Traces through time: a case-study of applying statistical methods to refine algorithms for linking biographical data

Mark Bell, Sonia Ranade
The National Archives,
Kew, London
E-mail: { mark.bell; sonia.ranade }@nationalarchives.gsi.gov.uk

Abstract

The Traces through Time project, which ran at The UK National Archives in 2015, developed algorithms and tools to link people appearing in historical records and to assign robust measures of confidence to the connections that are made. The method has application across the digital humanities, including for biographical research.

Fuzzy matching relies on the availability of background statistics on the population, the distribution of data values, data quality and the type and frequency of errors. This paper describes work to refine the original algorithms through implementation of a learning approach in which insights arising from one analysis are fed back into the algorithm to improve the baseline statistics for subsequent analyses. We find that this iterative approach delivers significant improvements over 'raw' scoring mechanisms. It enables us to carefully target the type and degree of fuzzy matching to be applied and can help balance the poor precision that results from allowing increased 'fuzziness' against the poor recall that arises from a more restrictive approach. Future work will extend the approach beyond names and dates of birth, and will embed these enhancements into the Traces through Time framework and tools.

Keywords: Record Linkage, String Similarity, Statistical Inference

1. Background: Approach to Record Linkage and confidence scoring

The identification of a link between two occurrences of an individual in the historical record is achieved through assessing the similarity between the individual attributes of the two entities to be compared. During this project, we have worked extensively with data from World War One service records from The National Archives collections.[1] The datasets in question were initially created by indexing the original paper documents, and our analysis is limited to those data attributes which were consistently captured by previous digitisation and transcription projects. For WW1 data we are generally restricted to linking records based only on names and either age or date of birth. Other attributes such as place of birth and service number are sometimes available but are not consistently captured across datasets.

Record Linkage is achieved using a probabilistic method based on the work of Fellegi and Sunter, (1969) and a variation suggested by Winkler (1990) to account for spelling differences between pairs of textual attributes. The basic approach is to find, for each attribute, the ratio between the probability that a pair of records refer to the same person and that of them referring to two different people. The Winkler variation allows the use of string comparison algorithms to accommodate spelling variations and applies a weighting to reduce the score for an attribute comparison if the attributes lie within a certain threshold of similarity but are not identical. Appendix A gives a brief outline of the calculation. Our work has taken this approach a step further and applied a range of weightings to achieve more fine-grained fuzzy matching.

Dates of birth, which are a key attribute for discriminating between different individuals in the records, are problematic for historical data. Often we have neither a date of birth nor an age. If only an age is provided it is not necessarily clear on which date that age applies - is it the individual's age at the date of the record? Or their age at the date of some other event mentioned in the record? And dates are often estimated or rounded. When a date of birth has been captured it is not necessarily accurate: consider the case of under-18s claiming to be older in order to enlist for military service. So, we require new techniques to derive confidence scores for dates, all based on estimated distributions.

In the case of a year being captured on the record, we create a probability distribution of likely values. This allows us to fuzzy match two different year values, adjusting for data quality and deriving a probability that the underlying values are the same. Instead of a single year, the record may state a range of years, possibly derived from the age. In this case the calculation is the same but the confidence scores returned will vary depending on the range of the stated ages. Finally, there are records with no indication of birth period. In this situation, the best we can do is to derive a frequency distribution for the whole dataset, drawing on external, expert knowledge if this information is not available in the data. The calculation is then the probability that the person in dataset A could be in dataset B.

The result of these individual attribute comparisons is a score on a logarithmic scale which can be used to assess our confidence that the pair of occurrences represent the same person.

[1] National Archives Discovery - http://discovery.nationalarchives.gov.uk/

2. Learning from record linkage results

This paper focuses on methods for refining the statistical model described above by learning from the results of matching many datasets. We describe an approach to identifying and incorporating common differences in textual information arising from factors such as: handwriting recognition errors, typographical errors and phonetic errors made when names are recorded. A different approach is described for dates of birth, where the algorithm must accommodate inaccuracies in recording such as mis-representation of age or rounding of declared ages[2]. In this case, the age distribution observed for each dataset is fed back into the algorithm to support a statistical approach to calculating the likelihood that two occurrences of a person with different recorded dates of birth, in fact, relate to the same individual. As each incremental enhancement of the algorithm improves the results of the matching process, these in turn, reveal further discrepancies in the data, from which the algorithm can learn. A number of distinct areas are being worked on, all building on previous research and reliant on the gathering of statistics over time. Here we highlight what has been done so far and which emerging ideas are being explored.

