Genealogical Record Linkage:
Features for Automated Person Matching

D. Randall Wilson
FamilySearch.org

Abstract. *This paper provides a high-level overview of how automatic person matching (genealogical record linkage) algorithms can be developed, and then provides a detailed explanation of many of the features used by FamilySearch in doing person matching. Empirical results show a dramatic improvement in accuracy by using these features trained with neural networks, when compared to traditional probabilistic record linkage with simple field agreement features.*

1. Introduction

Identifying matching (or duplicate) individuals is an important part of genealogy work. With vast databases of genealogical source and conclusion data coming on-line, it is important that computers become good at automatically detecting when two individuals in a database represent the same real person. This allows users of the system to resolve duplicates in a conclusional database and also find relevant data in source data to extend their knowledge of who has lived and how they are related.

Record linkage (Dunn, 1956; Newcombe *et al.*, 1959; Fillegi and Sunter, 1969; Newcombe, 1988) is a term often applied to the problem of identifying to database entries that refer to the same real entity. In the case of genealogical record linkage, the problem is to find pairs of individuals that represent the same real person.

This paper provides an overview of how genealogical record linkage can be done in general, and then provides a detailed examination of *matching features* that can be used to help a *match classifier* compare names, dates, places and other information between pairs of individuals. It concludes with empirical results from an 80,000-individual labeled data set showing dramatic improvement in accuracy using these features trained with neural networks compared to traditional probabilistic record linkage.

2. Record linkage overview

Blocking. When doing bulk record linkage, such as identifying all duplicates in a large database, it is usually impractical to compare every individual in the database with every other individual. Instead, records are typically grouped into *blocks* of records that have a few things in common (*e.g.*, same birth year and surname), and the records within each block are compared with each other. Since some matching records have missing or conflicting data elements, multiple blocking passes are often required to avoid missing many good matches.

When done one at a time, one individual is considered the *target*, and a query (called the *blocking query*) is issued to find a set of *candidate* individuals to compare against the target. These are individuals with enough in common that they are worth looking at, though most of them will not turn out to really be a match.

Scoring. Once a set of candidates has been found for a target individual, the next task is to use a *match classifier* to carefully compare the target with each candidate to determine how likely it is that they both represent the same real person. The classifier does this by computing a set of *features*, which will be discussed in detail below. The features identify aspects of how the records compare that are useful in determining whether individuals are the same person or not, such as how well the names agree, how well the dates and places in the events agree or conflict, and so on.

The classifier uses the feature values to calculate a *match score*, using an algorithm such as a machine learning algorithm, neural network, or statistical method. (With sufficiently powerful features,

a single layer neural network has turned out to be as accurate as anything else we have tried.) The resulting score is then compared with pre-determined *match thresholds* to decide what to do with the pair (*e.g.*, ignore it; present it to a user as a high, medium or low-confidence possible match; auto-link; etc.).

    Truth set. In order to train or evaluate a record linkage system, it is important to have enough *labeled data* to both train the system and to evaluate the accuracy of a match classifier. Labeled data consists of a set of pairs of individuals for which someone has decided whether the pair is a *match* (*i.e.*, both individuals represent the same real person), or a *differ* (*i.e.*, the two individuals represent two different real people). FamilySearch gathered a set of some 80,000 pairs of records that were labeled by genealogical experts. Typically 60% of this data is used for training, 20% used as testing, and the remaining 20% used for final evaluation after several iterations of testing and training (since during that process, the test data becomes somewhat "tainted", in that it was used to influence the development of the classifier that it is testing).

    Evaluating accuracy. The accuracy of a record linkage system is measured in terms of *recall* and *precision* at some threshold θ. *Recall* is the percent of the known matching pairs that get a score above θ. *Precision* is the percent of pairs with a score above θ that are matching pairs. Put another way, recall is the percent of real matches that the classifier calls a match, and precision tells what percent of pairs that the classifier *calls* a match really *are* a match. Thus, the false negative (or "missed match") rate is 100% - recall, and the false positive (or "bad match") rate is 100% - precision.

