Automatic Combining and Linking of Genealogical Records

By Thomas T. Wetmore IV

These are quick notes put together to try to give an overview of ideas on how automatic combining of genealogical records can be done. Disorganized and scattershot.

Introduction

Genealogical research is the quest to discover individuals who lived in the past. All available evidence that may apply to the individuals is gathered, and then the evidence is used infer them. A researcher makes conclusions by laying out person records extracted from the evidence, and then grouping the records into sets he believes represent individuals from the past. The process is based on good practices and experience.

Every year more and more genealogical data becomes available in digital, searchable form, easily accessible over the world wide web. Researchers can collect more evidence more quickly than ever before. So much evidence is now available that we should anticipate two major changes in the technical side of genealogy.

The first change is in genealogy programs. In the past programs were geared to record information about final individuals. Collecting evidence and making conclusions was done with little computer support. Newer programs can hold the evidence, the person records, and the individuals and families inferred from the evidence. And we can now anticipate programs that will support the process of combining and linking person records into individuals and families. Not only will the researcher be able to apply the manual process from within the program, but the program will apply statistics to the groups formed by the researcher and also be able to propose good groupings using internal algorithms. These algorithms make it possible to semi-automate the combining and linking process.

The second change is in services provided by organizations that make genealogical records available over the web, including the LDS, Ancestry.com and FamilyLink. These organizations have access to massive databases of evidence that has been indexed into hundreds of millions of records. Many real individuals are represented multiple times in these records, but it is now up to the researcher to gather those records and make the decisions about which ones refer to the same individuals. But it is now feasible for the organizations to run automated combination and linking algorithms to do most of the work of grouping person evidence records into individuals. Such a capability will revolutionize genealogical research. Imagine searching for a family on the 1920 census, and when found, having the same family instantly returned on the 1910 and 1930 censuses.

The purpose of this document is to consider the possibility of creating algorithms that can accomplish person record combining and linking on a massive scale, algorithms that result in the discovery of millions of individuals. There is an analogous system in existence today, created by a company that has extracted hundreds of millions of person records from the world wide web about employees of companies. Many individuals are represented many times, sometimes thousands of times, in the extracted records. Algorithms used by the company combine those hundreds of millions of records to a few million individuals and then link them to their companies. These algorithms are efficient and have a high accuracy level. The author of this documents knows a lot about those algorithms because he designed and implemented them.

Overview of the Combination Process

Combination is the process that takes person records extracted from indexed record sources and groups them together into sets that represent (or are likely to represent) the same real individual. The first part of the combination process is preparing the records for combination, and the second part is the combination itself. Here is a brief outline of what must be done to prepare person records for combination, followed by an outline of how combination can be structured.

Preparing Person Records. A person record is a record with contents designed for use in combination. A format is described later. Person records must be created from all the indexed databases one intends to include in a combination. These would include databases that index households from census records, and databases that index passengers from ship arrival records. Each database has its own indexing data already in its own format. During this step the data from all sources are converted to person records. This results in another massive database of person records, all in a consistent format, and taken from all sources to be used in the combination.

Indexing the Person Records. After creating the set of person records they must be indexed on all their internal fields so that later steps can quickly retrieve sets of records that match different indexing criteria. Indexing technology has reached the point that this is a straightforward task. Systems such as Lucene can handle databases of this massive scale.

Preparing the First Generation of Group Records. Combination is a process that combines sets of person records into larger and larger groups. At the end the final set of groups represents the final set of combined individuals. The process starts with person records and ends with a smaller set of group records. It makes the combining algorithms simpler if they are always dealing with group records. So this step simply creates the initial set of group records, one for each person record. The first generation of group records are also indexed by using the indexes on their contained person records.

Once the person records and first generation of group records are created and indexed the combination can be done. At its most basic combination requires the comparison of zillions and zillions of records. Effective combination strategies must be designed to avoid the tyranny of the O(n2) performance inherent in these comparisons, or combination will take zillions of years to run.

