relationship data instance is the spouse name
in person B’s kinship data. This situation can be understood as A may have different
marriages in his life.

When new_kinship data with different levels of certainty is obtained, it cannot be directly
added to the relationship table. It must be edited to the specific data instances format
in the relationship dataset. Then some different situations need to be considered. First
of all, if the new relationship has a top level of certainty, then it means A and B are
highly possible to be the same person in reality. Thus, it is very necessary to merge the
person A and B, that is, transfer all the B’s relationship data to the A and remove them
from the B. Secondly, because the person comparison process is cyclically performed in
the birth – person dataset, then there might be repeated comparisons of the same pair
of people, for example, AB and BA. At the same time, if both persons A and B have
multiple kinship data instances with the certainty level of A, then we need to compare
these kinship data instances one by one. The spouses and location information in these
data instances must be consistent, but the event time information must be different.
Therefore, when a certain kinship data instance of the person B is compared with different
kinship data instances of A, a new kinship with the same substantial data information but
different certainty level coefficients might be obtained. So before adding a new relationship
data instance to the person A, it is indispensable to check whether the kinship with the
same characters has been already saved for the person A. Finally, the corresponding
spouse and child in the new kinship data also need to be added or modified the same
kinship data instances. Besides, in the previous part of the article, it was mentioned that
if a person has more than one instance of kinship data, then the identification of the data
instances is composed of the person’s P_id and a suffix. For example, if it is the person’s
second relationship data instance, the suffix is −1. Therefore, every time a new kinship
data instance is added for the person A, then the identification of the new relationship
data is the P_id of person A plus the suffix −n. The n is the number of the sum of A’s
kinship data instances before adding the new one. The flow chart (Figure 4.10) below
shows the entire process.

In the process of adding the new kinship data instance, it also should be noted that if
the certainty level of the new data instance is not A, the original kinship records should
still be retained in the corresponding person. In other words, for the same person, it
is allowed to have two sets of different parent information. However, in the two sets of
different parents, either the father or the mother must be the same person, and the names
4.3. Genealogical Pedigree Construction

Case 2: If the role of person A is Father or Mother, and person B is Children.

The purpose of determining whether A and B are the same people, in this case, is to find possible parent information for a certain person A. For person A, only reliable kinship data for his role as a parent are used. Find the earliest birth date and address information of person A’s children in these data. Then this earliest date must be later than the birth date of B several years since the person only can have children after he gets married in adulthood. According to research by Tine de Moor et al. [dMBSG17], the average age of Dutch people’s first marriage in the 19th century was at the ages of 27 to 28 years. However, the sample size, standard deviation, or variance are not provided in the paper, and then we cannot get the 95% confidence interval to estimate the true distribution of first marriage age in the actual situation. Therefore, it can only be assumed that the age distribution range for the Dutch people to get married and have children for the first time in the 19th and early 20th century is 21 to 34 years. Afterward, if the age is between 35 and 45 years, it is also considered acceptable, but it will affect the level of certainty of the final kinship data. Meanwhile, if the birth places of A and B are the same, then the degree of similarity between them can also be increased. B must have hometown information, while A’s hometown information can only be inferred from known data. Also, based on the research of Tine de Moor et al. [dMBSG17], in the 19th century, as many as 55 percent of Dutch people had their first marriage in the place where they were born. At the same time, 17 to 18 percent of people have their first marriage less than 7 kilometers from their born place. Thus, the birth place of the first child of person

![Genealogical Pedigree Diagram](image-url)
A can be used as its birth place data. Besides, in the Dutch tradition, the name may carry birth place information. For example, people born in XXX will be called van de XXX. Then it is possible to obtain the birth place information by the name of person A.

In general, the essential data of A can be summarized as PersonA[name, gender, birth date of the first child, birth place of the first child], and PersonB[Person Remark, gender, birth date, birth place]. If the child dies at birth, the child will be marked as a dead infant in the original data. If the person B is a dead baby, then there is no need to continue with the further comparison. After that, the essential information of person A and B can be converted into an object with corresponding attributes, which are $A(x_{A1}, x_{A2}, x_{A3}, x_{A4})$ and $B(x_{B1}, x_{B2}, x_{B3}, x_{B4})$. The process of attribute matching and new kinship information generation methods are shown below.

$$D_c(A, B) = d(Y_1, Y_2, Y_3, Y_4)$$

$$f(d) = \begin{cases} 
0, & \text{if } Y_i = 0 \text{ for } i \in \{1, 2, 3\} \\
\sum_{i=1}^4 Y_i, & \text{otherwise}
\end{cases}$$

$$New\_kinship = \begin{cases} 
[A, father_B, mother_B, birth\_date_B, birth\_place_B, c,l : A] & \text{if } f(d) = 4 \\
[A, father_B, mother_B, birth\_date_B, birth\_place_B, c,l : B] & \text{if } f(d) = 3 \\
[A, father_B, mother_B, birth\_date_B, birth\_place_B, c,l : C] & \text{if } f(d) = 2.5 \\
[A, father_B, mother_B, birth\_date_B, birth\_place_B, c,l : D] & \text{if } f(d) = 2 \\
Empty & \text{Otherwise}
\end{cases}$$

After obtaining the $New\_kinship$ data about A’s parents, except for editing the data into a specific format, and different treatments are also required according to the situation. The first step is to check whether A already has kinship data about the parents, and if so, compare the certainty level of the new relationship data instance with the old relationship data instance. If the certainty level of the new data instance is higher, then add the new data instance and remove the old one. Otherwise, the old kinship data is retained. If the level of certainty is equal, then both instances are retained.
After that, a connection data instance needs to be added for person B to record that there is a possibility that A and B are the same person. This connection data uses the same certainty level as the new kinship data. The person B is allowed to have more than one connection data instance. The process is shown in the chart in Figure 4.11.

