A Green Form-Based Information Extraction System

for Historical Documents

Tae Woo Kim

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

David W. Embley, Chair
Stephen W. Liddle
Michael Jones

Department of Computer Science

Brigham Young University

ABSTRACT

A Green Form-Based Information Extraction System
for Historical Documents

Tae Woo Kim
Department of Computer Science, BYU
Master of Science

Many historical documents are rich in genealogical facts. Extracting these facts by hand is tedious and almost impossible considering the hundreds of thousands of genealogically rich family-history books currently scanned and online. As one approach for helping to make the extraction feasible, we propose GreenFIE—a "Green" Form-based Information-Extraction tool which is "green" in the sense that it improves with use toward the goal of minimizing the cost of human labor while maintaining high extraction accuracy. Given a page in a historical document, the user's task is to fill out given forms with all facts on a page in a document called for by the forms (e.g. to collect the birth and death information, marriage information, and parent-child relationships for each person on the page). GreenFIE has a repository of extraction patterns that it applies to fill in forms. A user checks the correctness of GreenFIE's form filling, adds any missed facts, and fixes any mistakes. GreenFIE learns based on user feedback, adding new extraction rules to its repository. Ideally, GreenFIE improves as it proceeds so that it does most of the work, leaving little for the user to do other than confirm that its extraction is correct. We evaluate how well GreenFIE performs on family history books in terms of "greenness"—how much human labor diminishes during form filling, while simultaneously maintaining high accuracy.

Keywords: green systems, self-improving systems, data extraction, regular-expression generation.

ACKNOWLEDGEMENTS

First of all, I would like to thank my committee chair, David Embley, for his patience and guidance which made this thesis possible. I owe it all to him. He not only taught me how to be a better student, but also how to be a better person. Thank you, Dr. Embley, for your great example.

I am grateful to my committee members, Stephen Liddle and Michael Jones. I am also grateful to Christophe Gerard Giraud-Carrier, Claire DeWitt, and Jen Bonnett for your patience, understanding, and help.

A very special gratitude goes out to the following members of BYU Data Extraction Group: Joseph Park, Peter Lindes, Deryle Lonsdale, and Scott Woodfield for their feedback from the proposal to the thesis.

I'm also grateful to friends who helped me along the way. Last but not the least, I would like to express my gratitude for family: my wife Kelly, my parents, and my other family members who have supported me in every possible way.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

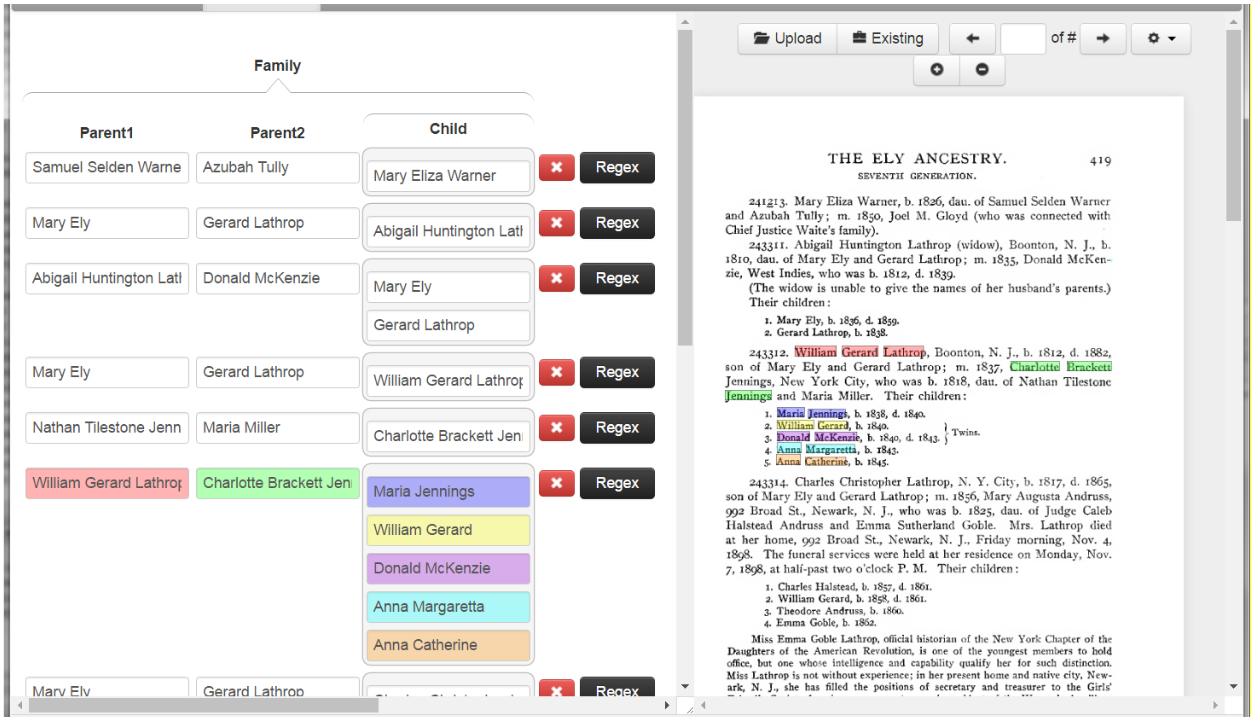
1. INTRODUCTION

The Church of Jesus Christ of Latter-day Saints has scanned and OCRed several hundred thousand family-history books all containing rich genealogical information about people and events in their lives. These books are keyword searchable, but not semantically searchable over events and family relationships. To make them semantically searchable the information needs to be extracted and indexed. However, both the volume of facts and the lack of structure of these facts make it impractical for the information to be extracted manually.

In responding to the demand, several information extraction tools are being developed. In this thesis we propose, prototype, and evaluate an information-extraction tool based on a form-filling paradigm. The tool "watches" users fill in forms, copying relevant information from OCRed historical documents into form fields, and thereby learns extraction rules and executes them, taking on as much of the extraction work as it can. The tool is "green" [Nagy12]—one that improves with use toward the goal of minimizing the cost of human labor while maintaining high extraction accuracy. We call our tool GreenFIE (**Green F**orm-based **I**nformation **E**xtraction).

Figure 1.1 shows the user interface for GreenFIE. A form to fill in is on the left and a page from an OCRed document is adjacent to it on the right. Users work page by page, finding all the information of interest on the page as specified on the form and then moving on to the next page. When the system initially loads a page for a form, GreenFIE fills out the form as best it can using the information-extraction rules in its repository. Initially, the repository may be empty, but GreenFIE populates it with extraction rules as it observes users fill in form records. Once a form record is filled in (e.g. the highlighted record in Figure 1.1) the user can click the

"Regex" button at the end of each record. Then, GreenFIE learns the extraction pattern from the annotated text—learns which record field is filled with which text along with its surrounding and delimiting text—and creates a regular expression that matches the pattern. GreenFIE then executes the regular-expression extraction rule to find more records that have the same pattern (if any).



Family

Parent1	Parent2	Child
Samuel Selden Warne	Azubah Tully	Mary Eliza Warner
Mary Ely	Gerard Lathrop	Abigail Huntington Latl
Abigail Huntington Latl	Donald McKenzie	Mary Ely Gerard Lathrop
Mary Ely	Gerard Lathrop	William Gerard Lathroj
Nathan Tilestone Jenn	Maria Miller	Charlotte Brackett Jen
William Gerard Lathroj	Charlotte Brackett Jen	Maria Jennings William Gerard Donald McKenzie Anna Margaretta Anna Catherine
Mary Ely	Gerard Lathrop	

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. 1825, 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'

FIGURE 1.1. GREENFIE USER INTERFACE.

We present the details of how GreenFIE operates as follows: Chapter 2 reviews related literature. Chapter 3 provides a system-level overview of GreenFIE (1) as a stand-alone system that could be used in any form-filling workflow and (2) as the particular tool we built for semi-automatic form-filling of genealogical information extracted from OCRed, historical, family-history documents. The chapter also describes in greater detail GreenFIE's user interface within the historical document processing system. The central processing component of GreenFIE is its ability to generate and generalize regular-expression extraction rules as explained in Chapter 4.

Chapter 5 documents the experimental work we have done to test GreenFIE's effectiveness. Chapter 6 discusses the results of the experiment, points to lessons learned, and makes recommendations. We draw conclusions and point to future work in Chapter 7.

2. RELATED WORK

We are not aware of any other system like GreenFIE—a system designed to allow users to annotate OCRed historical documents by filling in forms while at the same time synergistically working with users by learning from their annotation work and attempting to shift the burden of annotation as much as possible to itself. We are, however, aware of work on various aspects of GreenFIE: green systems that learn as users work, interaction reuse, information extraction, regular-expression generation, annotation tools and other supporting tools.

Green Systems. Our research is about making tools that improve with use. Researchers have long been interested in these kinds of tools (e.g. see [BloN12]). Nagy is a strong proponent of these types of tools [Nagy12b]. In his recent keynote address at the Family History Technology Workshop [Nagy12a], he called a pattern recognition system "green" if it observes human effort to approve or correct the output of a learning system and then improves itself. We designate our tool as being "green" in accordance with this idea.

Reusing Interactions. The fundamental idea of green systems is to reuse human interactions with a system to improve the performance of the system itself. Day, et al. argue for maximal reuse of every kernel of knowledge available at each processing step [DAHK97]. In their text annotation system for tagging NLP training data, they leverage the combined efforts of machine and user to produce domain specific annotation rules that can be used to annotate similar texts automatically. GreenFIE has some similarities with these efforts in the sense that it is about annotating text and leverages both machine and user to produce extraction rules that can be used to further annotate similar text automatically. Before he named his self-improvement systems “green,” Nagy had been developing systems that reuse interactions that demonstrate self-learning [ZouN04] and that through user interactions incrementally improve the quality of collections of badly recognized documents [LopN09].

Information Extraction. Our research falls in the general area of information extraction. Information extraction dates back to the early days of Natural Language Processing. Its purpose is to recognize named entities such as people or organizations from natural language. It has been growing ever since and has become a field of its own [TuAC06]. In 2008, Sarawagi published a lengthy treatise summarizing the work and describing the various different information extraction problems that interest researchers [Sara08]. At BYU, the Data Extraction Group has been active for two decades (e.g. see [EmLL11] which summarizes much of their work).

Semi-supervised Information Extraction. A work that somewhat parallels our own is OLERA [ChaK04]. Users interact with OLERA to generate extraction rules for their targets of interest. Instead of labeling training pages, users enclose an information block of interest and then specify relevant information slots for each field in the record. OLERA then generates a rule to extract the relevant information.

Regular-Expression Generation. GreenFIE generates and generalizes regular expressions from a given example. Constructing regular expressions from examples has long been an interesting research topic. [Blac00], as an example, references many of these efforts and adds its own contribution—an empirically evaluated visual interface that allows users to see and modify the effects of the supplied examples, several of which are needed to generate a regular expression. GreenFIE, however, generates regular expressions from a single example and for records in which field text varies while field identifiers and delimiting text are fixed. ListReader

[Pack14] generates these types of record-based regular-expression extraction rules, but does so by discovering re-occurring text-snippet patterns rather than from a single example.

