Comprehensive Evaluation of Name Matching Across Historic and Linguistic Boundaries

Patrick Schone, Chris Cummings, Stuart Davey, Michael Jones, Barry Nay, Mark Ward
FamilySearch, 50 E North Temple, Salt Lake City, UT
Patrickjohn.Schone@ldschurch.org, {CummingsCH, DaveySE, JonesMP, NayRB, WardRM}@familysearch.org

ABSTRACT

Personal names are the most critical elements for discovery and compilation of one's heritage. However, historical and multilingual names are subject to many alterations and variations which make searching and matching of such names a great challenge. Consequently, we have created a name matching corpus and evaluation which capture most of these variations and provide a means whereby different name matching systems can be compared as to their effectiveness on matching these historical and cross-lingual names. It is our plan to make this corpus and evaluation available to the public in order to provide a means for wide-scale improvements in historical name matching. We here describe the formation of the corpus, the evaluation methodology and metrics. Lastly we show the performance of a number of a name matching systems and identify potential directions for future enhancements.

1 BACKGROUND

When a genealogist searches for an ancestor, they will almost always start the search using the ancestor's name. It is therefore critical for genealogical search engines to be able to properly identify appropriate responses to personal name queries. Unfortunately, when personal name searches are made against databases of historical or multilingual records, it can be very difficult to provide appropriate responses. In particular, historical and multilingual personal name collections are replete with misspellings, sound-alikes, transliterations, nicknames, translations, initialisms, short-hand representations, partial-name representations, and best guesses. Names also change due to certain life events, such as marriage, immigration, military enlistments, and name popularity. Worse still, when modern-day annotators transcribe those historical documents, they are frequently confused by handwriting styles, image clarity, over-inking, word overlap, language unfamiliarity, image aging, and limited understanding of the names in use at the place and time.

Clearly, these sources of name variation represent serious difficulty for genealogical search processes. These issues also add to difficulty for a genealogists who has interest in learning if their ancestor, John Smythe, could potentially be the same person as one represented by a name like "Jonathan Smyth," or "George Smyth Jones."

A method for helping to overcome these difficulties is to provide resources which appropriately group similar names. These resources can be computational in nature, they can be human-derived, or they can be a hybrid between the two. Computational name-association resources might make judgments by computing statistics of name linkages, analyzing character edit distances, overlapping *n*-grams, ethnicity estimations, and so forth. Human-provided resources include knowledge bases (KBs) which store human-attested name variants. Hybrid resources can augment KBs with computational estimations when information is sparse or unclear. Any of these kinds of processes for identifying legitimate potential name variants are usually referred to in the literature as *name matching* processes. (Note that in many genealogical circles, "matching" may refer to entity disambiguation or linking, but "name matching" for the purposes of this paper follows the meaning from the more generic literature of bringing together reasonable name variants.)

Over the past approximately 40 years, the Church of Jesus Christ of Latter-day Saints has amassed huge KBs of name variants that were designed to support name matching across the centuries and across many geographical boundaries. Name matching algorithms derived from these KBs have been created and incorporated into various genealogical search products. However, in recent years, various competing approaches have arisen for converting these KBs into name matching tools. At the same time, the research community has developed new techniques for name matching which could be superior to using any of the KBs as they currently exist.

These changes suggest the need for a name matching evaluation for genealogical purposes – an evaluation that would allow for an understanding of what KB components, algorithms, and parameter settings provide optimal response to personal name queries. Analysis of full name conflation has been done by research organizations, such as at MITRE, for almost a decade [1].

For genealogical purposes, evaluation of full name conflation is insufficient. The evaluation must also consider partial names; the degree to which names conflate based on the era or the culture in which the name appeared; and name variation due to faulty representations and/or mis-transcriptions. As examples of historical variations, the strings "Catherine" and "Katherine" are known to be variants of each other, but "Kate" is a variant of more recent advent, and "Kitty," which was quite popular in the 1800s and early 1900s, is almost never used today. Likewise, linguistically, the Scandinavian "Cajsa" is an

alternative for "Catherine," which is almost unknown to English-speakers. "Sara" and "Lara" are examples of frequently mis-transcribed name pairs that might get overlooked by humans. Syntax issues also arise based on when and where the name was stored. For example, in a US census, a personal name may appear as "Arnold John" – does this mean the given name is "Arnold" or the family name is "Arnold" or the two names form a two-piece given name? All of these and more are issues for genealogical name matching.

We have created a name matching evaluation which incorporates issues of full and partial name renderings, temporal aspects, cultural boundaries, multilingual issues, storage, and transcription issues. Through the mining of data from our Common Pedigree (over 900M records, viewable at new.familysearch.org), transcriptions of our historical documents (about 2B records), and auto-identified people from Wikipedia (including titles, redirects, aliases, and appropriate cross-lingual links), we have created a database of 10M people. The database also includes *context*: of a gender, a place, and/or a date associated with most of the individuals.

We have also developed a related query set containing 10,000 people-plus-context. For each query person, a name matching system must attempt to find every name from the 10M-person corpus that could be perceived of as a reasonable name variant for the query. Systems can ignore the provided context or can use the context either at query time or indexing time, but not both (in order to preserve this task as a name matching evaluation as opposed to a full genealogical person search).

