Development of An Evaluation Paradigm for “RecordMatch” and its Application to GenMergeDB Clustering Results

Patrick Schone
Church of Jesus Christ of Latter-day Saints
50 East North Temple
Salt Lake City, UT, USA
Patrickjohn.Schone@ldschurch.org

Abstract

For years, the LDS Church has allowed genealogical patrons to search for their ancestors amongst large collections of indexed records where each record has been treated as independent of all other records. Recently, the Church has begun to consider the potential for providing patrons with clustered results of records. These clusters would attempt to identify the same entity across multiple record collections. This clustering process, which we will refer to as “RecordMatch” is, in essence, a very large entity disambiguation task across partially-populated records. We describe and implement a rigorous evaluation for RecordMatch by taking advantage of the groundwork lain by comparable evaluations in the research community. We then apply this evaluation framework to the results of an existing genealogical record-merging system, GenMergeDB, and demonstrate the high-level of success of existing technologies for the purpose of producing RecordMatch-type clusters.

1 Introduction

The LDS Church has significant interest in providing readily-accessible genealogical resources to anyone in the world with an interest in seeking after their ancestors. Since the advent of the personal computer, the LDS Church has attempted to digitize their genealogical information and make such data searchable. In the last few years, the Church has increased its stream of digital publications to include the presentation and search of digital historical images. With the click of a button, patrons today can search millions of digital images which the Church has transcribed and released, which include census records; birth, marriage, and death records; military records; and so forth.

Currently, as patrons search these digitized records, each image is treated by the system as being independent from any other image. Thus, if there is a “John Smith” in one image which is the same person as the “John Smith” mentioned in another record, the two records currently will be presented to the user separately. Though Church records could conceivably contain all known vital statistics for an ancestor, the patron would have to weave such information together by himself.

To alleviate this burden for the user, the Church has begun experimentation with a process that we will here call “RecordMatch” for lack of a better term. The notion of RecordMatch is to develop and use software systems which, when applied to the large cache of Church records, will attempt to discover mentions of entities which are *co-referent* (i.e., they refer to the same individual). When a patron then queries for an ancestor, RecordMatch could conceivably show them a compiled view of all known records associated with that particular individual. Such information could allow users to quickly grow genealogical trees from primary record sources in minutes – a feat which, only two decades ago, may have taken years and significant amounts of money.

RecordMatch-type software is an instance of a “record-linking” or “entity disambiguation” system. Such a system determines when two name strings refer to the same individual – usually by using collateral information such as textual context or other record information. It can be the case that two name strings can look exactly the same and yet refer to different people (where “John Smith is a common instance of this), or that two names can be quite distinct but refer to the same person, as with “Norma Jean Baker” and “Marilyn Monroe.”

A major difficulty for constructing any entity disambiguation system is the establishment of an evaluation collection (or "*corpus*") to confirm the accuracy of the system's results. This difficulty is particularly pronounced when the number of documents of issue is in the millions or in the billions, such as with the LDS Church image repositories.

In order for an evaluation to effectively address user needs, it must be able to answer two questions: (1) if the system proposes that name mentions A and B actually refer to the same person, how likely is it that the proposed connection is valid?; and (2) if the system does mention that A and B are coreferent, how likely is it that there is another mention, C, that is also coreferent but which was not identified by the system? The first of these issues refers to *precision*, or the probability that proposed connections are valid. The second is an issue of *recall*, or the probability that the connections that should have been found actually were found.

In this paper, we leverage processes attested to in the research community for establishing an evaluation for RecordMatch resources which can be used to evaluate both precision and recall. We also describe the methodology for implementing these processes on the LDS Church's family history data and we construct an actual evaluation using thousands of historical records. In particular, we apply this methodology to a collection of multiple image repositories specifically focused on Utah from 1850 to 1956.