Existing Annotation and Supporting Tools. The main tools on top of which GreenFIE is implemented are the Annotator [DEG14], FROntIER [Park15], and OntoES [EmLL11]. We used the Annotator to build the GreenFIE interface, and FROntIER and OntoES as the guiding pathway to creating the extraction rules that GreenFIE generates.

3. SYSTEM OVERVIEW

3.1 Architecture

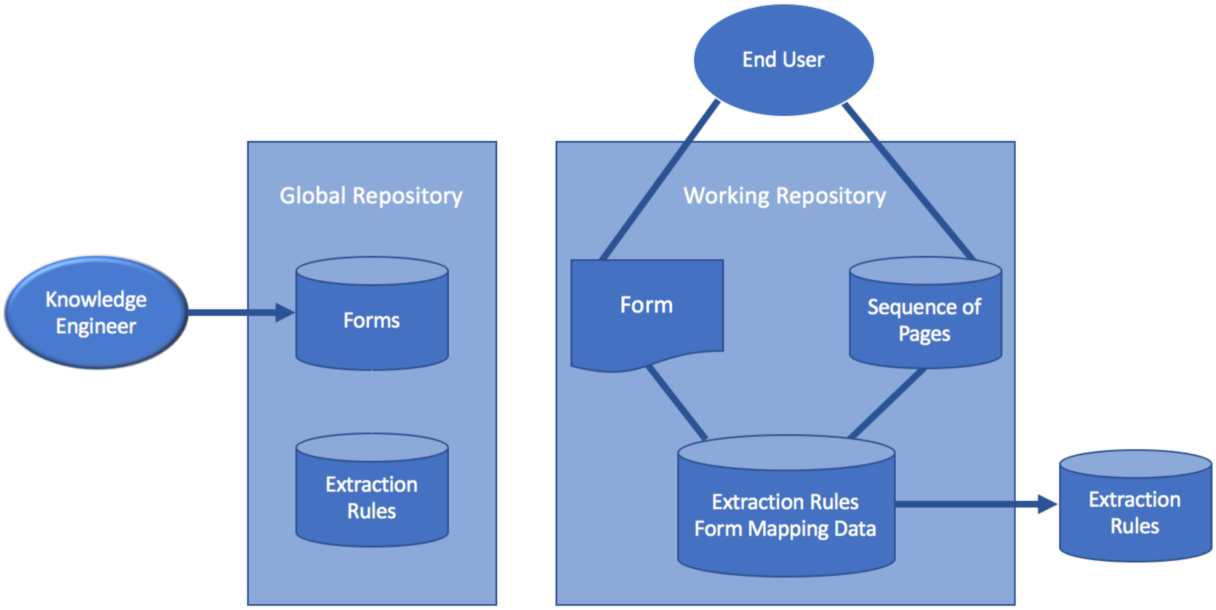


FIGURE 3.1. GREENFIE SYSTEM ARCHITECTURE.

In general GreenFIE can operate in a work environment in which users extract specific information from pages in documents as called for by forms into which the users copy the extracted information into form fields. Figure 3.1 shows the general architecture of a GreenFIE system. There are two types of users: a knowledge engineer and an end user. A knowledge engineer creates forms for the end user to work with to specify the information to be extracted. End users manage the extraction process by checking information filled into the forms by the extraction rules, correcting mistakes, and adding missing information. Results find their way into a data repository—a populated ontology (i.e. a populated database-like schema).GreenFIE generates extraction rules as it “watches” users perform these tasks. It stores rules in its working repository and executes them against as-yet-to-be-processed pages in the document. It also stores these rules in a global repository for possible use in new documents.

The global repository stores all the extraction rules GreenFIE has created as the system processes a multitude of documents. The working repository stores the extraction rules for the document GreenFIE is currently processing. When the system starts a new document, it runs the extraction rules in the global repository and copies rules that find matches in the new document into the working repository. For each new page, GreenFIE applies the extraction rules from the working repository to prime forms for the user. As mentioned, GreenFIE also adds new rules to the working repository as a page is processed.

3.2 Information Extraction from Historical Documents

Although GreenFIE can run as an assistant for any application in which the task is to fill in forms from information in a given document collection, we have only implemented for this thesis its use in extracting genealogical information from a collection of family-history books. As a result, the forms and ontological schema for GreenFIE are given and fixed in advance, and the knowledge-worker part of the system architecture in Figure 3.1 is not part of the prototype implementation.

The given forms are named *Person*, *Couple*, and *Family*. The *Person* form includes a name for a person together with birth and death information. The *Couple* form is for marriages, including the bride and groom and the date and place of a marriage. And the *Family* form (see Figure 1.1) is for parent-child relationships—the parents and a list of their children. The ontological schema is comprised of an integration of the fields in these forms.

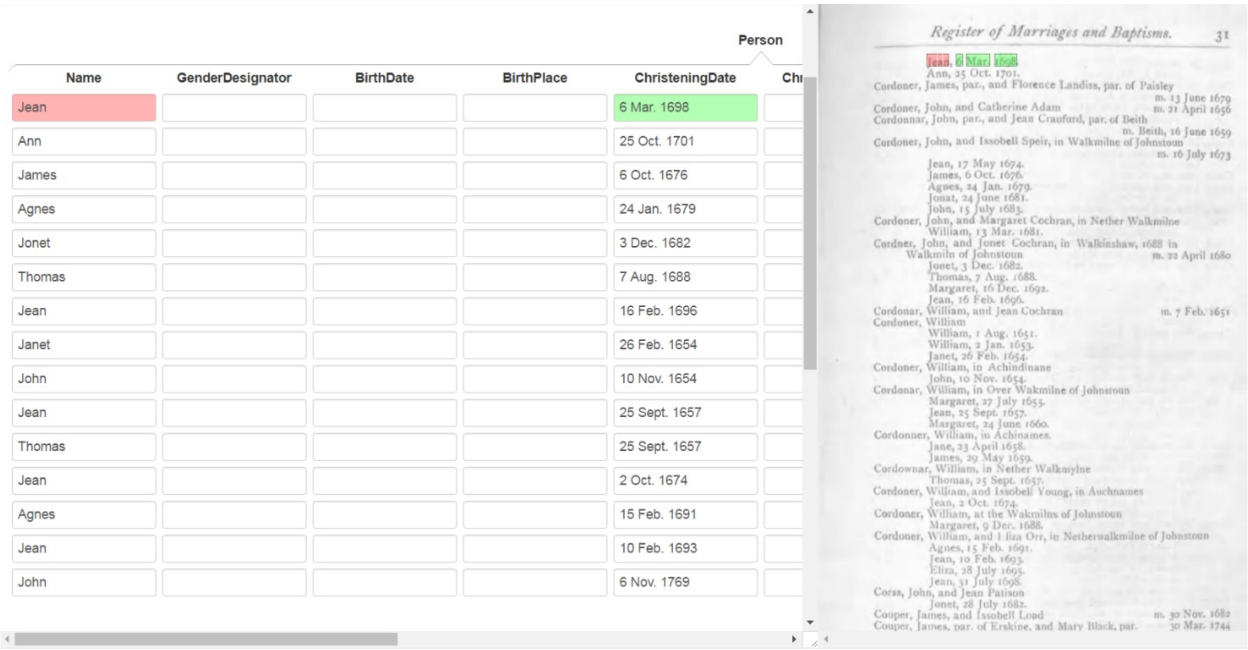
3.3 GreenFIE Interface

The interface for a GreenFIE user has been built on top of an existing annotation tool [DEG14]. Figure 1.1 shows the interface with a collection of populated Family-form records on the left populated from a page in a family-history book on the right.

A page in the interface can be either a PDF document or a plain text file, but is usually both: a PDF document superimposed over the hidden OCR in a plain text file. Thus, a user works with what appears to be an image of the original page in a historical document, but is actually working with the plain text file that is aligned with the document image. Once a user chooses a form and a document to work with, GreenFIE fills out the form for the page the best it can using rules it has in its working repository (if any). Hovering over a record causes the annotator system to highlight each field in the record and the corresponding text in the page. As Figure 1.1 shows, each field is highlighted with a different color for easy recognition. The user then examines the record and goes on to the next if it is correct. If the record is incorrect, the user fixes the error using the manipulation operations that allow field-instance data to be deleted, inserted, or modified. If the whole record is wrong, the user can click on the red-x button to delete it. If an entire record is missing, the user can add an empty record and fill it in. When a record has either been added or corrected, the user can click on the black "Regex" button, which causes GreenFIE to generate an extraction rule that would have correctly extracted the record. GreenFIE then generalizes the rule, executes it over the page to extract any additional records that satisfy the rule, and stores the rule in the working repository to be used for subsequent pages.

4. EXTRACTION RULE GENERATION AND EXECUTION

4.1 Rule Creation Basics



Person

Name	GenderDesignator	BirthDate	BirthPlace	ChristeningDate	Chr
Jean				6 Mar. 1698	
Ann				25 Oct. 1701	
James				6 Oct. 1676	
Agnes				24 Jan. 1679	
Jonet				3 Dec. 1682	
Thomas				7 Aug. 1688	
Jean				16 Feb. 1696	
Janet				26 Feb. 1654	
John				10 Nov. 1654	
Jean				25 Sept. 1657	
Thomas				25 Sept. 1657	
Jean				2 Oct. 1674	
Agnes				15 Feb. 1691	
Jean				10 Feb. 1693	
John				6 Nov. 1769	

Register of Marriages and Baptisms. 31

Jean, 6 Mar. 1698.
Ann, 25 Oct. 1701.
Cordoner, James, par., and Florence Landiss, par. of Paisley
m. 13 June 1679
Cordoner, John, and Catherine Adam m. 21 April 1656
Cordonnar, John, par., and Jean Craufurd, par. of Beith
m. Beith, 16 June 1659
Cordoner, John, and Issobell Speir, in Walkmilne of Johnstoun
m. 16 July 1673
Jean, 17 May 1674.
James, 6 Oct. 1676.
Agnes, 24 Jan. 1679.
Jonat, 24 June 1681.
John, 15 July 1683.
Cordoner, John, and Margaret Cochran, in Nether Walkmilne
William, 13 Mar. 1681.
Cordner, John, and Jonet Cochran, in Walkinshaw, 1688 in
Walkmiln of Johnstoun m. 22 April 1680
Jonet, 3 Dec. 1682.
Thomas, 7 Aug. 1688.
Margaret, 16 Dec. 1692.
Jean, 16 Feb. 1696.
Cordonar, William, and Jean Cochran m. 7 Feb. 1651
Cordoner, William
William, 1 Aug. 1651.
William, 2 Jan. 1653.
Janet, 26 Feb. 1654.
Cordoner, William, in Achindinane
John, 10 Nov. 1654.
Cordonar, William, in Over Wakmilne of Johnstoun
Margaret, 27 July 1655.
Jean, 25 Sept. 1657.
Margaret, 24 June 1660.
Cordonner, William, in Achinames.
Jane, 23 April 1658.
James, 29 May 1659.
Cordownar, William, in Nether Walkmylne
Thomas, 25 Sept. 1657.
Cordoner, William, and Issobell Young, in Auchnames
Jean, 2 Oct. 1674.
Cordoner, William, at the Wakmilns of Johnstoun
Margaret, 9 Dec. 1688.
Cordoner, William, and Iliza Orr, in Netherwalkmilne of Johnstoun
Agnes, 15 Feb. 1691.
Jean, 10 Feb. 1693.
Eliza, 28 July 1695.
Jean, 31 July 1698.
Corss, John, and Jean Patison
Jonet, 28 July 1682.
Couper, James, and Issobell Load m. 30 Nov. 1682
Couper, James, par. of Erskine, and Mary Black, par. 30 Mar. 1744

FIGURE 4.1. DISPLAY OF RESULTS AFTER GREENFIE RULE GENERATION AND EXECUTION.