We evaluate systems in terms of both mean average precision and F-score. To obtain a gold standard for this evaluation, we began with one or more known attested pairs for almost all queries (eg., an observed instance of a "John Smith" being equated to a "J. William Smith" in the pedigree, Wikipedia redirects/aliases/etc., or in multiple transcriptions of a single historical record). Then, we created an expanded truth set by having a single human vet (either by-hand or through a form of rudimentary automation) the top 20 name variants that any given name matching system proposed for any of the 10,000 queries. The "attested-variant truth set" consists of over 15000 entries, and the "human-vetted truth set" consists of 21.1M entries of which 1.7M are relevant. We also can use the attested variant truth set to score the human's ability to find valid name pairs. We will show results of this matching on multiple different systems.

It is our plan to release these data sets and evaluation tools to the world for rapidly improving historical name matching. We therefore describe in detail the composition of the datasets and provide system descriptions and performance as we have seen thus far.

2. NAME OBJECT DATABASE CREATION

We patterned our name-matching evaluation after a typical adhoc information retrieval (IR) evaluation such as TREC. In adhoc IR tasks, the system developers are given a large collection of documents which they are allowed to process in whatever fashion they choose as long as the processing is completed prior to running any queries. After initial processing, each system is given a set of queries which have not been used to train the system. The name matching results are evaluated to determine how responsive each system is to the information needs of the query. Responsiveness, or relevance, is usually determined by the person who issued the query. The top N results from each system are pooled together and tagged for responsiveness. This pooled "gold standard" provides a means for scoring systems not only for how precise they were at addressing user needs, but also, how well systems recalled all of the potentially-relevant documents.

In order to follow this same procedure, we needed to create "documents" based on names. These documents had to have properties of the kind that we would see in genealogical searches. The collection of these documents also had to be large enough to make it impractical for any system to memorize all potential query-answer responses, but not so large as to make evaluation impossible. We here describe each of the components and contributing features of our "10M Name Object Database."

2.1. Description of Name Objects

For the "documents" of our name matching corpus, we must obviously include people names. However, there are additional features that are provided with genealogical records that can provide more clues about how a name should be handled rather than just the name alone.

For examples, consider variants for the following:

* "George Martin," no additional context
* "George Martin," a FEMALE
* "George Martin," a person living in MEXICO
* "George Martin," a person living in the 1600s

In the first, one would assume that the person is a male whose family name is "Martin" and whose given name is "George." Potentially, one might assume that his name could also be found as "Georg Martin," "Geo. Martin," etc.

In the second case, knowing that the information represents a female, it would now no longer be appropriate to return "Georg" as a name variant for the given name. Instead, one might expect "Georgia" or "Georgeann."

In the third situation, one might know that "Martin" is probably "Martín" or perhaps "Martínez." The given name variant "Jorge" is now much more likely than "Georg" or even "Geo."

It is not necessarily clear what the temporal aspect provides in the fourth situation. Perhaps "Georgious" or some Latin name may be a more likely variant – certainly more so that "Geo." It can also be in the historical case that "Martin" is more likely to be another given name and that there is no family name associated with the individual.

To allow systems to use context if desired, we opted to create *name objects* which could provide both personal

name strings AND the additional context. These name objects are represented in the database and in the query set. Name-matching systems could use the context when they indexed the data prior to any search being performed, or they could incorporate context at query time, but as mentioned before, they could not do both or else the system would be doing a full genealogical query as opposed to just doing name matching.

Our name objects augment the name with, potentially, a place where the individual lived, a date from their life span, and/or their gender. The string "<empty>" is used when any of the context fields were missing. In many cases, we round dates to the nearest decade and places to two administrative levels (such as State, County).

Another component of each name object is metadata that may be provided by a genealogy patron. Since we do not know the level of expertise of the patron, this metadata could be very suspect. Even so, it is often the case that family names will be specifically marked by the patron. Thus, a name may appear as "George /Martin/," which suggests that "Martin" is indeed a family name – or is, at least as far as the patron was aware.

The following are examples of objects that may be stored in the database:

Francisco /Borbon/;Chihuahua;1920s;M

*vey /Cox/; Georgia;1860s;F

Stanley /Piotrowski/;Russia;1880s;M

Clara /Chambers/;Ohio;1860s;F

Lucila /Ocampo Campos/;Pateo, Michoacan, México;1853;F

Orvis /Jones/;Of Vermont;1821;M

/龔/鳳周夫人;中國湖南省;1469;F

Albert Edward /Cowan/;Nbru, Newbrunswick;1850s;M

2.2. 10-Million Name Object Database

In development of the database, we reason that a corpus with 10 million entries is very large – too large to allow for memorization. Yet if reasonable queries are created, it should be feasible to evaluate them even when applied to a database of this size.

We created a heterogeneous database designed to consider not only names in existing genealogical repositories, but also Wikipedia – which gives a great sampling of across-the-world names with cross-lingual components. We describe here how each of these collections contributes to our 10M-entry database.

2.2.1. Common Pedigree Scraping

For the past several years, the LDS Church has worked to create the largest family tree ever created. The Church's hope is that it will eventually be able to link together the entire human family into this tree. In 2012, this tree will be made available for use by the entire world, but up until recently, this tree has largely been created through contributions from LDS Church members.

This tree, often called the *Common Pedigree* or *new.familysearch.org*, contains about 900 million mostly-linked, multi-lingual person-instances. The word "instance" here is used to mean that Genealogist A may have submitted "George Martin" as an entry into the tree with a specified father and mother; Genealogist B may have also submitted him with the same facts; and Genealogist C may have submitted him as "Geo. Martin" with information about his spouse rather than his parents.

