Conceptual Modeling in Accelerating Information Ingest into *Family Tree*

David W. Embley[1,2], Stephen W. Liddle[1],
Tanner S. Eastmond[1], Deryle W. Lonsdale[1], Joseph P. Price[1], and Scott N. Woodfield[1]

[1] Brigham Young University, Provo, Utah 84602, USA
[2] FamilySearch, Orem, Utah 84097, USA

Abstract. *Family Tree* is a wiki-like shared repository of interconnected family genealogies. Because information ingested into the tree requires human authorization as verified in source documents, ingest is tedious and time-consuming. To significantly increase ingest efficiency while maintaining human oversight, we propose a pipeline of tools and techniques to transform source document genealogical assertions into verified information in the *Family Tree* data repository. The automation pipeline transforms pages of printed, scanned and OCRed family history books into a GEDCOM X conceptualization that can be ingested into *Family Tree*. All steps of the pipeline are fundamentally grounded in ontological conceptualizations. We report on the pipeline implementation status and give results of initial case studies in semi-automatically ingesting information obtained from family history books into *Family Tree*.

Keywords: conceptual modeling, information extraction, family history.

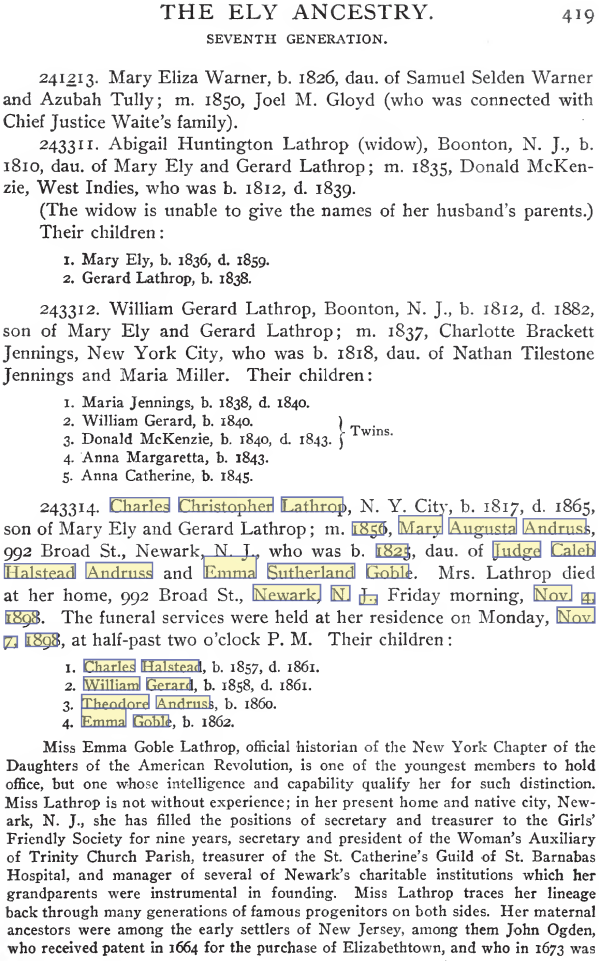
1 Introduction

FamilySearch [1] maintains a freely accessible collection of records, resources, and services designed to help people learn more about their family history. Its *Family Tree* allows users to collaborate on a single, shared, worldwide family tree. Currently *Family Tree* has information on about a billion people, including their names, birth and death data, and their marriage and parent-child relationships to others in the tree. Users can also attach to each person stories, photos, and images of documents from which the genealogical information is derived.

Users add persons one-by-one to *Family Tree* and update information already in the tree one item at a time. Users are expected to have verified the information they add to the tree, and their contact information is added to all updates they make. They should also document information they add by including source information—ideally images of documents that verify tree updates.

Using principles of automated conceptual-model-based information extraction [2, 3], we are building a system to accelerate ingest of information into *Family Tree*. As source documents, the system we are building targets the collection of several hundred thousand family history books, which are being scanned,

OCRed, and placed online by FamilySearch. The collection contains genealogical information about millions of people, many of whom are already in the tree, but many of whom are not. For those already in the tree these books may contain corroborating information, information not yet recorded in *Family Tree*, and in some instances conflicting information that needs to be resolved.



THE ELY ANCESTRY. 419

SEVENTH GENERATION.

241213. Mary Eliza Warner, b. 1826, dau. of Samuel Selden Warner and Azubah Tully; m. 1850, Joel M. Gloyd (who was connected with Chief Justice Waite's family).

243311. Abigail Huntington Lathrop (widow), Boonton, N. J., b. 1810, dau. of Mary Ely and Gerard Lathrop; m. 1835, Donald McKenzie, West Indies, who was b. 1812, d. 1839.

(The widow is unable to give the names of her husband's parents.) Their children:

1. Mary Ely, b. 1836, d. 1859.
2. Gerard Lathrop, b. 1838.

243312. William Gerard Lathrop, Boonton, N. J., b. 1812, d. 1882, son of Mary Ely and Gerard Lathrop; m. 1837, Charlotte Brackett Jennings, New York City, who was b. 1818, dau. of Nathan Tilestone Jennings and Maria Miller. Their children:

1. Maria Jennings, b. 1838, d. 1840.
2. William Gerard, b. 1840. } Twins.
3. Donald McKenzie, b. 1840, d. 1843. }
4. Anna Margaretta, b. 1843.
5. Anna Catherine, b. 1845.