Lastly, we apply the evaluation paradigm to the actual, blind results of a family-reconstitution system from Pleiades Software, Inc. called "GenMergeDB." The goal of the GenMergeDB software is to actually be able to piece together full families from across various record collections, but we use it here only to cluster individuals. We say that the evaluation is blind here because the evaluation for this work was established and developed independent of the GenMergeDB software or of its developers. Therefore, there was no tuning of system parameters on the test data and the results can be treated as quite legitimate. Through this evaluation, we show that existing software, such as GenMergeDB, when applied to a record collection of several million records, can truly address RecordMatch-type needs with very high precision and great recall. (Note that this paper does not attempt to predict performance of such technologies when applied to data collections that have orders of magnitude more records.)

2 Description of the Evaluation Paradigm

Entity disambiguation on large collections of data has been a subject of recent interest in academic, industrial, and governmental communities. As the amount of data in the world has grown, there has become a pressing need to provide users with tools that allow them to find specific pieces of information they are seeking. For example, if they are interested in "Michael Jordan," it is becoming more important for search tools to determine which specific "Michael Jordan" is desired since there are the famous people (such as the basketball player, the linguist, the politician, the mycologist, the footballer, the actor, etc.) and not-so-famous people that share that name. Therefore, there have been evaluations that have been stood up by various communities to address different disambiguation issues. These evaluations include the *WebPeople* challenge (Artiles, et al., 2008); NIST's *Automatic Content Extraction Cross-doc Coreference Task* (Przybocky 2008); and the entity-linking and slot filling tasks of *TAC-KBP* (McNamee, et al., 2010). Rather than inventing our own process for evaluation of RecordMatch, we attempt to leverage this prior work.

2.1 Use of Seed Entities as Starting Point

A starting issue for the establishment of an appropriate evaluation is to first make a determination of the information that must be vetted in order to have the evaluation be truly meaningful. It would be almost completely intractable for a human or even an army of humans to properly compare all of the instances (or *mentions*) of entities in a huge collection and determine which are co-referent and should be clustered versus those which should not be. Also, if the evaluation were to proceed by first having a computer system propose what it deemed to be correct and then having humans vet the output, the evaluation results would be seriously skewed in favor of precision and make the results less meaningful in terms of recall.

Starting with a small-ish, finite collection of seed mentions from which to grow full ancestral clusters is an appropriate strategy to provide both tractability and the proper analysis of precision and recall in an evaluation. A comparable strategy was used in the NIST Knowledge Base Population task (referred to as *TAC-KBP)*. In preparation for the TAC-KBP task, participants were given a million-plus document collection of raw texts. They were also given a knowledge base -- distilled facts about unique individuals. For the evaluation, they were then given two different kinds of entity disambiguation tasks, *entity linking* and *slot filling*. For entity linking, participants were given several thousand references to people, places, or organizations that were observed in the million-document collection and were asked "for each of these observations, can you tell which, if any, of the knowledge base entries are referring to the same entity as this document's mention." For slot filling, they were asked to use that seed mentions to further populate the appropriate knowledge base entry with any new information that could be gleaned from the entire millon-document collection.

We could build our evaluation by treating the RecordMatch task as being a combination of entity linking and slot filling. We vet results by beginning with a collection of seed mentions of people about whom we want to find more information (more specifically, we want to discover all other coreferent documents). A "seed" here is one record mentioning a particular individual. Note also that since each record is pre-distilled information from historic documents, that single record has some comparability to a weakly-populated knowledge base entry. ***Our first task, then, is to use a list of seed records to find all other records in the collection that are co-referent as well as to distinguish records that look co-referent but are not***. We will describe in more detail our actual approach for doing this in Section 3 of this paper.