    Since the recall and precision depend upon the threshold that is chosen, the threshold can be varied to create a precision-recall curve ("P/R curve"). This allows the selection of one or more useful thresholds that give different trade-offs of precision and recall. A high-recall (but low precision) threshold might be a useful cut-off score for deciding whether to show possible duplicates to a user; whereas a high precision (but lower recall) threshold can be useful for deciding whether to automatically combine two individuals in a database. Points in between might be useful for showing various confidence levels in a list of possible matches.

    When developing a match classifier, there are many decisions that can be made regarding the selection of data to train on, which features to use, parameters to use within those features, and so on. Having labeled data to compare against allows one to run experiments with different variations on the classifier and see how the changes affect the accuracy. Once a basic classifier has been developed, it is not uncommon for a new feature that seems like a good idea to actually hurt accuracy, so it is important to have data available for evaluation.

3. Features for matching individuals

    The FamilySearch match classifier uses a set of *features*, each of which can have several *feature values*, which correspond to various levels of *value agreement*. Each feature value can have a separate *weight*, which is added into a *match score* whenever a feature value "fires".

    For example, following are some of the feature value weights:

IndGivenName=-1: -2.2224	IndGivenName=6: 0.4946	IndSurname=-1: -1.8169
IndGivenName=1: 0.5968	IndGivenName=7: 1.2099	IndSurname=1: 1.4038
IndGivenName=2: 0.687	IndCommonGivenName=1: 1.0244	...
IndGivenName=3: 0.0743	IndCommonGivenName=2: 1.0773	Bias: -5.0982
IndGivenName=4: 1.5611	IndCommonGivenName=3: 1.1974	
IndGivenName=5: 0.686	IndCommonGivenName=4: 1.4942	

    The three most important types of features are those that compare names, dates and places. There are also a few other miscellaneous features included in the classifier. The name, date and place features are used for the individual and a variety of relatives, and the date and place features are used for several event types. In each of these cases, there are a number of different feature values corresponding to

various levels of agreement. Sometimes the relatives' features use less levels of agreement than the individual.

The feature categories and individual features are explained in detail below.

3.1 Names

Names are the most important genealogical data we use. They are varied enough to provide powerful identification, and most records contain at least this piece of information about a person.

Name variations. However, names are obviously not completely unique, so many people can have the same names. Furthermore, there are many reasons why the same person may appear in various records with different names, including:

- *Maiden vs. married name.* ("Mary Turner"/"Mary Jacobs"). Often a woman will take on her husband's surname at marriage.
- *Nicknames.* ("Bob" vs. "Robert"; "Mary"/"Polly"; "Sarah"/"Sally"; "Margaret"/"Peggy"; etc.)
- *Spelling variations* ("Elizabeth" vs. "Elisabeth"; "Speak"/"Speake"/"Speaks"/"Speakes")
- *Initials* ("John H. Smith" / "John Henry Smith")
- *Abbreviations* ("Wm."/"William", "Jas"/"James")
- *Cultural changes* (*e.g.*, "Schmidt" -> "Smith" when migrating to America; Scottish "Donul" can become either "Daniel" or "Donald" when Americanized).
- *Typographical errors* ("John Smith"/"John Smiht")
- *Illegible handwriting* (*e.g.*, "Daniel" and "David" both have the same number of "lumps" and "dots", and so can be confused).
- *Upper/lower case*. ("MARY", "Mary", "mary")
- *Spacing* ("McDonald"/ "Mc Donald"; "VanderGraff"/"van der Graff")
- *Articles* ("de la Cruz" / "Cruz")
- *Diacritics* ("Magaña", "Magana")
- *Script changes* (*e.g.*, "津村" (Japanese Kanji), "タカハシ" (Japanese Katakana), "Takahashi" (Romanized Japanese)).
  - Korean uses Hanja (Chinese characters), Hangul (a phonetic script), and Romanized forms of names, which can vary depending on Romanization method and personal preference.
  - Japanese has Kanji (Chinese characters), two phonetic scripts (Katakana and Hiragana), and also Romanized forms.
  - Chinese has traditional and simplified versions of its characters, along with various Romanized forms, including Pinyin.
  - Cyrillic (*e.g.*, used for Russian names) can be Romanized as well, as can Greek and other scripts.
- *Name order variations*. Sometimes first and middle names get swapped ("John Henry", "Henry John").
- *Given/surname swapped*. Other times the surname is given first ("Smith, John Henry"), especially in Asian cultures, where the surname is traditionally given first ("Kim Jeong-Su"). Yet for various reasons this can get swapped around.
- *Multiple surnames* (*e.g.*, "Juanita Martinez y Gonzales"). In Latin cultures, for example, people get a surname from their father and another from their mother, and one or both may be used in various cases.
- *Patronymic naming.* (*e.g.*, Johan, son of Sven, would be "Johan Svensen". His son Lars would be "Lars Johansen". But when migrating to America, all the children often took the father's patronymic name as their surname, so "Lars Johansen" might become "Lars Svensen," because his father was "Johan Svensen.")
- *Patriarchal naming*. (*e.g.*, "Fahat Yogol", "Fahat Yogol Maxmud", "Fahat Maxmud"). In some African cultures (*e.g.*, Somalia, Ethiopia) they use "patriarchal naming", where the person's

name is simply their own given name followed by their father's given name, then grandfather's, then great-grandfather's, and so on, depending on how "specific" they want to be. (A friend of the author's from Somalia has 17 generations memorized!). Some government records use three levels that get treated as "first, middle, last" in some cases, but when being less formal, they might use two, so the "first, middle" in one record might look like "first, last" in another one.

To handle at least some of these variations, the new FamilySearch pair classifier operates as follows.

Normalization and tokenization. Names are first normalized by removing punctuation, converting to lower case and converting diacritics to corresponding non-diacritic letters.

Names are also tokenized into name pieces, each of which is determined to be a given name piece or a surname piece. (Prefix and suffix name pieces like "mr.", "mrs.", "jr.", etc., are ignored, even in Asian languages). Often we know in advance which parts of the names are given name and surname parts (*e.g.*, because they were entered in separate fields or verified by a user). Otherwise, we have to parse the names and use our standardization routines to guess the piece type from the string, script and context.

Comparing name pieces. Because of the many ways in which names can vary for the same person, several levels of similarity are used on names. *Given name* pieces and *surname* pieces are compared separately. When comparing N name pieces from one record to M name pieces of the other, an N-by-M matrix is created, where the degree of agreement is stored.

The degree of agreement among two name pieces is determined as follows.
- *Exact:* the name pieces agree completely (ignoring difference in case or diacriticals), and the length is greater than 2 (or the name piece is Chinese, Japanese or Korean). ("john", "john")
- *Near:* the name pieces have a Jaro-Winkler similarity above 0.92. ("john", "johan")
- *Far:* the name pieces have a Jaro-Winkler similarity above 0.84; or one name "starts with" the other; or they are both a single letter. ("eliza", "elizabeth"; "elizabeth", "e")
- *Differ:* none of the above apply, so the name pieces actually conflict. ("john", "henry")

If N and M are of different sizes, then one of the names is said to have "missing" pieces compared to the other. Given this matrix, the pieces are lined up by finding the "exact" agreements first and removing the corresponding rows and columns from the matrix; then the "near" ones; then the "far" ones; and then seeing if there are any differing ones left.