The Similar Person Searching Between Birth and Marriage Event Datasets

After the first part is completed, each person in the birth – person dataset might already have many relationship data instances or still only one. In order to find more possible relationships, the second part is to use the data of persons and their kinship in the marriage event datasets. It works the same as the previous part, select a person A from the current birth – person dataset and compare him with all the person data instances in the marriage – person dataset. If it is found that there is a possibility that a person B is the same person with A, then some kinship relationships belonging to B would be added to A, which is the kinship information that was not in A’s stored relationship instances. Meanwhile, for the corresponding persons who appear in the newly added relationships, they also need to be saved into the birth – person dataset with the identical person data instance.

In the process of comparing whether A and B are the same people, it is necessary to use the personal information of A and B. Also, their kinship data with the A certainty level are required. In this part, after two persons A and B are found whose name similarity exceeds 95%, the comparison method is also divided into two cases according to different situations.

- Case 1: If the roles of person A and person B are both Father or Mother:
Chapter 4. Solution and Implementation

The difference with the comparison in the first part is that if the person’s role is the Father or Mother in the datasets of the marriage event, the temporal information stored in his kinship data is not the date of the child’s birth, but the time of the child’s marriage. The birth date of the child, groom, or bride in the marriage event is not mandatory information. Therefore, if the child’s date of birth is known, the year interval in the first part can still be used. However, if only the child’s marriage registration date is known, the year interval used in the comparison process needs to be modified. In the setting of the former case, the age difference between siblings in the same family is [2,12] years, and the average age of the first marriage for the Dutch people is 27 years. Then, it is possible to get that the year difference between the marriage date of one of the brothers or sisters and the birthday of the other is in [15,25] or [29,39]. The reason for the two different intervals is since one of the children is unsure of the date of birth, then it is impossible to determine which of the two is older. Assuming the child whose birth date is unknown is born earlier, then the year difference between his marriage date and the birth date of another child belongs to the range $27 - [ 2,12 ] = [ 15,25 ]$. Otherwise, the year difference range is $27 + [ 2,12 ] = [ 29,39 ]$. Therefore, the overall range can be set as [15,39].

The core data of person $A$’s relationship can be summarized as PersonA[spouse, child, childbirth date, childbirth place], and for the person $B$ it is PersonB[spouse, child, childbirth date, childbirth place, child marriage date, child marriage place]. For the person $B$, the data of childbirth date and childbirth place might be empty. Convert the person $A$ to the object $A(x_A1, x_A2, x_A3)$, which $x_A1$, $x_A2$ and $x_A3$ corresponding to childbirth date, spouse name, and childbirth place, respectively. Convert the person $B$ into the object $B(x_B1, x_B2, x_B3, x_B4, x_B5)$, which $x_B1$, $x_B2$, $x_B3$, $x_B4$ and $x_B5$ corresponding to childbirth date, child marriage date, spouse name, childbirth place, and child marriage place. Then the application of the attribute matching and new kinship determination process for $A$ and $B$ is shown below.

$$D_{pm}(A, B) = d(Y_1, Y_2, Y_3)$$

$$Y_1 = \begin{cases} 
1, & \text{if } |x_A1 - x_B1| \in [2,12] \text{ and } x_B1 \neq 'empty' \\
0.5, & \text{if } |x_A1 - x_B2| \in [15,39] \text{ and } x_B1 = 'empty' \\
0, & \text{otherwise}
\end{cases}$$

$$Y_2 = \begin{cases} 
1, & \text{if } \text{name\_similar}(x_A2, x_B3) > 95\% \\
0, & \text{otherwise}
\end{cases}$$

$$Y_3 = \begin{cases} 
1, & \text{if } x_A3 = x_B4 \text{ and } x_B4 \neq 'empty' \\
0.5, & \text{if } x_A3 = x_B5 \text{ and } x_B4 = 'empty' \\
0, & \text{otherwise}
\end{cases}$$

$$f(d) = \begin{cases} 
0, & \text{if } Y_1 = 0 \\
\sum_{i=1}^{3} Y_i, & \text{otherwise}
\end{cases}$$

(4.7) (4.8)
4.3. Genealogical Pedigree Construction

\[
\text{New\_kinship} = \begin{cases} 
[A, \text{spouse}_A, \text{child}_B, \text{birth\_date}_B, \text{birth\_place}_B, c.l : A] & \text{if } f(d) = 3 \\
[A, \text{spouse}_A, \text{child}_B, \text{birth\_date}_B, \text{birth\_place}_B, c.l : B] & \text{if } f(d) = 2.5 \\
[A, \text{spouse}_A, \text{child}_B, \text{birth\_date}_B, \text{birth\_place}_B, c.l : B] & \text{if } f(d) = 2 \\
& \text{and } Y_2 = 1 \\
[A, \text{spouse}_B, \text{child}_B, \text{birth\_date}_B, \text{birth\_place}_B, c.l : B] & \text{if } f(d) = 2 \\
& \text{and } Y_2 = 0 \\
[A, \text{spouse}_B, \text{child}_B, \text{birth\_date}_B, \text{birth\_place}_B, c.l : C] & \text{if } f(d) = 1.5 \\
& \text{and } Y_2 = 1 \\
[A, \text{spouse}_B, \text{child}_B, \text{birth\_date}_B, \text{birth\_place}_B, c.l : C] & \text{if } f(d) = 1.5 \\
& \text{and } Y_2 = 0 \\
[A, \text{spouse}_B, \text{child}_B, \text{birth\_date}_B, \text{birth\_place}_B, c.l : D] & \text{if } f(d) = 1 \\
& \text{or } f(d) = 0.5 \\
\text{Empty} & \text{if } f(d) = 0 
\end{cases}
\]

It should be noted that the birth date and place of the child in the new kinship might be empty since the information is not mandatory for the person whose role is the child in a marriage event. Besides, when adding the new relationship data instance to the person A, it also needs to follow the process introduced in the first case of the first part. However, the persons in the new relationship instances are from the marriage event data, so it is imperative to move these persons’ primary data instances from the marriage – person dataset to the birth – person dataset.