2.2 Scoring the Results

For each seed entity, *S*, in our evaluation, our truth data will need to include all mentions of that entity, and, where possible, other entities whose mentions may look like those of *S* but which are actually separate. To be more concrete, if *S* is a particular "Jane Smith," one would like to see that the truth data has a comprehensive inventory of all the mentions in the collection that refer to that "Jane Smith" (including name variants, misspells, maiden names, partial names, and so forth). Also, if there are other people who share the same name or one or more of the variants, it would be beneficial to have those included in the test set and know how they should cluster. By uniting these clusters, the truth data becomes a collection of sets, such as {A1,A2,A3,A4}, {B1,B2,B3,B4}, {C1,C2,C3},… where each set contains all mentions of a given entity and no other entity. With these sets, we say that A1 through A4 are separate mentions of person *A*; B1 through B4 are mentions of person *B*; etc. Some sets will have only one item, which means that there is only one mention of those people in the collection.

When a system proposes its cluster results, it will in essence be presenting a potentially different collection of sets than those of the truth data. For example, the hypothesis may have observed the same 11 mentions above, in addition to others from the collection, and clustered them into sets such as {A1,A2,B1}, {A3,C1,C2}, {A4}, {B2,B3,B4}, {C3,UV1,UV2}, … . An evaluation must be able to make a reasonable judgment about these hypotheses which takes into consideration both of the aforementioned issues of precision and recall.

There are various methods for scoring such results, but a frequently-use scoring technique is the B-cubed method (Vilain, et al., 1995). It is simple to describe and implement, and it is our method of choice here. The B-cubed algorithm begins by computing a precision and recall for each mention using intersections and unions between the mention's hypothesized and reference sets. Then, it computes the mean across all mention-wise precisions and recalls to form an average precision and average recall for the collection.

As stated, we will use the B-cubed method for our evaluation of RecordMatch. The mentions that we will use will be those which have been specifically vetted by a human evaluator. So for our purposes, if an entry in a hypothesis set has not been vetted by a human, we will throw it out of the evaluation since we have no evidence of whether the information is true or false.

For the hypothesized sets mentioned several paragraphs ago, let us suppose that the mentions UV1 and UV2 are unvetted (i.e., they have not been analyzed by a human as to whether they are true or false). We would therefore eliminate them from evaluation. Using the B-cubed scoring method for the remaining 11 mentions identified in those hypothesized sets, we would compute a mention-wise precision and recall for each of the individual 11 elements from the truth data and then compute the average thereof to form a final score. If we say that $\boldsymbol{T}a$ is the truth set for element *a*, that $\boldsymbol{H}a$ is the hypothesis for element *a*, and that |.| indicates the number of elements in a set (its *cardinality*), the mention-wise precision is defined to be

$$P = | \boldsymbol{T}a \cap \boldsymbol{H}a | / | \boldsymbol{H}a |$$

and the mention-wise recall is defined to be

$$R = | \boldsymbol{T}a \cap \boldsymbol{H}a | / | \boldsymbol{T}a |.$$

For example, since A1 should have been in the truth set {A1,A2,A3,A4} but was placed improperly in a hypothesis set {A1,A2,B1}, its precision is 2/3 because two of the three elements in its hypothesis set are correct, and it would have a recall of 2/4 because two of the four elements of its truth set are attested to in its hypothesis set. For the hypotheses of the 11 elements mentioned before, we would we obtain the mention-wise precisions and recalls indicated in Table 1.

Table 1: Mention-wise precision/recall information of Example Scenario

Entry	Prec.	Recall	Entry	Prec.	Recall
A1	2/3	2/4	B3	3/3	3/4
A2	2/3	2/4	B4	3/3	3/4
A3	1/3	1/4	C1	2/3	2/3
A4	1/1	1/4	C2	2/3	2/3
B1	1/3	1/4	C3	1/1	1/3
B2	3/3	3/4	(Unvetted/UV: deleted)		

In this example, we would average the precision columns and the recall columns to form final scores of average precision and average recall. In this example, the average precision would have been 25/33 and the average recall would have been 17/33. In synopsis, ***our second task in evaluation is to properly score the hypothesized clusters, and we do this using the B-cubed algorithm.***

3 Creating the Evaluation Set

The process we laid out in Section 2 for establishing an evaluation set required that we begin with a collection of seed entity mentions. From these, we would, by hand, attempt to find all possible variants for those entities while likewise clustering the mentions of other entities that share comparable names and variants with the seeds but which are, in fact, different. Our goal, in this process, is to have an evaluation which is very meaningful and yet can be tractably created. We will describe each of the processes we followed as we sought to establish such an evaluation set.