243314. Charles Christopher Lathrop, N. Y. City, b. 1817, d. 1865, son of Mary Ely and Gerard Lathrop; m. 1856, Mary Augusta Andruss, 992 Broad St., Newark, N. J., who was b. 1823, dau. of Judge Caleb Halstead Andruss and Emma Sutherland Goble. Mrs. Lathrop died at her home, 992 Broad St., Newark, N. J., Friday morning, Nov. 4, 1898. The funeral services were held at her residence on Monday, Nov. 7, 1898, at half-past two o'clock P. M. Their children:

1. Charles Halstead, b. 1857, d. 1861.
2. William Gerard, b. 1858, d. 1861.
3. Theodore Andruss, b. 1860.
4. Emma Goble, b. 1862.

Miss Emma Goble Lathrop, official historian of the New York Chapter of the Daughters of the American Revolution, is one of the youngest members to hold office, but one whose intelligence and capability qualify her for such distinction. Miss Lathrop is not without experience; in her present home and native city, Newark, N. J., she has filled the positions of secretary and treasurer to the Girls' Friendly Society for nine years, secretary and president of the Woman's Auxiliary of Trinity Church Parish, treasurer of the St. Catherine's Guild of St. Barnabas Hospital, and manager of several of Newark's charitable institutions which her grandparents were instrumental in founding. Miss Lathrop traces her lineage back through many generations of famous progenitors on both sides. Her maternal ancestors were among the early settlers of New Jersey, among them John Ogden, who received patent in 1664 for the purchase of Elizabethtown, and who in 1673 was

Fig. 1. Highlighted Data for Mary Augusta Andruss in *The Ely Ancestry*[4].

Figure 1 shows a paragraph from a page of one of these books, *The Ely Ancestry* [4]. The information of interest to be placed in the tree for Mary Augusta Andruss is highlighted—her birth date, her death date and place, her burial date, her parents, and her spouse along with their marriage date and their children. Figure 2 shows the information captured by our system for Mary. The

```
************************************
Person osmx393: Mary Augusta Andruss
************************************
Name:
      Conclusion Name: Mary Augusta Andruss
      Original Document Text: Mary Augusta Andruss
      Interpreted Document Text: Mary Augusta Andruss
      Married Name: Mary Augusta Andruss Lathrop
      Married Name: Mary Augusta Andruss Lathrop
Gender: Unknown
Facts:
      BirthDate:
           Conclusion: 1825
           Original Document Text: 1825
           Interpreted Document Text: 1825
      BirthPlace:
Marriage Relationships:
      Spouse: osmx334 (Charles Christopher Lathrop)
      MarriageDate:
           Conclusion: 1856
           Original Document Text: 1856
           Interpreted Document Text: 1856
ParentOf Relationships
      osmx260 (Charles Halstead Lathrop)
      osmx319 (William Gerard Lathrop)
      osmx168 (Theodore Andruss Lathrop)
      osmx434 (Emma Goble Lathrop)
ChildOf Relationships:
      osmx290 (Judge Caleb Halstead Andruss)
      osmx427 (Emma Sutherland Goble)
```

Fig. 2. Person Information Record.

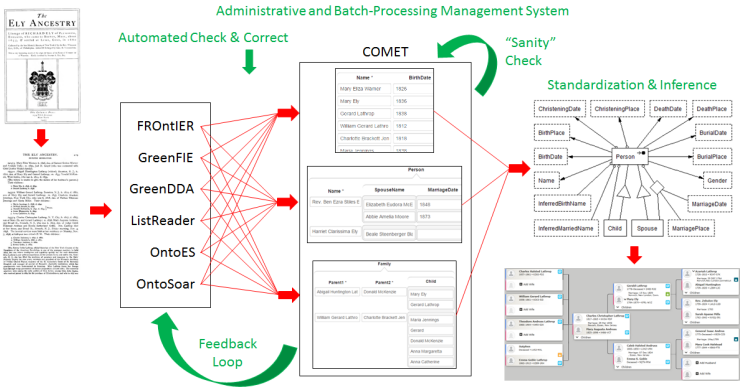


Fig. 3. Fe6 Pipeline.

captured data is ready to be automatically ingested into the *Family Tree* along with its source documentation, the text with highlights in Figure 1.

We call our system **Fe6** (**F**orm-based **e**nsemble with **6** extraction tools). Figure 3 shows the pipeline beginning with a source-document book and ending with the genealogical information from the book being ingested into *Family Tree*. Figure 3 illustrates the steps in the process:

1. Split the PDF document resulting from scanning a book into individual pages.
2. Apply an ensemble of extraction engines to each page.
3. Merge the extracted data and split it into three filled-in forms—*Person*, *Couple*, and *Family*, focusing respectively on individual, marriage, and parent-child information.
4. Check and correct the automatically filled-in forms.
5. Enhance the checked data by standardizing it and by inferring implied information such as gender and birth and married names.
6. Transform Fe6's internal conceptualization of the data into *Family Tree*'s internal conceptualization.

The remainder of this chapter describes details of the ingest pipeline (Section 2), which from beginning to end is fundamentally grounded in conceptual modeling [5–11]. We therefore particularly highlight the pipeline's connection to conceptual modeling. Next we give the status of our project (Section 3)—meeting FamilySearch's human-oversight requirements (Section 3.1), the implementation status of the pipeline (Section 3.2), and some preliminary results about our ingest experience (Section 3.3). We conclude by discussing potential impact (Section 4).

