Utilizing Stacking for Feature Reduction in Graph-Based Genealogical Record Linkage

Stephen Ivie, Yao Huang Lin and Christophe Giraud-Carrier
Department of Computer Science, Brigham Young University, Provo, UT 84602, USA
steve.ivie@gmail.com,yao.huang@cs.byu.edu,cgc@cs.byu.edu

***Abstract*— Genealogy research is centered on collecting records about an individual from various sources and combining the information to gain a larger historical perspective about that individual, commonly in the form of a pedigree. Data extraction, the internet, and other technological advancements have made large amounts of digital genealogical data more accessible. Discovering the relevancy of a digital record to a given pedigree involves determining if the individual described in the record is in actuality an individual within the pedigree. This process is called Genealogical Record Linkage (GRL). GRL can be automated through data mining and techniques by creating machine learned models from hand labeled comparisons. In this paper, we compare two such models-a tabular approach and a graph based stacking approach-and report the successful application of both on a large, post-blocking database. We also note the successful integration of these approaches in an open source distributed genealogy program that finds relevant machetes to a given pedigree from multiple online repositories.**

I. Introduction

At the core of genealogical research is the process of gathering related information from multiple sources and combining the information to gain a larger historical perspective. For example, birth, death, and marriage certificates can be combined to show a basic timeline of events for individual, the individual's parents, spouse, etc. Further information can be combined to form entire pedigrees, detailing many generations of ancestry and progeny.

Current technological advancements have contributed to an explosion in the amount and availability of digital genealogical information. Data extraction, optical character recognition (OCR), and other digitization techniques create many new records daily. Many internet sites make large repositories of genealogical information readily available to the public. With so many resources, finding and combining relevant information from multiple sources becomes a difficult problem. Data entry errors, unstandardized abbreviations, and discrepancies between sources further complicates the problem. For example, two records may have drastically different birthdays and names, but the records refer to the same person. In this paper, we address part of this problem: the need for an automated approach in determining the relevancy of a piece of genealogical information to a given pedigree. This approach is based upon record linkage.

Record linkage consists of discovering records such that records that are believed to refer to the same entity are treated as a single entity. Excellent overviews of techniques and research issues relevant to record linkage in general are in [9], [27]. In this paper, we focus on genealogical record linkage (GRL).

GRL is significant to genealogical research because it consolidates and links numerous databases, resulting in condensed search results that have a broad range of highly related information. GRL also helps genealogical researchers identify where their work overlaps with the work of others. Furthermore, GRL has application in medical genetics, where researchers identify the heredity of diseases and disorders such as cancer, Hunington's disease and febrile seizures, using medical pedigree charts [6], [2], [22], [13].

GRL differs from other record linkage problems in the quantity and nature of the attributes used to represent entities. Where most record linkage projects have records that consist of a small and finite number of densely populated attributes, GRL tends to have an infinite number of attributes. Comparing two records involves comparing only their common attributes, which results in sparsely populated, multi-valued comparisons. For example, an individual can have multiple spouses (due to remarriage, etc.), many children, many siblings, and a vast posterity and ancestry, each with numerous attributes.

In this paper we compare two data mining approaches to GRL both based on creating machine learning models from hand labeled comparisons. The first approach comes from our prior work, a Metric Based approach to Genealogical Record Linkage (MBGRL) [10], [11]. In this approach, each element of a pedigree pair was compared and scored using a similarity metric and inserted into a table. A machine learning model was then created using this table. In this paper we present an alternative approach: a Stacked Graph-based approach to Record Linkage (SGRL). In this method a pedigree is represented by a unidirectional graph. By utilizing the hierarchal nature of the graph, higher order metrics are made from basic comparators through stacking. Results on a large genealogical database show higher f-measure and accuracy for the graph-based approach.

II. Developing and Choosing Comparison Metrics

Most genealogical record linkage problems involve comparisons among primarily four types of basic data types: name, gender, date, and location. Combinations of these basic types can then form aggregates and collections of aggregates. A wide variety of metrics were tested in each of these basic comparison areas for both MBGRL and SGRL. To determine

which metrics was most appropriate on each data type, a metric performance evaluation was performed, as follows:

A. Metric Performance Evaluation Criteria

All metrics in a comparison category (name, date, location, etc.) were compared with each other using the following three criteria.