- Case 2: If the role of person A is Father or Mother, and person B is Children:

In this kind of situation, it is also to determine whether it is possible to add kinship data about parents to A based on the information of person B. Assume A and B are the same people, then the sequence of events for him must be married first and later childbirth. According to the description in the previous article, it is clearly that the average childbirth year range of the Dutch people after marriage is 1 to 12 years or 12 to 20 years. In addition, the person whose role is the child in the marriage event obviously has the spouse information. The person whose role is the father or mother in the birth event also has spouse information. Therefore, by comparing spouse information, the certainty level of the result can be significantly improved.

In this case, the core data of person A’s relationship can be summarized as PersonA[gender, spouse, childbirth date, childbirth place], and for person B it is PersonB[gender, spouse, marriage date, marriage place, father, mother, birth date, birth place]. Convert the person A to the object \(A(x_{A1}, x_{A2}, x_{A3}, x_{A4})\), which is \(x_{A1}, x_{A2}, x_{A3}\) and \(x_{A4}\) corresponding to the person A’s childbirth date, gender, spouse name, and childbirth place, respectively. Convert the person B to the object \(B(x_{B1}, x_{B2}, x_{B3}, x_{B4})\), which is \(x_{B1}, x_{B2}, x_{B3}\) and \(x_{B4}\) corresponding to the person B’s marriage date, gender, spouse name, and marriage place. Then apply the attribute matching determination process for A and B.
Chapter 4. Solution and Implementation

$$D_{cm}(A, B) = d(Y_1, Y_2, Y_3, Y_4)$$

$$Y_1 = \begin{cases} 
1, & \text{if } x_{A1} - x_{B1} \in [1, 12] \\
0.5, & \text{if } x_{A1} - x_{B1} \in [12, 20] \\
0, & \text{otherwise}
\end{cases}$$

$$Y_{i\in2,3,4} = \begin{cases} 
1, & \text{if } x_{Ai} = x_{Bi} \\
0, & \text{otherwise}
\end{cases}$$

$$f(d) = \begin{cases} 
0, & \text{if } Y_i = 0 \text{ for } i \in \{1, 2\} \\
\sum_{i=1}^{4} Y_i, & \text{otherwise}
\end{cases}$$

$$New\_kinship = \begin{cases} 
[A, father_B, mother_B, birth\_date_B, birth\_place_B, c.l : A] & \text{if } f(d) = 4 \\
[A, father_B, mother_B, birth\_date_B, birth\_place_B, c.l : B] & \text{if } f(d) = 3.5 \\
[A, father_B, mother_B, birth\_date_B, birth\_place_B, c.l : C] & \text{if } f(d) = 3 \\
[A, father_B, mother_B, birth\_date_B, birth\_place_B, c.l : D] & \text{if } f(d) = 2.5 \\
Empty & \text{if } f(d) = 1.5 \\
\end{cases}$$

The birth date and place values in the new kinship data might be empty, since the birth information is from marriage registration. Meanwhile, same as in case 1, when adding a new relationship data instance for the person A, we also need to transfer the primary data instances of the corresponding persons that appear in the new kinship to the birth – person dataset. The remaining operational requirements for adding a new relationship data instance are the same as the process introduced in the second case of the first part.

After this part, all possible kinship data and corresponding persons that exist in the marriage event datasets would be merged into the birth event datasets. In the following process, only persons and relationship data instances in the birth event datasets need to be used.

The Similar Person Searching Between Birth and Death Event Datasets

The third part is to use the information in the death event datasets to screen the existing kinship data from the previous two subsections. After the process steps over the last two parts, each person perhaps already has several relationship data instances with different certainty level. But the certainty level of a lot of data might be deficient, which is C or D. The true value of these low-credibility data is not high since these kinship relationships contain many uncertainty factors, these kinds of kinship information are only regarded as some reference data. Therefore, if it is possible to reduce some of them, the overall quality of the data can be improved.

In the death event datasets, the person whose role is the child has important information about the death date. If it is possible to find the corresponding person of the deceased in the birth
event dataset, then the relationship in which the date is beyond the time of death should be classified as erroneous data. Deleting these inaccurate data can help to increase the overall reliability. In this part, only when the similarity between a deceased person and a person in the birth event data set is very high, they can be confirmed as the same person in reality. In the comparison process, only those who have both parents and spouse-child relationship data with A certainty level would be used. A part of the person in the death events have an additional piece of relative information, some of them are spouse information, and others are undetermined relatives. Through the data processing in the previous two parts, some parts of the person in the birth event dataset might also have two types of kinship data with A certainty level. Afterward, if two persons $A$ and $B$ whose name similarity reaches 95% and also meet the above condition are found, then the core data used to determine their likeness can be summarized as $\text{Person}_A[$birth date, gender, father, mother, spouse, birth place] and $\text{Person}_B[$death date, gender, father, mother, relative, death place]. Also, according to data from the 'Demograph of the Netherlands' [Wik20a], and 'Statista' [Sta20], the average life expectancy of the Dutch in the 19th century was 35 to 45 years and rapidly increased to 50 to 65 years from the 20th century.