Given the count of exact, near, far and differ results found, along with an indication of whether one name had "missing" pieces, a level of similarity for the entire name is determined as follows.
- 7: One "exact" name piece agreement, and at least one more piece that is exact or at least near. No "missing" pieces.
- 6: One "exact" name piece agreement, and at least one more piece that is exact or at least near. At least one "missing" piece.
- 5: One "exact", no "missing".
- 4: At least one "near", no "missing".
- 3: One "exact", at least one "missing".
- 2: At least one "far"; no "missing"
- 1: At least one "far" or "near"; at least one "missing"
- 0: At least one of the two names has no name pieces at all (so there is nothing to compare)
- -1: At least one "differ" (regardless of whether there were also "exact", "near" or "far" agreements)

These levels of similarity were arrived at by looking at pairs of records that were labeled as match or differ and seeing where the ratio of matches to differs tended to change across the data.

For names of relatives, there tends to be less data available for training, so a slightly simplified list of similarity levels are used. In these cases, 2 and 3 are combined, as are 5 and 6, so that we end up with a similar list of levels of similarity:
- 5: One "exact" name piece agreement, and at least one more piece that is exact or at least near. No "missing" pieces.

- 4: One "exact" name piece agreement, and either (a) no "missing", or (b) at least one "missing" and at least one more piece that is exact or at least near.
- 3: At least one "near", no "missing".
- 2: One "exact", at least one "missing".
- 1: At least one "far"; or at least one "near" with at least one "missing"
- 0: At least one of the two names has no name pieces at all (so there is nothing to compare)
- -1: At least one "differ" (regardless of whether there were also "exact", "near" or "far" agreements)

This abbreviated set of levels is used for the father's and mother's given and surnames; for the spouse's surnames; and for children's given names. The full set is used for the individual's own given and surnames; and for the spouse's given names.

Name Frequency (odds). Agreement on an uncommon name like "Merlin" is less likely to happen by chance than agreement on a common name like "John". In order to account for this, a table of given names and surnames was used to determine who often each name piece appears in the data. One table was built for given name and another for surname, for each of 14 cultures in the world. The frequency was inverted in order to be expressed in terms of odds. For example, "john" appears as one of the given name pieces in approximately one out of every 25 males in the "North America" culture. The table contains only names that occur more frequently than 1 in 1500 for given names, and 1 in 100,000 for surnames. (Note that there are far more surnames than given names in most cultures).

When there are one or more "exact" name piece agreements, the odds of each is looked up in the table (for the culture for both persons, if different), and the highest (most "rare") value is selected. The odds are mapped (*discretized*) into one of four groups as follows:
- Given names
  - 1: Odds <= 40
  - 2: 40 < Odds <= 300
  - 3: 300 < Odds <= 1500
  - 4: Odds > 1500 (*i.e.*, the name is not in the list of given names)
- Surnames
  - 1: Odds <= 4000
  - 2: 4000 < odds <= 10,000
  - 3: 10,000 < odds <= 100,000
  - 4: Odds > 100,000 (*i.e.*, the name is not in the list of surnames)

Odds are calculated only on the individual's own given names and surnames. Odds are not used for relatives' names.

3.2 Dates

Dates are another important type of feature that can be used to decide whether two records refer to the same real person. But as with names, we can only treat them as a piece of the puzzle. On the one hand, differing pairs of individuals will often have the same date for the same event, because of course many people were born on the same day (and even more born within the same year), for example.

On the other hand, it is also possible for matching pairs of individuals to disagree on an event date due to various reasons:
- *Estimated years.* (*e.g.*, "3 Jun 1848" vs. "about 1850"). One or both dates can often be estimated from an age in a census or some other partial information.
- *Auto-estimated years.* Sometimes algorithms have been used to estimate dates. For example, Ancestral File estimated dates based on relatives' dates, and used angled brackets to indicate that it was an estimate, *e.g.*, "<1852>". These dates could be off by decades.
- *Errors in original record.* For example, census records record the ages of individuals, but these are quite often off by up to 5 years. Sometimes this is due to "rounding to the nearest 5 years",

but other times the kid (or dad) at the door being interviewed may not remember everyone's ages right.