3.1 The Evaluation Corpus

The Church has annotated or acquired the annotations of several image collections associated with the state of Utah. These collections were presented to us for use in this evaluation, and we acknowledge, with appreciation, the work of those that built and assembled the collections. This collection involved six Utah census collections (1850, 1860, 1870, 1880, 1900, 1920); a collection of Utah marriage records; a collection of Utah death records through 1956; some birth records; a collection of Indian Affidavits from Utah; and military records of Utah Veterans.

In any record from any collection, there may be more than one person mentioned. There is always a principal person for whom the record was specifically created. Yet there could be references to other people in that record, such as through specifications of parental information, spousal information, references to children, or even references to in-laws or grandparents. Each principal is identified by a record identifier, and we create identifiers for related individuals mentioned in the document whose identification is not already specified in the document. For example, if person A has an identifier X and if his or her mother is included in the document without an index, we will refer to her by some identifier such as X-M. The entire collection of records contains about 1.8 million identifiers.

3.2 Identifying Reasonable Seeds

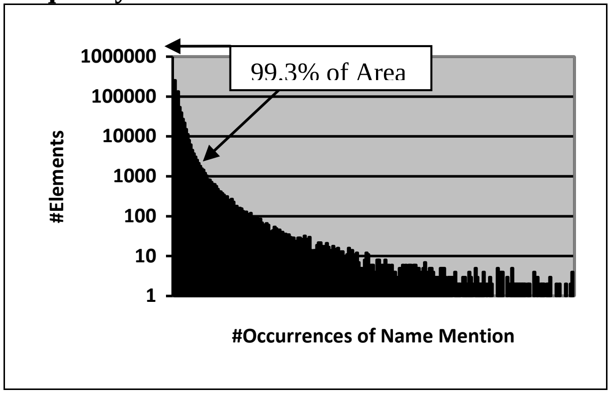
In order to select reasonable seeds, we build a table consisting of every database mention for each unique lower-case version of a name string. For example, the name “john marshall baird” would be a row header in this table, and the row contains three mentions: 334065598-F, 513764213-F, and 233792086-F. The string “john marshall ericson,” also is a row header and its row contains three mentions as well. The row for the name string “john marshall” has over 60 entries, but that row does not include the above indices. That is, that row only contains a list of those identifiers whose full and exact name is “John Marshall.”

We selected about 100 seeds from this table. Appendix A includes the list of the selected seed entries. To prevent an inordinate amount of human vetting, we drew seeds only from rows containing

20 or fewer entries. The vast majority of these seeds were selected at random, though a small number were taken from known family members of evaluators and co-workers. Only a single mention was used as a seed for any particular name string, and that mention was largely drawn at random from among the set of identifiers of people having that name mentioned.

One may wonder about the legitimacy of selecting seeds that do not span the whole space of occurrence frequencies. As will be mentioned in a moment, this evaluation did not focus solely on the seeds, but also on potential variants for that seed. It should be noted that a seed that has relatively few occurrences can in fact give rise to or require the processing of a highly frequent name variant (such as "Ellen Louise Miller" giving rise to the variant "Ellen Miller") or may be a woman whose maiden or married name is quite common (such as "Miss Alice Strauss" giving rise to "Alice Allen"). In addition, Chart 1 illustrates the occurrences of given name strings (x-axis) and the number of name strings that have that occurrence (y-axis). Only 0.7% of name strings have occurrences of greater than 20 in this collection.