2 Fe6 Pipeline

The objective of the Fe6 pipeline is to populate the conceptual-model diagram on the right in Figure 3 and then to transform the data in this conceptualization into *Family Tree*. We begin by automatically extracting data into the conceptual model diagrammed in Figure 4, which models the target data directly extractable from a text document. The views superimposed on this diagram correspond to Person, Couple, and Family forms, which are automatically filled in so that a user can check and correct the output generated by the tool ensemble. We refer to our conceptual models as ontologies, which emphasizes the philosophical notion of "the nature of being"—the reality of the existence of families and individuals.

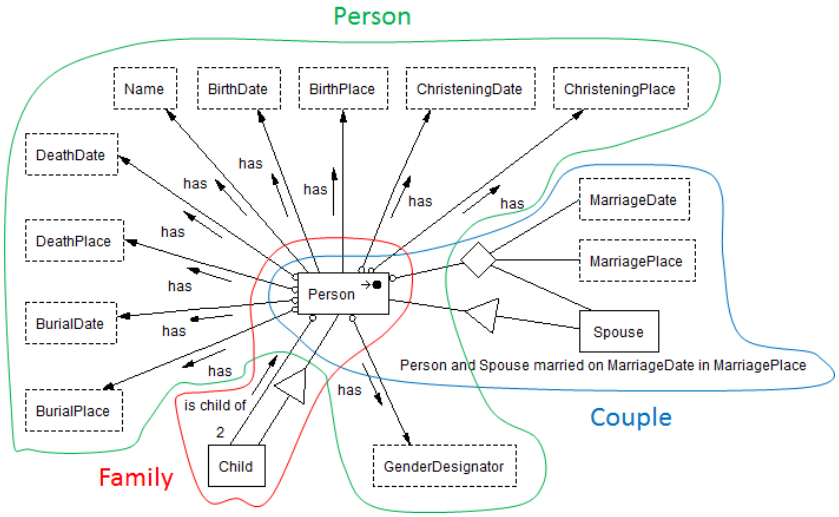


Fig. 4. Target Ontology for Extraction Engines.

2.1 Import Book

Given a printed historical book containing genealogical information, it is scanned, OCRed, and rendered as a PDF document. It is then split into pages and for each

page we produce five files: (1) a single-page PDF document; (2) a PNG image of the page; (3) a `.txt` file with the OCRed text; (4) an XML document containing bounding-box information for every character, word, and line of the OCRed text in the PNG image; and (5) an HTML web page that renders the PNG image superimposed over hidden OCRed text for use in the user interface that allows for checking and correcting automatically generated extraction results.

2.2 Run Extraction Tools

Conceptual modeling is the underlying formalism of all six of the Fe6 ensemble's extraction tools. In essence, the tools "read" the text on a page by converting word sequences into conceptual entities and relationships among the entities. Categorically, the extraction tools stem from work in expert systems, natural language processing, and machine learning. Spanning across these categories helps the ensemble work with document types that range from those that are highly structured (e.g. cemetery records that are near table-like in structure) to those that are free running text (e.g. narrative family history stories) and everything in between (e.g. the page in Figure 1 from the 830-page Ely book).

FROntIER [12] extends our work on conceptual-model-based data extraction [2]. It extracts and attempts to organize data, reasoning about the extracted information to infer facts not explicitly stated in the underlying text and deduplicating extractions of different references to the same person. FROntIER extraction rules are solidly based on conceptual modeling. Each lexical object set s has a collection of regular expression extraction rules that identify instances in running text that belong to an extension of s. Nonlexical object sets such as *Person* are instantiated by *ontological commitment*—a relationship between language and an object postulated to exist by that language, so that when a person name is extracted, a *Person* object is instantiated. Relationships between and among entities are instantiated by regular expression recognizers with embedded entity instance recognizers. For example, the rule "*person-name* was born on *date*", where *person-name* and *date* are any of the regular expression recognizers in the collection of recognizers for person names and dates, can instantiate a relationship in the *Person-has-BirthDate* relationship set in Figure 4.

OntoES is another extension of [2] based on extracting ontology snippets, which let users specify extraction rules for a collection of object and relationship sets. An ontology snippet is a view over a conceptual model, and each ontology snippet regular expression recognizer identifies and extracts some or all of the objects and relationships for the view in a single execution of the rule. For our application, we tailor these views to our three forms: Person, Couple, and Family. Thus, ontology snippet extractors are an efficient way to fill in the fields of these forms. We note that there is a strong relationship between forms and these ontology snippet views; indeed, for any view we can derive a form and from any collection of related forms we can derive a conceptual model [13].

GreenFIE [14] "watches" users fill in form-records, namely records for our Person, Couple, or Family form, and can generate ontology snippet extraction rules from each of the filled-in records it "sees." It then executes these extraction

rules on subsequent pages to prepopulate forms for users to check and continue to fill in for record patterns not yet encountered. GreenFIE is "green" in the true sense of the word, which in this context stands for tools that improve themselves as they are used in real-world work [15].