2.1 Learning from record linkage results

In our work so far we have discovered benefits in deriving an age profile for a dataset which lacks dates of birth by linking to one which does. We have also improved linkage results by allowing for discrepancies in ages.

In order to improve on this technique, we analyse the dates of birth for high confidence matches to build a statistical profile of common differences. For example, in WW1 records this approach will highlight that for soldiers in the 16-20 age range it is more common for two records referring to the same person to have different years of birth than for those in, say, the 30-35 range. Therefore, if we had two records with years of birth 1882 and 1883 (age 33-34 in 1916) then we would have less confidence that they are a match than if they were 1897 and 1898 (age 18-19 in 1916). This behaviour is particular to WW1. Examination of another dataset such as the GRO death registers[3] shows that it is quite common for the deceased's age to have been guessed at the time of registration. We would therefore want to accept a different profile of differences in this dataset, where there is a higher likelihood of discrepancies in dates of birth for older people as their deaths are less likely to have been registered by a close relative.

2.2 Fuzzy name comparisons

In the case of name comparisons the variation between name transcriptions for records representing the same person can be thought of as a function of several factors (list not exhaustive):

- Regional spelling variations.
- How the recorder hears the name, particularly with unfamiliar names and regional accents.
- The recording medium – hand written vs. typed.
- Involuntary errors during data capture, spelling mistakes while writing or typing the original document.
- Involuntary errors during transcription, including those caused by difficult handwriting.

A commonly seen example of a transcription error caused by handwriting is the cursive 'T' being misread as the letter 'J', due to the similarity between those letters in that style of writing. By analysing the frequency of high confidence matches which have this specific difference in their names we can refine the confidence scores returned when this difference is encountered. Without this more nuanced approach we could miss perfectly good matches which only differ on a single initial or increase the rate of false positives by allowing any single initial difference. The key to the approach is to capture the results not just for a single dataset but to associate the difference with metadata connected to a collection as a whole – for example, records in a particular format from some defined time-period – allowing accumulation of generalised statistics based on many examples which are typical of a type of record. The misreading of 'T' and 'J' is far less likely in typed records since the typed letters have a distinct appearance but there are likely to be other typical differences arising from keyboard layout.

Simply informing the model of the probability that Ts and Js have been interchanged has delivered good results, so the next step is to use the data to identify a wider range of commonly occurring transcription errors. We use the Jaro-Winkler measure to find similar strings and weightings to assign confidence depending on this measure. Our aim now is to look at methods for using the difference itself to increase accuracy.

2.3 Name frequency statistics

In the absence of high volume name data, such as a census, it can be difficult to accurately calculate the frequency of occurrence of a particular name in the population that appears in the records under consideration. This is especially the case if the data sources being matched are relatively small (< 10,000 records). Consider a dataset with 1,000 records including Messrs. Taylor and Zephania. From this alone, we might surmise that these surnames have equal probability of 0.001 while, in reality, the former is more common and the latter is rare. Accumulating match results over multiple datasets allows us to create a larger population of individuals from which to derive probabilities. The

[2] The 'age heaping' effect is observed in datasets which record age (rather than date of birth). The resulting distribution is skewed, typically showing peaks at 'round' ages (e.g. 10, 20, 30…). For an illustration, see the 1911 census graphic from the ONS data visualisation centre (http://www.bbc.co.uk/news/uk-18854073).

[3] General Register Office death registrations supplied by http://www.gro.gov.uk/gro/content/

difficulty arises from the fact that the most common names will, by nature, belong to lower confidence linked pairs, which are therefore more difficult to associate as referring to the same individual. We have also identified a caveat to the assumption of attribute independence in the general linkage model. By clustering forenames and surnames together we have identified groups of names that typically occur together and which appear to align with national or ethnic groups - e.g. Irish, Italian, Hispanic/Portuguese. Although the names themselves are still independent of one another, there is an implicit dependence with a third variable, nationality, which is not directly expressed in the data. As a result, name matches such as Patrick Murphy, a common Irish name, and Angus MacDonald, common in Scotland, are assigned higher confidence scores than is warranted because each name part, considered individually, is not particularly common in the population as a whole. We mitigate this type of association in the Traces through Time approach by arbitrarily reducing the population used in the probability calculation by a factor of ten. This has some basis in the data, as Ireland has 10% of the population of England, for example. However, it is a blunt tool. We are now working on refining this technique, again by gathering statistics through linking multiple datasets.