- ***Information Gain.*** The formula for information gain is given by:

$$Entropy(S) = -p_M \log p_M - p_m \log p_m$$

$$Gain(S, A) = Entropy(S) - \sum_{v \in V_A} \frac{|S_v|}{|S|} Entropy(S_v)$$

  where S is the collection of normalized scores a particular comparison metric generates (together with their associated target value), p_M is the proportion of matches, and p_m is the proportion of mismatches, V_A is the set of possible values of attribute A, and $S_v = \{s \in S : A(s) = v\}$ (the subset of S where attribute A has value v). Information gain measures how well a given attribute (consisting of the results of a metric comparison) separates the training data according to its target classification (match, mismatch).
- ***F-score.*** The formula for F-score is given by:

$$Precision = \frac{TP}{TP + FN}$$

$$Recall = \frac{TP}{TP + FP}$$

$$F\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

  where TP is the number of true positives (i.e., correctly labeled matches), FN is the number of false negatives (i.e., matches incorrectly labeled as matches), and FP is the number of false positives (i.e., mismatches incorrectly labeled as matches). The F-score tries to combine precision and recall into a single measure. F-scores were calculated using 10-fold cross-validation.
- ***Overall Accuracy.*** The overall accuracy is simply the ratio of the number of correctly labeled pairs to the total number of pairs in the training set. Overall accuracy was computed using 10-fold cross-validation.

A metric was considered to be superior only if it outperformed the other metrics in its category on all 3 of the above criteria

B. Metric Performance Evaluation Results

A number of metrics were tested for each basic data type and evaluated based on the criteria of the previous section. The following comparison metrics were found to be superior in their respective comparison groups.

TABLE I
NAME COMPARISON METRIC SCORES

Comparator Metric	Accuracy	F-Score	I-Gain
NamesNeuralNet	***91.5%***	***.85***	***.461***
JaroWinkler	89.7%	0.821	0.412
EnsembleCJMN	89.2%	0.8	0.368
EnsembleOfEditDistance	88.7%	0.79	0.354
NeedlemanWunch	87.6%	0.76	0.32
ChapmanMatchingSoundex	80.3%	0.695	0.242

1) Name Comparison Metric: The two most common categories of metric-based approaches for comparing names are phonetic comparisons algorithms and pattern comparison matching algorithms. Common phonetic algorithms we explored include: Soundex [29], Chapman Soundex, Phonex [14], Phonix [7], and Double-metaphone [24]. Common pattern and edit distance based algorithms we explored include: Levenshtein [18], Needleman & Wunch, Monge-Elkan [17], and Jaro-Winkler [12]. A weighted ensemble of some of the above string metrics —Monge-Elkan, Jaro-Winkler and ChapmanSoundex— was also tested. These comparators have consistently shown high accuracy on name matching tasks [3], [23]. Scores were calculated using the name comparisons of the base individuals in 10,349 labeled pedigree pairs. An aggregate name comparator was created using outputs from various metrics as inputs into an artificial neural network. Accuracy, f-score, and information gain are shown in Table I. A precision-recall curve for the highest scoring metric, the name neural network, is shown in figure 1.

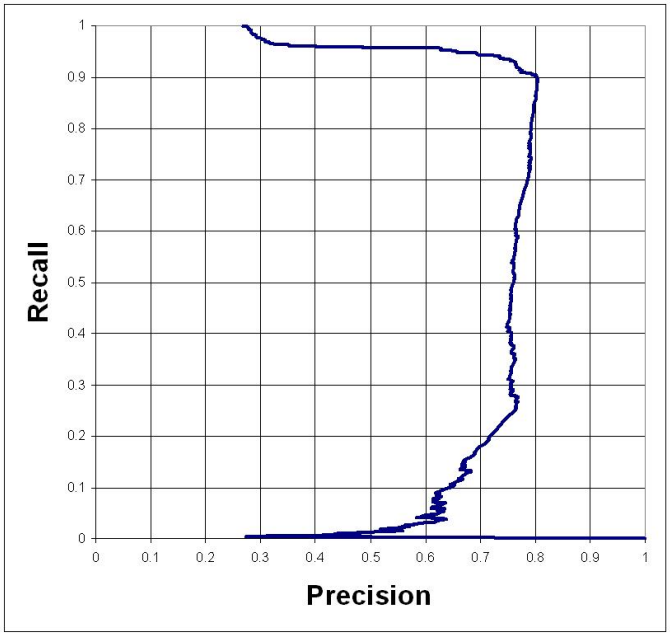

Fig. 1. Precision Recall Curve of Neural Network Name Comparison

2) Date Comparisons Metric: Three types of date comparators were originally evaluated. First, various edit distance