Because common pedigree contains linked records with multiple variant contributions, it provides a ready-source of names and contexts that can be used for genealogical and cross-lingual name matching. We drew five million entries fairly randomly from the common pedigree (though there was some exception to the randomness in trying to ensure enough cross-lingual components to the evaluation).

2.2.2. A-B-Arbitrate and Historical Records

Another very large genealogical corpus is found at *FamilySearch.org*. This corpus is a collection of historical records, such as vital and census records.

When these records are transcribed, the LDS Church follows a three-transcriber process. The first two transcription volunteers, call them A and B, are given the task of independently transcribing all of the content from the historical images that they are presented with. A third transcriber, called ARB, is given the task of looking at the outputs from A and B and determining which is right, or providing additional edits.

As was hinted at earlier, historical record analysis gives rise to a lot of the name-matching problems for genealogists and machines. Thus, this corpus is ideal for representing common errors that need to be contended with in the name matching process. We drew 4.217 million entries from this corpus as contributions to our DB.

2.2.3. Utah-Specific Corpora

Since many of the LDS Church's genealogical patrons are located in the US state of Utah, there has been a heavy push to acquire as many possible Utah record collections as possible. Consequently, the Church has acquired over a dozen such collections representing about two million name instances from the mid-1850s to the mid-1950s. For reasons that will be more clear momentarily, we selected a small set of 12.8K of these records for incorporation into our 10M corpus.

2.2.4. Wikipedia Mined Names

Research has illustrated how one can use Wikipedia as a name-entity-tagged training corpus for creating entity taggers [2]. We replicated the first stages of this process and categorized all of the English Wikipedia page titles. Approximately 1M names of Wikipedia are devoted to different people, and we used a subset of these in our collection. In addition, we collected Wikipedia redirects and cross-lingual links that relate to people pages which will be described in a moment and used a subset of those. The subset links will be described in the next section.

Since Wikipedia data is not the same as genealogical data, we categorized Wikipedia infoboxes and attempted to determine which fields could be useful for providing the

desired genealogical contexts of place and date. We also did an analysis of the pronouns used in the document in order to estimate the gender. This estimation proved to provide reasonably reliable gender tags.

Using these various processes, we were able to augment our name object corpus with 755K name objects from Wikipedia. The use of the redirects and cross-lingual links will be described in Section 3.

3. QUERY SET AND MUST-FINDS

After having constructed the name object database (which from this point will be referred to as NODB), the next component of the evaluation was to create the query set. For our evaluation, we wanted again to have a query set that would be large enough to allow for long-term reuse of the queries. We opted therefore to create an evaluation set of 20,000 queries. After some initial tests, we found that the 20,000 queries proved to be unwieldy for systems and evaluators, so we later reduced the set to 10,000.

In creating the query set, we drew from the same collections over which we built the NODB. However, we wanted to be able to create the set in such a way as to avoid a serious, common problem with name-matching evaluations; namely, how does one know which matches are right? Name matching evaluations are typically subjective and though this evaluation must also contain an aspect of subjectivity, we wanted to provide a component of the evaluation that was based on indisputable facts to ensure the integrity of the evaluation.

3.1 Attested Pairs in Evaluation

To provide this integrity, we created our query set by ensuring that most queries have at least one attested pair. For example, if the data sources from which we drew our queries have multiple variants that allow us to know that, say, "Fred /Smith/;1890s;Utah;M" is the same as "Fredrick Smith;1900s;Idaho;M," then we can add "Fred /Smith/;1890s;Utah;M" to the query set and either add "Fred /Smith/;1890s;Utah;M" or "Frederick Smith;1900s;Idaho;M" to the KB and know that it is an attested match for the "Fred /Smith/;1890s;Utah;M" query.

When a system queries for "Fred /Smith/," it may return answers like "Frederick /Smith/;1940s;Wisconsin;M" "Frederick /Smith/;1860s;Utah;M" and "Fred J /Smith/;<empty>;<empty>;<empty>" – all which may be valid and will have to be subjectively marked by hand. Yet if the system fails to find "Fredrick Smith; 1900s;Idaho;M" then we will know that it did not find all the results that it should have; and if it did find the particular attested "Fredrick Smith;1900s;Idaho;M" then we will know that at least that pairing is correct.

3.2. Discovering Attested Pairs

We wanted to ensure that we had attested pairs for almost the entire name matching query set. Note that this is somewhat of an artificial constraint because it can be the case that a name query should return NO answers, but, as stated, we wanted to ensure that there was some non-subjective foundation to the evaluation.

Our goal in query creation, then, was to identify attested pairs, use these for queries, and, as necessary, augment the NODB to ensure it contains at least one member of each pair. This process added 13.9K more entries to the NODB and made the NODB a perfect 10M-name-object set. The following sections indicate how we use the existing corpora to identify attested pairs.

3.2.1. Mining Attested Pairs from Common Pedigree

Common pedigree was described previously as a mostly-linked massive family tree. Part of the linking in the common pedigree is a stitching of families together. Yet other links are of particular interest for finding attested pairs. In particular, many of the links in common pedigree are used to stitch together multiple instances of the same individual. In our example of Section 2.2.1, "George Martin" was linked to "Geo. Martin." These cross-instance linkages provide a perfect source for finding attested pairs and we use many of the common pedigree pairings in our queries.

3.2.2. Attested Pairs in A-B-Arb and Utah Corpora

The LDS Church's historical record collection is not currently disambiguated as a whole. However, the previously-mentioned Utah corpus has been disambiguated automatically. Through a 2010 evaluation (see [3]), it was determined that this automatic process combines records with an accuracy of about 99%. Consequently, we draw some of our attested pairs from this Utah collection.