3. Identifying common differences

3.1 Differences in names

Our approach for identifying common differences in transcription and spelling is to look at matched records which differ in a single attribute. For example, the matched pair “Robert Adrian Gardner, born 17/11/1898” and “Bob Adrian Gardner, born 17/11/1898” have different first names but are otherwise identical. If the first names were the same we would consider this to be a high confidence match. However, in our existing statistical model the first name would contribute a negative weighting to the calculation. By looking at record pairings which only differ on a single attribute and which would score above a certain threshold, T, indicating a high confidence match, if the difference was not there we can ascertain patterns in these differences. So in the example above, if we see a number of record pairings with the same pattern, we may deduce that Bob is an alternative form of Robert.

In the following definition, when we refer to a transformation we mean some difference in spelling has been encountered between two attributes which could be due to common spelling variations (‘Phillip’ and ‘Philip’), spelling errors (‘Roland’, ‘Rolend’) or diminutive forms (‘Bob’, ‘Robert’).

DEFINITION 1: We define a Transformation Pattern (TP) as a function which transforms a string S_1 to a string S_2 by substituting any substring ss_1 of length l in S_1 with another string ss_2 also of length l. We shall also say that for a transformation pattern T_x that T_x on S_1 **yields** S_2 if applying the pattern T_x to S_1 results in the string S_2.

DEFINITION 2: We consider a transformation pattern to be Common (a CTP) if it occurs above a certain percentage of the time. More concretely, if we take n record pairs where each pair has a record which contains an attribute, a_1, containing the string ss_1, then applying TP to the equivalent attribute, a_2, on the linked record will yield a_1 at least c% of the time. An attribute pair is defined as having attributes a_1 and a_2, so the set of n pairs P where either a_1 or a_2 contain the string ss_1 is:

$$\boldsymbol{P = \{p_{1<a_1,a_2>}, \quad p_{2<a_1,a_2>}, \quad \cdots, p_{n<a_1,a_2>}\}}$$

T_x is the transformation pattern that transforms ss_1 to ss_2.

$$\boldsymbol{\frac{\sum_{i=0}^{n} \begin{cases} 1 \ \ if\ T_x : p_i\langle a_1\rangle \xrightarrow{yields} p_i\langle a_2\rangle \\ 0 \ \ otherwise \end{cases}}{n} \geq c}$$

In order to find common patterns of transformation we must calculate all n-grams around each character difference between $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$, where the corresponding n-grams in the two attributes are different, and the n-gram length is up to the length of the longest string. In order to normalize different length strings we use the Needleman-Wunsch (NW) (Needleman and Wunsch, 1970) alignment function to find the maximal alignment of two strings and then pad any gaps in alignment with ‘-‘ symbols or an ‘@’ symbol at the end of a string to differentiate between characters being inserted within a string and those added to the end.

For example, `NW('needle', 'nedle')` produces the aligned strings:

```
needle
ne-dle
```

And the resulting set of n-gram pairs is: { (‘e’, ‘-‘), (‘nee’, ‘ne-‘), (‘need’, ‘ne-d’), (‘needl’, ‘ne-dl’), (‘needle’, ‘ne-dle’), (‘ee’, ‘e-‘), (‘eed’, ‘e-d’), (‘eedl’, ‘e-dl’), (‘eedle’, ‘e-dle’), (‘ed’, ‘-d’), (‘edl’, ‘-dl’), (‘edle’, ‘-dle’) }

The next step is to represent these n-gram pairs in a tree structure where the parents of a pair are the pairs which are produced by adding one character to each n-gram in the original pair. So (‘ed’, ‘-d’) is a parent of (‘e’, ‘-‘) where ‘d’ has been appended to each entry in the child pair. The resulting tree is shown in figure 1.

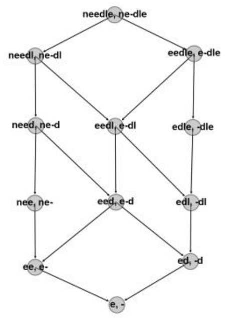

Figure 1: n-gram tree for name pair "needle, ne-dle"

We can now generate n-gram trees for every pair of different attributes from a list of matched pairs of records. These are then stored in a Directed Graph structure with each node representing an n-gram pair and edges having a weight equal to the number of instances of their parent <$\boldsymbol{a_1}$, $\boldsymbol{a_2}$> (the root node in the n-gram tree).

We have made the graph generated from the results of linking a number of WW1 collections together available to view online.[4]

The reasoning behind loading the n-gram trees into a graph is that we are aiming not just to identify single letter transcription errors but also multi-character transformations, which may be phonetic in nature – for example, 'f' for 'ph', or the prefix 'Mc' for 'Mac'.