The most effective way to run combination is as a series of phases in which the number of comparisons is managed at all times. The structure of each phase is as follows:

Gather Batches of Group Records to be Considered by this Phase. The input to each phase is the set of indexed group records that were the result of the previous phase. Each phase will work on this set of records and produce a smaller set of group records for the next phase. There are always too many group records to consider at the same time, so phases are structure to run on a series of batches. The indexes on the group records are used to gather the batches. For example, a batch might consist of every group with a first name from a specific closed nickname set with surnames starting with 'J' who were born in the second decade of the 19th century.

Keying Group Records and Bucketing. The second part of each phase is bucketing the group records. This is done for each batch of group records seprately. Bucketing consists of creating a key for each group record, creating a bucket for each key, and placing all group records with the same key into the same bucket. This is an O(n) algorithm. Bucketing is the key (pun intended) to controlling the O(n2) tyranny.

Comparisons within Buckets. Comparisons are done only after bucketing and only between records that are in the same bucket. Some phases don't require combinations at all if the statistics on the keys are such that is overwhelmingly probable that all records with the same key refer to the same individual. In this case the records in the buckets are simply joined together with an effective

comparison cost that is O(n). If phases with this property are run early it combination it has the benefit of reducing n when it is the largest without the need for record by record comparison. However, this is not a condition that applies very often, and in most phases the records within each bucket must be compared together in some pair-wise fashion. If the keying and bucketing step has created a large number of buckets with average size much smaller than the number of records in the batch this results in a great reduction in the number of comparisons required. If there are n groups in a batch, if those n groups are fairly evenly distributed across b buckets, then the performance of the algorithm is (O(n) + b*O(m2)) where m is equal to n/b which is obviously less than n. The critical design issue of each phase is to assure that b be as large as possible to that n/b is much less than n. The comparisons within buckets decide when group records refer to the same final individual; when they do the group records are combined into larger groups. At the end of each phase the groups that have been combined are removed from contentions and the new groups are indexed in preparation for the next phase.

Previous work on combination algorithms have shown that the structure of the phases and the ordering of the phases can have great impact on the overall performance of the algorithms. Much of the critical work in making combination feasible lies here.

After all phases have been run the result is a final list of combined group records. These group records represent the final individuals. All person records within the final group records are believed to refer to the same real person.

Linking Algorithms

Linking deals with relationships between persons, groups and individuals. This is the only part of the genealogical application that is meaningfully different from the other application, so the only part that needs more development. At high level there are some obvious points. When groups are combined the other groups that contain the person records those combined groups are related to should also be combined. This has serious performance implications – whenever it is possible to combine groups without doing comparisons there is a great benefit. Linking may be as simple as implied here, but I'm not yet convinced that there are not some lurking gotchas.

Over and Under Combinations and Other Errors

At the end of the combination phases there is a final set of group records, each assumed to represent a real individual. There are errors that may have occurred that make this assumption incorrect. Here are the main errors.

Over-combination. Over-combination occurs when a final group holds more than one individual. The combination phases went too far in combining groups.

Under-combination. Under-combination occurs when more than one final group refers to the same individual. The combination phases did not go far enough in combining groups.

More Complex Cases. More complex cases can occur when the person records for multiple individuals are distributed into a number of groups simultaneously.

In the employee application we found we could tune the algorithms to both equalize the amount of over and under combination and to reduce the overall amount of errors.

However errors must occur when combining data as sketchy and as diverse as genealogical records. This must be expected and even embraced from the beginning. The final person groups should never be treated as definitive individuals by the organizations doing the combination. Instead the different person records that made up the individuals should be made available to researchers as a list of indexed

records from their original source databases in the context that these are the person records most likely to refer to the same individual. This alone is enough to revolutionize genealogical research. Maybe in the future these algorithms can be used to reconstruct the tree of mankind, but at present they should be limited to providing valuable services to researchers.