ListReader [16] discovers record patterns in text. It abstracts the text of an entire book, replacing, for example, words that begin with an uppercase letter like "Mary" by the symbol "[UpLo]" and digit sequences like 1836 by the symbol "[DgDgDgDg]". It then groups text into the patterns it encounters. For example, in Figure 1, it groups children with a birth and death date like "1. Mary Ely, b. 1836, d. 1859" whose pattern is "[Dg]. [UpLo] [UpLo], b. [DgDgDgDg], d. [DgDgDgDg]" into one group and children with just a birth date into another group. A user then labels a ListReader-chosen prototypical example by filling in a form—in this Mary Ely example, by putting "Mary Ely" in the Person-form's *Name* field, "1836" in the form's *BirthDate* field, and "1859" in the form's *DeathDate* field. This form filling process establishes a correspondence between the record in the group and a form and thus also the ontology because of the correspondence between form and conceptual model [13]. It also labels every other record in the group. Thus, with one record labeling, all the information for all the records in the group is extracted into the conceptual model—usually hundreds of records in books like *The Ely Ancestry*.

OntoSoar [17] extracts data using NLP techniques to segment and parse the text, and a cognitive reasoner (Soar [18]) to semantically analyze the parse of each segment and map results of the analysis to an ontology. OntoSoar's segmenter chunks semi-structured text like that in Figure 1 into clauses which may or may not be sentential in structure but are nevertheless parsable by its Link Grammar parser. The analyzer in our implementation has 240 Soar production rules. These rules build meaning using ideas inspired by construction grammars, which (1) pair textual forms with meaning; (2) construct knowledge structures with inference rules; and (3) map knowledge structures to ontologies by comparing their common entities and relationships. The mapping provides a conduit for populating the ontological conceptualization in Figure 4 with data.

GreenDDA is an experimental tool, with which we are investigating the use of standard machine learning, but requiring only a minimal amount of clean training data. It is "green" in the sense that it takes its clean training data from user-checked and -corrected filled-in forms for a page. Its DDA (Decision Directed Adaptation) [19] component then trains a classifier, applies it to a subsequent page, takes the results and adds them to its set of training data, and then repeats this process on additional pages. If the process converges to a stable state, the trained classifier is then applied as part of the ensemble to unprocessed pages in an attempt to improve the extraction.

2.3 Merge Extracted Information

The next step in the Fe6 pipeline is to merge the results obtained from the extraction engines. Merge proceeds by noting the position on the page of extracted text strings. Identical strings appearing at the same location on a page

are merged, as are strings with significant overlap. For example, if one tool extracts "Judge Caleb Halstead Andruss" from the page in Figure 1 and another tool omits the title, "Judge", extracting only "Caleb Halstead Andruss", they are nevertheless merged as one. Since persons are instantiated by ontological commitment with names, name merge implies person-object merge as well.

We keep multiple string values for each lexical object. First is the text of the extracted string itself along with its page location. Second is a cleaned string in which we attempt to (1) fix common OCR errors such as the "i" in "i860" in the birth year of Theodore Andruss in Figure 1 and (2) resolve end-of-line hyphens so that "McKen-\nzie" in Figure 1 becomes "McKenzie". Third is a mapping of the date values into a Julian date string which can easily be converted into an integer for date comparison operations. Thus, for example, the death date of Mary Augusta Andruss in Figure 1, which is "Nov. 4, 1898", becomes "1898308".

We next evaluate the merged/cleaned data and fix egregious anomalies. Unlike most databases which require data to be valid with respect to declared constraints, we allow our conceptual models to be populated with invalid data, preferring to specify ontologically correct constraints and let violations stand until they can be resolved. For example, the model instance in Figure 4 declares that a *Person* has exactly one *BirthDate* as specified by the functional arrow and the absence of an "o" (an "o"ptional indicator) on its tail connection. But the extraction engines may find zero or several birth dates for a person. Min-violations of a cardinality constraint [20] merely mean that information is unknown, but max-violations are egregious and should be fixed. Consider the participation constraint *2* in Figure 4 declaring that a *Child* has exactly two parents. This is a commonly encountered violation because of the difficulty of specifying how far ahead to look for a child list for a couple. In Figure 1, the amount of text to skip between Mary Augusta Andruss and her first child, Charles Halstead, is greater than the amount of text to skip between Joel M. Gloyd, who has no children, and the next couple's first child, Mary Ely. To not miss parent-child associations, the extraction engines need rules with both short and long skip-lengths. The result in this example is that Mary Ely has four parents, Mary Eliza Warner, Joel M. Gloyd, Abigail Huntington Lathrop, and Donald McKenzie. This egregious anomaly can be reliably and automatically fixed by discarding *Child–is_child_of–Person* relationships for all but the closest couple.

2.4 Check Quality

Figure 5 shows the user interface for COMET, our **C**lick-**O**nly, or at least **M**ostly, **E**xtraction **T**ool, which allows users to fill in forms on the left from a document on the right. Users click on text tokens in the document to fill in a field of focus in a form. The document is an image of a scanned page superimposed over hidden OCRed text. Users may edit field values, for example, to correct OCR errors. They may also move to previous or subsequent pages to enable annotating records that cross page boundaries such a list of children that continues onto a subsequent page. As Figure 5 shows, hovering over a filled-in record highlights the fields of the record and the corresponding extracted text in the document.