Convert the person $A$ to the object $A(x_{A1}, x_{A2}, x_{A3}, x_{A4}, x_{A5}, x_{A6})$, these attributes corresponding to the person $A$’s core data. Also, convert the person $B$ to the corresponding attribute object $B(x_{B1}, x_{B2}, x_{B3}, x_{B4}, x_{B5}, x_{B6})$. Then apply the attribute matching determination process for $A$ and $B$.

$$D_{bd}(A, B) = d(Y_1, Y_2, Y_3, Y_4, Y_5, Y_6) = \begin{cases} 1, & \text{if } x_{B1} - x_{A1} \in [35, 45] \text{ and } x_{A1} \leq 1900 \\ 1, & \text{if } x_{B1} - x_{A1} \in [50, 65] \text{ and } x_{A1} > 1900 \\ 0, & \text{otherwise} \end{cases}$$

$$Y_i \in \{2, 3, 4, 5, 6\} = \begin{cases} 1, & \text{if } x_{Ai} = x_{Bi} \text{ and } x_{Ai}, x_{Bi} \text{ are not empty,} \\ 0, & \text{otherwise} \end{cases}$$

$$f(d) = \begin{cases} 0, & \text{if } Y_i = 0 \text{ for } i \in \{1, 2, 3, 4, 5, 6\} \\ \sum_{i=1}^{6} Y_i, & \text{otherwise} \end{cases}$$

$$\text{Death date for the person } A = \begin{cases} \text{Death date of the Person } B & \text{if } f(d) = 6 \\ \text{Nothing} & \text{otherwise} \end{cases}$$

If it is possible to determine the death date of the person $A$, the next step is to delete the data instances of $A$’s kinship, which event time is later than the date of death. In essence, all the relationships that could be deleted are the person $A$’s spouse and child kinship data instances. Therefore, the relationship instances with the same information which belong to the corresponding spouse and child also need to be removed. Meanwhile, it should be noted that when deleting the kinship data of the person $A$, the identification of some kinship data instances should be modified. Because except for the first relationship data instance of $A$, the identification of the remaining data instances is composed of $P_{id}$ of $A$ and a suffix. The suffix
corresponded to the number of the sum of the person A’s kinship relationship when the data instance was added. If some of the kinship data instances are deleted, then the number of suffixes in the identification of the remaining data instances needs to be modified in time.

In conclusion, through the processing of the above three parts, all valuable information in the marriage and death event datasets can be integrated into the person and relationship dataset of the birth event. In the subsequent parts of the project, it only needs to use the data in these two datasets to provide users with genealogical pedigree search and visualization functions.

4.4 Genealogical Pedigree Visualization

According to the survey, the most reliable and widely applicable genealogy visual layout is the tree structure layout. Therefore, in this project, genealogical pedigree visualization is also based on the classical tree structure. The main reason for not choosing time concentration or radial layout is that both structures require detailed temporal information in the original data resource. The situation of datasets and data will be fully introduced in the next section. Furthermore, because the data information contained in each character object is not complicated, the use of nodes to store messages can fully meet the current requirement.

The traditional root tree structure has certain limitations when displaying the complex relationships between genealogical characters, so the visualization structure in this project is based on the hourglass tree. Figure 4.12 shows a basic example of the genealogical pedigree structure.

![Figure 4.12: Layout of genealogical pedigree visualization](image)

In the dataset containing the genealogical data, although each person’s roles and relationship composition is different, the substantive information of each kinship data instance can be summarized as the same structure. It is a couple with a child. Meanwhile, several different instances of relationship may hold information about different children of the same couple. In the visualization structure, the couple and their children are connected to the root node of an hourglass tree. Every individual node except the root node can connect two different hourglass trees. According to the kinship relationship, the parents will be placed above the children. After that, all the hourglass trees are combined to form a complete genealogical pedigree. Rectangle and ellipse nodes are used for males and females in the structure, respectively. If the gender is unknown, diamond nodes are used. Meanwhile, different edge formats can be used according to...
the uncertainty of the relationship. When the certainty level of the relationship data is higher, the edge will be more solid. Otherwise, the edge line will be more dotted. Besides, due to the root node connects everyone in a family and each relationship data is also associated with family members, then the certainty level of the kinship relationship can also be stored and displayed in the root node.

4.5 User Interaction

The ultimate goal of the project is to provide users with a search function for genealogy. Therefore, user interaction must be built. Firstly, an interactive user interface must be provided. Users can enter the relevant information of the person who they want to check about the genealogical pedigree. Then the interface will return and display all the people in the system related to the information entered by the user. Depending on the amount of the information supplied by the user, search results with varying degrees of accuracy can be provided. For example, if the user provides only a part of the name information, the returned result may include many people, and it is difficult to sort the search results. If the user can provide detailed information, such as full name, date of birth, birth place, spouse, parents, and so on, then the scope of the search results can be greatly narrowed. Then the result of people can be sorted according to the different degrees of matching the search information. After the user selects any person in the result list, it is necessary to display the direct kinship information and the corresponding genealogical pedigree, which belongs to the chosen person. Furthermore, the user can select any person in the pedigree chart. If he has kinship that does not appear the current pedigree, then these relationships need to be added to the genealogy, and the previous visualization result should be updated and re-displayed to the user. The below figure shows a use case for user interaction.

Figure 4.13: Use case for user interaction
The search keywords that need to be provided by the user is the name, gender, date of birth, year of life (19th century or 20th century), place of origin, place of life, spouse, father’s name, mother’s name, and child’s name. Among this information, except for the name of the searched person, the rest is all optional input. Regarding the name, the user can input the full name of the person who they want to search or only a part of the name, such as the last name or first name. Afterward, the searched person and all relevant information can be converted into a person object with ten attributes. For the attribute that the user does not enter the information, the corresponding values are empty. Then for each person in the dataset, through their necessary information and the kinship data with A certainty level, they will also be established as a person object with the same ten attributes. The matching degree computation between two persons is shown below:

\[
S(A, B) = s(Y_1, Y_2) = \begin{cases} 
  \text{similar}_\text{name}(x_{A1}, x_{B1}), & \text{if } \text{similar}_\text{name}(x_{A1}, x_{B1}) \geq 0.4 \\
  0, & \text{otherwise}
\end{cases}
\]

\[
Y_1 = \begin{cases} 
  \text{if } \text{similar}_\text{name}(x_{A1}, x_{B1}) \geq 0.4 \\
  0, & \text{otherwise}
\end{cases}
\]

\[
Y_2 = \sum_{i=2}^{10} l(x_{Ai}, x_{Bi}), \quad l(x_{Ai}, x_{Bi}) = \begin{cases} 
  1, & \text{if } x_{Ai} = x_{Bi} \text{ and } x_{Ai}, x_{Bi} \text{ are not empty} \\
  0, & \text{otherwise}
\end{cases}
\]

\[
f(s) = \begin{cases} 
  0, & \text{if } Y_1 = 0 \\
  Y_1 \times Y_2, & \text{otherwise}
\end{cases}
\]

The \( f(s) \) value is the final similarity level between two person objects, \( A \) and \( B \). The name similarity comparison also uses the Levenshtein distance algorithm. The Levenshtein distance algorithm returns a decimal between 0 and 1. The higher the similarity of the name string, the higher the returned value. The name plays a key role in finding similar people. If the name similarity is too low, then it is not necessary to compare the two persons. Only when the name matching degree is higher, the effect of the remaining attribute matching will be more significant.