The final processing stage is to coalesce nodes with only one parent as, if we have found a pattern of transformation, we do not need to see that pattern repeated in longer n-grams which are not encountered in other attribute pairs. For the "needle" example, if there is no other pairing <$\boldsymbol{a_i}$, $\boldsymbol{a_j}$> where

$$\boldsymbol{p}: \boldsymbol{a_i} \xrightarrow{\boldsymbol{yields}} \boldsymbol{a_j} \text{ for } p = \text{“}e\text{”} \rightarrow \text{“-“}$$

then we can connect the root node "needle, ne-dle" to the node "e, -" and remove all intermediary nodes in the tree.

3.2 Differences in years of birth

Applying the n-gram method to years of birth identified lots of common differences but did not unearth any patterns in transcription errors, such as 1 for 7, as we may have expected. We found a more effective approach was to use the arithmetic differences between years.

We compared one series of naval records, ADM337, against two other naval series – ADM339 and ADM188. The method was to analyse pairs of records which were identical in every way apart from the year of birth. The results were intriguing and suggest a pattern of behavior in the ADM188 series which was not present in ADM339. Linking ADM337 and ADM339 returned results that we would have expected for WW1 records – the rate of 1 or 2 year differences was between 0.17% and 1.44% for years of birth up to 1897 (taking the higher year of birth of any record pair), increasing to 11.69% and 25.58% for years of birth 1898 and 1899 respectively. Additionally we found that in the case of 1898 7.78% of pairs had a difference of 1 year, while for 1899 23.26% of pairs had a difference of 2 years. This tallies with our expectation of 16 and 17 year olds inflating their ages in order to join the war effort from 1916.

When we linked to ADM188 we discovered a different pattern. There was still a peak of 28.91% of 1897 births with 1 year difference. However, we saw a consistent 10-20% 1 year difference rate for all other years of birth. One theory for why this should happen is that perhaps the application form asked for day and month of birth plus age at application. The year of birth was then calculated from the age which introduced a high proportion of errors in the year of birth.

4. Results of CTP identification

Figure 2 shows the graph of attributes which have undergone the transformation p = "e" → "i".

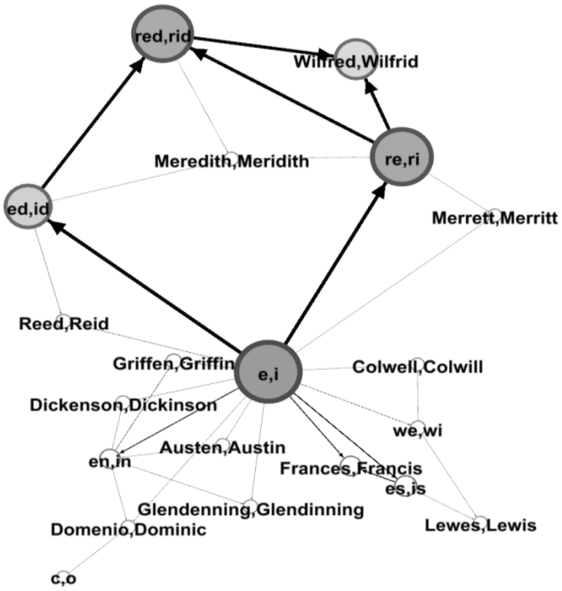

Figure 2: n-gram tree for attribute pairs where an e is replace by i

This diagram highlights a number of patterns of interest. The node sizes represent the weighted degree of the node so we can see that the pairing 'Wilfred, Wilfrid' is the most common. If we look at the most common spelling differences overall, we find that "e" → "i" occurs very frequently, but figure 2 suggests this result may be skewed by the very common spelling variation of Wilfred/Wilfrid.

An even stronger example is given by figure 3 which demonstrates that the seemingly frequent transformation

[4] https://www.dropbox.com/s/8xndg8o26g9096d/ngram_trees_2.svg?dl=0

“i” → *“y“* is almost entirely due to the variant spelling of Sidney/Sydney which accounts for 100 of the 108 occurrences of this transformation.

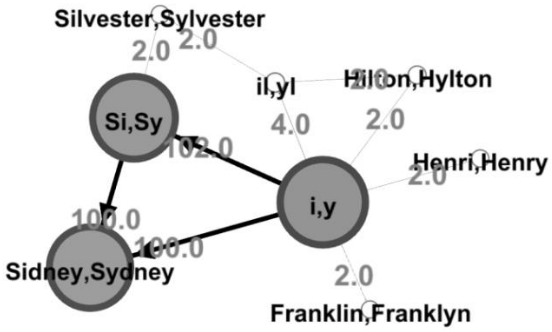

Figure 3: n-gram tree for attribute pairs where an ‘i’ is replaced by ‘y’

There are a number of CTPs which are worthy of further investigation and we will examine the effect of capturing four of these patterns in our statistical model below.