Extraction rules for the GreenFIE prototype are regular-expression information-extraction rules. Figure 4.1 shows the GreenFIE interface with some record fields on the left filled in for Page 31 of a transcript of the Kilbarchan Parish Record [Gr1912] on the right. In Figure 4.1, a user has entered the highlighted information into the first record and clicked on the *Regex* button (hidden in Figure 4.1, but accessible as a user moves to the right with the slider-bar at the bottom of the screen). As a result GreenFIE created and filled in the following 14 records.

For the highlighted text in Figure 4.1 (“Jean, 6 Mar. 1698.”) a regular expression matching the information is:

```
\n([A-Z]{1}[a-z]{3}),\s(\d{1}\s[A-Z]{1}[a-z]{2}\.\s\d{4})\.
```

When a regular expression matches a text string, the part of the matched string that corresponds to a parenthesized sub-regular expression is captured and associated with a capture-group number—numbered by counting opening parentheses reading left to right, omitting those specifically marked as non-capture groups by `(?: …)`, if any. GreenFIE then associates the capture-group number with the field name in the form, which after some internal processing causes the captured text to be displayed in the named form field. Capture-group 1, `([A-Z]{1}[a-z]{3})`, causes "Jean" to be placed in the Name field in the record, and Capture-group 2, `(\d{1}\s[A-Z]{1}[a-z]{2}\.\s\d{4})`, causes "6 Mar. 1698" to be placed in the ChristeningDate field.

When the regular expression above for the "Jean" record is applied to the text of the full page, only two more records are captured. To be more effective, GreenFIE generalizes regular-expression recognizers, being careful not to over-generalize. If, for example, GreenFIE increases the span of quantifiers of `[a-z]` and `\d` by the ceiling of plus and minus 50%, the generalized rule becomes

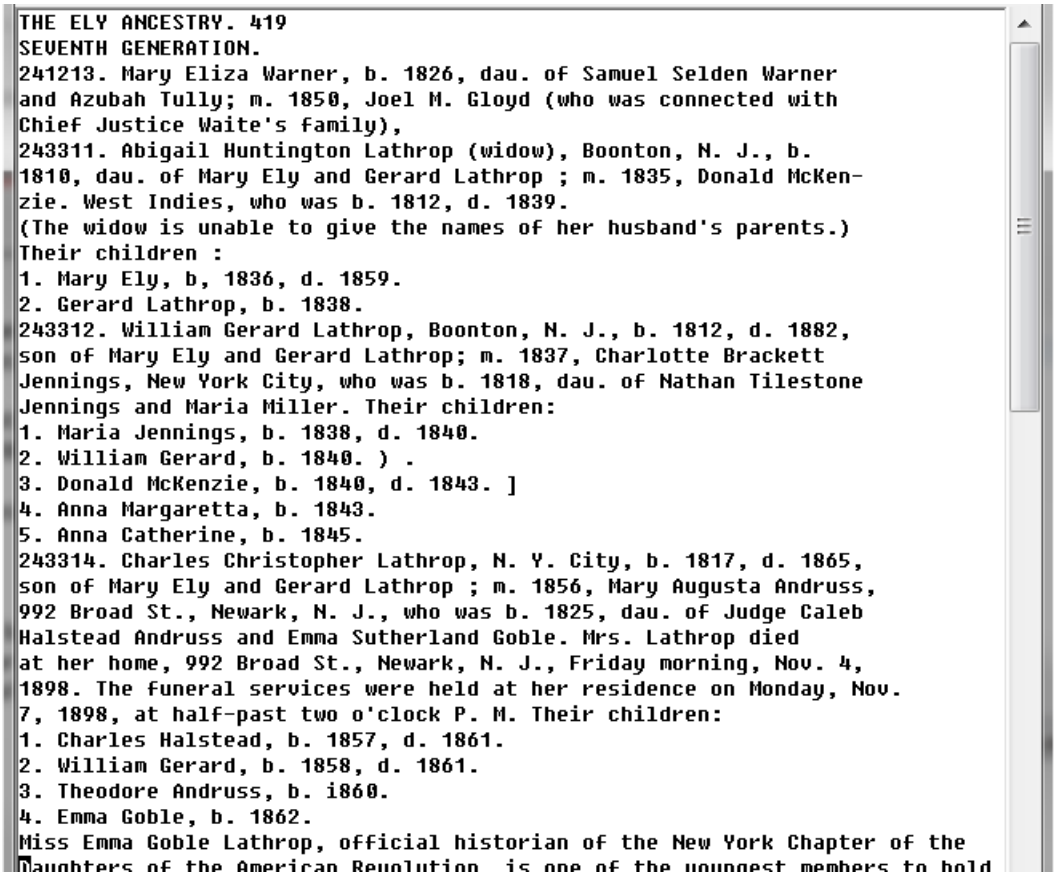
```
\n([A-Z]{1}[a-z]{1,5}),\s(\d{0,2}\s[A-Z]{1}[a-z]{1,3}\.\s\d{2,6})\.
```

which extracts the additional 14 records in Figure 4.1.

Because GreenFIE knows the type of each form field, it can take advantage of this knowledge to better generalize capture-group expressions. Generalizing given names and day-month-year dates and substituting these for the capture-group expressions yields

```
\n([A-Z][a-z]+),\s(\d{1,2}\s[JFMASOND][a-z]{2,4}[.]?\s\d{4})\.
```

which extracts all 32 of the christening records on Page 31.



```
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 McKen-
zie. 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. ) .
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. 1825, 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. i860.
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
```

FIGURE 4.2. SAMPLE OCRED TEXT.

Figure 4.2 shows the underlying OCRed text of the document image in Figure 1.1 over which GreenFIE constructs and executes its regular expressions. Note that in the OCR all white space to the left of a line has been removed but that each line remains intact so that a newline character "\n" appears between every line. Note also some of the OCR errors: The brace and "Twins" designator for William Gerard and Donald McKenzie has been replaced by noise. More troublesome for GreenFIE, however, are some of the more subtle OCR errors—"i860" in place of "1860" as Theodore Andruss's birth year and the comma following Mary Ely's birth-year designator "b," in opposition to all other birth-year designators "b." which terminate with a period. GreenFIE has to function well in spite of all these errors. To help, we supply it with knowledge of common OCR error confusions (e.g. "i" and "1" and "," and ".").

Summarizing, GreenFIE's approach to extraction-rule generation is to create a regular expression with capture groups that will recognize the given record with its filled-in fields and generalize it, enabling it to recognize similar patterns and extract the information in these patterns into appropriate form fields. GreenFIE generalizes regular expressions for form fields using known patterns for field types when these are available and otherwise uses its basic generalization for field values. As we will discuss in the remaining sections in this chapter, this basic generalization is a bit more sophisticated than the simple ±50% we exemplified here in this motivational section. Type-dependent generalizations are also more complex than indicated here. Sets of regular expressions for common forms of names, dates, and places are needed, along with a mechanism to select appropriate expressions for the particular case under consideration. GreenFIE also includes knowledge of common OCR errors in its generalizations for fields and text that appears before, between, and after field text. Text surrounding the field values is critical to generating good recognizers, and we will explain how GreenFIE avoids accepting and generalizing too much of the field-surrounding text, making its extraction rules more specific than necessary while at the same time not over-generalizing which leads to extracting incorrect information. We will also show in Section 4.3 that GreenFIE has an interesting way of generalizing for lists like the child lists in the Ely page in Figure 1.1 or the Kilbarchan page in Figure 4.1. Knowing about lists, GreenFIE generalizes the number of list elements to an expected maximum size, so that if the list for which the regular-expression rule is being generated has only a couple of list elements, the generalized rule is able to extract any number of list elements up to the expected maximum.

4.2 Rule Creation for Simple Records

A record that has only single-entry fields is *simple*. For purposes of rule creation, we also consider form records that have multiple-entry fields but only one multiple instance to be *simple*. Records in the Person form (see Figure 4.1) are always simple since the form consists only of single-entry fields. In Figure 1.1 the highlighted child list in the Family form is complex. The first record, however, has only one child, Mary Eliza Warner, and for rule creation we consider it to be simple.

For a simple record, GreenFIE generates an extraction rule of the form:

<before>(<field>)<right><skip><left>(<field>)<right><skip> … <left>(<field>)<after>

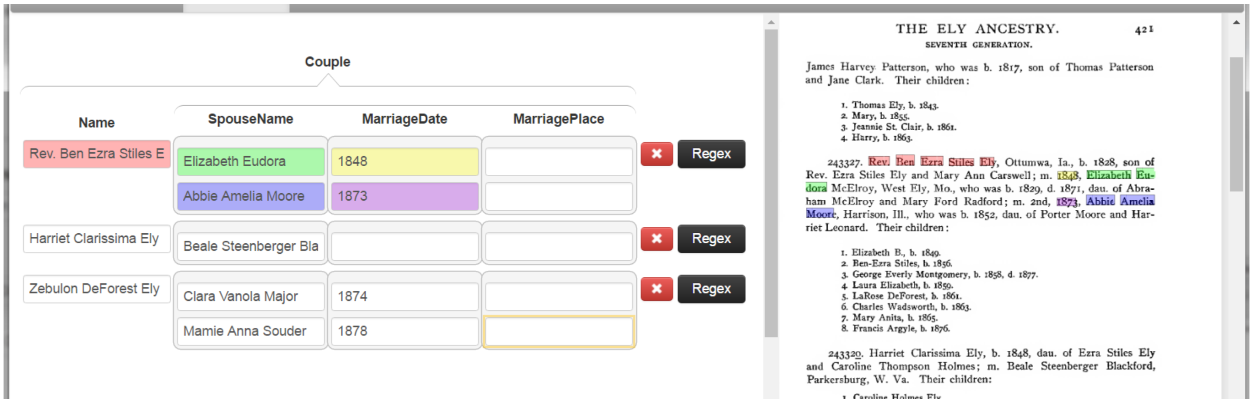
For <before>, GreenFIE finds the text snippet immediately preceding the text of the first field up to and including the first space in the preceding text that comes before some non-white-space text or up to and including a newline character "`\n`", whichever comes first. It then replaces spaces with "`\s`" and protects special characters such as a question mark by using an escaping backslash "`\?`", and it accommodates the common OCR error of mistaking a comma for a period and vice versa by placing both symbols within character-class brackets when either of the symbols is encountered in the text snippet. Further, if an all-digit string is encountered in the text snippet, GreenFIE generalizes it to "`\d{`*n*`,`*m*`}` where *n* and *m* are set to the ceiling of ±10% of the actual length. GreenFIE treats <after> similarly for the text immediately following the text of the last field except that it does not include the newline character "`\n`", if any. GreenFIE also treats <left> and <right> respectively in the same way it treats <before> and <after>. If <left> and <right> overlap between successive fields, however, GreenFIE uses the text snippet between the fields with appropriate replacements as a delimiter between the fields.