### 4.6 Implementation

The programming language used to implement the entire project is Python. The overall programming system is composed of three parts: preliminary data preparation, genealogical data processing, and a user interface with a genealogical pedigree visualization. The first part is mainly to convert raw resources obtained from BHIC into an appropriate format that is suitable for data processing. The second part is to merge the useful genealogical data in three different registration events and create the datasets for person characters and corresponding kinship relationships. The third part is to provide users with search and interactive functions as well as display genealogical pedigrees to users. The implementation structure is shown in Figure A.3 below.
4.6. Implementation

4.6.1 Preliminary Data Preparation

The raw data material provided by BHIC is in the XML documents. The essence of the XML file is a tree-like structure that is easy to interpret and supports a hierarchy. The largest top-level element is called the root, which contains all other sub-elements. Elements are defined by start and end tags. The characters between the start and end tags build the content of the element. The attribute consists of value pairs that exist in the start tag or empty element tag. To process XML documents in Python, it can be achieved by using the lxml-XML tool library. Firstly, import the ElementTree package, which has the function of reading and manipulating XML. In the ElementTree library, it can use the parse function to directly parse the XML file into an element tree. But the parse function needs to parse and cache the entire XML file at once. In this project, the average size of each XML file provided by BHIC is around 5 to 6 gigabytes. If the entire XML is parsed into an element tree at once, it will directly explode the computer’s memory. For this reason, it has been tested, even if the cloud processor using Google Colaboratory’s 25GB RAM still cannot meet the requirement of parsing the complete element tree at once. Therefore, in this project, iterparse function was
used, which can help for stepwise parsing. iterparse works similarly to parse in that it also builds a tree. But it safely rearranges or deletes parts of the tree after it has been parsed. For example, to parse a large file, it can divide the entire tree structure into element parsing, parse only a single element at each time, and delete them immediately after processing the elements. The iterparse function activates the processing of an element by a start or end event, which means the start or end tag of the element. The disadvantage of this operation is that it can only get each element, the content of the element and the attribute of the element, but it cannot store the hierarchical relationship between the elements, for example, which child elements are contained in an element. The feasible solution is that first parse the tag names, attributes, and values of each element, and store the value and attributes of each tag under the tag name, then associate with the name of each tag and the structure of each element in each XML file (introduced in Section 4.1.2). The data can be reintegrated into the CSV format that satisfies the project requirements.

### 4.6.2 Genealogical Data Processing

The implementation process of this part follows the logic and theory introduced in the previous section. It should be mentioned that the dictionary was used in the Python programs for caching the data from CSV files instead of the list. Because comparing with the list, the dictionary has more dynamic query capabilities. When querying relationship data instances for a specific person, the dictionary can help locate the corresponding position faster.

### 4.6.3 User Interface With Genealogical Pedigree Visualization

The user interface was implemented by using the Tkinter [Shi13] tool library. Tkinter is the de-facto standard GUI (graphical user interface) package for Python. Tkinter is not the only Python GUI toolkit. However, it is the most commonly used. Tkinter provides functions that can quickly build an interactive interface, which can help developers rapidly construct a variety of functional modules, such as input entry, button items, display entry, and so on. At the same time, it can help developers quickly with typesetting. At the same time, these functional modules can be easily integrated and typeset.

The genealogical pedigree implementation is based on the Python Graphviz library. Graphviz is an open source graph visualization software. Graphic visualization is a method to represent structural information as abstract graphs and network graphs. The Graphviz layout can obtain graphic descriptions and structures in a simple text language and makes graphs in useful formats, such as images and SVG for web pages. It is very suitable for helping to build genealogical pedigrees in the current project. It can provide nodes with various shapes and colors, and also edges with different types of thickness. Meanwhile, it is also really helpful in setting different hierarchies for the nodes.
4.6.4 Result Example

The screenshots below show an example of using the current implementation to illustrate the resulting program to query and generate a genealogical pedigree. The user can search for an ancestor in his family based on some information he knows. After that, all possible people matched in the current dataset are displayed in the upper left box. The first person is one with the highest matching degree in all results. Then the user can select a person in the result list, and the kinship relationships related to this person will be displayed in the text in the display box below. When the user selects a person in the search results, in addition to the kinship description in the text form, a genealogical pedigree is also displayed to the user. It is a picture file and opens directly after the user makes a selection. In the example in Figure 4.15, the person Pitronille Naeijkens was selected. According to the resulting genealogical pedigree, in the current genealogical datasets, she only has one type of kinship relationship.

The user can select one person who appears in the current genealogy from the display box on the lower right. The genealogical pedigree will be refreshed and re-displayed to the user. If the selected person has a kinship that is not included in the current genealogical pedigree, the new kinship and related people will be added to the updated result. If all the kinship of the selected person already exists in the current family tree, there will be no change in the result. Some examples are shown in the following sets of pictures.

Figure 4.15: The user interface for searching

![Figure 4.15: The user interface for searching](image)

Figure 4.16: Genealogical pedigree example 1.

