Extraction Rule Creation by Text Snippet Examples

David W. Embley (Brigham Young University & FamilySearch)
George Nagy (Rensselaer Polytechnic Institute)

Abstract A rule-based information extraction tool for semi-structured family history books is introduced. Users program the tool by giving discriminating text snippet examples and designating a substring of the text snippet as belonging to a specified class (e.g. birth date, marriage place, child in family, …). A designated grouping class facilitates record formation, creating a group of class-value pairs as the record's fields that describe a record's objects (e.g. persons with their birth and death information, families with parent-child relationships, …). Once initialized, the extraction tool can suggest candidate rules for a user's consideration. Initial experimental results indicate that a user working synergistically with the tool can quickly achieve high quality extraction results.

1. Introduction

FamilySearch has scanned and OCR'd more than 360,000 family history books [1]. Indexing them for semantic search by hand, even with thousands of volunteers, is untenable, causing us to turn to automated information extraction for a solution. Automated extraction tools are based on a variety of techniques including machine learning, natural language processing, and expert rule systems. Each technique has its strengths and weaknesses, and ultimately the best solution may be an ensemble of these tools working together to extract genealogical information from the large variety of documents: family histories, parish registers, funeral-home burial records, ship manifests, and many more.

Adding to our ensemble of extraction tools, we proffer GreenQQ—a rule-generation system programmed by giving examples. Further, once initialized by a few examples, GreenQQ generates candidate rules for the user to assess, possibly edit, and add to its collection of extraction rules. It is "Green" because like other "green" tools [2], it improves with use while doing real-world tasks. And, as we show in our initial experimental results (Section 3), it manifests two Q's: (Q1) rule sets can be developed Quickly and (Q2) the rule sets produce Quality results.

2. By-Example Programming

We humans readily identify patterns wherever we see them. In Figure 1, for example, the family groups are immediate, and we have an expectation that other family groups will appear in a similar arrangement. On closer examination, meaningful text snippets emerge. In Figure 1 fathers appear in last-name-first form at the beginning of a family group and spouse names follow shortly thereafter following an "and". Marriage places and dates are preceded by "m.", and marriage proclamation dates (banns) are preceded by "p.". Children in a family are in indented lists. Christening dates immediately follow child names, and birth dates are specified with "born".

A user programs GreenQQ by giving a text snippet example that includes the text to be extracted and sufficient surrounding context to identify the extract and also specifies the extract's class (Parent, Child, BirthDate, …). GreenQQ then builds an abstraction of the pattern and extracts and classifies extracts in all similar text snippets. Thus, given an example, GreenQQ first does NER (Named Entity Recognition [3]). NER, however, is insufficient because we not only need classified text values but also the relationships among the various values. For example, we need to know not only that some text value is a birth date, we also need to know whose birth date it is.