As an example, the OCRed text from the third line in Figure 4.2 is “241213. Mary Eliza Warner, b. 1826, dau. of Samuel Selden Warner” where we annotate “Mary Eliza Warner” as a Name and “1826” as BirthDate. Here, <before> is “`\n\d{5,7}[.,]\s`”, and <after> is “`[.,]\s`”. Between the name and birth year, <right> and <left> overlap and thus the regular-expression text becomes the delimiter “`[.,]\sb[.,]\s`”.

The <skip> in the extraction-rule pattern does not come into play in this example, but would for “Mary Eliza Warner, who was born in 1826”. GreenFIE encodes a <skip> as “`[\s\S]{n,m}?`” which skips over at least *n* and at most *m* characters including white-space characters. The “?” at the end makes the skipping “lazy” so that as soon as the regular-expression pattern following the <skip> is encountered, the regular-expression processor stops skipping over characters. GreenFIE computes *n* and *m* by counting the characters between <right> and <left> and computing *n* as the floor of *p*% less than the count (but not less than 0) and *m* as the ceiling of *p*% greater than the count. The percentage *p* is fixed empirically, and GreenFIE currently has it set at 300% when the length is less than or equal to 15, 150% when the length is greater than 15 but less than or equal to 30, 75% when the length is greater than 30 but less than or equal to 60, and 37% when the length is greater than 60. Thus, for the 12 “who was born” characters in our example between <right> and <left>, *n* = 0 and *m* = 36.

4.3 Rule Creation for Complex Records

A *complex* record contains at least one multiple-entry field that has more than one multiple instance. For example, records in the Family form (see Figure 1.1) may have child lists with several children, and there are two spouses in two of the records of the Couple form in Figure 4.2.



Name	SpouseName	MarriageDate	MarriagePlace
Rev. Ben Ezra Stiles E	Elizabeth Eudora	1848	
	Abbie Amelia Moore	1873	
Harriet Clarissima Ely	Beale Steenberger Bla		
Zebulon DeForest Ely	Clara Vanola Major	1874	
	Mamie Anna Souder	1878	

THE ELY ANCESTRY. 421

SEVENTH GENERATION.

James Harvey Patterson, who was b. 1817, son of Thomas Patterson and Jane Clark. Their children:

1. Thomas Ely, b. 1843.
2. Mary, b. 1855.
3. Jeannie St. Clair, b. 1861.
4. Harry, b. 1863.

243327. Rev. Ben Ezra Stiles Ely, Ottumwa, Ia., b. 1828, son of Rev. Ezra Stiles Ely and Mary Ann Carswell; m. 1848, Elizabeth Eudora McElroy, West Ely, Mo., who was b. 1829, d. 1871, dau. of Abraham McElroy and Mary Ford Radford; m. 2nd, 1873, Abbie Amelia Moore, Harrison, Ill., who was b. 1852, dau. of Porter Moore and Harriet Leonard. Their children:

1. Elizabeth B., b. 1849.
2. Ben-Ezra Stiles, b. 1856.
3. George Everly Montgomery, b. 1858, d. 1877.
4. Laura Elizabeth, b. 1859.
5. LaRose DeForest, b. 1861.
6. Charles Wadsworth, b. 1863.
7. Mary Anita, b. 1865.
8. Francis Argyle, b. 1876.

243329. Harriet Clarissima Ely, b. 1848, dau. of Ezra Stiles Ely and Caroline Thompson Holmes; m. Beale Steenberger Blackford, Parkersburg, W. Va. Their children:

1. Caroline Holmes Ely

FIGURE 4.3. COUPLE FORM WITH EXTRACTED INFORMATION.

The fundamental pattern for regular expressions for complex records is

<base><skip>(<anchor><list-fields><skip>)$^{n\text{-}1}$<anchor><list-fields>

The <base> component is for the non-repeating part of the record. In the Family form (see Figure 1.1) <base> is for the parents, and in the Couple form (see Figure 4.3) the name of the person who may have multiple spouses is the <base>. The first <skip> skips over commentary text between the <base> and the beginning of the list. The list component is an <anchor>-<list-fields> pair repeated *n*-1 times with a <skip> between list elements. An <anchor> marks the beginning of a list component, for example, a new line for each child in a Kilbarchan family (see the document in Figure 4.1) or a child number for each child in an Ely family (see Figure 1.1). The fields in a list component, <list-fields>, identify the fields to be extracted for the list—just the child's name for the Family form and for the Couple form the spouse name, marriage date, and marriage place. The *n* designates the expected maximum size of the list. The *n* in the fundamental pattern varies and is 1 for the first list element, 2 for the second, 3 for the third and so forth. The maximum value for *n* is set a priori based on application knowledge. After perusing the Ely and Kilbarchan lists, we set *n* = 5 for the Couple form and *n* = 12 for the Family form.

Interestingly, both <base> and <list-fields> expressions expand to be of the same form as the expression for simple records except that there is no <before> for the <list-fields>—it having been replaced by the <anchor>. Thus, for example, the <base> expression generated for a Family-form record for the highlighted annotation in Figure 1.1 is

```
\n\d{5,7}[.,]\s
([A-Z][a-z]+(?:\s[A-Z](?:[c][A-Z][a-z]+|[a-z]+))?(?:\s[A-Z][a-z]+)?)
[.,]\s[\s\S]{45,99}?\d{4}[.,]\s
([A-Z][a-z]+(?:\s[A-Z](?:[c][A-Z][a-z]+|[a-z]+))?(?:\s[A-Z][a-z]+)?)
[.,]\s
```

and the <list-fields> expression is

```
([A-Z][a-z]+(?:\s[A-Z](?:[c][A-Z][a-z]+|[a-z]+))?(?:\s[A-Z][a-z]+)?)
[.,]\s
```

where the name capture-group expressions all recognize Ely-like names consisting of one, two, or three capitalized words.

The <anchor> at the left of <list-fields> is a text snippet and is obtained in the same way as is <before> for the simple-record extraction rule. Then, however, GreenFIE checks to see if the text (if any) is one of the known <anchor> texts. Known <anchor> texts are ways of numbering list elements such as "1., 2., …" or "1st, 2nd, …", or "i., ii., iii., iv., …" and so forth. If GreenFIE recognizes the <anchor> text, it will use the numbering scheme to generate regular-expression text for the first, second, third, and so on up to the *n*th <anchor>. In our prototype implementation of GreenFIE, we predefined only the <anchor>s that appear in our development test set, namely "`\n1\.\s`", "`\n2\.\s`", …, "`\n12\.\s`" which is the style used to list children in Ely pages (see the document page in Figure 1.1) and "`\n`" which is the style used to list children in Kilbarchan pages—i.e. each child starts on a new line (see the document page in Figure 4.1). The capture groups for obtaining field values are set in the same way as they are for the simple-record expression with one exception. Since only the information in the last <list-fields> is to be captured, GreenFIE marks all other would-be capturing groups as non-capturing groups by changing the opening parenthesis "`(`" to "`(?:`". As an example, GreenFIE generates the regular-expression extraction rule in Figure 4.4 for the third child highlighted in the Family form in Figure 1.1.

```
\n\d{5,7}[.,]\s
 ([A-Z][a-z]+(?:\s[A-Z](?:[c][A-Z][a-z]+|[a-z]+))?(?:\s[A-Z][a-z]+)?)
 [.,]\s[\s\S]{45,99}?\d{4}[.,]\s
 ([A-Z][a-z]+(?:\s[A-Z](?:[c][A-Z][a-z]+|[a-z]+))?(?:\s[A-Z][a-z]+)?)
 [.,]\s[\s\S]{62,136}?
\n1[.,]\s(?:[A-Z][a-z]+(?:\s[A-Z](?:[c][A-Z][a-z]+|[a-z]+))?(?:\s[A-
Z][a-z]+)?)[.,]\s
 [\s\S]{0,43}?
\n2[.,]\s(?:[A-Z][a-z]+(?:\s[A-Z](?:[c][A-Z][a-z]+|[a-z]+))?(?:\s[A-
Z][a-z]+)?)[.,]\s
[\s\S]{0,43}?
 \n3[.,]\s([A-Z][a-z]+(?:\s[A-Z](?:[c][A-Z][a-z]+|[a-z]+))?(?:\s[A-
Z][a-z]+)?)[.,]
```

FIGURE 4.4. A GREENFIE-GENERATED EXTRACTION RULE FOR A COMPLEX RECORD.

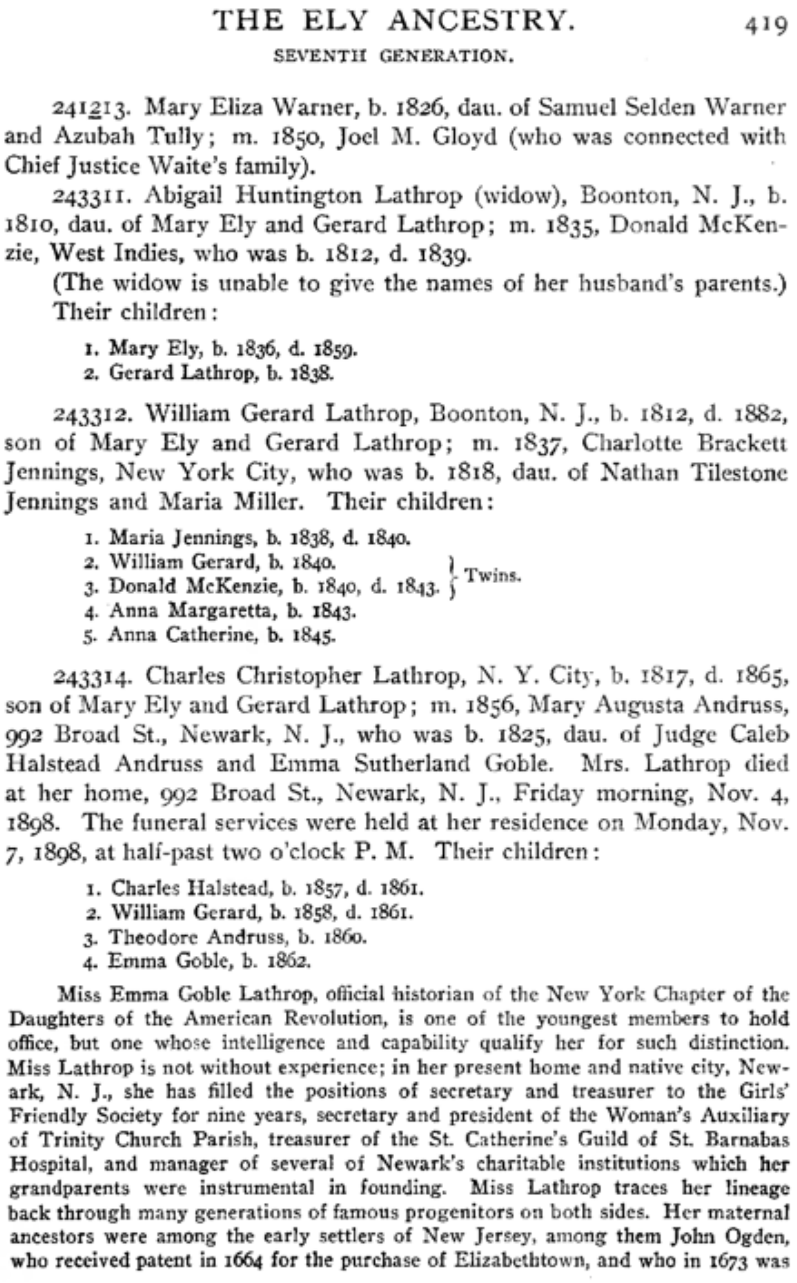
As Figure 4.4 emphasizes, each generated extraction rule targets only one list element—e.g. extracts only one of the children in the Family form, the third for the extraction rule in Figure 4.4. To get all list elements, GreenFIE executes all *n* of its extraction rules—e.g. executes all 12 generated extraction rules for the Family form. After obtaining the results for each extraction rule, GreenFIE stitches them together to form a single complex record like the highlighted record in Figure 1.1.