5. Using CMTs in record linkage

5.1 Approach

We will further analyse a method for building four patterns, which emerged from the method described above, into our probabilistic linkage model.

The patterns are:

1. *“Henry”* → *“Harry“*
2. *“Mac”* → *“Mc“*
3. *“ll”* → *“-l”*
4. *“J”* → *“T”*

In our existing record linkage process the names “Henry” and “Harry” are considered different enough that a negative weighting is applied to our confidence score. The effect of this is that the score changes from +1.84 (where both records use the name “Henry”), to -2.06 (for “Henry”/“Harry”), a swing of -3.9. We consider a total score above 7.5 to be a high confidence match, so we will look for records where the names are identical and have a score of > 7.5 or the names are different only as a result of the transformation “Henry” → “Harry“ and have a score of > 3.6.

We follow a similar process for patterns 2 and 3, but this time taking the swing to be only -1.5 as the resulting strings under these transformations are only one character different and this has a smaller negative effect in our calculations.

Finally for pattern 4, which is the scenario where an initial ‘J’ or ‘T’ has been incorrectly transcribed (as a ‘T’ or a ‘J’), we calculate using a swing of -6. This is a default value in our model for different initials.

We can then derive the probability of each transformation occurring by comparing the number of records which have undergone the transformation against the number which are un-transformed , as shown in table 1:

TP	S1 == S2	$tp: S1 \xrightarrow{yields} S2$	% transformed
1	4482	43	00.95
2	1800	78	04.15
3	19076	163	00.85
4	424	31	06.81

Table 1: Percentage of names undergoing the transformations. TP column refers to the numberings at top of section 5.

These percentages are fed into our statistical model as probabilities. We will test the effectiveness of this by comparing three methods:

- Winkler - Use the current process of applying a weighting to the probability score based on a string similarity.
- Probability - When one of the 3 TPs is encountered we multiply the probability score by the appropriate percentage according to table 1.
- Equivalent - We treat any string S2 which is the result of applying the TP to S1 as equivalent to S2 and therefore consider the strings to be equal in our linkage algorithm – i.e. we do not apply the weighting in the first scenario.

Ideally we would use a golden record set with known results to compare the results of applying each method. Due to the resource intensive nature of creating golden record sets of sufficient volume for record linkage, we ran linkage exercises using records which had both a name and date of birth. Only name was used to derive links, date of birth was used for later verification of these links. Since most of testing was with files of circa 13k records there are unlikely to be enough pairs of different people with exactly the same name and date of birth to have a significant effect on results. With this method we have at least a pseudo-golden result set.

5.2 Results and discussion

The use of probabilities in our record linkage algorithm gave good results, especially for patterns 1 and 4 which, under our existing method, have a large negative impact on the match score. If we consider a strong match to have a score above 7.0, then in a model where an incorrect transcription of the letter ‘J’ to ‘T’ causes the score to be lowered by 4, only records where the other attributes total at least 11.0 would pass this threshold. In the absence of a date of birth, that individual would have to have quite an unusual name for the link to be discovered.

Table 2 shows the results for each experiment. We ran each experiment with a scoring threshold of >5.0. Since the 'equivalent' method treated every transformation as an agreement it will consistently score higher than the other two methods, so for the purposes of this test we consider it to provide a baseline of results. This explains why it produces no false negatives in our results table.

The first thing to note is that both the Winkler and Probability methods produce far fewer false positives than the Equivalent method in all tests. This is to be expected but is important to note since a high number of false positives could waste a considerable amount of time and effort if the linking approach is used to identify potential matches for further research. We can find all True Positive links by lowering the scoring threshold but there is always a balance to be made with the False Positive rate. In this respect the Winkler method was the best performer for patterns 2 and 3, but the Probability method was a close second.

Both Winkler and Probability failed to find some of the links but, as discussed above, they could always be found by reducing the scoring threshold. We should also remember that the threshold chosen was an arbitrary one for the purpose of comparing the methods under investigation, so these results might be more reasonably interpreted as the Equivalent method scoring some matches too highly. This is an example of the recall-precision trade-off. The threshold allows the user of the application to choose between seeing many possible results, high recall, and restricting the results to only the most likely matches, high precision.