- *Confusion between similar events*, such as birth & christening; marriage bands & marriage; and death & burial. These "birth-like", "marriage-like" and "death-like" events sometimes get entered into each others' fields, and can be off by several days to a few weeks.
- *Lag between event and recording of event.* For example, sometimes a child is born, and a few days later the birth is recorded in the church or government records. Sometimes the date of recording is confused with the date of the actual event.
- *Entry or typographical errors.* Someone could have written the data wrong, or someone else might have read the data wrong. ("1910"/"1901"; "1720"/"172"!)
- *Calendar changes.* Calendar systems have changed on occasion, and sometimes a date is recorded different in different places. Of particular note, the Gregorian calendar started being used in 1582, but it wasn't until the 20th century that the last adherents of the Julian calendar (including Greece and Russia) finally made the switch. This sometimes causes a date to be off by 10-14 days, and, since the beginning of the year moved from March to January, can cause the year to be off by one. (See Wikipedia's Gregorian Calendar article). Sometimes a "dual date" is used to avoid ambiguity when a date comes from a time and place where the Gregorian calendar may not yet have been adopted.

In order to deal with the different kinds of variations in dates that are found among matching and differing pairs, the following levels of date matching are used in the new FamilySearch classifier.

- 3: Exact. Day, month and year are present in both dates and agree exactly. This is a powerful piece of evidence that two individuals might be the same person. Not only is a specific date more specific than a year (and thus less likely to agree by chance), but the very existence of these fields often indicates higher confidence in the data itself, since dates estimated from other day are typically no more precise than a year.
- 2: Year. Year agrees, and the day and month are either missing or within 1 day. (We have also experimented with allowing up to two weeks of difference, which seems to help).
- 1: Near. Within 2 years, and the day and month don't conflict (*i.e.*, agree or are missing).
- 0: Missing. At least one of the dates does not exist in the record, so there is nothing to compare. (Has no effect).
- -1: Differ. Year disagrees or day and month are present and off by more than a day.

The above levels are used for an individual's birth (including christening), marriage and death (including burial) events, as well as for the spouse's birth (and christening) and death (and burial) dates. Since the individual's christening is also compared with an individual's birth date, the individual's christening date (*i.e.*, comparing only christening with christening) uses an abbreviated set of levels: *Exact*, *Missing* and *Differ*.

For children's birth dates, the above set of levels did not make as much sense. Many pairs of individuals that are compared have one child in one record and another child in the other. So while having exactly agreeing child's birth dates does make a match more likely, disagreeing dates are also quite reasonable among matching pairs. Therefore, for child birth and christening dates, there is a positive weight for exact date agreement, but no weight for similar or differing dates.

Child date difference. It is rare for an individual to have children very far apart, so the child date difference feature looks at the closest year of birth between one individual's list of children and the other's. Then a different weight is applied if this difference is less than 10, 16, 22, 30 or greater than 30 years, respectively.

Date Propagation features. It is often the case that two individuals do not have date information about the same event, but do have some date information somewhere in their record. One of the most common cases involves one individual who has their own birth date (such as the child in a birth or christening record or either spouse from a marriage record); and another individual who has a child with

a birth date (*e.g.*, an individual who was a parent in a birth or christening record). To help avoid "unreasonable" matches for pairs like this, two features compare the birth date of one individual with the birth dates of children of the other, in order to see how old the first individual would have to be with the other one was having children:

- *Early child birth*. It is rare for a parent to be too young when having children (although data errors in records can cause biologically impossible situations to appear in the data). So the early child birth feature compares the birth year of one individual with all of the child birth years of the other, and finds the smallest difference. A weight is then applied if this difference is less than 5 (*i.e.*, biologically dubious), 15 (very rare), 18 (reasonable) or greater than 18 (common), respectively.
- *Late child birth*. Similarly, it is rare for a parent to be too old when having children, so the late child birth feature determines the greatest difference between birth year of one individual and child birth year of the other. A weight is then applied if this difference is less than 45 (*i.e.*, reasonable), 55 (rare), 65 (unlikely), or greater than 65 (dubious).