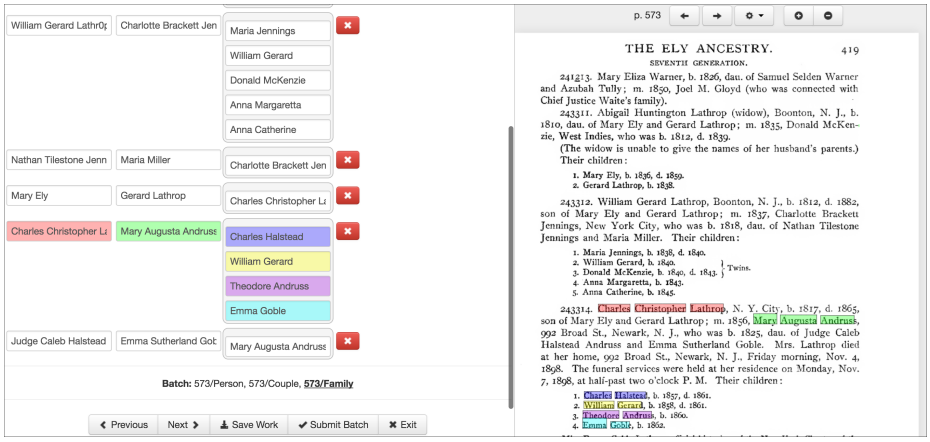


Fig. 5. COMET Screenshot.

Form-records in COMET correspond precisely with ontological conceptualizations [13]. Thus, when a user fills in a *Family* form record in Figure 5, the underlying system populates the *Family* view in Figure 4 with the data. Conversely, when the extraction engines populate the target extraction ontology in Figure 4, the form records for any of the various views are filled in so that a user only has to check the work of the ensemble of extraction engines and make corrections—e.g. delete erroneous records with the red-x button in Figure 5, add a missing record, or click on a filled-in field to edit or replace the field value.

Users work on a batch of pages at a time as controlled by the buttons in the lower left of the interface. After clicking on *Submit Batch*, the system invokes a semantic check of the data to find violations of ontologically declared constraints and missing person or place names in authority lists. Declared constraints consist not only of the conceptual model's cardinality constraints but also of Datalog-like general constraints declared over the model's object and relationship sets [21]. Authority lists comprise tens of thousands of person and place names known to FamilySearch. When irregularities are found, icons are added to fields in question and the batch is returned to the user for further review.

Users can click on the icons to obtain explanations. For example, clicking on the question-mark icon for the child Francis Argyle in Figure 6 yields the pop-up, which explains that Elizabeth Eudora McElroy cannot be her mother since Elizabeth died before Francis was born. After resolving raised issues, users again click on the *Submit Batch* button, and the system accepts the results. If accepted person or place names are missing in the authority lists, the system adds these missing names to local, book-specific authority lists, which are checked along with global, FamilySearch-provided authority lists so that when checking subsequent pages, the system will not mark these names as possible errors.

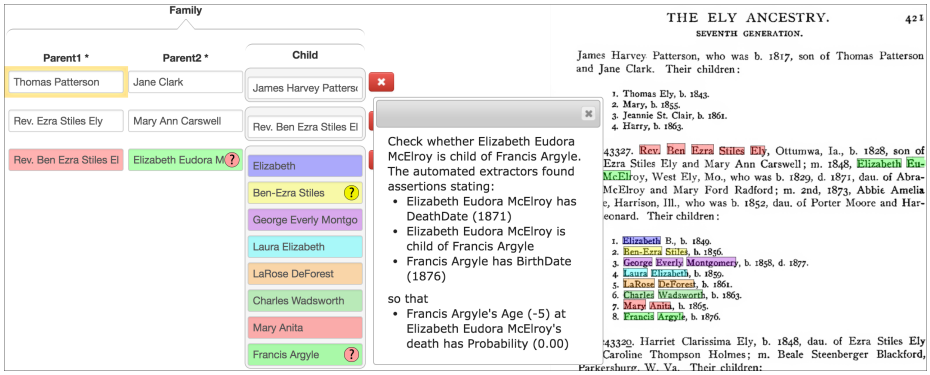


Fig. 6. Screenshot of Constraint Violation: Child Born After Mother's Death.

2.5 Enhance Data

At this point in the Fe6 pipeline, the data is assumed to have been correctly extracted. The data, however, is not necessarily in a preferred form and desired data that is not directly extractable but is strongly inferred is not present.

We standardize dates and person and place names. For example, Mary Augusta Andruss's death date in Figure 1 is extracted as "Nov. 4, 1898" and standardized as "4 November 1898". We standardize a name by ordering its components with title(s) first, followed by given names, surnames, and suffixes, and we use standard upper- and lower-case nomenclature. Place names are taken from FamilySearch's place-name authority when a match can be found, and, in any case, are ordered by administrative levels, local to global.

Gender is almost never directly extractable in family history books because authors do not normally use the words "male" or "female". Instead they expect readers to infer gender by context. We can reliably do the same inference automatically. We do directly extract "gender designators" such as "he", "she", "Mrs.", etc., and we use them as reliable indicators of gender. A married person in a historical document whose gender is unknown but whose spouse's gender is known can also be reliably inferred. Lastly, first given names are good indicators of gender and can be used as a last resort. Drawing from the billion-plus persons in *Family Tree*, FamilySearch has a 92-megabyte file of names paired with their probability of being male. Using a threshold of above 0.95 for males and below 0.05 for females, we can be quite sure of the gender. If there is insufficient information, we leave gender unknown.

Inferring birth and married names is tricky because we do not know which name form has been extracted. In Figure 1, the listed child names consist only of given names (no birth surnames); parent names are birth names that may or may not have a title like "Judge"; one of the names, namely "Mrs. Lathrop", has no birth-name components at all; another, namely "Miss Emma Goble Lathrop", includes the full birth name and adds a title. In other documents names like "Mr. and Mrs. Charles Christopher Lathrop" appear in which no part of the birth

name of the female spouse is included, and married female names appear with and without maiden surnames, e.g. either of "Mary Augusta Andruss Lathrop" or "Mary Augusta Lathrop". However, given enough information about father and male spouse names, birth and married names can reliably be sorted out.