![Figure 4.16: Genealogical pedigree example 1](image)

In the first example, the person Corneille Joosen was selected. He has a relationship that does not appear in the old genealogical pedigree. Then the new kinship was added into an updated genealogy.
Because the certainty level of the second kinship of Corneille Joosen is really low, the spouse of Corneille Joosen in the second kinship was selected. After that, the kinship with A certainty level was displayed. Actually, this A level kinship is the source of Corneille Joosen’s C level kinship. From the names, Corneille Joosen and Corneille Goosens are very similar, and the birth date interval of the children in the two relationships is within a reasonable range, which is two years. However, the child’s birth place and the names of Corneille Joosen’s spouse in the two relationships are different. Therefore, the certainty level of the possible kinship found for Corneille Joosen is pretty low. Meanwhile, it can be seen that data with different certainty levels correspond to edges in different formats.

In the third instance, the person Antoinette Joosen was selected. There is another person named Antonetta Joosen who might point to the same person with Antoinette Joosen. Their names are very similar, and the slight difference can be considered as an input error in the original record. Meanwhile, the birth date between Antoinette Joosen and Antonetta Joosen’s children is 30 years, which is a very reasonable interval. There is a possibility that Antonetta Joosen and Antoinette Joosen is the same person. But they lived in different places, so the certainty level of that they are the same person is B instead of A. If the user selects a person in the result list and the genealogical pedigree does not have any update after re-displayed, it proves that all the kinship relationships contained in the person are already in the current genealogy. In the following image, it shows a more complicated structure with more persons and relationships.
More examples of genealogical pedigree are shown in the appendix A. In these complicated genealogical structures, in order to ensure that the same generation is placed at the same level, the unavoidable problem is the intersection of edge segments. This is the biggest limitation of the current solution that can be observed intuitively. The reason for this problem is that in the current implementation, it used the tool library Graphviz to build the visual pedigree diagram. Still, in this tool library, the connection layout and setting of the edge cannot be customized. Therefore, it will cause some intersecting edges. The solution to this aspect will be one of the focuses of future work.
Chapter 5

Result Discussion

This chapter mainly introduces the results of the project, the problems and challenges encountered during the implementation, the discussion of the results, and the possible future work of the project.

5.1 Answer of Research Questions

Through the relevant information and knowledge research and combined with the analysis of the actual situation of the project, now it is able to answer all research questions.

- **What kind of data does BHIC provide?**
  In summary, the BHIC provides historical multi-type civil registration records data of the Noord-Brabant area. It includes many types of civil registration events such as birth, death, marriage, military service, etc. The data resources contained in each event record are the event registration time, location, important person, and various relationships. The kinship data between the persons can be found in parts of the records. However, these data are only the most foundation kinship information. These relationships exist independently and are not related to each other. There are only two types of relationships which are parents with children and spouses, and every person only contains one relationship. These kinship data can only be regarded as the most basic genealogical data. In other words, they can be the most fundamental part of the genealogical pedigree. But it is not enough to construct a complete and complicated family tree by relying on these kinship data.

- **How to visualize the genealogy?**
  Many different methods can be used for genealogy visualization. The most commonly used in contemporary society are tree layout, radial layout, and time concentration layout. Each of these modes has its own advantages, and they also apply to different situations. In the current project, the visualization function is based on the tree structure. This is the most traditional and reliable way to build pedigree charts. The advantages of the tree
layout are clear structure and simple construction algorithm. The disadvantage is that the data that each node can display is limited, and also the genealogical timeline information cannot be highlighted well. The amount of primary data for each person in our current genealogy is not large, and the temporal information is not enough. Therefore, the tree structure layout can fully meet the functional requirements of the current project, which is a very appropriate option.

**What are the necessary information to establish the genealogical pedigrees?**
The essential points of establishing a genealogical pedigree are people and the connection between people, which means kinship relationships. All persons in the same genealogy have direct or indirect links, that is, they are close or distant relatives of each other. Therefore, for the establishment of a complicated genealogical pedigree, it must first determine the most fundamental kinship relationships. In most cases, the starting point of genealogy is a person or a simple kinship such as father-mother-child. These persons are the initial characters. Then determine the kinship relationships of the initial characters. Then save relationships that do not exist in the current genealogy. Then repeat the previous step to find new kinship information based on the new relative persons. It is much easier to determine the closest and direct kinship than to find complicated multi-person relationships. In fact, each person in the genealogy can be regarded as a node, and the kinship is the edges between the nodes. The graph that connects all the nodes through the edges is the genealogical pedigree.

**How to retrieve genealogical data from a huge database?**
From the result of the previous research question, it is clear that the core of the genealogical data it the kinship information. Meanwhile, immediate family relationships are the most useful data. Immediate family relationships include parents with children, spouses, and siblings. The spouses and sibling information are both parts of the parents-children relationship. That means as long as the kinship information of each person’s parents and children is available, then it is possible to build a genealogical pedigree. Thus, when mining and retrieving data from a vast database to construct a genealogy, the primary goal is everyone’s child and parent information. If there is already a wealth of immediate family relationship information in the database, then these data are beneficial and can be used directly. If there are distant relationships, then this information can be collected and processed to obtain valuable data. However, if the information of each person’s immediate family is not sufficient to form a multi-person pedigree, then it is necessary to mine the possible relationship data from the database. In this situation, it is necessary to collect relevant personal information and some specific temporal information for each person, such as place of birth, date of birth, time of marriage, etc. Combining this information with everyone’s existing kinship relationships, we can find hidden kinship data that may exist in the database.

**How to set up a computing model to measure the uncertainty of the relationship between people based on people’s basic information?**
The actual situation of the current project is that the basic information of each person obtained from civil registration events is different. Meanwhile, most people have only
one instance of kinship data. Therefore, in order to build a complicated multi-person genealogy, it is necessary to find more kinship relationships for each person. Because these found kinship relationships do not have reliable documents or data as references, their reliability must be measured, which is the level of certainty. The method for finding hidden kinship for a particular person is to check whether the person appears in other relationships, that is, to determine whether some people that appearing in two different kinship data instances are the same person in reality. The way to judge whether two characters are the same person is to compare their attributes and the temporal information of some important events related to them. The attributes of each person include name, gender, hometown, and so on. The temporal information of important events covers the date of birth, the date of birth of the child, the age of marriage, and so on. These data are reliable data that are given by BHIC resources. Based on comparing the consistency of the attributes of the characters, and the rationality of the order in which some critical events occur, the similarity between the characters can be obtained. Depending on the similarity, the certainty level of the data can be determined when adding new kinship. When the similarity is higher, the certainty level of the added new kinship data is naturally higher, otherwise the lower the certainty level.