TABLE II
DATE COMPARISON METRIC SCORES

Comparator Metric	Accuracy	F-Score	I-Gain
NeuralNetDates	***91.4%***	***.856***	***.473***
DateDifference	89.9%	.833	0.429
EnsembleEditDist	89.6%	.83	0.429
EnsembleCJMN	89.5%	.828	0.426
JaroWinkler	89.5%	.828	0.426
NeedlemanWunch	89.4%	.829	0.431

TABLE III
LOCATION COMPARISON METRIC SCORES

Comparator Metric	Accuracy	F-Score	I-Gain
LocationsNeuralNet	***79.6%***	***.566***	***.138***
EnsembleEditDistance	66.7%	0.048	0.006
EnsembleCJMN	66.4%	0.048	0.007
GeoLocationComparator	35.3%	0.401	0.002
JaroWinkler	34.3%	0.198	0.074
NeedlemanWunch	30.0%	0.21	0.091

based metrics were used which made allowances according to common data recording errors. For example, the comparison of 21 June 1800 and 12 June 1800 will score slightly lower (having a greater probability) than 10 June 1800 and 19 June 1800, because the common error of reversing date digits implies a slightly higher probability of being a match (i.e., it is more likely that "21" matches "12" than that "10" matches "19", even though the difference in number of days is the same. Second, time based comparators in which a similarity score is calculated primarily according to the absolute value of the difference in number of days between the two dates. For example 1 June 1800 and 10 June 1800 would conceptually result in a score of 9 because there is a 9-day difference between the two dates. Hence, a score of 0 means an exact match and a high score implies a low probability that the two dates match. Finally, an aggregate date comparison type was created as a combination of these metrics as inputs into an artificial neural network, since the metrics preform differently on different examples. This combination date comparator had the highest accuracy, f-measure, and information gain as shown in table II, and its precision recall graph is shown in figure 2.

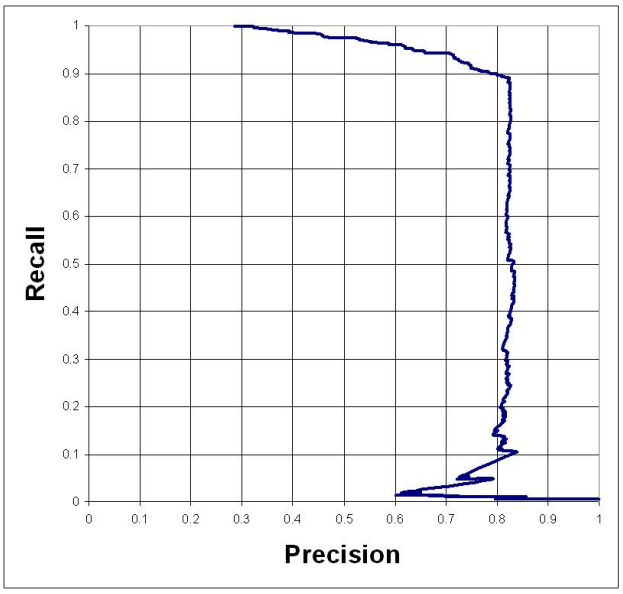

Fig. 2. Precision Recall Curve of Neural Network Date Comparison

3) Location Comparison Metric: As stated earlier, the locations in the dataset have already been standardized. This means that misspellings, variations and abbreviations have been previously resolved to actual locations (past and present). Location strings are compared initially to see if there is an exact match, assuming all four parts of a location are present (i.e., city, county, state, and country). If they are not a match, traditional string comparison metrics are rendered useless because the location names have already been standardized. For example, it makes no sense to compare the string similarity of "Manhattan" and "New York City." Instead, a physical distance metric was created.

Using Yahoo Maps online services, literal distances are calculated between two locations (cities). Using a physical distance metric allows for greater sensitivity of determining common data entry errors. For example, one birthplace may erroneously list a larger city like Salt Lake, rather than the actual suburb, like Sandy. Another common location discrepancy exists between a pedigree that lists the city of the hospital an individual was born in, and another pedigree lists the city the individuals parents lived in when the individual was born (e.g., someone was born at the Provo hospital, but lives in and is from Orem).