Chart 1: Count of distinct names with a given frequency of occurrence



3.3 Finding Variants

Establishment of variants was done in two phases. During the first phase, the evaluator created an information retrieval index where each "document" in the index was a unique name string from the collection, and the contents of that document would be the individual name components of that name and the overlapping trigrams of the name. Using the example from above of "john marshall baird", the index would contain a document named "john marshall baird" whose contents were "john marshall baird joh ohn hn_ n_m _ma mar ars rsh …." All unique names were converted in the same fashion and indexed using Solr (see Apache Solr, 2010).

Suppose, then, that the evaluation had a seed named "john marshall ericson." The evaluator would find potential variants of this name using the Solr index. This name would be converted into a collection of name components and overlapping trigrams as mentioned above, and that set of tokens would be used to form a query to the Solr index. Solr would return a list of unique name mentions which the greatest word-and-trigram commonality with the input string. Aa actual run of this example through the Solr index yields a set of potential name variants: john marshall ericson, marshall y. ericsson, marshall g. ericson, marshal ericson, louis marshall ericson, marshall graham ericson, john marshall, john ericson, john marshall egan, john m. ericson, john m ericson, …. The evaluator would analyze (by hand) each of the top 100 to 150 returned name strings and would assess if any of these could potentially be name variants for the seed. This process was reasonably successful at finding variants, particularly for seeds that mentioned men, though it often had to be repeated when the seed was determined to be a woman and her maiden or married name was discovered which was not in the original seed mention.

Hindsight suggests that also including acronyms and nicknames into the index may have helped increase the mean average precision of this top-150 list. For example, "Henry" was often abbreviated as "Hy" in Utah during the time frame 1850-1950. "Hy" and "Henry" do not share either name parts or overlapping trigrams, so it is quite possible that one might miss Henry while looking for Hy. Using the Church's name authorities could have also been beneficial in this process. Even so, the trigrams helped to surface a number of spelling errors and other name variants which may not have appeared from a word-based search and would not have been considered to be typical name variants that one might look up in a list.

3.4 Vetting Seeds and Variants

Once the evaluator identified the potential legal variants, a separate process distilled out the legal variant rows from the table described in Section

3.2 in order to produce the list of all mentions that could be related to the seed entry. For each element of the list, the process obtained the original record containing that name mention and it would re-focus the information from that record on the individual of interest. That is, if the principal person in the record was X, but the seed variant of interest was the mother, X-M, the system would convert the record about X into a record that looked to be about X-M. So if X's birth date was included in the original record, it would now be listed as child birth information in X-M's artificial record and a rule-based approximation would be made for the birth year of X-M. If X's father was included in the record about X, it would now be listed as X-M's husband. As much information as possible from the X record would be distilled into the artificial X-M record. Then, all records from all variant mentions grown from the seed mention would be compounded into the same table.

The table of information was then carefully reviewed by hand. Any entry in the table that looked to have the same information as the seed was then labeled with "Y" (for "yes.") Any variants that were determined to be unrelated to the seed were then clustered. The first non-related variant was identified as "N1" ("not related"/ first different unique entity). The next non-related variant was either associated with N1 or was named a new entity, N2. When the information in the table was insufficient to make a full determination, the evaluator used the LDS Church's genealogical sites new.familysearch and beta.familysearch.org to search for the potential of already-compiled information or raw documents about the individual as auxiliary information.

By the completion of this vetting process, over 3300[1] mentions had been considered – all based upon the 100 initial seeds. With only three exceptions, each *seed* actually belonged in a set with other mentions. However, given that there was no special effort to discover all variants and mentions of people whose name looked like the seed person's but were not (i.e., the people that formed the various N-sub-i clusters), there were many such collateral mentions that fell into singleton clusters.