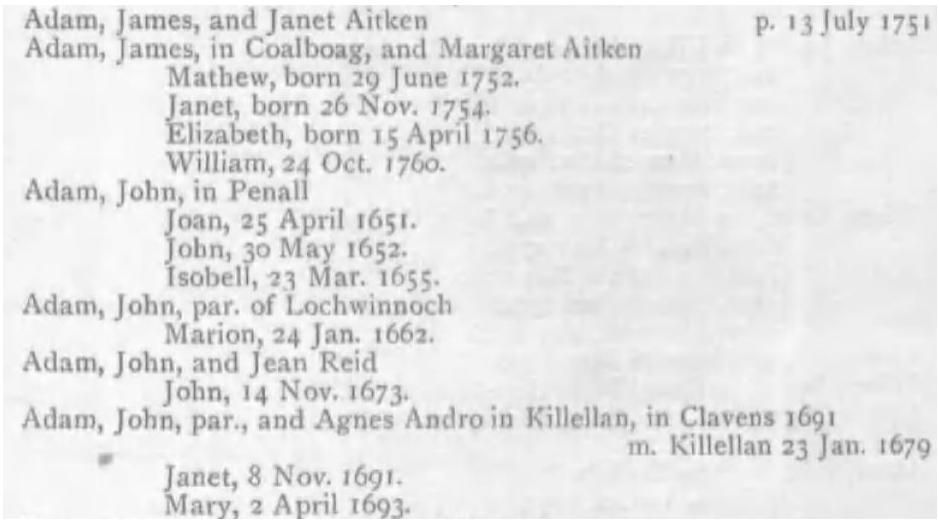

Adam, James, and Janet Aitken p. 13 July 1751
Adam, James, in Coalboag, and Margaret Aitken
Mathew, born 29 June 1752.
Janet, born 26 Nov. 1754.
Elizabeth, born 15 April 1756.
William, 24 Oct. 1760.
Adam, John, in Penall
Joan, 25 April 1651.
John, 30 May 1652.
Isobell, 23 Mar. 1655.
Adam, John, par. of Lochwinnoch
Marion, 24 Jan. 1662.
Adam, John, and Jean Reid
John, 14 Nov. 1673.
Adam, John, par., and Agnes Andro in Killellan, in Clavens 1691
m. Killellan 23 Jan. 1679
Janet, 8 Nov. 1691.
Mary, 2 April 1693.

Figure 1. Families in the Kilbarchan Parish Register [5].

In semi-structured text, particularly, and even in free-running text, authors tend to group object-descriptor text close to the object being described. A group of classified text values that all pertain to the same object forms a *record*—a set of field-value pairs, where for GreenQQ the field name is the class and the value is the extracted text. For a run over a document, GreenQQ is programmed for a single record type. All the records of interest can be extracted, of course, in multiple runs. GreenQQ programmers specify record types by enumerating the classes (the fields) that belong to the record and designating one of the classes as the grouping entity. GreenQQ forms records by grouping together all the field-value pairs that fall between the grouping entity in the document's textual sequence. For our family history application, the records and their fields can be pre-specified. Hence, the only requirement for programming GreenQQ for extracting information from family history books is to give it discriminating text snippet examples and say what part of the text snippet is to be extracted as the value for a field. Additional details about the abstraction underlying snippet representation, which is invisible to the user, can be found in [4].

In Figure 1, for example, to program for Person records consisting of a person's name and either a birth date or a christening date, a user would specify the following and only the following (nothing else!):

[Name] [^ Mathew, born] [Mathew]
[Name] [^ Joan, 25] [Joan]
[BirthDate] [born 29 June 1752. $] [29 June 1752]
[BirthDate] [, 25 April 1651. $] [25 April 1651]

In these examples, the first element is the record's field name, the second is the text snippet which includes discriminating context, and the third is the value to be extracted. Notationally, the carrot denotes a line beginning and a dollar sign designates a line ending. Also, "born" needs to be designated as a literal, so that when the text is abstracted as a pattern template, it remains unchanged.

GreenQQ creates templates for each text snippet. The text snippet in first statement above becomes [^ Cap , born] where "Cap" denotes a capitalized word and born is a literal, and the text snippet in the last statement becomes [, Num Cap Num . $] where "Num" is a number. These

templates recognize all eleven Person records in Figure 1 and more than a thousand additional Person records in the 143-page Kilbarchan Parish Register [5].

Although GreenQQ finds many hundreds more Person records, it does not find all of them. In the Kilbarchan book, and almost assuredly in every book with semi-structured text, there are exceptions to basic patterns. Authors add explanatory notes, accommodate exceptions, and are not always self-consistent. Moreover, typesetting errors creep in and OCR errors often abound. GreenQQ programmers rarely have problems giving discriminating text snippets; considerably harder is finding examples for the exceptional cases, which are typically sprinkled sparingly throughout the document. A useful feature of GreenQQ is its ability to find these exceptional cases and generate candidate rules for the programmer to assess, edit if necessary, and include in their rule set. GreenQQ finds these exceptional cases by compiling a list of keywords (proper nouns and literals) that appear in both matched (and therefore classified) and unmatched (still to be classified) portions of the text. Then, for unmatched keyword sequences, it creates a text snippet example that includes a few text tokens to the left and right of the text to be extracted and classifies it according to how it is most often classified in the matched text.