Over a period of time, a database was created with every unique location in the database and its corresponding geo-coordinates. Distances were calculated as follows.

$$D = r.[\sin La_1 . \sin La_2 + \cos La_1 . \cos La_2 . \cos(Lo_2 - Lo_1)]$$

where r is the radius of the earth in kilometers, La is latitude in radians, and Lo is the longitude in radians. Over 94% of the locations could be resolved to coordinates (the remainder are cities that no longer exist, are not yet indexed by yahoo, etc).

This standardized, literal-distance location metric shows minor improvements in performance when compared to edit distance string comparison metrics. This metric is also 'future-friendly', as many genealogy programs allow users to enter GPS coordinates for locations, such as grave sites [1]. When used in combination with other string comparison metrics, higher performance levels are achieved, as shown in table III. However, location attributes are less indicative of the target value, and so only low levels of precision and recall were achieved (see Figure 3).

III. STRUCTURE DIFFERENCES IN MBGRL AND SGRL

In GRL, a record is a pedigree consisting of a base individual, his/her siblings, spouse, progeny and ancestry, all with basic information about major lifetime events including

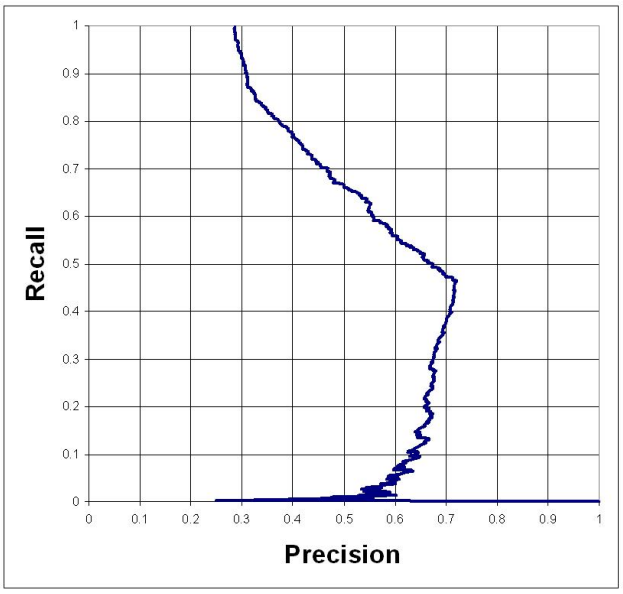

Fig. 3. Precision Recall Curve of Neural Network Location Comparison

dates and places. GRL primarily focuses on determining whether or not a genealogical record refers to the same person as the base individual of a pedigree. In each case, the genealogical record can be treated as a partial pedigree. Each pedigree in such a comparison may be very unique due to spelling errors, data entry errors, variations between two or more databases, missing values, etc. As such, GRL considers more than exact-match pedigrees; it considers pedigrees that may differ drastically, but in actuality refer to the same individual. Both MBGRL and SGRL utilize the same basic comparison metrics listed above. However, they are very different in structure. MBGRL utilizes a "flat table" approach where every basic information type in a pedigree is compared with its corresponding basic information type on another pedigree, and the result of each comparison becomes an attribute in an instance row of the table. For example, the comparison score of the individuals name becomes the first attribute, the fathers' name comparison score becomes the second, etc. This results in sparse table with a large number of attributes. Due to the sparesness of many attributes, any chosen machine learning algorithm will be sensitive to using low weights or ignoring attributes that are significant for a match.

A. A Graph Based Approach

SBGRL utilizes a graph based data structure. At the base of this structure is a date and a location. Combined, these form an significant event. A significant event is a unique occurrence in one's life, e.g. a birth, marriage or death. Assuming the probability that two unique people share this exact event is relatively the same among all given significant events, then each significant event that matches between to pedigrees should carry approximately the same amount of weight in a prediction model. This can be accomplished by creating a collection of events and using summarative scores to represent the collection. In SGRL, we chose the following summarization scoring mechanisms:

- Maximum score
- Minimum score
- Standard deviation
- Number of comparison scores above a threshold
- Number of comparison scores below a threshold

In GRL, there are many comparisons that have one-to-many relationships that can be combined to form aggregates (i.e. location and date from a significant event) or collections(i.e. a set of events). For example, a person may remarry multiple times and thus have a number of spouses; a person may also have a large number of children, many siblings, etc. By combining basic comparison types to form aggregate comparators, then combining aggregate comparators to form groups comparators, more advanced comparators can be introduced.