2.6 Update Tree

At this point in the Fe6 pipeline, we will have the conceptual model in Figure 3 populated with information—one instance for each page that contains genealogical information in a given family history book. The information collected will have been automatically extracted by the ensemble of extraction tools, checked and edited as needed by a human to ensure accuracy, and automatically enhanced by inferring critical information that is not directly extractable. Further, all of the extracted and inferred lexical data will have been converted to a standard form acceptable for input into *Family Tree*.

In preparation for ingesting this generated information into *Family Tree*, we next transform the data from the pipeline's conceptual model to GEDCOM X [22]—a standard conceptual model for exchanging genealogical information. Each GEDCOM X document contains the information for one page and may include some information from prior and subsequent pages when the focus page has cross-page annotations. We also gather into each GEDCOM X document citation information for the book and bounding-box coordinates for each extracted data instance on the focus page and on any surrounding pages.

For ingest into *Family Tree*, we generate a person information record (see Figure 2) for each person listed in a GEDCOM X document. Taking a person's record document as input, we programmatically fill in a form with the information and invoke a search for the person in *Family Tree*. The search form has fields for title, first names, last names, suffix, gender, living or deceased status, date of birth, birth place, date of death, death place, father first names, father last name, mother first names, mother last name, spouse first names, and spouse last name. From the record in Figure 2, we can fill in 13 of these 16 fields. When executed, possible matches are returned, ordered best first according to FamilySearch's matching algorithm. We programmatically scrape information from the top three possible matches and compare it with the person's information record. Each field that matches for a given search result increases the score. For Mary Andruss, first names, last names, gender, deceased status, father first names, father last name, mother first names, mother last name, spouse first names, and spouse last name all match individually, and our match algorithm declares that the Mary Augusta Andruss whose extracted information is in Figure 2 matches Mary Augusta Andruss whose ID in *Family Tree* is K4B6-VCT.

Having found Mary Andruss in *Family Tree*, we can now automatically add any missing information, add any alternative conflicting information, and add a source document to validate these updates. Our proposal for automating actual updates to *Family Tree* while also satisfying FamilySearch's human oversight requirements is in Section 3.1. Here, we note that by hand, we added Mary's death date, burial date, and married name, which were all missing, and we

changed Mary's birth date from "about 1831" to "1825". To document these tree updates, we also added the image in Figure 1 as a source document.

3 Project Status

3.1 Human Oversight of Automated Updates

The oversight for ensuring that the information is correct with respect to the source document is centered in COMET along with the pipeline's interactive quality checking procedures. Thus, so long as the downstream inference and standardization algorithms function properly, the information presented for ingest should be considered as having had sufficient human oversight.

The automated search for matches in *Family Tree* can have several outcomes: (1) insufficient evidence to be confident of any match, (2) sufficient evidence to be confident of (2a) zero matches, (2b) one match, or (2c) several matches. For (2b), which is like the Mary Andruss example above, automatic ingest removes the tedium of adding facts and source documentation by hand. When merging conflicting information, a new fact should replace an existing fact only if the new fact properly subsumes the existing fact or if the existing fact is specifically marked as being questionable (e.g. "about 1831"). For (1) and (2a), automatic ingest is straightforward, but the decision to create a new person depends on policy. An alternative would be to create a new node in a tree for the book outside of *Family Tree*. Then, upon completion of the book, node clusters with links to *Family Tree* nodes can be automatically ingested as can node clusters deemed by policy to be large enough to add to *Family Tree*. For (2c) a human must be in the decision-making ingest loop. Interestingly, as we explain next, a conceptual-modeling view of the results of running the pipeline can aid the decision-making process.

Figure 7 shows a conceptual-modeling view laid out as proposed in D-Dupe [23], a visualization tool aimed at helping users integrate new information into a database and deduplicating information already in the database. Each named rectangle is an object set derivable as a role-specialization of the *Person* object set in the conceptual model in Figure 3. *Father*, for example, is a male person who has a child. The relevant objects for the question at hand appear inside the object sets. Objects are denoted by their internal ID's and, since they are all persons established by ontological commitment, their names also appear to make the view human readable. Lines denote relationships and together with the objects form a subgraph of the larger underlying graphs of both the pipeline's conceptual model and *Family Tree*'s conceptualization. Attribute values for persons in the two *Person* object sets provide additional information for determining duplicates.

A D-Dupe view for integration and deduplication can be generated whenever the automated search returns several matches—Case (2c) above. For example, "Mary Ely (osmx161)", the first Mary Ely in the page in Figure 1, matches two persons[3] in *Family Tree*, Mary Ely (KFRL-WXZ) and Mary Eli (MGV1-9BJ).

[3] Two person instances of the ever evolving *Family Tree* instance on June 5th, 2017.