The A-B-Arbitrate data also provides a nice source of "attested pairs." Independent annotators mis-transcribe the names they see in approximately 10-20% of the cases (though arbitration helps to significantly improve these numbers). These variations can serve as attested pairs because they reflect the kinds of errors that really may occur in the transcription of images as well as the errors that patrons may have in trying to distill information from their own genealogical studies.

3.2.3 Wikipairs

The last source of variation is derived from Wikipedia. These variations come from first paragraph name re-statement, cross-lingual links, and page redirections.

First paragraph restatement: The first paragraph of many Wiki pages often starts by re-stating the name of the person that the page is about. Sometimes, these restatements actually include a name variant rather than a copy of the original title. Additionally, there may be a parenthetical representation in the first paragraph which provides a non-English representation of the person's name.

Cross-lingual Links: Many Wiki pages have cross-lingual links to pages from other languages that are about the same subject – these cross-lingual links provide additional name variants. However, to ensure that the name matching evaluation makes practical sense, we only select from among those multilingual name variants where there is the possibility that a historical record could have been created. That is, if a person is from China, it is reasonable to believe

that his or her name could also be observed in some document in Chinese or in a Chinese transliteration.

Redirects: Wikipedia redirects allow a Wikipedia searcher to be looking for a Wiki page using one title and be directed to the (hopefully) proper Wiki page having a different title. Some of these redirects are the result of spelling errors, different presentation of the name, different parsing of the name, and so forth.

Consider the page "Andrey Kolmogorov." At the time of the creation of this paper, the first paragraph of that Wiki page starts with a bold-faced "Andrey Nikolaevich Kolmogorov." The page also includes a parenthetical representation of the name in Russian, namely "Андре́й Никола́евич Колмого́ров." Both of these name variants can serve as attested pairs. Also, the Wiki page indicates that Andrey Kolmogorov is from Russia, so if we look at the Russian cross-lingual link, it indicates that another variant is "Колмогоров, Андрей Николаевич." Lastly, the redirect of "Andrei Kolmogorov" takes the user to this page, which suggests that this also is an attested pair.

3.3. On Buck Danner and Marilyn Monroe

One last comment needs to be made about the name matching query collection. There are individuals such as "Marilyn Monroe" whose real name is "Norma Jeane Mortensen" or "Norma Jeane Baker." Although one would expect an entity disambiguation system to determine that these are the same individual, such a determination would not be made by name matching but through matching of other facts. Consequently, we worked to eliminate the majority of these kinds of pairings from the query set.

On the other hand, there are individuals who use generic nicknames as did Henry Frederick Danner, who went by "Buck Danner." We believe that a name matching system should reasonably be able to identify people by generic nicknames, and we have retained such attested pairs.

4. SYSTEMS IN INITIAL COMPARISONS

The purpose of this paper is to describe the evaluation that was created for historical and cross-lingual name matching for genealogical purposes. However, a goal is that this evaluation will foster the development of other name matching systems which can take advantage of these resources. It seems reasonable, then, to provide a brief summary of the kinds of algorithms that have been used thus far in this evaluation so that other researchers can know to some degree what has and has not been tried.

4.1. Knowledge Base Systems

One of the main system types that are tested in this paper are based on knowledge bases of names. We will describe these systems briefly.

4.1.1. Structure of Knowledge Bases

One of the key ingredients of several name matching systems tried in this paper is a knowledge base of personal names variations. It was mentioned previously that the LDS Church has been working over a 40-year period to assemble names that conflate with each other. The LDS Church has also developed software to "serve up" these name variations either for patron or computer use (see https://labs.familysearch.org/stdfinder/). We will refer to these name variation tools and knowledge bases as "S&A's Name Standards."

In recent years, the knowledge base has also been used by others to create derivative name standards. One competing standard that is in use within FamilySearch is the "DRS Name Standard." Several key distinctions between these are (1) the S&A has a larger knowledge base; (2) the S&A accommodates culture-based conflations of names; and (3) the S&A version strictly computes variations that are identified in the knowledge base whereas the DRS system will compute variations "on the fly" when they are not observed in the knowledge base.

Another name standard, which we will here call "FOL" has also recently become available. Though it was not derived from S&A knowledge bases per se, it is similar in nature and also has static name groupings.

4.1.2. Searching for Names Using KBs

For the name matching experiments to proceed, the knowledge bases must be inserted into a name query system. The personal name knowledge bases are built on understandings about individual name tokens, not on whole names. Therefore, to create the required name matching systems, the system builders applied software to each name of the NODB to: (1) tokenize each name into name pieces; (2) determine where name pieces refer to parts of surnames, given names, either, or neither; (3) identify the appropriate conflation set for each surname or given name piece; and (4) index all name pieces and name conflation sets in a search engine.

Given the query name, each particular system would repeat each of the four steps just mentioned. Suppose that C(X) represents the numeric identifier for conflation set of X and that Wt(X) represents a weight that will be applied to a system for finding word "X." If the query name string were "X Y," this could conceivably be converted into an expanded query {Wt(X)*X OR Wt(C(X))*C(X)} AND {Wt(Y)*Y OR Wt(C(Y))*C(Y)}.

4.1.3. Parameterization of KB Systems

There are some principal differences between the various knowledge-based name-matching systems. These fall into five kinds of variations:

(a) KB Choice: Which knowledge base is used (S&A, DRS, FOL, none)? If "none", then one can only match on the exact name piece.