5. EXPERIMENTAL RESULTS

Since there is no other tool that GreenFIE directly competes with, we did not run an experimental evaluation against a competing tool. Instead, we ran field experiments to determine whether GreenFIE reduces the amount of user input it takes to annotate historical documents. In particular, we measured its "greenness"—how precision and recall are affected by new regular-expression extraction rules GreenFIE generates and executes as it "watches" a user annotate a document.

5.1 Experimental Setup

In our field experiments, we used two historical books that contain semi-structured genealogical information: *The Ely Ancestry* [Va1902] and *The Register of Marriages and Baptisms in the Parish of Kilbarchan* [Gr1912]. Figures 5.1 and 5.2 respectively display a page from the Ely and Kilbarchan books. The Ely page in Figure 5.1 is typical of the Ely book with a mix of semi-structured running text and formatted lists. The semi-structured text has good field identifiers. Figure 5.1, for example, shows the author's use of "b." and "d." for birth and death dates, "m." for marriage dates followed by the spouse name, and also shows numbered child lists. On the other hand, Kilbarchan pages (see Figure 5.2) are more spatially formatted and have relatively few distinct field identifiers. Kilbarchan pages do have "m." for marriage dates and "p." for proclamation of banns (required declarations of an intended marriage, which we can use as approximate marriage dates). But special layouts are the only indicators for fathers of families (which are on the left margin), children (which are indented in a list), and christening dates (which follow children's names). Exceptions are noted specifically, such as "born" for children not christened and "(father dead)" for a father who died before the christening or birth of his child.



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. 1825, 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:

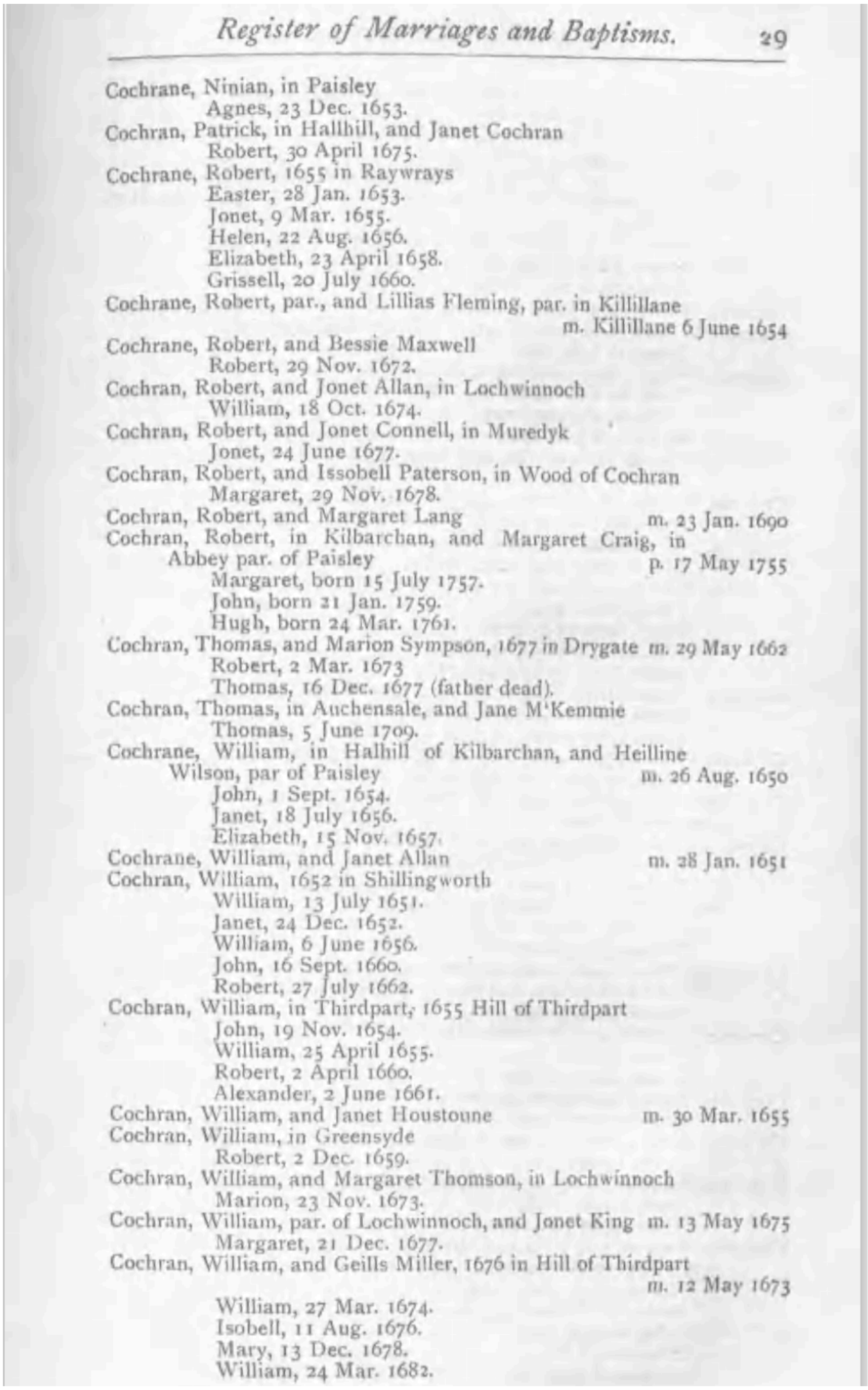
1. Charles Halstead, b. 1857, d. 1861.
2. William Gerard, b. 1858, d. 1861.
3. Theodore Andruss, b. 1860.
4. Emma Goble, b. 1862.

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

FIGURE 5.1. A SAMPLE PAGE FROM THE ELY BOOK.

Consistent use of field identifiers and layout patterns make it possible for GreenFIE to successfully generate regular-expression extraction rules. The Ely and Kilbarchan books satisfy these semi-structured expectations and yet are quite different, with Kilbarchan being highly structured and Ely being a mix of structured lists and free-running text. We purposely do not investigate documents consisting mostly of free-running text that do not have these semi-structured characteristics because they are not within the realm of documents GreenFIE is expected to be able to process.

Although it would be preferable to process entire books, because of resource and time constraints, we only processed a selected set of pages in a book. We selected as a blind test set a sequence of three pages from each of the two books. Omitting blank pages, pages with pictures, and non-genealogical commentary pages, every three-page sequence of both books has a lot of information that corresponds to the three forms we use: Person (see Figure 4.1), Couple (see Figure 4.3), and Family (see Figure 1.1). Our experimental task over the blind test set consisted of filling in these three forms with the aid of GreenFIE to capture the information in three pages for the two books. Altogether several thousand facts were extracted, and we measured the accuracy of the extraction and the speed-up provided by GreenFIE.



Register of Marriages and Baptisms. 29

Cochrane, Ninian, in Paisley
Agnes, 23 Dec. 1653.
Cochran, Patrick, in Hallhill, and Janet Cochran
Robert, 30 April 1675.
Cochrane, Robert, 1655 in Raywrays
Easter, 28 Jan. 1653.
Jonet, 9 Mar. 1655.
Helen, 22 Aug. 1656.
Elizabeth, 23 April 1658.
Grissell, 20 July 1660.
Cochrane, Robert, par., and Lillias Fleming, par. in Killillane
m. Killillane 6 June 1654
Cochrane, Robert, and Bessie Maxwell
Robert, 29 Nov. 1672.
Cochran, Robert, and Jonet Allan, in Lochwinnoch
William, 18 Oct. 1674.
Cochran, Robert, and Jonet Connell, in Muredyk
Jonet, 24 June 1677.
Cochran, Robert, and Issobell Paterson, in Wood of Cochran
Margaret, 29 Nov. 1678.
Cochran, Robert, and Margaret Lang m. 23 Jan. 1690
Cochran, Robert, in Kilbarchan, and Margaret Craig, in
Abbey par. of Paisley p. 17 May 1755
Margaret, born 15 July 1757.
John, born 21 Jan. 1759.
Hugh, born 24 Mar. 1761.
Cochran, Thomas, and Marion Sympson, 1677 in Drygate m. 29 May 1662
Robert, 2 Mar. 1673
Thomas, 16 Dec. 1677 (father dead).
Cochran, Thomas, in Auchensale, and Jane M'Kemmie
Thomas, 5 June 1709.
Cochrane, William, in Halhill of Kilbarchan, and Heilline
Wilson, par of Paisley m. 26 Aug. 1650
John, 1 Sept. 1654.
Janet, 18 July 1656.
Elizabeth, 15 Nov. 1657.
Cochrane, William, and Janet Allan m. 28 Jan. 1651
Cochran, William, 1652 in Shillingworth
William, 13 July 1651.
Janet, 24 Dec. 1652.
William, 6 June 1656.
John, 16 Sept. 1660.
Robert, 27 July 1662.
Cochran, William, in Thirdpart, 1655 Hill of Thirdpart
John, 19 Nov. 1654.
William, 25 April 1655.
Robert, 2 April 1660.
Alexander, 2 June 1661.
Cochran, William, and Janet Houstoune m. 30 Mar. 1655
Cochran, William, in Greensyde
Robert, 2 Dec. 1659.
Cochran, William, and Margaret Thomson, in Lochwinnoch
Marion, 23 Nov. 1673.
Cochran, William, par. of Lochwinnoch, and Jonet King m. 13 May 1675
Margaret, 21 Dec. 1677.
Cochran, William, and Geills Miller, 1676 in Hill of Thirdpart
m. 12 May 1673
William, 27 Mar. 1674.
Isobell, 11 Aug. 1676.
Mary, 13 Dec. 1678.
William, 24 Mar. 1682.