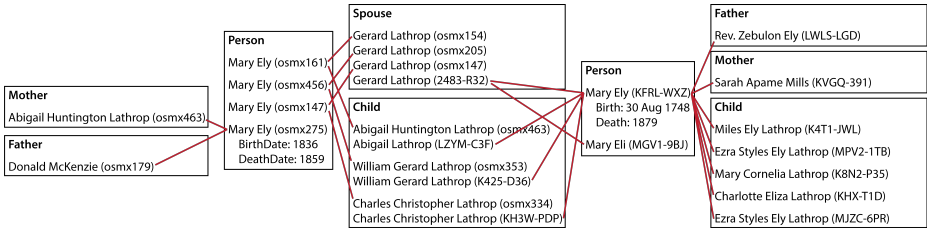


Fig. 7. Integration and Deduplication of Mary Ely.

The D-Dupe view in Figure 7 has the two *Family Tree* Mary Ely instances in the *Person* object set on the right and the extracted Mary Ely instance in the *Person* object set on the left. Also in the *Person* object set on the left are other Mary Ely instances judged by FROntIER-like inference [12] as potential duplicates. As Figure 7 shows, all one-hop person-person relationships also appear. The object sets *Spouse* and *Child* between the two *Person* object sets hold groups of objects judged by our match algorithm to be the same.

To make merge decisions, a user has, in addition to a D-Dupe view like the one in Figure 7, access to all the information about persons in *Family Tree* by clicking on a person's FamilySearch ID, and access to source document information including the page of interest and the entire book by clicking on a person's extraction-assigned ID. Once a decision is made, a user can alter the contents of the *Person* object sets and then click on a "go" button to request the ingest. In our example, a user would remove "Mary Ely (osmx275)" from the left *Person* object set and then request the ingest. The system would react by automatically directing the user to FamilySearch's merge page where "Mary Ely" and "Mary Eli" would be merged using FamilySearch's merge procedure and would then automatically ingest each of the extracted "Mary Ely"s.

3.2 Pipeline Implementation

The pipeline is coded in Java up to the point of information ingest, which is coded in Python using Selenium [24] to automate interaction with the FamilySearch web site and update *Family Tree*. The pipeline runs from beginning to end, and the code is being improved as we gain experience and encounter new edge cases. The given-name/male-probability list and the name-authority list have been curated and are used in the pipeline, but the place-authority list has not yet been created. The D-Dupe-like integration and deduplication tool is only in the proposal stage.

The ensemble of extraction engines, COMET, and the user interface for the pipeline management system are coded using Java, PHP, JavaScript, jQuery, CSS, and HTML5 and make use of a variety of off-the-shelf tools, including Soar, the LG Parser, and Stanford Core NLP packages. The extraction engines are all in their individual academic prototype stage. They all run, but considerable work will be required to tech-transfer them into tools usable by anyone besides

ourselves. COMET has been used by subjects in some experimental evaluations; they generally find it usable after a few minutes of training. We have only begun to build a management system that will control the processing of books through the pipeline.

3.3 Initial Field Tests

Ely [4]. To compare the effort between manually and automatically ingesting information, we updated *Family Tree* by hand according to the information in Figure 1. We filled in search forms with the genealogical data from the generated person information records (e.g. see Figure 2), identified matching *Family Tree* records, merged duplicates (if any), checked the matching records, and added to them source documentation and missing information. Of the 31 unique person information records, 28 matched exactly one *Family Tree* person record. The record for Mary Ely married to Gerard Lathrop matched two, as Figure 7 shows, and we merged them. Donald McKenzie's and Abigail Huntington Lathrop's person information records each matched three records that were themselves duplicates, and in both cases we merged the three records. We added highlighted source documents like the one in Figure 1 for all 31 matched tree records. Overall, we (1) replaced two primary names with more complete names (e.g. "Emma Sutherland Goble" in place of "Emma S. Goble"); (2) replaced six uncertain BMD (Birth/Marriage/Death) facts (e.g. "about 1831" or merely "deceased") with certain facts; (3) added two missing BMD facts, and (4) added eight supplementary facts such as married names or alternate spellings of names. All of this work, which could have been done fully automatically within seconds of compute time, took more than five hours of tedious typing, checking, clicking, and waiting for responses from the FamilySearch web site.

Kilbarchan [25]. In a fully automatic extraction run over the 143 pages of the Kilbarchan, Scotland, parish record, the ensemble created person information records like the one in Figure 2 for 8,539 individuals. The automatic extraction's F-score was judged to be near 95%. Our matching algorithm found that 38% of these individuals were already in *Family Tree*. In a sample of 150 person information records, we checked our match-scoring algorithm, and for those that matched correctly, we determined how much and what kind of information could be immediately added to the tree. For match scores of 8 or more, meaning roughly that the person in the Kilbarchan data and the person in *Family Tree* matched on eight separate items of information, our match algorithm correctly matched 100% of the time and correctly matched 64% of those with match scores between 5 and 7. Of those correctly matched, 20% had information in the Kilbarchan data that could be immediately added to the tree to improve the data, including adding or fixing first and last names, birth and marriage dates, and parent-child relationships.

Miller [26]. Similar to our Kilbarchan field test, in a fully automatic extraction run over the 396-page Miller Funeral Home Records from Greenville, Ohio, we extracted information for 12,226 individuals. The match rate of individuals already in *Family Tree* for the Miller records was lower than for the Kilbarchan

book—just over 10% compared to 38% for Kilbarchan. Of the 1,280 individuals our matching algorithm found, the Miller records provided information that could be automatically added to 57% of them—a complete name, full birth date, full death date, or names of an individual's spouse, parents, or children.