3.3 Places

The other main type of data found in genealogical records is place data. Knowing that two individuals are from the place makes it more likely that they are the same real person. However, there are, of course, many different people from the same place (including different people who are relatives or family members, and thus have similar names). So again, place information is just one piece of evidence that the individuals might be the same real person. However, the more specific of a place two individuals can be placed in, the smaller the pool of individuals there is in which they could be confused by chance, so this is an important piece of information to consider.

One real person can of course appear in different places for different events, due to moving, visiting or migration. They may have a single event recorded with different places for various reasons, too, including:

- *Different places for similar events*. They may have been born in one place and christened in another; or died in one place and buried in another; etc. And the place for one event may have been reported for the other similar type of event.
- *Multiple marriages*. People can remarry in a new location.
- *Estimated places*. Sometimes a place is not stated in an original record, but is estimated from other data, and the estimate could be wrong.
- *Data errors*. Sometimes one or the other places are just wrong, even though the two individuals really do refer to the same real person.

Even the same place can appear differently in different records for several reasons, including

- *Abbreviations* ("USA", "United States", "United States of America"; "VA" vs. "Virginia")
- *Different numbers of levels included*. ("Rose Hill, Lee, Virginia, United States", "Virginia, United States", i.e., one place is a subset of the other place).
- *Inclusion of place level indicators* such as "county" or "city". ("Lee, VA", "Lee Co., VA"; "Salt Lake City, Salt Lake Co., Utah"; "Salt Lake, Salt Lake, Utah")
- *Inclusion of commas* to indicate "missing levels". (", Lee, VA" vs. "Lee, VA". The latter might be interpreted as a city name instead of a county name. The former includes an extra comma as a hint that the city name is not included).
- *Changing boundaries*. A person might have been born in a town that was once in one county or country, but which later existed in another county or country when boundaries changed.
- *Place name change*. Sometimes the name of a place really does change. (*e.g.*, Istanbul was Constantinople. New York was once New Amsterdam. Why they changed it, I can't say...)

In order to handle some of the above variations, places are *standardized*, *i.e.*, parsed and looked up in catalogs, in order to come up with a fully qualified standard place (*e.g.*, "Rose Hill, Lee, Virginia,

United States"), with an associated place id for each level of the place. The country is considered the "level 1" place. The place below that (*e.g.*, state in the United States; or province in Canada; etc.) is the "level 2" place, and so on. For example, a typical place in the United States will have a city at level 4; county at level 3; state at level 2; and country at level 1.

Places for a particular event are compared to see what levels agree, and the following levels of agreement are used for place matching features:
- 8: Agreed down to level 4 (*i.e.*, levels 1, 2, 3 and 4 all have the same place id).
- 7: Agreed down to level 3, disagreed at level 4. ("Riverton, Salt Lake, Utah, United States" vs. "Draper, Salt Lake, Utah, United States")
- 6: Agreed down to level 3, no data at level 4. ("Riverton, Salt Lake, Utah, United States" vs. "Salt Lake, Utah, United States" [*i.e.*, Salt Lake County])
- 5: Agreed down to level 2, disagreed at level 3.
- 4: Agreed down to level 2, no data at level 3.
- 3: Agreed at level 1 (country), disagreed at level 2 (*e.g.*, state)
- 2: Agreed at level 1 (country), no data at level 2 (*i.e.*, at least one of the places had only a country)
- 1: Disagree at level 1 (*i.e.*, country disagrees)
- 0: No place data for one or both of the individual's places being compared. (No weight)

The above list of place agreement levels is used for
- the individual's birth place;
- spouse's birth place; and
- marriage place

Spouse family places. It is often the case that one individual (or one of their close relatives, *i.e.*, father, mother, spouse or child) has a place for one event, and the other individual has a place for a different event (or relative's event). Since people often stay in one place or at least stay near that place, it is useful to see whether the set of places overlaps between one individual and their close relatives and the other.