5.2 Limitation

During the implementation of the project, it has been found that the current solution has some limitations. Some of them are due to resource constraints, and others are due to defects in our solution. However, owing to time constraints, these limitations cannot be fully resolved. But solving these limitations can be the direction of future work. The limitation problems are mainly the following.

- The accuracy of the person matching model cannot be detected.

In the research process, that most genealogy developments and researches are based on reliable family documents. It contains detailed family history information such as birth, death, and migration of people. They can easily determine the kinship between people, which is the basis of building a genealogy. But for the current project, it needs to create a genealogy for a certain person whose kinship information is scarce. So there is not much reliable relationship data available. However, most of the researches are on whether there is a kinship relationship or blood relationship among the personnel, which is medical studies based on genes or genetic factors. Therefore, the solutions provided by these materials cannot be applied to the current project.

Combining the actual situation of the project, a function was proposed for comparison and determination of similarity between persons. The whole process is based on the reliable data of the person to establish the person attribute features, then set a comparison model for attribute features. However, some of the parameters in the model are derived from our assumptions, which means there is no reliable resource to prove them, for example, the average childbearing age of Dutch families and the average age interval of each child.
And what required is relevant information in the 19th and early 20th century. At that
time, demographics and censuses were not well-developed, and the format data document
storage was not uniform. Besides, these questions belong to a relatively small research
direction. Thus, there is not much research or study on Dutch historical data in this
regard. It also makes us unable to find parameters suitable for the project model. In
addition, in reality, a person’s life may have many changes, such as changing the name or
living in a place that is far away from the birth place. The impact of these factors is not
well considered in our model. At least, the original data resources cover only around 100
years starting from the 19th century to the early 20th century, and only in the area of
North-Brabant. For a large family, the scope of area and time is both too small. In such a
limited range, the genealogical data information that can be found is naturally restricted.
Although we set different levels of certainty for the obtained kinship relationships, but
there is not a reasonable way to help to verify the accuracy of these kinship and certainty
levels, assuming the user does not know any information about his family. Perhaps with
the help of our method, the user can find the genealogy pedigrees that may be related to
his family. Still, because of the lack of reference to side evidence, the similarity between
the results and the actual situation cannot always be measured.

- **The current data processing time exceeds the acceptable range.**
The biggest problem encountered in the implementation of the project is the program
operation time. As mentioned in the previous part of the article, the method that used
for collecting genealogical data is to find and save all the kinship data instances for
each person in advance. The advantage of this method is that it can save the time of
generating the genealogy pedigree. But the disadvantage is that the data preparation
time in the early stage is exceptionally long. The data preparation of the project is
divided into three parts. In the first part, the person data instances in the birth event
datasets need to be compared in pairs. The Birth – person table obtained from the
original BHIC resource has 1.8 million instances of personal data, and each of them has
at least one kinship relationship. Assuming that the total number of data instances
in the table is \( n \), then the complexity of pairwise comparisons is \( O(n^2) \). Furthermore,
in the remaining two parts, it needs to compare the persons in the Birth – person
dataset with persons in the Marriage – person and Death – person datasets, respectively.
The total number of person instances in the Marriage – person or Death – person
table is greater than 1.8 million. Thus, the complexity of the second part and the third
part is at least \( O(n^2) \). Then the overall complexity of the data preparation part is
\( O(n^2) + O(n^2) + O(n^2) = 3O(n^2) = O(n^2) \), and the \( n \) is more massive than 1.8 million.
This caused the running time of the implemented program to be too long and beyond
the acceptable range. We have conducted tests on Google Cloud using 25 GB memory
servers to perform the program execution. After 30 hours, only less than 30% of the data
processing is completed (Google only allows the use of servers up to 30 hours at once). But
using our local computer for testing, after 48 hours, only less than 40% completed, and
the running speed is getting slower and slower. So far, it cannot complete the processing
of all data, and only a part of the kinship data can be collected. However, according
to the results that have already obtained, it can prove that our solution is reasonable
and feasible. It can generate the data with expect and also can be used to create the genealogy pedigree with the designed layout.

- **User interaction function is not complete.**
  The biggest limitation of the user interaction function is that it cannot allow users to operate on the generated genealogy pedigree chart directly. Because in the current implementation results, the final pedigree results will create a picture format and display. The user’s interaction with the genealogy is done through the user interface. The user needs to select the person item in the interactive user interface, and then the program will update the genealogy according to the user selection then re-display the new result to the user. Such a solution is not very user-friendly. The best way is to provide interactive user functions directly on the pedigree chart. For example, the user can directly select a person node in the pedigree chart, and then the pedigree chart can directly refresh the results. The reason causing this result is that in the project implementation phase, we really want to maintain the consistency of the program, so that each part of the project was implemented using the Python programming language. However, currently, there is not any suitable Python tool library that can be used to help create pedigree charts that provide user interaction. In most contemporary cases, the best interactive data visualization programs are built with the JavaScript programming language. Therefore, using the JavaScript programming language to construct genealogical pedigrees that can directly provide user interaction functions will be the focus of future work.