The probability method is particularly strong when tested on patterns TP 1 and TP 4. Here Winkler performs badly since it treats "Henry" and "Harry", and "J" and "T", as different strings. We could pick up "Henry" and "Harry" as similar strings by lowering the Jaro-Winkler threshold in our string matching but this has a knock-on effect of creating more false matches in our overall linkage results and additionally reduces performance by generating more candidate pairs for matching. For TP 1 we can provide a good example of the effect of lowering the threshold. Reducing it to >4.8 results in a 100% True Positive rate, albeit at the expense of 7 extra False Positives. Again the Equivalent method creates a high number of False Positives, although performs better on TP 1 than for other patterns.

	True Positives			False Positives			False Negatives		
TP	Winkler	Prob.	Equiv.	Winkler	Prob.	Equiv.	Winkler	Prob.	Equiv.
1	1	5	7	0	1	17	6	2	0
2	8	8	11	7	10	82	3	3	0
3	11	10	12	10	10	33	1	2	0
4	0	5	6	0	17	71	6	1	0

Table 2: Results of testing three scoring methods for the four transformation patterns

6. Name Independence

6.1 The independence assumption

The probability model we use in TTT assumes that the attributes within a record are independent of each other. It can easily be shown that this isn't always correct by considering the relationship between forename and gender, for example, –a 'Mary' is far more likely to be female than male. However, for record linkage, using several variables it has been found to be a reasonable assumption which maintains simplicity in the model without compromising the accuracy of results. In the case of matching historical records, where we often only have the name of the person as a linking key, we have found that the assumption does not hold. In particular we have identified a relationship between the national/cultural background of a person and their name. In the matching results this is manifested in the form of unexpectedly high scores for some names. Consider the name Angus which is typical to Scotland. In a series of 582k naval service records there are 218 Anguses which suggests a probability of 0.00037 of being called Angus. If we were to imagine for a moment that only people born in Scotland could be called Angus and only 20% of our population are from Scotland, then this probability becomes 0.0019. When fed into our logarithmic scoring algorithm, this represents a difference of 0.7 in the scores obtained. Thus, if a name is strongly dependent on country of origin then it makes sense to calculate the probability of that name based on the population of that country, not the population of the entire United Kingdom.

6.2 Calculating dependence

In the example above, external knowledge would readily identify 'Angus' as a Scottish name. However, these associations are also evident in the data. We took a selection of the most common forenames and surnames from our series of 582k records and classified them as English, Scottish, Irish, Hispanic or Italian. Then we selected records from the series where the person's name was comprised only of names in these forename and surname lists. Figure 4 shows the result of cross referencing the classifications of forename and surname for these people. The x-axis represents the classification of the forename, each bar represents that of the surname, and the height of the bars represents the percentage of people. So we can interpret the tallest bar as – "93% of people with an Italian forename also have an Italian surname".

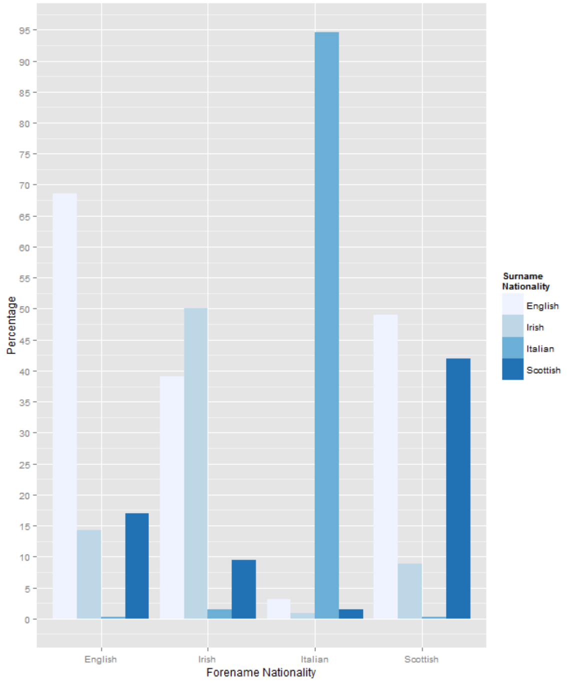

Figure 4: The relationship between forename and surname nationalities

6.3 Incorporating dependence into the model

Consider a simplified form of our model for all names in a population *P* which are comprised of one forename and one surname.

The score we calculate for a link between two records having name "X Y", assuming independence, is:

$$-\log_{10}\left(\frac{f(X)}{P}\right) - \log_{10}\left(\frac{f(Y)}{P}\right)$$

f(n) being the frequency of name "n" in population P.