(b) Clustering/Culture: If DRS, is auto-clustering used? If S&A, does one use cultural conflations? Is this feature off?

(c) Initials: If initials are not in name clusters (which they typically are not), should they be expanded to allow any name that starts with the same letter?

(d) Missing Data: If a query is for X Y /Z/, is "/Z/" by itself an acceptable answer?

(e) Weight Set: What are the weights that are used to score the results? For the purposes of this paper, we will say that there are two weight sets: W1 and W2.

We will refer to particular database-driven name matching systems by naming the parameters they use. A system involving the DRS system that uses DRS-clustering, expands initials, uses weight set one, but does not allow for missing data is "DRS,+CLUS,+INIT,-NOMISS,W1."

4.2. Alternative Systems

Knowledge bases are very helpful in name matching for ensuring that name searches bring together names that are the same but do not look alike, such as "Peggy" and "Margaret;" and for divorcing names that look or sound alike but are unrelated, such as "Mary" and "Barry." However, static name-piece knowledge bases have several potential limitations which hamper name matching:

(a) they are not comprehensive, so spelling errors, emergent names, and close re-spellings may not be included;

(b) they do not address the structure of the whole name, so they cannot state whether order-swapping of names is allowed or if it is acceptable to change more than one name piece for a conflation;

(c) if they do not allow multi-token units, they cannot account for the merging of name pieces such as "Mary Beth=Marybeth" or "Mc Lean=Mclean;"

(d) if they are unweighted, they allow all name variants to be brought together without regard for the strength of their affinity (for example, allowing "Mike" and "Michael" to conflate with the same strength of affinity as "Mike" and its Russian equivalent, "Chiel");

(e) the knowledge bases are likely to disallow "generic nicknames" which can be used by potentially anyone (such as Buck, Slim, or Ace) since they do not conflate with any particular given name; and

(f) the knowledge base may allow "Sam" to conflate with "Samuel" or with "Samantha" since either is a possibility; but it should not be the case that "Sam John /Smith/" should be allowed to become "Samantha /Smith/."

We are pursuing various name-matching alternatives and extensions to knowledge base systems. We have created an initial version of such a system which tries to take advantage of the benefits of knowledge bases, but which can also address many of the weaknesses mentioned above. The system currently has the following components:

*information-retrieval-based weights for name pieces as well as for overlapping trigrams (across full names), where finding name pieces will be weighted higher than trigrams, but trigrams diminish the effects of name segmentation issues and provide greater scores for order-preserving name matches

*Transliteration to handle cross-lingual name pairings

*Name variability is allowed only if probability of affinity is greater than a threshold

*Names with no marked surname are permitted to have potentially any name piece to count as given names or as surnames

*Allowance is made for a set of generic nicknames and search is structured so as to allows generics to be found for a non-generic query name

*Gender-specific conflations are used at indexing time, and gender estimation is used at query time.

A major limitation of this system as it currently stands is that it has the potential of predicting far too many results due to the trigrams. We are exploring parse-based filtering of names as a mechanism for cleaning up bogus results (such as described in [4]), but those are not available at this time. Nevertheless, the system as it stands is interesting and will be applied to our test corpus and its performance scored. We will refer to this system as "KITCHEN-SINK."

5. NAME-VETTING PROCESS

It was mentioned previously that part of the gold standard for this evaluation would be derived using attested pairs. However, attested pairs do not provide the entire picture for name matching. If there is a query for "John /Smith/+context," there may be one attested pair referring to the same individual as "John Q /Smith/." However, since the context is only provided for conditional handling of the name, it should be that every John /Smith/ and John Smith in the data is an appropriate name matching response. It may be the case that every "Johnathan /Smith/" is also a legitimate response, but this is subject to review. Therefore, a human review stage is needed to get the full name matching picture.

Like other information retrieval (IR) evaluations such as TREC (see trec.nist.gov), we have generated the human-created gold standard *after* systems have produced their results. The following sections describe this labor-intensive vetting process which was carried out by a single individual for consistency. Also like other IR evaluations, the sole individual was the same person as the creator of the queries.

5.1. Description and Difficulties of Name Vetting Task

The human subjectively marked the results of systems with a score ranging from 0 to 5. A zero is an indication that the particular result did not reasonably match the query, and 1-5 indicate various levels of matching. A '5' indicates that the annotator has 100% confidence in the match, and a '1' suggests very little (but greater than 0%) confidence in the match.

When the evaluation first began, it was assumed that the human doing the query vetting would be responsible for marking each individual system response. However, a query like "John /Smith/+context" might identify 1000 different instances of "John /Smith/" in addition to all other variants and non-variants that would need to be considered. This massive multiplicity of results suggested that some simplifications would be necessary.

5.2. Simplification of Name Vetting Task

To accelerate the tagging process, the annotator is allowed to look at the full query (including context) and if the annotator believes that one instance of "John /Smith/" is a valid system response, then all responses with John /Smith/ would be treated as valid. That is, if we say that there is a "John /Smith/" class, which represents all name objects whose full name is "John /Smith/," then either the entire name class or none of the name class satisfies the query.

This assumption poses some potential concerns, such as whether or not a female "George /Jones/" should be allowed to match with a male "George /Jones/," but since the name matching system does not know the gender for both the query individual and the NODB individual, it seems completely appropriate to allow these matches.