It is possible to compute statistics for each of the combination events that build the group records. The format I recommend makes it possible. This allows metrics to be given to the likelihood that any pair of person records in the group are the same individual. Of course all these likelihoods should be quite high anyway based on the combination algorithms that brought them together. Having this data available can be used effectively in user interfaces which can bring together the records of near certainty of being the same individual. Say at the first level of inquiry only the records most likely to be the same individual are presented. The interface can then allows a wider breadth of records to be shown. The researcher never sees the combined individuals as an entity; he only sees a list of evidence records that are very likely to refer to the same individual and in the majority of cases does. As far as the researcher is concerned it's pretty close to magic.

Experts

The comparison phases compare some pretty fuzzy attributes, including such things as names and places. Names are spelled differently, they have different forms in different languages, they have short forms and diminutive forms, they belong to nickname sets, and so on. Names have different statistics in terms of their occurrence within different populations at different times. All these issues should be taken into account when comparing names.

In the other application I was involved with it proved indispensable to encapsulate all kinds of special knowledge about these fuzzy properties in software entities we called experts. This approach allowed a clean separation in development between the combination algorithms and the expert algorithms. As needs arose in the combination software new interfaces to the software experts were added and implementation of those interfaces were developed independently. Though I was the designer and implementor of the combination algorithms I was also one of the authors of the name expert.

Another area where a software expert is required is in the geographic and location area. Person records contain location information in many forms. Places may have one or many components (e.g., city, county, state, country, province, region, etc), the same location may be described in many ways (e.g., with or without a county, with or without a state, different forms of the city or village name). And beyond these simple issues are issues of how place names are expressed in different languages and how they have evolved over time and through history. For the combination problem being discussed here the problem to be solved can be expressed in very simple terms: given two strings that represent places, what is the likelihood that they are the same genealogical place. Easy to express but potentially very difficult to implement. It is mandatory to have a good expert in place for this issue. It would probably be useful to provide a geography sub-expert for each record source being used. These experts could map locations as they existed then to their names today. Thus each record source could provide its own normalization scheme. A good example to consider here are the various U.S. censuses. At the interval of each census the set of place names within the U.S. has changed and evolved. But it is possible to normalize the place names used in each of the censuses to the names in use today. There is considerable work to get this done well, but a lot can be done quickly in order to get started.

Person Record Format

Person records are the inputs to the combining algorithms. Each person record holds the information about one person taken from one evidence record. Person records may hold the following types of

information:

Personal Information. This includes a unique id for the record (UUIDs should be used), and the link or reference back to the evidence record this record comes from. It includes the name of the person and birth date and place, and death date and place.

Relationship Information. This includes information about the other persons mentioned in the same evidence that generated this record. Person records either contain references to these other records or include data directly from those records or both. Whichever implementation is used, each person record provides access to information about any parents, siblings, spouses or children who were mentioned in the same evidence.

Location Information. Much genealogical evidence locates persons to a time and place. Person records hold this location information.

This format strikes a flexible balance between a complex and simple format. Most person records will hold a sparse subset of the possible information. Many will be just a name with location information. Records taken from census evidence will be richer, possibly containing many relationships and location information.

Group Record Format

During combination person records are iteratively combined into larger and larger group records. From an abstract point of view a group record is a set of person records, and the information in a group record (the same kind of information listed above for person records) are weighted distributions of the values found in the person records.

Two possible implementations of group records are:

List of Person Records. In this implementation a group record holds the list of the person records making it up, and it contains a weighted set of information items computed from the person records. This is a simple representation and requires the minimum number of group records. A distinct disadvantage of this approach is that it cannot record the combination history that brought the person records into the group record.

Node in a Tree of Group Records. In this implementation a group record is a node in a tree of group records. The record at each node holds the list of child group records that are one level below in the tree. As in the first implementation each group record (this can be limited to just the root group records) holds the weighed set information items from the person records. Each combination step joins pre-existing trees by creating a new root group record and subsuming the existing trees under it. The weighted information of the new root node are easily computed from the information from the subsumed roots. The advantage of this approach is that full history and statistics of combination is kept with the tree. The disadvantage is that a larger number of group records is required, though this is in fact very minor (the first implementation requires n group records, the second requires a potential maximum of 2n records, not a substantial difference).