In SGRL, this is done through stacking. Basic comparisons are combined to make higher order comparisons by using the basic comparisons' output scores as input into a Multilayer Perceptron, commonly referred to as an artificial neural network. The output of the neural network is a regression probability score that summarizes its inputs. Output scores from multiple models of the same type can then be summarized using the summarative scores listed above. These summarative scores then serve as inputs to a neural network that can classify the collection. The complete stacking framework is shown in figure 4.

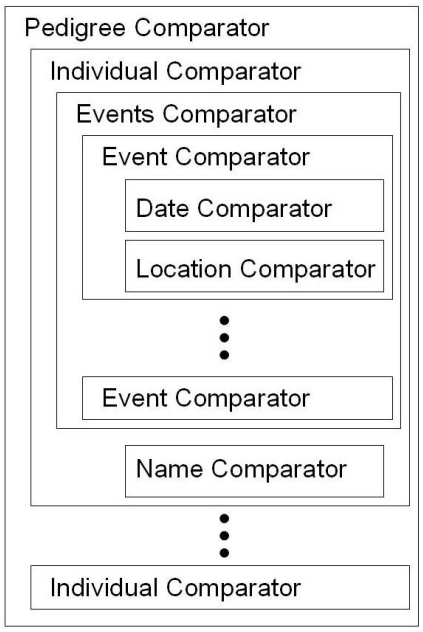

Fig. 4. Aggregates of SGRL

Training examples are labeled such that a match has target value 1 for the MATCH node and target value 0 for the MISMATCH node, and the reverse for mismatches.

IV. Experimental Results

MBGRL was created using Weka's backpropagation learning algorithm(artificial neural network), with disproportional weighting towards false negatives (type II error), due to the bias inherent in the data (the 1:3 ratio of non-matches to matches). A validation set of 10% of the training set was applied to preserve generality and discover the optimal model. This model preformed significantly better than other learning algorithms on the same flat table; however, it was sensitive to the large number of unknown values, as well as having too large of an attributes per records ratio.

As a result, we turn our attention to a graph based stacking approach as an alternative. By treating the data as a graph, rather than a flat table, we are able to generalize rules to combine attributes at each level, thereby preforming feature reduction. Attributes that may have been ignored due to the spareness of instances with that attribute were able to carry a heavy influence on such statistics as best event score. Furthermore, the model is more general, and can handle new attributes that fit into the higher level aggregates (i.e. a bar mitzvah record can be compared to a pedigree, even though the model had no training instances of that particular event since the model treats all events the same). The resulting set of attributes in the final stage is significantly reduced, and the resulting model suffers the effects of over-fitting less due to too many attributes.

The genealogical database used in our experiments was provided by the Family and Church History Department (FCHD) of The Church of Jesus Christ of Latter-day Saints. The database consists of a set of pedigree comparisons, where each pedigree comparison is labeled as either being a "match" or "non-match." The distribution of matches to non-matches is approximately 1:3 (i.e., 1 match for every 3 non-matches), or approximately 25% matches. The database contains over 16,000 labeled pedigree comparisons, split evenly into 3 sets. Sets 1 and 2 are used for all training, and initial testing purposes using 10 fold cross validation. Set 3 is held as a final test set used to verify the algorithms that show promise in sets 1 and 2.

The database consists of names of people (e.g., "Jane Doe"), relationships (father, mother, sibling, child, spouse), and events (birth, christening, marriage, burial, etc.). Blocking on the database was performed previously by the FCHD so that only pairs that are very similar are left in the provided database. The database has also been heavily standardized, meaning it has gone through several data cleaning and attribute-level reconciliation algorithms that have made every attribute conform to some standard form. For example, all abbreviations and misspellings in the city attribute have been converted to actual, full and unabbreviated city names. Finally, as explained previously, SGRL consolidates over 300 original attribute comparisons to 12 normalized attributes.

Once the multi-tiered artificial neural network has been induced (see above), the test set is run through it (the first time that set is used for any purpose). The combination of our metric-based algorithms results in high accuracy, F-score, precision, and recall as shown in Tables IV and V. Table IV is the confusion matrix (0 stands for Mismatch and 1 stands for Match), and Table V summarizes the main quantities of interest.