An additional simplification for the annotator was to follow standard IR practices and only evaluate the top *N* system results. In this case, the reviewer evaluated the top 20 name classes from each system for each query. It is important to recognize that tagging these 20 classes may be equivalent to tagging 2000 separate results (although Zipf's law suggests that the bulk of name classes would have only 1-3 name objects).

5.3. Creation of Support Tools

As tagging continued, it was evident that the human tagger was redundantly processing the same name variants and the same name constructs. For example, if the query is for "John /Smith/+context" and the response name class is "John /Smith/", it is obvious that this is a match and that the human should not have to do the vetting. Likewise, if the human accepts "Jon /Smith/" as a variant for "John /Smith/", then it seems reasonable and consistent that the human would choose "Jon /Brown/" for "John /Brown/." Therefore, the annotator created software which implements the rules that he would follow for simple decisions, and would allow him to identify and weight name variants that appeared in the data as being invalid or valid with varying strengths.

The core of the automatic algorithm was based on weighted minimum edit distance. The support tool aligned the words from the query name with those from the system response. During the alignment, the system used an error table created by the human that suggested "if the query is for X and the response is Y, this should constitute an E-point error." If "Y" was the same string as "X," the error would be zero. If "(X,Y)" had not been seen as a pair before, the system performed a character overlap analysis between X and Y, and, if there was sufficient overlap, the system would treat this as a small error.

In the alignment also, if a first name from the query was dropped and became a middle name in the result, there would be a two-point error. The converse was also true.

After the alignment was performed, if the number of errors was greater than a threshold, the score was set to zero. If the overall error was small, then the system would provide a score, such as "(5-error)" for a two-word query. If the error was large but reasonable, and if (X,Y) was brought together because of character overlap, the query result was presented to the human to judge the result. The system also presented results to the human where the surname was not explicitly marked. If the human weighted the result favorably, the system asked the human to determine if any high-character-overlap word pairs (X,Y) should be added to the word-pair error table and with what error value.

5.4. Data Outcomes of the Vetting Process

The 10,000-query set produced tens of millions of results. The human, coupled with support tools, was able to tag 21.2M results with a 0-up score, and another 100K outputs were in a language that the annotator could not readily interpret and were marked with an "S". 806K results were produced by systems but were not assessed by the human – these were marked with a "-1."

It was mentioned earlier that the human judge rated results with a '5' if the output was almost a perfect match, down to '0' as a non-match. Table 1 represents all judgments for the 10K queries broken down by score and by whether the human vetted the result or the score was auto-generated. Numbers in parenthesis were not judged, but were produced by systems.

Table 1: *Score Break-down for Judged Results*

	Human-Touched	**Automatic Only**
5	2,261	82,834
4	6,130	178,270
3	9,660	113,431
2	14,419	506,106
1	14,135	496,826
0	252,070	19,534,983
-1	0	(882,413)
S	0	(99,278)

To give the reader a sense of what the human tagger viewed as worthy of a particular scoring level, the following is an example.

Example query: Matea /Pedersdr/

Level-5 Score: Matea /Pedersdr/

Level-4 Score: Matea /Pedersen/

Level-3 Score: Matea Olava /Pedersen/

Level-2 Score: Mattie /Peterson/

Level-1 Score: Alice M /Peterson/

Level-0: Marie Emma /Pedersdr/

For the evaluation component of this paper, a system result of 1 through 5 will be treated as "relevant" – a reasonably valid name matching response. A score of a 0, -1, or S will be treated as "not relevant" unless it is an attested pair. There are some metrics that could potentially take advantage of these weights in scoring, but for the purposes

of this paper, we will let the final relevance judgments be binary decision as was just described.

5.5 Quality of the Human+Tool Vetting Process

Due to the vast amount of data and the quantity of results that were generated through automatic means, one might question how "clean" the gold standard really is. If the human judge has done an inadequate job identifying valid name matches, then the whole evaluation comes into question. Moreover, given that there is a great deal of subjectivity to the evaluation, it could be that the particular human tagger was especially aggressive at throwing out responses -- many of which may have been valid.

Fortunately, when the original queries were created, they were created with associated attested pairs. This means we can "score" the human's performance as compared to attested pairs.

Suppose a query has three attested pairs: two are predicted by at least one system as being valid matching names and the third is not predicted. Suppose further that one of the predicted answers is scored automatically with "4" and the other is given a score by the human judge as "2" (and, let us say that a '2' given by a human is a "2H"). We will say that the *best-scored attested pair* for this query is "4." Table 2 shows the cumulative counts of the best-scored attested pairs across all of the 10,000 queries.

Table 2. *Cumulative Best-scored Attested Pair Per Query*

Best-Scored Attested Pair	Cumul. #Queries	Best-Scored Attested Pair	Cumul. #Queries
5H	572	S	8489
5	4407	S as 0H	8655
4H	4879	S as 0	8657
4	6355	0H	8952
3H	6711	0	9299
3	7106	-1	9432
2H	7396	Not Found/S	9604
2	7789	Not Found	10000
1H	8121		
1	8391		

In Table 2, if the best-scored attested pair was '4', then we say the cumulative number of queries is the total number of unique scores that were given a score of 5H, 5, 4H, or 4.

Of the 9299 queries with non-negative best-scored attested pairs, the human identified 90.2% of the attested pairs. 98 queries, or 1% of those that were missed, were not in a language that the reviewer could process and were given a score of "S." 168 additional queries, or 1.8%, had cross-lingual components that the human judge tried unsuccessfully to judge. If we consider only Latin-script errors that the human or system should have been able to judge, 3.2% of the error was contributed by the human and the support system made an additional 3.7% error.