3.5 After-the-fact Extensions/ Phase Two

The first phase of evaluation was done with complete autonomy from the developers producing hypothesis clusters, so the automatic systems were in no way optimized for the evaluation. However, a second phase of evaluation used system output for the purpose of better ensuring completeness. If the automated system proposed connections between a seed and some mention which was not contained in the first-phase truth table, the additional information would be evaluated and included in the phase-two truth table whether true or false. Interestingly, despite significant human efforts to be thorough and complete in phase one as described in the previous sections, there were still 102 instances proposed by automation as being associated with seed instances but which were not discovered in any of the aforementioned processes. These errors were largely due either to (a) the human's inability to find the married or maiden name for a seed female mention though the system did so successfully, or (b) a legitimate variant was found but it happened to be a frequently-occurring name string making it quite hard for the human to vet.

Lastly, to help ensure integrity of the results, the primary evaluator and system designers reviewed and agreed upon any clusters, missing mentions, or falsely-included information for which there was initial disagreement. Through this extensive process, the final truth set was determined to be clean and likely to have few if any remaining errors. Those that may exist are unlikely to be of significant consequence in the overall evaluation.

4 Evaluation Applied to GenMergeDB

The final step in this evaluation is to apply the processes and truth tables to the output of an existing system. As mentioned previously, a commercial tool called GenMergeDB was applied to the Utah test collection for the purposes of establishing clusters between *all* mentions in the collection. GenMergeDB was built to help individuals and organizations to discover potential links in genealogical data. The software designers had no notion at all of how this evaluation would proceed nor of

[1] 1100 of the 3300 entries were overgenerated indices that were created in order to provide the evaluator with better information about family structure and were not used in the end scoring process. These extra indices were family-type items such as X-M for a mother identified in her child's record even when her official identifier was also included in that record. Such overgeneration provided for a one-to-one correspondence between IDs and name mentions.

which seeds would be selected prior to providing their results to the evaluator. We illustrate the performance of this system below.

4.1 Comparison Systems/ Shatter-All

It is also beneficial to have the output of other algorithms to compare results to. At the current time, efforts are under way to test the potential of applying the LDS Church's existing record-linkage software to perform the RecordMatch task (see, for example, Wilson and Quass, 2008), but the application of those tools has not been completed. Therefore, we use as the Shatter-All baseline (Poessio, et al., 2007) as a comparison. The Shatter-All baseline is a one where every mention gets placed into a cluster of its own (which is essentially the Church's current status in this area). One would hope that a disambiguation system can improve beyond just doing nothing, but there have been demonstrated occurrences in the literature where the results from algorithms have been worse that having done nothing.

4.2 Score Type 1: Using only Truth Set Y's

We evaluate in two parts. Since the construction of this truth set placed heavy emphasis on seeds, it seems appropriate to have at least one component of the evaluation that focuses also on seeds. Therefore, in this first set of scores, we compute the B-cubed scores of algorithms by only averaging across those elements that are co-referent to the seed entities. Suppose A1 is a seed mention of entity A, that A2- A4 are other mentions of A, and that B1 shares a name with A1 but actually is a different entity, B. If a hypothesis set were {A1,A2,B1}{A3,A4}, we factor in the B1 when we are trying to compute the mention-wise precisions of A1 and A2, but we will not factor its own cluster into the overall score during this first wave of scoring. We apply the B-cubed algorithm to generate the scores for both the GenMergeDB output and the Shatter-All baseline (Tables 2A and 2B). Additionally, we show the result when scoring on GenMergeDB output for only those sets which they claimed to be non-singleton sets and whose entries are in the truth sets (Tables 2C and 2D). In Tables 2A-2D, we show performance in terms of B-cubed average precision (AverageP) and average recall (AverageR) as well as the number of clusters proposed by the system and the number of mentions whose sets contributed to the score.

Table 2A: Score of GenMergeDB Output Averaged across Truth Table Y's Only

Average P	Average R	#Clusters	#Mentions
0.989	0.690	201	712

Table 2B: Score of Shatter-All Baseline Averaged across Truth Table Y's Only

Average P	Average R	#Clusters	#Mentions
1.000	0.143	712	712

Table 2C: Score of GenMergeDB Output Averaged across Truth Table Y's Only (no GM Singletons)

Average P	Average R	#Clusters	#Mentions
0.987	0.800	122	633

Table 2D: Score of Shatter-All Baseline Averaged across Truth Table Y's Only (no GM Singletons)

Average P	Average R	#Clusters	#Mentions
1.000	0.134	633	633

From these results, it seems clear that the GenMergeDB software is providing a huge benefit over the Shatter-All situation (which Shatter-All emulates the current status in Church-provided results).