One feature we use is called the *spouse family places* feature. It compares the set of places related to where one individual was born with the set of places related to where the other individual was likely to be when they were having children. In particular, using the sample place similarity levels above, it compares:
1. The individual's birth place, christening place, and the spouse's birth and christening place, with
2. the other individual's marriage place and their children's birth and christening places.

All places. Another feature is called the *all places* feature. It compares all of the birth and christening places for one individual and their close relatives (as well as marriage place for the individual) with those of the other individual and their close relatives. This tends to ask the question "Did these individuals cross paths?"

One special thing about this feature is that it only fires when the spouse family places feature does not find a good agreement (since, if it did, this feature would be redundant). Therefore, if the spouse family places feature gets a zero (meaning that one or both individuals had no place data for the relevant events), then the list of all places is compared, though only down to level 3 (*i.e.*, place agreement values 6, 7 and 8 are grouped together into value 6 in the above list).

Finally, if the spouse family places feature got a value of 1, 2 or 3 (*i.e.*, disagreed on country, or agreed only at the country level), and the "all places" set agreed at level 2 (*e.g.*, state) or better, then another special value is used. So the all places values are:
- 7: Agreed down to level 2 or more, and spouse family places was 1, 2 or 3.
- 6: Agreed down to level 3 or more, and spouse family places was 0 (*i.e.*, no relevant place data available)
- 5: Agreed down to level 2, disagreed at level 3, and spouse family places was 0

- 4: Agreed down to level 2, no data at level 3, and spouse family places was 0
- 3: Agreed down to level 2, agreed at level 1 (country), disagreed at level 2, and spouse family places was 0
- 2: Agreed down to level 2, agreed at level 1 (country), no data at level 2, and spouse family places was 0
- 1: Disagreed at level 1 (country), and spouse family places was 0
- 0: None of the above (*i.e.*, spouse family places already handled it, or there was no relevant data to use)

3.4 Miscellaneous features

In addition to name, date and place features, there were a few more that were included to improve accuracy.

Gender. One "hard-coded" rule is that individuals with a different gender were not allowed to match. There are actually cases where someone estimated the gender and got it wrong, so in reality this feature is not always correct. However, since "new FamilySearch" did not allow the gender of an individual to be changed (or to be multi-valued), the match algorithm used a hard-coded large negative weight to force pairs with conflicting genders to never be considered a match.

Sibling. One of the most common situations in which records are incorrectly considered to be a match is among siblings. Siblings typically have the same surname, father name, mother name and birth place, and are often born within a few years of each other. Therefore, unless there is good information on both individuals with regards to birth date, spouse information or death information, it is easy for siblings to get a good score. Twins pose a special problem, because the birth dates are the same. One of the trickiest cases is the "same-name sibling" case. It was common if one child died to give the next child the same name. Sometimes this was due to a tradition of naming the oldest child after that father's father, or similar traditions.

One way to protect against this type of matching is to compare the parent IDs. If the actual ID of the father or mother of one individual matches that of the other one, then we have a situation where the structure of the data is telling us that these are siblings. We used this feature during bulk link, when person records had not yet been combined, to avoid merging siblings in the original data. Once data was combined in new FamilySearch, however, we had many situations where a family had been partially combined, leaving a copy of a child from one original family and another copy from another original family sitting in the same family. Since the parent ids then matched, this feature prevented their being brought together, so this feature was later removed. However, we continue to use different confidence levels (match score thresholds) for individuals who have the same parent IDs when compared to those who do not, which is similar to continuing the use of this feature.