The "-1" scores indicate that a person would have to dig deeply to find the result, which is typically a system issue rather than a human issue. This represents 1.3% of the total failure to find attested pairs. The remaining 5.7% of the total error is attributable to the systems' inability to find the pairs, with 1.7% of the total failure due to cross-lingual issues and 4% being other issues.

Overall, when one considers that the human+tool only failed to find 6% of the Latin-script attested pairs from among over 21M items presented to them, the judgments seem very credible. Moreover, since the final gold standard couples the human results with the actual attested pairs, the overall scores are likely to be quite trustworthy.

6. EVALUATION AND INITIAL SCORES

After having described the evaluation paradigm and the systems involved, as well as the process for vetting results, we now show the performance of the various algorithms.

6.1. Description of Metrics

In information retrieval, two key ingredients to system performance are *precision* and *recall*. If a system predicts N answers and p of them are correct, its precision is p/N. If there are R possible correct answers in the entire collection, recall is p/R. It is possible to "game" precision or recall by producing very few answers or far too many answers, respectively.

Two common metrics exist which attempt to simultaneously consider precision and recall. One of these is Mean Average Precision, or MAP, which computes the area under the precision-recall curve for each query, sums these, and then normalizes by the number of queries. MAP considers system results in order, and systems that return correct responses earlier in the list are rewarded higher. A weakness of MAP is that if one system returns N answers and another system returns $10N$ answers, and if the $10N$-set has correct responses at the same ranks as the N-set, the two systems produce the same MAP.

Another balancing metric is F-score, which is the harmonic mean of precision and recall. More specifically, F-score is (Precision*Recall)/(0.5*(Precision+Recall)). F-score does not consider order, which is a potential deficiency. In F-scores case, if two systems each produce N results with k correct results ("hits"), the two systems will have the same F-score even in the first system produced its k correct hits at the top of the list and the other produced the k at the bottom of its list.

Another recently-introduced metric [5] is Adjusted Mean Average Precision, or AMAP. This metric preserves the benefits of order that are derived from MAP, but penalizes systems which produce extended lists of output with no relevant components. A limitation of AMAP is that one needs to have some sense for the "irritation level" of users to properly calculate the penalty—that is, how deep a user is willing to look without successfully finding any valid hits to their query and without getting annoyed. For names, we set the value as 5.

We illustrate performance using all three of these metrics. MAP will show which systems are better at finding the correct results earlier in their list of outputs. F-scores will indicate which systems tend to not over-produce non-correct results. AMAP will also balance these issues.

6.2. System Performance Compared to Human Judge

Table 3 shows performance of multiple name matching systems as they are compared to the "gold standard" produced by combining human judgments and attested pairs. The descriptions for each system were provided in Section 4, where it was indicated that the knowledge-base-derived systems would be described by the five features of Section 4.1.3 and the one alternative system, "KitchenSink."

Table 3: *Scores using Human Judgments*

System Description	MAP	F-Scr	AMAP
NONE,+OFF,+INIT,+MISS,W1	23.0	23.9	22.2
NONE,+OFF,-INIT,-MISS,W1	17.0	17.0	20.7
NONE,+OFF,+INIT,-MISS,W1	22.7	27.3	22.6
FOL,+OFF,+INIT,+MISS,W1	33.3	20.5	28.8
FOL,+OFF,+INIT,-MISS,W1	33.5	28.9	31.4
FOL,+CLUS,+INIT,+MISS,W1	33.7	20.6	29.1
FOL,+CLUS,+INIT,-MISS,W1	33.9	29.2	31.7
DRS,+OFF,+INIT,+MISS,W1	32.8	26.8	30.6
DRS,+OFF,+INIT,-MISS,W1	33.1	34.8	32.7
DRS',+OFF,+INIT,+MISS,W1	32.9	27.8	31.0
DRS',+OFF,+INIT,-MISS,W1	33.1	34.9	32.7
DRS,+CLUS,+INIT,+MISS,W1	33.7	26.4	31.1
DRS,+CLUS,+INIT,-MISS,W1	33.9	35.0	33.3
DRS',+CLUS,+INIT,+MISS,W1	33.7	27.5	31.5
DRS',+CLUS,+INIT,-MISS,W1	33.9	35.0	**33.3**
S&A,+OFF,+INIT,+MISS,W1	30.8	27.7	29.2
S&A,+OFF,+INIT,-MISS,W1	31.0	34.5	30.7
S&A',+OFF,+INIT,+MISS,W1	30.8	28.6	29.4
S&A',+OFF,+INIT,-MISS,W1	30.9	34.6	30.7
S&A,+CLUS,+INIT,+MISS,W1	32.4	27.8	30.5
S&A,+CLUS,+INIT,-MISS,W1	32.6	**35.5**	32.3
S&A',+CLUS,+INIT,+MISS,W1	32.4	28.9	30.8
S&A',+CLUS,+INIT,-MISS,W1	32.6	**35.5**	32.3
DRS,+OFF,-INIT,-MISS,W2	**35.7**	34.4	**35.8**
S&A,+OFF,-INIT,-MISS,W2	33.8	**37.9**	33.0
KITCHEN-SINK	**44.2**	18.6	31.0

6.3. System Compared to Attested Pairs

Since the human judgments are subjective, it is also beneficial to consider the scores as if the attested pairs themselves formed the gold standard. Of course, this gold standard may count "X Y /Z/" to be a match for "X /Z/" and may disregard "X /Z/" itself since it may not have been attested. Yet even so, the metric is not subjective and can be reflective of the kind of performance a genealogical search might experience while searching for an ancestor.