4.3 Score Type 2: Using Whole Truth Set

In a second type of scoring, we now factor in ALL mentions from the truth set of data. Recall that in the first type of scoring, we would not have averaged B1's mention-wise precision and recall into the overall averages. Tables 3A-3D use the same headings as Tables 2A-2D and show the evaluation information with significantly more mentions.

Table 3A: Score of GenMergeDB Output Averaged across All Truth Table Vetted Information

Average P	Average R	#Clusters	#Mentions
0.980	0.806	1049	2197

Table 3B: Score of Shatter-All Baseline Averaged across All Truth Table Vetted Information

Average P	Average R	#Clusters	#Mentions
1.000	0.382	2197	2197

Table 3C: Score of GenMergeDB Output Averaged across All Truth on Non-Singleton GM Clusters

Average P	Average R	#Clusters	#Mentions
0.975	0.870	561	1709

Table 3D: Score of Shatter-All Baseline Averaged across All Truth on Elements from Table 3C

Average P	Average R	#Clusters	#Mentions
1.000	0.295	1709	1709

These new tables still show that GenMergeDB is highly precise even across a much wider range of data, though it is relevant to note that the Shatter-All performance increased here due to the shallower treatment of non-seed entities. It seems fairly clear from these results, though, that patrons who were to use RecordMatch based on the GenMergeDB process would be presented with a fairly comprehensive collection of original records pertaining to the ancestors they seek, and the precision from such clusters will be high.

Acknowledgments

The author would like to thank Chris Cummings for vetting several of the names of the evaluation corpus, and for helping to provide background on the various image collections. The authors would also like to Randy Wilson, Scott Smith, David Barss, and Ray Madsen for providing additional necessary resources or information for this evaluation. The author would especially like to thank Sue Dintelman and Tim Maness of Pleiades-Software Development, Inc. for providing evaluators with the results of their GenMergeDB process as applied to the Utah Collection, and for the many hours they spent to verify the validity of the truth tables used in this evaluation.

References

Apache Solr (2010). http://lucene.apache.org/solr

Javier Artiles, Satoshi Sekine, Julio Gonzalo (2008). Web people search: results of the first evaluation and the plan for the second. *Proceedings of the 17th Int'l World Wide Web Conference*, Beijing, China.

Paul McNamee, Hoa T Dang, Heather Simpson, Patrick Schone, Stephanie M. Strassel. (2010). An Evaluation of Technologies for Knowledge Base Population. *Proceedings of LREC-2010*, Malta, pp. 369-372.

Maximo Poessio, David Day, Ron Arstein, Jason Duncan, Valdimir Eidelman, Claudio Giuliano, Rob Hall, Janet Hitzeman, Alan Jern, Mijail Kabadjov, Gideon Mann, Paul McNamee, Allessandro Moschitti, Simone Pnzetto, Jason Smith, Josef Steinberger, Michael Strube, Jian Su, Yannick Versley, Xiofeng Yang, Michael Wick. Exploiting Encyclopedic and Lexical Resources for Entity Disambiguation. Johns Hopkins Summer Workshop, Tech'l Report, 2007.

Mark Pryzbocky (2008). Automatic Content Extraction 2008 Evaluation Plan. National Institute of Standards and Technology, Gaithersburg MD. http://www.itl.nist.gov/iad/mig//tests/ace/2008/doc/ace08-evalplan.v1.2d.pdf.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman (1995). A model-theoretic coreference scoring scheme. *MUC6*. 45--52. Morgan Kaufmann.