  Furthermore, in the current user interaction interface, it chose to cache the data table of persons and relationships into the program. Currently, the dataset used in the program was just the test dataset. Due to the problem of program running time, it cannot complete the genealogical data preparation part. Even if the size of the test dataset is much smaller than the final full version, it already took quite a long running time on starting the program and help users find relevant personnel. Meanwhile, if it is the full version of the genealogical datasets, it should contain more than 1 million person data instances and millions of kinship relationship data instances. If put all these data in the cache of the program, it will have high requirements on the computer memory, and it will also take an extremely long time to start the program. Thus, if we want to make a user-friendly program, then this issue needs to be solved.

### 5.3 Future Work

The future research direction of the project is focused on optimizing the limitations of the current solution. In the previous section, it introduced in detail the limitations of the current project implementation results. Based on these issues, there are some designs and plans for the future work direction of the project.

- **The first is to increase the types of data used in the project.** The current solution only used three types of civil registration event records as data resources, which are birth,
death, and marriage. Using only these data as a person’s attributes to find the same person appearing in different event records is very limited. If more available data types such as military service registration and immigration registration are introduced, then it is possible to find essential information belonging to the same person. In this way, when processing the person comparison, more attributes or features can be adopted to improve the accuracy of the results. Besides, the use of data might not be limited to Noord-Brabant. The migration of the population is very ordinary. When a person moves to another city or region to live, all event registration information related to him will also be kept by the new region. Using only local data resources might cause many possible kinship data to be missed. Therefore, if the data from the entire Netherlands can be used as the resource, it will be more helpful to the project. Furthermore, find some family that has detailed information records of the family history and use these data to make a corresponding genealogy, then using our program to create the genealogy pedigree for the same family. By comparing the results of the two genealogies, it is able to verify the performance of our solution. If there is a consistent or similar part, it can prove the rationality of our method. At the same time, it can also help us check the accuracy of the results. Based on the accuracy results, problems in the solution method can be discovered and improved.

• The second point is to deal with the computation complexity problem in the solution method. The previous chapter introduced that the current computation complexity is \(O(n^2)\), and the value of \(n\) is tremendous. In the implementation result, the program running time is completely beyond the acceptable range. Therefore, think about how to reduce the running time is an unavoidable question. There are two ways to reduce the computation complexity of the solution. The first is to modify the solution to find a new method to search hidden kinship relationships. The other is to reduce the value of \(n\). The first aspect will be the main direction of future research. So far, we have not come up with a better idea. The second aspect can start by dividing the people in the original datasets. Because in the process of finding the same person, the first thing to do is the person name comparison. The current method needs to compare a certain person with the rest of the people in the dataset, so the value of \(n\) will be immense. If people with similar names are directly classified at the beginning, then when looking for similar names, it only needs to be compared among people in the same category. To classify the person names, it requires an algorithm with a complexity of \(O(n)\). Then the complexity of comparing names among the same type of person is \(O(m^2)\), which \(m\) being the number of total person data instances in that particular type. Still, here the value of \(O(n)\) and \(O(m^2)\) will be much smaller than the \(O(n^2)\). It can help us achieve the purpose of shortening the running time of the program.

• The third point is about the improvement of user interaction. The current user-interactive interface and function design are not very user-friendly. And many missing functions still need to be completed. The focus of future work is to develop an interactive user interface in the web form to replace the current graphical user interface. The advantage of using a web-based interactive page is that the genealogy pedigree can be directly set on the page. In this way, the user can directly perform operations on the pedigree chart, which
is a more convenient and fast mode for users. In addition to the existing interactive channels, new interactive functions can also be added. For example, after generating a genealogy pedigree for the user, the user can modify the genealogy according to the family information he knows, such as deleting some wrong kinship relationships or adding new relationships to the genealogy. After that, data of people and kinship in the corresponding datasets also need to be modified depending on operations.

Besides that, the issue of how to use the final data was also mentioned in the limitation introduction. If the final person and kinship data tables are cached in the user interaction program, then it will be a big test for the computer processor memory, because each table contains more than 1 million data instances. Even if the program can be executed, the running and replying speed of the program will be extremely slow. In actual operation and testing, such a running speed is unacceptable. Therefore, the best solution is to make a database model for the final data result and process the search-related information entered by the user into the form of SQL. Use SQL to retrieve the required data from the database, then display them to the user and generate the corresponding genealogy pedigree. This method not only reduces the unreasonable steps of caching data in the program but also speeds up the process of searching data. Therefore, this direction can also be used as the focus of future work.
Chapter 6

Conclusion

In general, the most crucial part of the entire project is preliminary data preparation and genealogical data construction. These two parts also consumed most of our time and concentration. The current solutions and implementation results can help users complete tasks of genealogical data determination and measurement of uncertainty in kinship. But there are still many aspects that need improvement and modification. The ultimate goal that proposed from the beginning of the project was to make a universal solution that can provide a method that is suitable for most users in most situations, rather than only helping one or two specific families to find genealogical information. The data that currently using is only regional data for the Noord-Brabant area, and with 140 years around the range, which is from the 19th century to the early 20th century. The applicable scope of the data resource itself is relatively narrow. Therefore, the genealogy that can be produced has great limitations in terms of reliability, comprehensiveness, and complexity. However, the solution we have adopted is not just for these specific data with constraints. If there are richer data sources, the current solution does not only continue to apply to new resources but also can improve the current solution and lead to better results. Meanwhile, due to time issues and defects in the early design of the project, the current implementation of the visualization part does not make us very satisfied. It can be considered as only a testing result of genealogical data, but not as a very qualified result that can adequately meet requirements from real users. Thus, this aspect needs to be further improved.

In summary, it is a very challenging project. The project involves many aspects of research and investigation. It is not only about data analysis and data processing but also contains a lot of demographic and genealogical knowledge. The creation of genealogical pedigrees and the collection of family information are totally new fields for me, so I encountered many difficulties and problems in the process of exploring solutions and implementations. But I tried my best to solve and overcome these issues and challenges. Through the development and practice of this project, it has dramatically helped with personal ability improvement.
Bibliography


Appendix A

Examples of genealogical pedigrees

Figure A.1: Example 1
Figure A.2: Example 2

Figure A.3: Example 3