Table 4: *Scores using Attested Pairs as Gold Standard*

System Description	MAP	F-Scr	AMAP
NONE,+OFF,+INIT,+MISS,W1	27.1	19.0	22.1
NONE,+OFF,-INIT,-MISS,W1	25.5	**27.2**	24.8
NONE,+OFF,+INIT,-MISS,W1	27.0	25.0	25.0
FOL,+OFF,+INIT,+MISS,W1	34.4	10.1	18.4
FOL,+OFF,+INIT,-MISS,W1	34.4	19.2	24.8
FOL,+CLUS,+INIT,+MISS,W1	35.0	10.2	18.6
FOL,+CLUS,+INIT,-MISS,W1	35.0	19.5	25.1
DRS,+OFF,+INIT,+MISS,W1	33.7	16.0	22.6
DRS,+OFF,+INIT,-MISS,W1	33.7	25.3	28.3
DRS',+OFF,+INIT,+MISS,W1	33.7	16.7	23.1
DRS',+OFF,+INIT,-MISS,W1	33.7	25.3	28.3
DRS,+CLUS,+INIT,+MISS,W1	34.2	14.9	22.4
DRS,+CLUS,+INIT,-MISS,W1	34.2	24.7	28.3
DRS',+CLUS,+INIT,+MISS,W1	34.2	15.7	23.0
DRS',+CLUS,+INIT,-MISS,W1	34.2	24.8	28.3
S&A,+OFF,+INIT,+MISS,W1	33.3	18.0	23.9
S&A,+OFF,+INIT,-MISS,W1	33.3	**27.0**	29.0
S&A',+OFF,+INIT,+MISS,W1	33.3	18.8	24.5
S&A',+OFF,+INIT,-MISS,W1	33.3	**27.0**	29.0
S&A,+CLUS,+INIT,+MISS,W1	34.1	17.9	24.2
S&A,+CLUS,+INIT,-MISS,W1	34.1	**27.2**	29.6
S&A',+CLUS,+INIT,+MISS,W1	34.1	18.8	24.8
S&A',+CLUS,+INIT,-MISS,W1	34.1	**27.2**	29.6
DRS,+OFF,-INIT,-MISS,W2	**44.1**	22.4	**30.0**
S&A,+OFF,-INIT,-MISS,W2	42.8	25.8	**32.1**
KITCHEN-SINK	**58.6**	3.6	18.3

6.4. Brief Analyses of the Tables

The tables have been colored with green to indicate best results per column, and with yellow to identify the second best results. From these comprehensive tables, we can discern a number of issues about the systems being tested. It is worth identifying these issues briefly.

First, the "baseline" system, which uses no knowledge base, provided a significantly poorer result on the human gold standard than did any of the other systems. Its MAP and AMAP were also sizably lower than others at finding attested pairs. Interesting, though, it gave the best F-score

result at finding attested pairs – perhaps because it frequently fails to find matching names, but when it does, it is usually correct.

The KitchenSink system could be a significant improvement over other systems if its long-tailed response could be truncate to prune out bogus name matches. As was mentioned earlier, work is under way to create a name-parsing filter (see [4]).

We can also see that the W2 weight set seems to provide better results that the W1 weight set. It would be advantageous to see more comparisons between these two systems to see if the weight is the key ingredient that makes the difference.

From the table, it is clear that when the MISSING parameter is allowed, the MAP increases very little if any, but the F-score drops significantly. This suggests that the MISSING parameter is adding new outputs to the bottom of the list, but most of the additions are not valid matches.

Lastly, we can see that when S&A's culture processing is turned on, there is a slight gain in performance for all three measures. When clustering is turned on with the DRS knowledge base, the MAP/AMAP scores also increase at with very little or no sacrifice to the F-score.

7. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we illustrated the creation of a comprehensive, historical and cross-lingual evaluation of name matching systems. We also identified a number of initial algorithms and showed their performance. The performance for now is certainly usable and genealogical patrons are able to discover their ancestors with current levels of name matching accuracy.

However, this evaluation has provided us a mechanism for determining how to move forward in creating our genealogical search. We also believe that there is sufficient amount of work in this area that we would like for the research community to get involved. In order to enable more rapid improvements in this area of genealogical name matching, we will be seeking to open this evaluation to the wider research community in the upcoming months.

8. REFERENCES

[1] K Miller, M Arehart, C Ball, J Polk, A Rubenstein, K Samuel, E Schroeder, E Vechhi, C Wolf (2008) "An Infrastructure, Tools and Methodology for Evaluation of Multicultural Name Matching Systems", MITER, McLean, VA.

[2] P Schone, T Allison, C Giannella, C Pfeifer (2011) "Bootstrapping Multilingual Relation Discovery Using English Wikipedia and Wikimedia-Induced Entity Extraction," ICTAI 2011, Boca Raton, FL pp 944-951.

[3] P. Schone (2011) "Development of An Evaluation Paradigm for 'RecordMatch' and its Application to GenMergeDB Clustering Results," FHTW at RootsTech 2011, Salt Lake City, Utah

[4] P. Schone, S. Davey (2012) "A Multilingual Personal Name Treebank to Assist Genealogical Name Processing," FHTW at RootsTech 2012, Salt Lake City, Utah (to appear)

[5] P. Schone, M. Jones (2011) "Genealogical Search Analysis Using Crowd Sourcing," CIR@SIGIR 2011, Beijing, China.