In my opinion the advantages of the second approach outweigh the advantages of the first. In the earlier implementation I opted for the first approach. The loss of combination history this caused made it much harder to trace and understand exactly what was happening during combination. In order to trace and evaluate the combination phases I had to bolt on a tracing package that included a complete implementation of the group tree approach just to maintain the history for use in debugging.

Another advantage of the group tree approach comes in a statistical sense. Each combination event comes with statistical likelihood that the combination is joining groups representing the same

individual. Earlier combinations, those lower in the tree, should have much higher likelihoods, while those closer to the root typically have lower likelihoods. In any user interface used to present the results of combination the likelihoods and the groups behind them can be used when showing the person records to users in lists of person records that may be the same individual.

Definitions

Person Record – A person record is a record of information extracted from evidence that includes a person's name and whatever other information about that person found in the evidence. Person records are the starting points for nearly all genealogical work.

Combination – Combination is the process of combining person records that refer to the same individual (or are highly likely to refer to the same individual) into more inclusive group records.

Linking – The process of establishing relationship links between Individuals based on the links that exist between person records and group records.

Group Record – A record that represents a collection of person records. At one extreme a group record can contain a single person record. At the other extreme a group record can contain all the person records believed to refer to the same Individual.

Individual – A term used to represent a single real person. The term is used instead of person to avoid the ambiguities around the person record concept.

Persons, Groups, Personas, Individuals

When talking about person combination and linking (and genealogy in general) there is plenty of potential for confusion in the terms we choose for the different concepts. The term *person* may be the most overloaded. In this description a person record is at the most basic level and is the information that can be gleaned about an individual from a single item of evidence. There is a real individual who lived that the person record refers to but it is important to keep the two concepts separate. Genealogists attempt to find as much information about the individuals they are researching as they can, gathering a large number of person records with the hope that many sets of those person records can be inferred to be the same real individuals.

In my earlier work in this area I used the same distinction between persons and individuals and also used the same group concept The concepts were easily embraced by the other developers. What was a little confusing was the that the software uses the same classes to hold persons and individuals.

In earlier work in family reconstitution from church registers the person records were called *nominal records* or *nominal person records* (and combination process, done manually by hand using strict rules, was called *nominal record linking*). Though a little awkward this term does convey the idea that the record centers around a single name found in the evidence and the information that was directly associated with it.

Sometimes the word persona is used to convey one meaning or another, and a number of terms have been used to refer to different types of group records.

What are being called groups records here are collections of person records that some researcher or some algorithm has brought together because of the belief that the person records contained refer to the same individuals.

Discovering and Correcting Errors in Indexed Data Sources

After combination it is possible to consider going back and finding errors in the original indexed data sources. In some handwritings the names *Daniel* and *David* can appear almost identical. Because these two names are not related in any way, normal combination phases would never key these person records into the same buckets to give them an opportunity to be compared. But if they had the same surname, same birth and death information, were related in the same ways to the other records, then statistics can be computed about the likelihood that a transcription error were made.

For any type of error that is consistently found in indexed of record sources, it is possible to design a post-combination, error-correction phase to address it.

Automation, Semi-Automation, Manual Processes

When dealing with millions of records only automatic algorithms that run without human intervention are reasonable. This presents a major problem in the areas of quality control. Testing these algorithms is nearly impossible. After six years of working with these algorithms in the other application, this remains the single largest issue we face. In normal cases of testing these kinds of algorithms you would gather ground-truth data for a number of randomly selected test cases and see how the algorithms perform on them. In the case of the previous application and in the current application, gathering of ground truth is difficult because there is no truth. The testers gathering the data have to make the same inferences about what person records refer to the same individuals as do the algorithms they are going to test, so testing is only done between human-generated conclusions and algorithm-generated conclusions.

Another issue is how to deal with errors that are discovered by users (given that is that their conclusions are correct!). We faced this problem in the other application and put together a system that allowed users to register and then change our individuals. These corrections can be treated as just another type of person record to be taken into account during the next cycle of combination.