In order to incorporate dependence on the cultural provenance of names into the calculation we will use the conditional probability formula:

$$P(B \mid A) = \frac{P(A\ and\ B)}{P(A)}$$

For our person "X Y" where both "X" and "Y" originate from country C, with population P_c, we can revise the formula to:

$$-\log_{10}\left(\frac{f(X)}{P}\right) - \log_{10}\left(\frac{d.f(Y)}{P_c}\right)$$

where *d* is a multiplier to give us the probability of a person from country C having the name "Y". To simplify we can use the average percentage from Figure 3 for nationality C as the multiplier *d*.

We can now put this formula into the same form as our original formula to obtain:

$$-\log_{10}\left(\frac{f(X)}{P}\right) - \log_{10}\left(\frac{f(Y)}{P}\right) - \log_{10}\left(\frac{d}{P_c/P}\right)$$

Since we have an estimate of d we only need to estimate P_c for each nationality group to adjust the score to account for dependence.

6.4 Estimating national populations

In order to estimate the population *Pc* for nationality C we need to look at names which are common enough to be linked to several candidates. We matched together two lists of names, A and B, with 50k and 582k records respectively, and filtered out the matches for a single instance of each unique, two item (i.e. forename, surname) name in A. We then further filtered the results to include only names comprised of the most common English, Irish and Scottish forenames and surnames.

For English only names (English forename and surname) which attracted at least 4 possible matches we performed a linear regression, as seen in Figure 5:

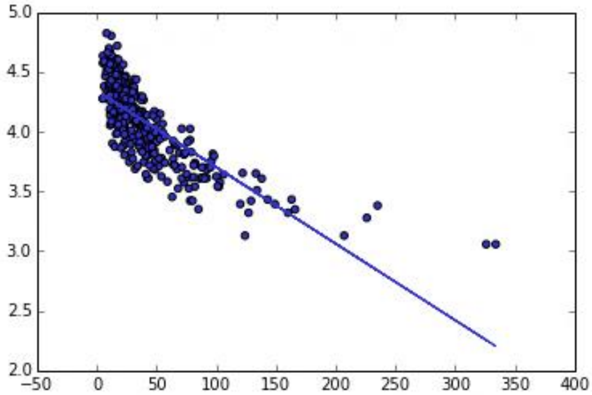

Figure 5: Plot of matches against score for English names

This regression provides a mechanism for calculating the expected score based on the number of matches. We can use this to calculate an expected score for the Irish and Scottish names which, in turn, allows us to estimate the population-sizes to be used to adjust our scoring for Irish and Scottish names. Our adjusted score is derived from the intercept (4.34) and slope (-0.006) from the linear regression and the numbers of matches from B for each person in A with an Irish or Scottish name. We then compare this to the actual score for the match and calculate the difference, D

$$D = S_n - 4.34 + (-0.006).M_n$$

$\boldsymbol{M_n}$ being the number of matches for name $\boldsymbol{n}$ and $\boldsymbol{S_n}$ being the actual score for exact matches on name $\boldsymbol{n}$.

By averaging these differences we were able to calculate the ratio $Pc\ /\ P$ to feed into the dependence formula.

Taking the probabilities from Figure 4 for Irish/Irish and Scottish/Scottish we arrive at figures of 1.58 for Irish names and 1.54 for Scottish names.

This means that whenever we come across a person with

Irish forename and surname or Scottish forename and surname we will subtract these adjustment figures from the score.

6.5 Results of adjusting for name dependence

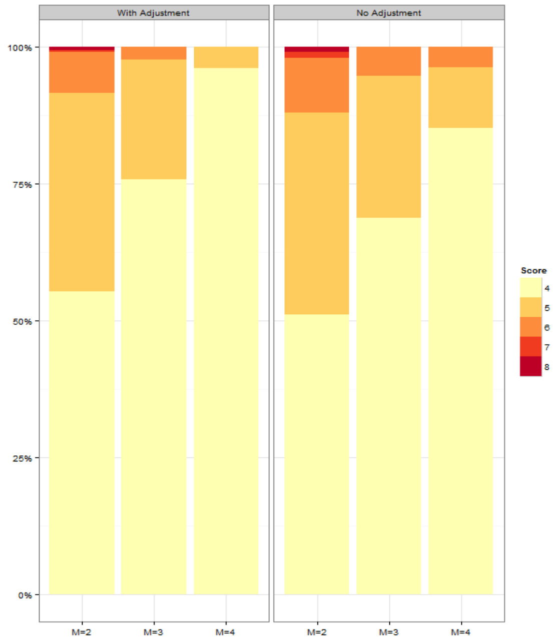

Figure 6: Number of matches by score, with and without adjustment

To test the nationality adjustment outlined above, we linked together two de-duplicated sets of records, A and B, with and without the adjustment for Irish and Scottish names. We then counted the number of matches from B against each unique name in record set A. Figure 5 shows a plot of the match counts against the integer score, with and without adjustment.