4 Concluding Remarks

The Fe6 ingest pipeline is fundamentally grounded in conceptual modeling: The principles of ontological modeling and ontological commitment facilitate the identification and extraction of individuals and their genealogical information from semi-structured text. The strong correspondence between forms and conceptual models provides coherent user views that ease the human check-and-correct of results produced by the ensemble of extraction engines. Inference rules written with respect to conceptual object and relationship predicates drive the semantic sanity checks and the inference of critical data that cannot be directly extracted. And human oversight of entity resolution via deduplication and record integration is likely best achieved by viewing a relevant graph of the entities and their relationships embedded in conceptually derived object sets.

The Fe6 pipeline can accelerate ingest into *Family Tree* while simultaneously maintaining FamilySearch-required oversight. With COMET we can guarantee human-level accuracy of extracted information. Depending on the outcome of automatically matching extracted data with the tree, information can either be automatically attached or, when human oversight is required for entity resolution, can be presented in a generated view of the information that facilitates a quick and accurate resolution. As a rough estimation of expected acceleration, it took about 5 hours to ingest the genealogical information from the Ely page in Figure 1 manually into *Family Tree*. Using COMET, it took less than 30 minutes to annotate the information from scratch and less than 10 minutes when the form records were prepopulated with data by the ensemble of extraction engines. Except for assessing duplicates, the ingest can be fully automatic. Thus, we can estimate a potential 10-fold speed-up without the involvement of the ensemble of extraction engines and a 30-fold speed-up with them.

References

1. FamilySearch. `http://familysearch.org/`.
2. D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng, and R.D. Smith. Conceptual-model-based data extraction from multiple-record web pages. *Data & Knowledge Engineering*, 31(3):227–251, 1999.
3. D.W. Embley, S.W. Liddle, and D.W. Lonsdale. Conceptual modeling foundations for a web of knowledge. In D.W. Embley and B. Thalheim, editors, *Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges*, chapter 15, pages 477–516. Springer, Heidelberg, Germany, 2011.
4. G.B. Vanderpoel, editor. *The Ely Ancestry: Lineage of RICHARD ELY of Plymouth, England, who came to Boston, Mass., about 1655 & settled at Lyme, Conn., in 1660.* The Calumet Press, New York, New York, 1902.

5. D.W. Embley, B.D. Kurtz, and S.N. Woodfield. *Object-oriented Systems Analysis: A Model-Driven Approach*. Prentice Hall, Englewood Cliffs, New Jersey, 1992.
6. D.W. Embley. *Object Database Development: Concepts and Principles*. Addison-Wesley, Reading, Massachusetts, 1998.
7. B. Thalheim. *Entity-Relationship Modeling: Foundations of Database Technology*. Springer, Berlin, Germany, 2000.
8. A. Olivé. *Conceptual Modeling of Information Systems*. Springer, Berlin, Germany, 2007.
9. D.W. Embley and B. Thalheim, editors. *Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges*. Springer, Heidelberg, Germany, 2011.
10. D. Dori. *Model-based Systems Engineering with OPM and SysML*. Springer, 2015.
11. ER web site. `http://conceptualmodeling.org/`.
12. J. Park. FROntIER: A framework for extracting and organizing biographical facts in historical documents. Master's thesis, Brigham Young University, Provo, Utah, 2015.
13. C. Tao, D.W. Embley, and S.W. Liddle. FOCIH: Form-based ontology creation and information harvesting. In *Proceedings of the 28th International Conference on Conceptual Modeling (ER2009)*, pages 346–359, Gramado, Brazil, November 2009.
14. T.W. Kim. A green form-based information extraction system for historical documents. Master's thesis, Brigham Young University, Provo, Utah, 2017.
15. G. Nagy. Estimation, learning, and adaptation: Systems that improve with use. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, Hiroshima, Japan, November 2012.
16. T.L. Packer. *Scalable Detection and Extraction of Data in Lists in OCRed Text for Ontology Population Using Semi-Supervised and Unsupervised Active Wrapper Induction*. PhD thesis, Brigham Young University, 2014.
17. P. Lindes. OntoSoar: Using language to find genealogy facts. Master's thesis, Brigham Young University, Provo, Utah, 2014.
18. J.E. Laird. *The Soar Cognitive Architecture*. The MIT Press, Cambridge, Massachusetts, 2012.
19. G. Nagy, DDA: Decision Directed Adaptation. personal communication.
20. S.W. Liddle, D.W. Embley, and S.N. Woodfield. Cardinality constraints in semantic data models. *Data & Knowledge Engineering*, 11(3):235–270, 1993.
21. S.N. Woodfield, D.W. Lonsdale, S.W. Liddle, T.W. Kim, D.W. Embley, and C. Almquist. Pragmatic quality assessment for automatically extracted data. In *Proceedings of ER 2016*, volume LNCS 9974, pages 212–220, Gifu, Japan, November 2016.
22. GEDCOM X. `http://www.gedcomx.org/`.
23. H. Kang, L. Getoor, B. Shneiderman, M. Bilgic, and L. Licamele. Interactive entity resolution in relational data: A visual analytic tool and its evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 14(5), September/October 2008.
24. SeleniumHQ: Browser automation. `http://www.seleniumhq.org/`.
25. F.J. Grant, editor. *Index to The Register of Marriages and Baptisms in the PARISH OF KILBARCHAN, 1649–1772*. J. Skinner & Company, LTD, Edinburgh, Scotland, 1912.
26. Miller Funeral Home Records, 1917 – 1950, Greenville, Ohio, 1990.