FIGURE 5.2. A SAMPLE PAGE FROM THE KILBARCHAN BOOK.

Prior to running against the blind test set, we "trained" GreenFIE on a development test set from each book consisting of the two-page sequence immediately preceding the blind test-set pages. The document page in Figure 5.1 was the first of the Ely-book two-page sequence, and the document page in Figure 5.2 was the second page for the Kilbarchan book. Training consisted of fixing the rules for generalizing regular expressions (as explained in Chapter 4) so that GreenFIE performed well on the development test set. For known field value-type generalizations, we populated the set of generalization expressions for names and dates using only patterns for the names, dates, and places that appeared in the development test set. For names, the generalization expressions were:

```
([A-Z][a-z]+,\s[A-Z][a-z]+)

([A-Z][a-z]+(?:\s[A-Z](?:[c][A-Z][a-z]+|[a-z]+))?(?:\s[A-Z][a-z]+)?)
```

For dates there was only one:

```
((?:\d{1,2}|I|[[]\d{1})

\s[JFMASOND][a-z]{2,4}[.,]?\s(?:\d{4}|i\d{3})|\d{4})
```

For places, we choose to have none because only a few place names are in our test set and for those that were (e.g. “Killillane” in Figure 5.2 and similar names) the basic generalization was sufficient.

The only <anchor> enumerator that appeared in the development test set for children in the Family form or multiple spouses in the Couple form was the sequence: “1. … 2. … 3. …”. Hence this was the only enumerator established for the experiment. Kilbarchan child lists are indented (but not enumerated), and GreenFIE handles these formatted lists by observing newline indicators, `\n`’s.

5.2 Blind Test-Set Experimental Results

A run over the blind test set in the experiment consisted of four user actions: annotating, completing, trimming, and deleting a record for a form. These actions were repeated until all the information in each page in the three-page sequence was correctly captured. User actions proceeded in page order, top to bottom. When encountering information on a page to be extracted, the user either (1) added a new record and filled it in or (2) for records created by GreenFIE accepted it if correct or edited or deleted it. Record edits consisted only of (1) completing: adding entries to empty fields in records that were correct but incomplete and (2) trimming: deleting children beyond the end of the list of actual children in the Family form when the record properly captured the parents and all children but incorrectly added additional children. Record delete was only for removing a generated record that subsumes an extracted record that was correct. Other generated records were either correct or incorrect in some other way. We could have deleted these incorrect records along the way, but left them intact so that we could measure precision from beginning to end. After each user action except trimming or deleting, the user clicked the "Regex" button alerting GreenFIE to generate a regular-expression rule for the record, generalize it, execute it over the page currently being processed, and create additional records recognized by the generated extraction rule. Except for duplicates, created records for information appearing on the page subsequent to the record information for which the new extraction rule had been generated were then added. GreenFIE automatically deleted any properly subsumed subsequent records. Incorrectly generated records both before and after, if any, were added. Before beginning to work on the second or third page, GreenFIE initialized the form by executing all rules accumulated so far in the run. The form for the first page in the run always started empty with no records being filled in.

To validate the “greenness” of the system, we tracked and plotted recall and precision as a function of the number of GreenFIE-created regular-expression extraction rules. A ground truth for each of the pages was created, and after each rule-creation/rule-execution cycle, we compared the records extracted for the page with the ground-truth records. We counted a record correct only if it was a perfect match with a record in the ground truth. Records with more or fewer fields filled in or with conflicting instance data in a field were counted as being incorrect. No GreenFIE-generated record was ever deleted and those that were wrong (as opposed to being incorrect simply by being incomplete) were not edited; instead they were left in place as false positives. (OCR errors in fields were corrected both in the ground truth and in extracted records, but otherwise no editing of field values occurred.)

Figure 5.3 shows a graph of the results of running Ely pages 575 to 577 for the Person form. The x-axis gives the annotation cycle number for each page. An annotation cycle consists of (1) annotate a record (either by adding a new record and filling it in or by editing an existing record by adding the field values) and (2) click the “Regex” button. The zeros mark the beginning of a new page in which all regular-expression extraction rules collected so far in the run are executed. The graph (Figure 5.3) is plotted across the three pages: 0 to 15 is for page 575, 0 to 8 is for 576, and 0 to 7 is for 577. The second and third “0” show the initial precision and recall that GreenFIE found with the extraction rules that it learned from the previous page(s). Since users annotate records in a page until all records are correctly included, the recall curve for each page necessarily increases monotonically until it reaches 100%. A precision curve decreases whenever a GreenFIE-generated rule extracts an incorrect record and increases whenever a user comes to and fills in fields for an incomplete and thus an incorrectly generated record. Since we do not delete incorrect records, genuine false positives persist to the end of the run.

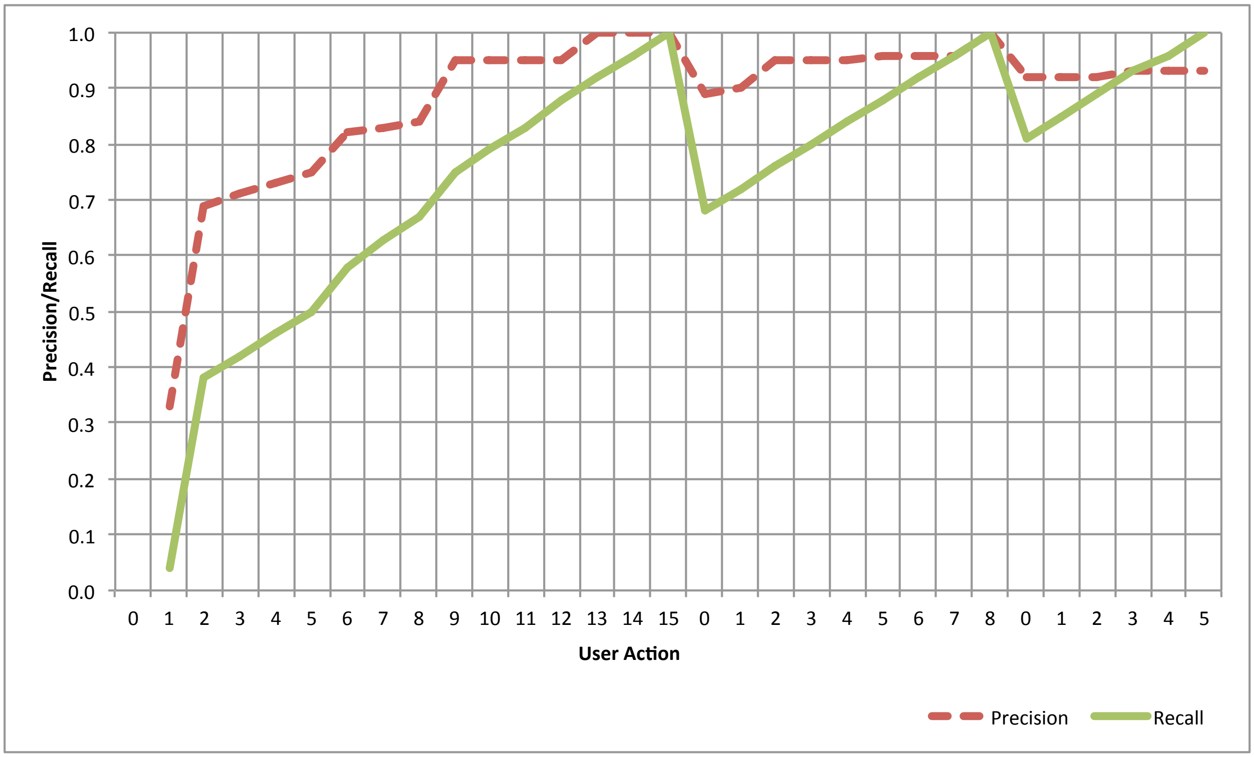


FIGURE 5.3. GRAPH OF PERSON FORM RESULTS FOR ELY PAGES 575–577.

Table 5.1 gives the raw data for the graph in Figure 5.3. Although the graph is visually appealing and makes the overall results easy to grasp, it lacks the detail of the raw data in Table 5.1. Since the graphs for all six experimental runs have the same form as the graph for the first run in Figure 5.3, we omit the remaining graphs and instead give the raw data for the remaining runs in Tables 5.2–5.6. In the tables, *Total* is the number of records on the page, *Found* is the number of records extracted by the working set of regular-expression extraction rules, *Correct* is the number of records that have a perfect match with the ground truth, *Incorrect* is the number of generated records that are not *Correct*, *Precision* = *Correct* / (*Correct* + *Incorrect*), *Recall* = *Correct* / *Total*, and *F-score* = 2 × (*Precision* × *Recall*) / (*Precision* + *Recall*).

TABLE 5.1. PERSON FORM RESULTS FOR ELY PAGES 575–577.

Ely 575								
User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	0	24	N/A	N/A	N/A	N/A	N/A	N/A
1	1	24	3	1	2	0.33	0.04	0.07
2	2	24	13	9	4	0.69	0.38	0.49
3	3	24	14	10	4	0.71	0.42	0.53
4	4	24	15	11	4	0.73	0.46	0.56
5	5	24	16	12	4	0.75	0.50	0.60
6	6	24	17	14	3	0.82	0.58	0.68
7	7	24	18	15	3	0.83	0.63	0.71
8	8	24	19	16	3	0.84	0.67	0.74
9	9	24	19	18	1	0.95	0.75	0.84
10	10	24	20	19	1	0.95	0.79	0.86
11	11	24	21	20	1	0.95	0.83	0.89
12	12	24	22	21	1	0.95	0.88	0.91
13	13	24	22	22	0	1.00	0.92	0.96
14	14	24	23	23	0	1.00	0.96	0.98
15	15	24	24	24	0	1.00	1.00	1.00
Ely 576								
User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	15	25	19	17	2	0.89	0.68	0.77
1	16	25	20	18	2	0.90	0.72	0.80
2	17	25	20	19	1	0.95	0.76	0.84
3	18	25	21	20	1	0.95	0.80	0.87
4	19	25	22	21	1	0.95	0.84	0.89
5	20	25	23	22	1	0.96	0.88	0.92
6	21	25	24	23	1	0.96	0.92	0.94
7	22	25	25	24	1	0.96	0.96	0.96
8	23	25	25	25	0	1.00	1.00	1.00
Ely 577								
User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	23	27	24	22	2	0.92	0.81	0.86
1	24	27	25	23	2	0.92	0.85	0.88
2	25	27	26	24	2	0.92	0.89	0.91
3	26	27	27	25	2	0.93	0.93	0.93
4	27	27	28	26	2	0.93	0.96	0.95
5	28	27	29	27	2	0.93	1.00	0.96

Table 5.2. Person Form Results for Kilbarchan Pages 33–35.