The effect of using the adjustment has been to lower the scores of many records which have multiple matches. Without adjustment 85% of records with 4 matches had a match score of <5, whereas with adjustment this increased to 96%. We had one instance of a record with a score above 6, a score suggesting a medium confident match, with 4 matches which was for a person with a Scottish name. For records with 3 matches 5% had a score of 6, reducing to 2.3% with the adjustment.

As with the CTP experiments, the score provides a means of balancing precision and recall in the results. In our record linkage results, a score above 7.0 suggests a high confidence match where we wouldn't expect to see two different people with the same name occurring in the same context. Below 7.0 we begin to see more names shared by two different individuals, and below 6 more names shared by three different people. In our experience, when we see more matches than we expect for a particular score, these tend to be for people with names not originating in England. Using this technique of adjusting scores based on a population size derived from nationality, which is in turn derived from a person's name, we have reduced the number with more matches than we would expect for the score that is observed.

As an example, in the match results we found a single match to "Angus McLeod" with a score of 6.4. Without the adjustment this score would be 7.9 indicating a very high confidence match. In reality this isn't such an uncommon name so we shouldn't consider our match to be quite so definite and therefore the adjusted score of 6.4 seems more appropriate.

7. Conclusion and future work

We have discussed two enhancements to the Traces through Time record linkage model. The first was the use of comprehensive statistics of common differences in the spellings of names to incorporate the probability of a name being spelled two different ways between a pair of candidate records. This proved to be an effective addition to our model, especially for variations which can not necessarily be captured by standard string similarity measures, such as errors in transcribing initials or name variants which are very different, like 'Jack' and 'John'. We found an advantage in compiling statistics from matching many different datasets in that the use of initials is uncommon enough in many of the datasets that no CTPs for initials were found until we matched one particular series that had a high incidence of initials. We can now apply the statistics derived from matching that one series to matching any series in the same format and from the same period. Unfortunately we didn't have enough examples of typed records to find any patterns which were specific to that medium, but we hope to explore this further in the future. We also plan to apply the pattern detection algorithm to records from different historical periods to see how this effect varies through time.

Our investigations into year of birth differences returned very interesting results about how the forms in one particular series were filled in. This is another avenue for further exploration.

The second enhancement was to incorporate an adjustment to match scores depending on the national or cultural origin of names. This is something we already do in our model but only by applying an arbitrary adjustment. We demonstrated a data driven method for calculating an expected score based on the number of matches a particular name attracts. This seems to work well for Irish and Scottish names. We would now like to extend the model to names which originate further afield, which are likely to have smaller populations within our data. This also will involve the development of a more robust method for identifying such names, as it will be time consuming to manually compile lists. We have already explored a clustering approach which we will continue to develop.

8. References

Fellegi, I. P., and Sunter, A B. (1969), *A Theory for Record Linkage, Journal of the American Statistical Association*, Vol. 64, No. 328 (Dec., 1969), pp. 1183-1210

Winkler, W. E. (1990a), *String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage, Proceedings of the Section on Survey Research Methods, American Statistical Association., 354-359.*

Needleman, Saul B.; and Wunsch, Christian D. (1970). *A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology 48 (3): 443–53.*

Appendix A

We present here a brief description of the probabilistic linkage method used and how the Jaro-Winkler score is used to cater for inexact string matching. We refer to this as the 'Winkler' method in our paper.

The Fellegi-Sunter method calculates the ratio of the probability of two records representing the same person versus that of them representing two different people. These are referred to a P(M) and P(U), for 'Matched' and 'Unmatched', respectively. Furthermore they calculate this ratio differently depending on whether the attributes being compared are the same or different, giving two scores PA (for agreement) and PD (for disagreement). When comparing two attributes a1 and a2 we calculate a score, S, based on the following equations:

$$S = P_A = \frac{P(M)}{P(U)} \; if \; a_1 == a_2$$

$$S = P_D = \frac{1 - P(M)}{1 - P(U)} \; if \; a_1 <> a_2$$

In order to handle spelling errors, Winkler proposed finding a point somewhere between PA and PD depending on the Jaro-Winkler score for a1 and a2.
If J is the result of passing a1 and a2 into the Jaro-Winkler algorithm then our calculation becomes:

$$S = \max(P_a - (P_A - P_D).(1 - J).\rho), P_D)$$

The constant ρeffectively controls how much tolerance to string difference is allowed before the disagreement score is reached.