For example, after a run of the initial rules above, GreenQQ finds a text snippet in the Kilbarchan book for which it generates the candidate rule:

[Name] [Robert, in Hilhead $ ^ James (daughter), 8 June] [James].

The user should now should edit this rule, cutting back the context to be less restrictive, yielding:

[Name] [^ James (daughter)] [James].

As it turns out, the abstract template for this instance not only properly classifies names which are parenthetically daughters but also those that are parenthetically any other single word such as “natural” and “posthumous” which also appear in the Kilbarchan book.

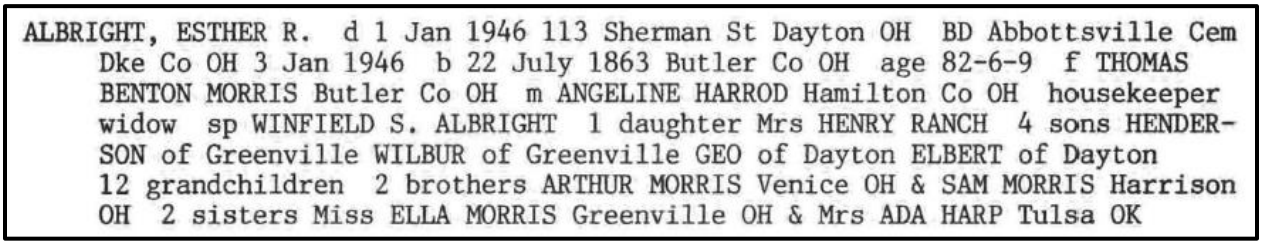
3. Initial Experimental Results

For the 143-page Kilbarchan Parish Record [5], we programmed GreenQQ with ten unique examples. Running the generated rules over the book, GreenQQ extracted 44,996 field values and organized them into 19,436 records (see Table 1). Every field value it extracted was 100% correct. GreenQQ extracted the records with a “Soft” precision of 0.97—meaning that 97% of the field values were correctly grouped into records although a few field values were missing from some of the records. The “Hard” record precision score of 0.87 along with the 1.00 precision score for field value extraction means that 87% of the records were perfect, having all their field values correctly extracted and properly grouped.

It takes only a minute or so to find an example, type in its class name, copy-and-paste the text snippet for the example, and designate the part of the text snippet to be extracted. And, it takes only a few seconds to type in each literal and each syntactic marker designating literals and rules. Programming GreenQQ for the experimental run in Table 1 (14 rules and 4 literals) took about half an hour. Thus, with a half hour’s work, we obtained 19,436 records, 97% of which were correct and 87% were also complete (as judged in a sampling of three randomly chosen pages). In these records, 44,996 field values were associated with 3,959 persons with birth or christening dates, 7,908 marriages, and 7,569 families including 3,959 children.

Table 1. Kilbarchan Experimental Results.