TABLE IV
Confusion Matrix For MBGRL

		Predicted	
		0	**1**
Actual	**0**	3,719	65
	1	102	1,276

TABLE V
Summary Statistics

Summary	MBGRL	SGRL
Actual Matches	1,378	[Pending]
Actual Mismatches	3,784	[Pending]
Total Comparisons	5,162	[Pending]
Match/Mismatch Ratio	0.3642	[Pending]
Accuracy	0.9676	[Pending]
Precision	0.9260	[Pending]
Recall	0.9515	[Pending]
F-score	0.9386	[Pending]

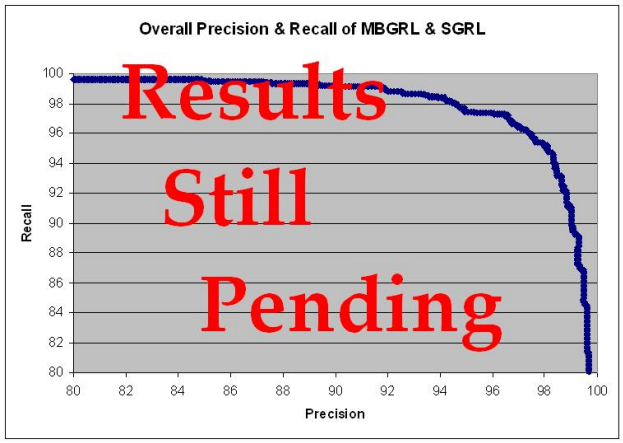

Fig. 5. Precision-Recall Curves of SGRL & MBGRL

Experiments with other machine learning, including artificial neural networks, logistic regression, C&R Tree, and CHAID, result in lower accuracy and F-score than the backpropagation algorithm in MBGRL..

V. Integration In Genesis

After testing and finding a SGRL model that had high performance measures, the SGRL model was serialized and

TABLE VI
Confusion Matrix For SGRL

		Predicted	
		0	**1**
Actual	**0**	[Pending]	[Pending]
	1	[Pending]	[Pending]

included as an updateable plugin to the open source genealogy project names Genesis. Genesis uses SGRL by searching numerous online genealogical repositories for information related to a pedigree, and then scores the relevancy of the information according to SGRL. Information with a high score is automatically combined, low scores are discarded, and a range of scores in between are ranked and flagged for human review. Source code is available at www.dtfproject.org.

VI. Conclusion

Details of the results of SGRL are still pending. Our work on record linkage is ongoing. Finding ways to increase sensitivity on the basic metrics of name comparisons, date comparisons, and location comparisons may show further improvements. Innovative solutions to the one-to-many comparison problem are also promising. We are also exploring the use of Markov Logic Networks [4] within our sparse and uncertain genealogical context. Finally, choosing 3 target attribute values of *match*, *mismatch* and *unknown: needs human intervention*, and optimizing for these values, may prove more appropriate for many genealogical record linkage contexts.

Acknowledgments

Data for our experiments was graciously provided by the Family & Church History Department of the Church of Jesus Christ of Latter-day Saints. This work is funded in part by a BYU Mentoring Environment Grant.

References

[1] Booth, M.T. (2006). Enhancing Your Genealogy Using GPS. Available online at http://www.personalhistorian.com/Support/ EnhancingYourGenealogyWithGPS.pdf. Retrieved March 1, 2007.

[2] Bourret, P. (2005). BRCA Patients and Clinical Collectives: New Configurations of Action in Cancer Genetics Practices. *Social Studies of Science*, **35**(1):41-68.

[3] Cohen, W.W., Ravikumar, P. and Fiendberg, S.E. (2003). A Comparison of String Distance Metrics for Name-Matching Tasks. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*.

[4] Domingos, P., Kok, S., Poon, H., Richardson, M. and Singla, P. (2006). Unifying Logical and Statistical AI. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, 2-7.

[5] Fellegi, I.P. and Sunter, A.B. (1969). A Theory for Record Linkage. *Journal of the American Statistical Association*, **64**:1183-1210.

[6] Frezzo, T., Rubinstein, W., Dunham, D. and Ormond, K. (2003). The Genetic Family History as a Risk Assessment Tool in Internal Medicine. *Genetics in Medicine*, **5**(2):84-91.