Own Ancestor (aka "I'm my own Grandpa"). Sometimes a father and a son have a lot of information in common, especially if the son is named after the father. Therefore, we put in another feature with hard-coded weights to make it impossible to match two individuals if we can tell that one is the parent or grandparent of the other. This is done by seeing if either individual's ID is in the set of parent or child IDs of the other; or if any of the child IDs of one individual is in the set of parent IDs of the other. If so, then one is the ancestor of the other, and the match is not allowed.

No Names. One more thing we found over time was that it was common to find an individual with no names randomly matching several attributes during a match operation, and thus getting a decent score without being penalized for a conflicting name (as would usually be the case when comparing two randomly-selected individuals). Since no-name records were rare in our labeled data, we added a feature that fired when an individual had no names, and hard-coded its weight.

4. Results

The features described in this paper have been used by FamilySearch for doing bulk linking of a billion-individual database, and are used in ongoing duplicate checking. Weights for these features were trained using a single-layer neural network (Rumelhart & McClelland, 1986) on 48,000 labeled pairs, and tested on 32,000 pairs. These results were significantly higher than those achieved by using traditional Probabilistic Record Linkage (Newcombe, 1959; Fillegi & Sunter, 1969; Newcombe, 1988)—which is equivalent to the "naïve Bayes" machine learning method (Michie *et al.*, 1994; Quass & Starkey, 2003)—using the same features. Both of these were more accurate, however, than using simple "normalized field agreement" features as was used by earlier systems like TempleReady and Ancestral File.

For example, at a precision level of 99%, Probabilistic Record Linkage (PRL) with simple features had a recall of 41%; a neural network with simple features had a recall of 45%; PRL with the features described in this paper had a recall of 69%; and these same features with weights trained by a neural network had a recall of 91%. The results clearly show the power of using a combination of neural network training and powerful features to maximize accuracy.

5. Conclusions

The features presented in this paper give a genealogical record linkage algorithm the discriminating power necessary to provide powerful person matching.

Additional features have been tried, such as better handling of patronymic name stems (*e.g.*, "-sen", "-son", "-se", "-sdatter", etc.); using name standardization catalogs to improve name comparison; using data propagation more generally to distinguish between those who lived at conflicting time frames and those whose data is compatible; and so on. Some of these have shown some promise. One other area of promise that needs more exploration is "graph-matching" (Wilson, 2001), which takes into account how well an entire segment of a relationship graph lines up between two trees before a final decision on whether each individual is a match.

Accurate matching helps users of a system find individuals they are looking for without having to wade through so much irrelevant (non-matching) data. Accurate matching also helps to avoid incorrectly combining non-matching individuals. Since data in one system often ends up finding its way into others, it is in the best interest of the genealogical community for everyone to do matching as accurately as possible. Hopefully the features described in this paper can serve as a starting point for further improvements in the state of the art in matching algorithms.

References

Dunn, H. L. 1956. Record Linkage, *American Journal of Public Health*, vol. 36, 1412-1416.

Fillegi, I. P. and Sunter, A. B. 1969. A Theory for Record Linkage, *Journal of the American Statistical Association*, vol. 64, 1183-1210.

Michie, D., D. Spiegelhalter, and C. Taylor. 1994. *Machine Learning, Neural and Statistical Classification*, Ellis Horwood, Hertfordshire, England. Book 19.

Newcombe, H. B., Kennedy, J. M., Axford, S.J., and James, A.P. 1959, Automatic linkage of vital records. *Science*, vol. 130, 954-959.

Newcombe, Howard B. 1988. *Handbook of Record Linkage*, Oxford University Press, New York.

Quass, Dallan, and Starkey, Paul. 2003. Record Linkage for Genealogical Databases, *ACM SIGKDD '03 Workshop on Data Cleaning, Record Linkage, and Object Consolidation,* August 24-27, 2003, Washington, D.C.

Rumelhart, D. E., and J. L. McClelland. 1986. *Parallel Distributed Processing*, MIT Press.

Wilson, D. Randall, 2001. Graph-Based Remerging of Genealogical Databases. *Family History Technology Workshop (FHTW2001)*.