Kilbarchan 33								
User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	0	35	N/A	N/A	N/A	N/A	N/A	N/A
1	1	35	21	21	0	1.00	0.60	0.75
2	2	35	29	29	0	1.00	0.83	0.91
3	3	35	31	31	0	1.00	0.89	0.94
4	4	35	34	34	0	1.00	0.97	0.99
5	5	35	35	35	0	1.00	1.00	1.00
Kilbarchan 34								
User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	5	32	32	32	0	1.00	1.00	1.00
Kilbarchan 35								
User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	5	31	31	31	0	1.00	1.00	1.00

Table 5.3. Couple Form Results for Ely Pages 575–577.

Ely 575								
User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	0	11	N/A	N/A	N/A	N/A	N/A	N/A
1	1	11	2	2	0	1.00	0.18	0.31
2	2	11	3	3	0	1.00	0.27	0.43
3	3	11	4	4	0	1.00	0.36	0.53
4	4	11	5	5	0	1.00	0.45	0.63
5	5	11	6	6	0	1.00	0.55	0.71
6	6	11	7	7	0	1.00	0.64	0.78
7	7	11	8	8	0	1.00	0.73	0.84
8	8	11	9	9	0	1.00	0.82	0.90
9	9	11	10	10	0	1.00	0.91	0.95
10	10	11	11	11	0	1.00	1.00	1.00
Ely 576								
User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	10	15	7	7	0	1.00	0.47	0.64
1	11	15	8	8	0	1.00	0.53	0.70
2	12	15	9	9	0	1.00	0.60	0.75

3	13	15	10	10	0	1.00	0.67	0.80
4	14	15	11	11	0	1.00	0.73	0.85
5	15	15	12	12	0	1.00	0.80	0.89
6	16	15	13	13	0	1.00	0.87	0.93
7	17	15	14	14	0	1.00	0.93	0.97
8	18	15	15	15	0	1.00	1.00	1.00
Ely 577								
User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	18	17	9	8	1	0.89	0.47	0.62
1	19	17	12	10	2	0.83	0.59	0.69
2	20	17	13	11	2	0.85	0.65	0.73
3	21	17	13	12	1	0.92	0.71	0.80
4	22	17	14	13	1	0.93	0.76	0.84
5	23	17	15	14	1	0.93	0.82	0.88
6	24	17	16	15	1	0.94	0.88	0.91
7	25	17	17	16	1	0.94	0.94	0.94
8	26	17	17	17	0	1.00	1.00	1.00

TABLE 5.4. COUPLE FORM RESULTS FOR KILBARCHAN PAGES 33–35.

Kilbarchan 33								
User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	0	5	N/A	N/A	N/A	N/A	N/A	N/A
1	1	5	1	1	0	1.00	0.20	0.33
2	2	5	2	2	0	1.00	0.40	0.57
3	3	5	3	3	0	1.00	0.60	0.75
4	4	5	4	4	0	1.00	0.80	0.89
5	5	5	5	5	0	1.00	1.00	1.00
Kilbarchan 34								
User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	5	12	0	0	0	N/A	0.00	N/A
1	6	12	5	4	1	0.80	0.33	0.47
2	7	12	7	6	1	0.86	0.50	0.63
3	8	12	8	7	1	0.88	0.58	0.70
4	9	12	9	8	1	0.89	0.67	0.76
5	10	12	10	9	1	0.90	0.75	0.82
6	11	12	12	11	1	0.92	0.92	0.92
7	12	12	13	12	1	0.92	1.00	0.96
Kilbarchan 35								

User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	12	13	10	9	1	0.90	0.69	0.78
1	13	13	11	10	1	0.91	0.77	0.83
2	14	13	13	12	1	0.92	0.92	0.92
3	15	13	14	13	1	0.93	1.00	0.96

Table 5.5. Family Form Results for Ely Pages 575–577.

Ely 575								
User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	0	13	N/A	N/A	N/A	N/A	N/A	N/A
1	1	13	1	1	0	1.00	0.08	0.14
2	2	13	2	2	0	1.00	0.15	0.27
3	3	13	3	3	0	1.00	0.23	0.38
4	4	13	4	4	0	1.00	0.31	0.47
5	5	13	5	5	0	1.00	0.38	0.56
6	6	13	6	6	0	1.00	0.46	0.63
7	7	13	7	7	0	1.00	0.54	0.70
8	8	13	8	8	0	1.00	0.62	0.76
9	9	13	9	9	0	1.00	0.69	0.82
10	10	13	10	10	0	1.00	0.77	0.87
11	11	13	11	11	0	1.00	0.85	0.92
12	12	13	12	12	0	1.00	0.92	0.96
13	13	13	13	13	0	1.00	1.00	1.00
Ely 576								
User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	13	11	2	2	0	1.00	0.18	0.31
1	14	11	3	3	0	1.00	0.27	0.43
2	15	11	4	4	0	1.00	0.36	0.53
3	16	11	5	5	0	1.00	0.45	0.63
4	17	11	6	6	0	1.00	0.55	0.71
5	18	11	7	7	0	1.00	0.64	0.78
6	19	11	8	8	0	1.00	0.73	0.84
7	20	11	9	9	0	1.00	0.82	0.90
8	21	11	10	10	0	1.00	0.91	0.95
9	22	11	11	11	0	1.00	1.00	1.00
Ely 577								
User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	22	14	3	3	0	1.00	0.21	0.35

1	23	14	5	5	0	1.00	0.36	0.53
2	24	14	6	6	0	1.00	0.43	0.60
3	25	14	7	7	0	1.00	0.50	0.67
4	26	14	8	8	0	1.00	0.57	0.73
5	27	14	9	9	0	1.00	0.64	0.78
6	28	14	10	10	0	1.00	0.71	0.83
7	29	14	11	11	0	1.00	0.79	0.88
8	30	14	12	12	0	1.00	0.86	0.92
9	31	14	13	13	0	1.00	0.93	0.96
10	32	14	14	14	0	1.00	1.00	1.00

TABLE 5.6. FAMILY FORM RESULTS FOR KILBARCHAN PAGES 33–35.

Kilbarchan 33								
User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	0	17	N/A	N/A	N/A	N/A	N/A	N/A
1	1	17	1	1	0	1.00	0.06	0.11
2	2	17	16	4	12	0.25	0.24	0.24
3	2	17	16	5	11	0.31	0.29	0.30
4	3	17	16	6	10	0.38	0.35	0.36
5	4	17	31	12	19	0.39	0.71	0.50
6	5	17	32	13	19	0.41	0.76	0.53
7	6	17	32	11	21	0.34	0.65	0.45
8	6	17	32	12	20	0.38	0.71	0.49
9	6	17	32	13	19	0.41	0.76	0.53
10	7	17	46	14	32	0.30	0.82	0.44
11	7	17	46	15	31	0.33	0.88	0.48
12	7	17	46	16	30	0.35	0.94	0.51
13	8	17	46	17	29	0.37	1.00	0.54
Kilbarchan 34								
User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	8	14	26	0	26	0.00	0.00	N/A
1	8	14	26	1	25	0.04	0.07	0.05
2	8	14	26	2	24	0.08	0.14	0.10
3	9	14	28	3	25	0.11	0.21	0.14
4	9	14	27	3	24	0.11	0.21	0.15
5	10	14	29	4	25	0.14	0.29	0.19
6	10	14	29	5	24	0.17	0.36	0.23
7	11	14	31	6	25	0.19	0.43	0.27

8	12	14	35	7	28	0.20	0.50	0.29
9	12	14	35	8	27	0.23	0.57	0.33
10	12	14	35	9	26	0.26	0.64	0.37
11	12	14	35	10	25	0.29	0.71	0.41
12	13	14	36	11	25	0.31	0.79	0.44
13	13	14	36	12	24	0.33	0.86	0.48
14	14	14	37	13	24	0.35	0.93	0.51
15	15	14	37	14	23	0.38	1.00	0.55
Kilbarchan 35								
User Action	Regexes	Total	Found	Correct	Incorrect	Precision	Recall	F-score
0	15	17	43	1	42	0.02	0.06	0.03
1	15	17	43	2	41	0.05	0.12	0.07
2	16	17	44	3	41	0.07	0.18	0.10
3	17	17	45	4	41	0.09	0.24	0.13
4	17	17	45	5	40	0.11	0.29	0.16
5	17	17	45	6	39	0.13	0.35	0.19
6	17	17	45	7	38	0.16	0.41	0.23
7	18	17	46	8	38	0.17	0.47	0.25
8	18	17	46	9	37	0.20	0.53	0.29
9	19	17	47	10	37	0.21	0.59	0.31
10	20	17	47	11	36	0.23	0.65	0.34
11	20	17	47	12	35	0.26	0.71	0.38
12	20	17	47	13	34	0.28	0.76	0.41
13	20	17	47	14	33	0.30	0.82	0.44
14	21	17	48	15	33	0.31	0.88	0.46
15	21	17	48	16	32	0.33	0.94	0.49
16	22	17	49	17	32	0.35	1.00	0.52

TABLE 5.7. RESULTS SUMMARY.

	User Action	Correct	Correct / User Action
Ely	86	157	1.83
Person	28	76	2.71
Couple	26	43	1.65
Family	32	38	1.19
Kilbarchan	64	176	2.75
Person	5	98	19.60
Couple	15	30	2.00
Family	44	48	1.09
Overall	150	333	2.22

Table 5.7 gives the rate at which 100% recall is achieved. The rate is the total number of records correctly extracted divided by the total number of user actions. The maximum rate, for example, is 19.60, which tells us that working on just five records is sufficient to extract all the information in the 98 records for the Person form on the three Kilbarchan pages. Table 5.7 also tells us the total number of records extracted, 333, and the total number of user actions, 150. We estimated the number of data values extracted in these 333 records to be about 10,000 so that approximately 65 data values were obtained for each user-extracted or user-edited record.

6. DISCUSSION

6.1 Experimental Results Observations

Semi-Structured Text. GreenFIE works better with more structured text. Kilbarchan records for names and birth or christenings dates are simple and well-structured. It required only 5 user actions to extract all 98 records. The layout of spouse and parent text also appears to be well structured. However, the variations due to inserted commentary, OCR errors, and inconsistent punctuation turned out to be more than initially perceived and required GreenFIE to make a regular expression for each of the many variations. In contrast to the 19.6 records per user action extracted for the Kilbarchan Person form, 2 records per user action were extracted for the Couple form and only 1.09 records for the Family form. Ely records were also deceptively less regular than initially perceived, again because of inserted commentary, OCR errors, and inconsistent punctuation. Over all three forms, the rate of automatic annotation was 1.83 records annotated per user action.

Precision. Except for the Kilbarchan Family form, the total of 106 GreenFIE-generated regular-expression extraction rules led to relatively few precision errors—none for the Kilbarchan Person form and for the Ely Couple and Family forms; only one for the Kilbarchan Couple form; and only two for the Ely Person form.