[7] Gadd, T. (1990). PHONIX: The algorithm. *Program: Automated Library and Information Systems*, **24**(4):363366.

[8] GENMERGE. http://www.genmerge.com/ (demonstrated at the *4th Annual Workshop on Technology for Family History and Genealogical Research*, Provo, UT.

[9] Gu, L., Baxter, R., Vickers, D. and Rainsford, C. (2003). Record Linkage: Current Practice and Future Directions. CMIS Technical Report No. 03/83.

[10] Ivie, S., Pixton, B. and Giraud-Carrier, C. (2007). Metric-Based Data Mining Model for Genealogical Record Linkage. *In Proceedings of the IEEE International Conference on Information Reuse and Integration*.

[11] Ivie, S., Henry, G., Gartell, H. and Giraud-Carrier, C. (2007). A Metric-Based Machine Learning Approach to Genealogical Record Linkage. *In Proceedings of the 7th Annual Workshop on Technology for Family History and Genealogical Research*.

[12] Jaro, M.A. (1995). Probabilistic Linkage of Large Public Health Data File. *Statistics in Medicine*, **14**:491-498.

[13] Johnson, W.G., Kugler, S.L., Meulener, M.C., et al. (1998). Pedigree Analysis in Families with Febrile Seizures. *American Journal of Medical Genetics*, **61**(4):345-352.

[14] Lait, A. and Randell, B. (1993). An Assessment of Name Matching Algorithms. Technical Report, Deptartment of Computer Science, University of Newcastle upon Tyne, UK.

[15] Lee, C., Rey, T., Mentele, J.W. and Garver, M. (2005). Structured Neural Network Techniques for Modeling Loyalty and Profitability. In *Proceedings of SAS User Group International (SUGI 30)*.

[16] Lendaris, G.G., Zwick, M. and Mathia, K. (1993). On Matching ANN Structure to Problem Domain Structure. In *Proceedings of the World Congress on Neural Networks*, 488-493.

[17] Monge, A. and Elkan, C. (1996). The Field-Matching Problem: Algorithm and Applications. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 267-270.

[18] Navarro, G. (2001). A Guided Tour to Approximate String Matching. *ACM Computing Surveys*, **33**(1):3188.

[19] NeSmith, N.P. (1992). Record Linkage and Genealogical Files. *Utah Genealogical Journal*, **20**(3-4):113-119.

[20] Newcombe, H.B., Kennedy, J.M., Axford, S.J. and James, A.P. (1959). Automatic Linkage of Vital Records. *Science*, **130**:954-959.

[21] Newcombe H.B. (1967). Record Linking: The Design of Efficient Systems for Linking Records into Individual and Family Histories. *American Journal of Human Genetics*, **19**(3 Pt 1): 335-339.

[22] Nukaga, Y. (2002). Between Tradition and Innovation in New Genetics: The Continuity of Medical Pedigrees and the Development of Combination Work in the Case of Huntington's Disease. *New Genetics and Society*, **21**(1):39-64.

[23] Pfeifer, U., Poersch, T. and Fuhr, N. (1996). Retrieval Effectiveness of Proper Name Search Methods. *Information Processing and Management*, **32**(6):667-679.

[24] Philips, L. (2000). The Double-metaphone Search Algorithm. *C/C++ Users Journal*, **18**(6).

[25] Pixton, B. and Giraud-Carrier, C. (2006). Using Structured Neural Networks for Record Linkage. In *Proceedings of the Sixth Annual Workshop on Technology for Family History and Genealogical Research*.

[26] Quass, D. and Starkey, P. (2003). Record Linkage for Genealogical Databases. In *Proceedings of the the KDD-2003 Workshop on Data Cleaning, Record Linkage and Object Consolidation*.

[27] Winkler, W.E. (2006). Overview of Record Linkage and Current Research Directions. Research Report Series (Statistics #2006-2). Available online at http://www.census.gov/srd/papers/pdf/rrs2006-02.pdf.

[28] White, D. (1997). A Review of the Statistics of Record Linkage for Genealogical Research. In *Record Linkage Techniques: Proceedings of an International Workshop and Exposition*, 362-373.

[29] Zobel, J. and Dart, P. (1995). Finding Approximate Matches in Large Lexicons. *Software–Practice and Experience*, Vol. 1, 331-345.