Randy Wilson, Dallan Quass (2008). Beyond Probabilistic Record Linkage Using Neural Networks to Improve Genealogical Record Linkage. Submission to *AAAI-2008.*

APPENDIX A: List of Seed Entities

Seed Identifier	Seed Name Mention
55512093M	Lizzie/Watson
55552047M	Birdie/Price
60729376M	Jennie/Lund
77916025	Robert Cross Osmond
77991813F	Daniel C./Ressler
78001600M	Arminda/Thompkins
78007227F	M. K./Parsons
78037395S	Max/Schmidt
79625298S	Elvina/Holt
79625579	Nellie Hogenson
79626074S	Mary/Siebert
79630447M	Sarah/Haines
79631440	Infant Sharp
195103041S	Miss Alice/Strauss
233756158F	Petter M. Fife
233764135F	Henry Moyle
233794975F	Salud Orozco
233845235	Mabel Ella Lewis Niver
233847218	Ezra Taft Hatch
233854483	Zilpha Wood Urie
240370277F	David O Mckay
240420547	Katherine Riley Wilcox
241406186S	Annie Mary/Dunford Munson
254148595S	Lizzie/Bocklund
287742810S	Estella/Ahlstrom
294667486S	Marion Ethel/Johansen
294667806S	Ellen Louise/Miller
295420636S	Mary Ellen/Wardle
295439796	Washington Lemmon
296632298	Uno Peterson
296701506SM	Jean W./Purdie
296704904SF	Henry/Carsey
296778257	Phillip M. Kellogg
296842949SM	Minnie/Donner
296892214F	Bent Hansen
297043996S	Roxey/Nickerson
297102744SM	Elizabeth/Cram
297122359S	Nellie M./Evans
298102675SF	John/Groberg
298103168SF	Peter/Larsen Jr
298214322	Chas Vernon Palmer
298545125SM	Mary/Burningham
298715963	James C. Jenson
298861929	Fred Scott
333976982F	Edw. D./Woolley
333981050M	Frances/Vance
333983302F	William G./Crawford
333987745M	Jennie/Campbell
333995294	John Christian Sandberg
333998594	Gottfried Schoene
334052912S	Ellen Nora/Julian Shields
334054917	Mary Margo Brissell
334063768F	Gus/Kitsopoulos
334063781S	Blanche/Arnold
334064327S	Maggie/Shelledy
334064685S	Chloe Young/Baxter
334065114S	Nicoline/Sorensen
334068334M	Ethel/Adams
334204642	Rose Hannah Lester Lewis
334472595S	Laura/Calegory
334475218S	Albert/Galloway
334476915S	Maude/Potter Meeks
338229823	Matilda Jeffs
351866858	Rufus Call Willey
438470940S	Rula/Broadbent
438471345S	Signora/Powell
464644936	Alace D Wilson
464681939S	Eliza Weston
466467796F	Phillip Paskett
466486142M	Jerusha Maughan
466576726	Bertel Johnson
466582410	Jacob Dentin
466582452	Jno Kerkman
466614851	Flowers Whorton
487547776	Russell Stevenson
518657862	Miles Skenzich
638862336	Ida Wooley
652090921	Anna M Lunds
652102718	Thomas Chipman
652111307	Edwin Hy Hooker
684645939	Ann E Cummings
684670675	W J Wright
848075501	James L Carter
848160043	Willim H Gagon
848257302	Fred Williams Hyke
849078795S	Maren Christina Elton
849089696	Alma W Weed
849498281S	Hellen Preggastis
849504996	Rose Paoletti
849521052M	Francis C Peacock
849543879S	Blanch H Ahern
849556398	Susie Mary Schettler
849564522	Alice L Phillips
849594069	Cleo C Beckstead
852027374	Boon Monson
852039310	Hannah C Lawson
1000153141060F	Lemuel T. Steele
1000153149629S	Alice Leigh
1000153203695M	Anne Humble
1000153217503	Mary O. Sullivan
1000408592286	Kathryn Edna Thurgood