Kilbarchan Family Form Precision. In processing the Kilbarchan Family form, 132 records were found. However, only 48 were correct, leaving 84 incorrect and thus a precision of 0.36. Looking at the document, it is evident why the precision is so low. The main feature that reveals child lists is the anchor that enumerates the children. The Ely enumerator for children in the Family form is the sequence of numbers, 1, 2, …, up to 12 since we chose 12 as our upper bound. Because of this distinguishing enumerator, the precision for the Ely Family form was

100%. With no standard enumerator, GreenFIE takes whatever it sees on the left of the child names and uses it for counting. For Kilbarchan the left context is a newline (`\n`), and the *n*th child's name appears immediately after the *n*th newline character, counting from the beginning of the list. Unfortunately, since child names are formatted the same as father surnames and the anchors for both are the same—the newline character, GreenFIE found father surnames as child names, thus creating lingering lists of children as precision errors. Moreover, the lack of right context for some mother names plus "`\n`" being the enumerator for a child list caused GreenFIE's generated regular-expression rule for names to take the first child's name as part of a mother's name.

6.2 Lessons Learned

Value-type generalizations. We generalized name and date types across the two books, Ely and Kilbarchan. The results were much better than they would have been without these value-type generalizations. However, the cost of creating regular expressions for these generalizations has to be taken into account. In fact, to compose a regular expression that will match all names in different languages and cultures is almost impossible. However, it is possible to have library collections of regular expressions for many of the most common names and dates for each different language and culture. Even so, it may not be best to use these fully general regular expressions. As we saw in the Kilbarchan Family form, using Ely-like names that have more than two components caused the problem of sometimes being unable to separate the first child's name from the mother's name. Kilbarchan mother names never have more than two name components, and if we had generalized just for Kilbarchan names, this particular precision error would not have arisen. Value-type generalizations are good, but book-specific value-type generalizations are likely to be even better.

OCR errors. OCR errors cause problems for record fields, for delimiters, and for left and right context. OCR errors in known value types and character classes such as punctuation can be accommodated more readily than when value types and character classes are not known [Pack11]. For example, when "1" is recognized as "I", "i", or "l" in a numeric part of a date type or commas are recognized as periods and vice versa, these alternatives can be built into GreenFIE's regular-expression rules. Interestingly, in GreenFIE's interaction with users, OCR errors can be learned by observing corrections users make when editing out OCR errors in field values. Although, not an OCR error in the usual sense, when the OCR output fails to include formatting information such as tabs for child lists or extra-long dashes for missing child names, downstream processes like GreenFIE are handicapped, having even less information than a human for interpreting the text.

End-of-line hyphens. End-of-hyphens are problematic, especially when they appear in field values. It becomes unwieldy to generalize regular expressions to accommodate them. For example, if a hyphen appears in a name, a regular expression needs to be general enough to capture names without a hyphen or with a hyphen in different places, but not all places, within a name. One possible way to accommodate end-of-line hyphens without having to make regular expressions account for them might be to make a second copy of the text in which words with end-of-line hyphens are closed up and then process regular expressions against this second copy of the text.

Skip lengths. Finding the right length for delimiting commentary is complex. A main problem encountered in the experiment was skips allowing groups of children to be assigned to two different sets of parents in both Ely and Kilbarchan. In Figure 5.1, for example, the length of the text between Mary Augusta Andruss and her first child, Charles Halstead, exceeds the length of the text between Joel M. Gloyd and the child Mary Ely, who is not his child. Hence, it is impossible to choose a skip length that works for all cases. In this example execution of the rule for obtaining Mary Augusta Andruss's children incorrectly assigns Mary Ely and her brother Gerard Lathrop to be children of Joel M. Gloyd and Mary Eliza Warner. Moreover, GreenFIE's skip-length generalization that properly assigns Mary and Gerard to be children of Abigail Huntington Lathrop and Donald McKenzie is insufficient to allow Mary Augusta Andruss's children to be assigned to her. Hence, both rules are generated, and, in this example, Mary and Gerard are assigned to two sets of parents.[1] Skip lengths for Person and Couple records in our experiments did not cause precision errors. They did, however, cause a proliferation of regular-expression rules.

6.3 Recommendations

Choice of Records to Annotate. In the experiment, we specified the rules for record annotation to be done in certain way—in page order, top to bottom, with no record deletions, and required rule-creation for every positive annotation. However, a GreenFIE user is actually free to annotate, delete, and modify records as best suits the application using any strategy that works well. It is probable that annotating the most prevalent patterns first would produce better performance. In Kilbarchan, for example, choosing a well-structured family or in Ely, choosing a person with birth and death dates would have been better than the ill-structured Kilbarchan family and Ely person with only a birth date we were forced to choose by our experimental protocol. It is also probable that creating rules for seldom-occurring patterns or exception-case patterns such as for a child with an unknown name will not help much and may do more harm

[1] We note that in the larger ensemble of extraction tools in which GreenFIE is intended to work, the problem of a child having too many parents can be reliably resolved automatically [WLL+16].

than good. Extraction rules need not be generated if they are perceived to have relatively little use or are perceived to potentially extract incorrect information.

White-Space Formatting. It would be better to prevent the loss of tab-indentation in the output of OCR (if possible) or recoup it from the OCR bounding-box information. In documents like Kilbarchan that format lists by tab-indentation, the information is as valuable to GreenFIE as it is to a human reader.

Book Specific Value-Type Generalizations. We would recommend that rather than using value-type generalizations of all possible forms of person names, dates, and place names, use book-specific generalizations. In Kilbarchan, for example, father names are always "<surname>, <first name>", mother names are always "<first name> <maiden name>", and children all have just their given names; also, dates are all formatted the same. In our experiment, even having Ely-like name generalizations for Kilbarchan caused some problems that could have been avoided by book-specific value-type generalizations.

Taking our own advice for the Kilbarchan Family form, we added tabs for children in the Kilbarchan document, used only Kilbarchan-specific name generalizations, chose a better annotation strategy, and re-ran the first cycle of the experiment[2] on a typical family found on the first page in the blind test set. When we applied the regular-expression rule GreenFIE would have generated to the tab-altered documents, the regular expression correctly recognized four families on Page 033, two on Page 034, and six on Page 035. More importantly, the rule did not extract any incorrect families. Because of the tab anchor being part of the regular-expression extraction rule, GreenFIE-generated extraction rules would never produce precision errors with

[2] It was not feasible to actually create a document for the GreenFIE annotator with tabs as needed. So this experiment was done outside of GreenFIE using a regular-expression development site.

run-on children who do not belong to the family nor would they produce any other of the precision errors caused the absence of tabs. An analysis of the Family-form precision errors in Table 5.6 found that all 84 remaining at the end of the experimental run were due to the absence of tabs.

7. CONCLUSION

GreenFIE is a Form-based Information-Extraction tool which is “green” in the sense that it improves with use toward the goal of minimizing the cost of human labor. It learns based on user feedback, adding new extraction rules to its repository as work is accomplished, and it saves labor by executing newly created extraction rules immediately, filling-in records in advance so that the user need only check the automated form fill-in.

The results of our experiments indicate that GreenFIE does help diminish human labor. There were a total of 333 records to be extracted which would have required 333 user actions to extract. However, with GreenFIE, the task required only 150 user actions. GreenFIE works better when the text is well-structured and has strong and unique anchors for each field to be extracted. Person-form data in Kilbarchan is the most organized and has a unique anchor for each field; thus it had the best result—19.6 records found per user action. The Kilbarchan book is generally more structured than the Ely book. The experimental results show that in Kilbarchan, GreenFIE found 2.75 records per user action while in Ely, it found 1.83. However, Family-form data in Kilbarchan suffered; in fact it did the worst, getting only 1.09 records per user action because of the high variability of parent data and the lack of unique anchors for child lists. Moreover, the Family-form precision was extremely low—only 36%. In a separate test in which missing tab anchors were added, however, GreenFIE performed much better and attained 100% precision.

As for future work, we would like to do a tech-transfer of GreenFIE and add it to the ensemble of extractors being developed for semi-automatically extracting genealogical information from a large collection of scanned and OCRed family-history books. As part of the tech transfer, we would also like to upgrade the interface for better real-world use and for allowing knowledgeable users to manage rule capture, curation, and reuse across books.

REFERENCES

[Blac00] A.F. Blackwell, SWYN: A Visual Representation for Regular Expressions, in *Your Wish is My Command: Giving Users the Power to Instruct their Software*, H. Lieberman (ed.), Morgan Kaufmann, 2000, 245-270.

[BloN12] D. Blostein and G. Nagy, Asymptotic Cost in Document Conversion, *Proceedings of the 19th Conference on Document Recognition and Retrieval* (DRR'12), San Francisco, California, USA, 24–26 January, 2012, 82970N–82970N-9.

[ChaK04] C.-H. Chang and S.-C. Kuo, OLERA: Semisupervised Web-Data Extraction with Visual Support, *IEEE Intelligent Systems*, 19(6):56–64, November/December, 2004.

[DEG14] BYU Data Extraction Group, Annotator, http://dithers.cs.byu.edu/annotator2, 2014 (last accessed May 2017).

[EmLL11] D.W. Embley, S. W. Liddle, and D. W. Lonsdale, Conceptual Modeling Foundations for a Web of Knowledge, Chapter 15 in *The Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges*, Springer, 477–516, 2011.

[Gr1912] 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.

[LopN09] D. Lopresti and George Nagy, Tools for Monitoring, Visualizing, and Refining Collections of Noisy Documents, *Proceedings of the Third Workshop on Analytics for Noisy Unstructured Text Data* (AND 2009), Barcelona, Spain, July 23–24, 2009, 9–16.

[Nagy12a] G. Nagy, Keynote Address: Back to the Future, *Family History Technology Workshop*, Salt Lake City, Utah, February, 2012.

[Nagy12b] G. Nagy, Estimation, Learning, and Adaptation: Systems that Improve with Use, *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, Hiroshima, Japan, November, 2012, 1–10.

[Pack11] T.L. Packer, Performing Information Extraction to Improve OCR Error Detection in Semi-structured Historical Documents, *Proceedings of the 1st International Workshop on Historical Document Imaging and Processing* (HIP'11), Beijing, China, 16–17 September, 2011, 67–74.

[Pack14] T.L. Packer, *Scalable Detection, Recognition, Extraction, and Structuring of Data from Lists in OCRed Text for Ontology Population using Semi-Supervised and Unsupervised Active Wrapper Induction*, PhD Dissertation, Department of Computer Science, Brigham Young University, December 2014.

[Park15] J. Park, *FROntIER: A Framework for Extracting and Organizing Biographical Facts in Historical Documents*, Master's Thesis, Department of Computer Science, Brigham Young University, January 2015.

[Sara08] S. Sarawagi, Information Extraction, *Foundations and Trends in Databases*, 1(3):261–377, March 2008.

[TuAC06] J. Turmo, A. Ageno, and N. Catala, Adaptive Information Extraction, *ACM Computing Surveys*, 38(2):1–47, July 2006.

[Va1902] 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.

[WLL+16] W.N. Woodfield, D.W. Lonsdale, S.W. Liddle, T.W. Kim, D.W. Embley, and C. Almquist, Pragmatic Quality Assessment for Automatically Extracted Data, *Proceedings of the 35th International Conference on Conceptual Modeling* (ER 2016), Gifu, Japan, 14–17 November 2016, 212–220.

[ZouN04] J. Zou and G. Nagy, Evaluation of Model-Based Interactive Pattern Recognition," *Proceedings of International Conference on Pattern Recognition XVII*, vol.II, Cambridge, UK, August 2004, 311–314.