Kilbarchan Accuracy Results (Pages 20, 41, 123)													
Person							Soft: Correct = Correct+Partial			Hard: Incorrect = Partial+Incorrect		
Class / Record	Nr. Rules	Nr. Found in Book	Nr. in Pages	Correct	Partial	Incorrect	Recall	Precision	F-score	Recall	Precision	F-Score
Name	2	3959	92	87	0	0	0.95	1.00	0.97	0.95	1.00	0.97
BirthDate	1	2823	24	18	0	0	0.75	1.00	0.86	0.75	1.00	0.86
ChristeningDate	1	7461	68	57	0	0	0.84	1.00	0.91	0.84	1.00	0.91
Field Values	4	14243	184	162	0	0	0.88	1.00	0.94	0.88	1.00	0.94
Record	1	3959	92	75	12	0	0.95	1.00	0.97	0.82	0.86	0.84
Couple							Soft: Correct = Correct+Partial			Hard: Incorrect = Partial+Incorrect		
Class / Record	Nr. Rules	Nr. Found in Book	Nr. in Pages	Correct	Partial	Incorrect	Recall	Precision	F-score	Recall	Precision	F-Score
Name	1	7908	67	65	0	0	0.97	1.00	0.98	0.97	1.00	0.98
SpouseName	1	4324	55	49	0	0	0.89	1.00	0.94	0.89	1.00	0.94
MarriageDate	3	2610	28	24	0	0	0.86	1.00	0.92	0.86	1.00	0.92
MarriagePlace	1	67	3	1	0	0	0.33	1.00	0.50	0.33	1.00	0.50
Field Values	6	14909	153	139	0	0	0.91	1.00	0.95	0.91	1.00	0.95
Record	1	7908	67	55	9	2	0.96	0.97	0.96	0.82	0.83	0.83
Family							Soft: Correct = Correct+Partial			Hard: Incorrect = Partial+Incorrect		
Class	Nr. Rules	Nr. Found in Book	Nr. in Pages	Correct	Partial	Incorrect	Recall	Precision	F-score	Recall	Precision	F-Score
Parent1	1	7569	69	63	0	0	0.91	1.00	0.95	0.91	1.00	0.95
Parent2	1	4316	58	52	0	0	0.90	1.00	0.95	0.90	1.00	0.95
Child	2	3959	91	85	0	0	0.93	1.00	0.97	0.93	1.00	0.97
Field Values	4	15844	218	200	0	0	0.92	1.00	0.96	0.92	1.00	0.96
Record	1	7569	72	61	1	4	0.86	0.94	0.90	0.85	0.92	0.88
Summary							Soft: Correct = Correct+Partial			Hard: Incorrect = Partial+Incorrect		
Class / Record	Nr. Rules	Nr. Found in Book	Nr. in Pages	Correct	Partial	Incorrect	Recall	Precision	F-score	Recall	Precision	F-Score
Field Values	14	44996	555	501	0	0	0.90	1.00	0.95	0.90	1.00	0.95
Records	3	19436	231	191	22	6	0.92	0.97	0.95	0.83	0.87	0.85
Unique	10	29152										



ALBRIGHT, ESTHER R. d 1 Jan 1946 113 Sherman St Dayton OH BD Abbottsville Cem Dke Co OH 3 Jan 1946 b 22 July 1863 Butler Co OH age 82-6-9 f THOMAS BENTON MORRIS Butler Co OH m ANGELINE HARROD Hamilton Co OH housekeeper widow sp WINFIELD S. ALBRIGHT 1 daughter Mrs HENRY RANCH 4 sons HENDERSON of Greenville WILBUR of Greenville GEO of Dayton ELBERT of Dayton 12 grandchildren 2 brothers ARTHUR MORRIS Venice OH & SAM MORRIS Harrison OH 2 sisters Miss ELLA MORRIS Greenville OH & Mrs ADA HARP Tulsa OK

Figure 2. Record from the Miller Funeral Home in Greenville, Ohio [6].

Table 2. Miller Experimental Results.

Miller Accuracy Results (Pages 296, 318, 363)												
							Soft: Correct = Correct+Partial			Hard: Incorrect = Partial+Incorrect		
Class / Record	Nr. Rules	Nr.Found in Book	Nr. In Pages	Correct	Partial	Incorrect	Recall	Precision	F-score	Recall	Precision	F-score
Name	2	15265	30	29	1	0	1.00	1.00	1.00	0.97	0.97	0.97
DeathDate	1	12629	30	30	0	0	1.00	1.00	1.00	1.00	1.00	1.00
DeathPlace	3	5940	26	19	0	0	0.73	1.00	0.84	0.73	1.00	0.84
BurialPlace	2	4865	7	6	0	0	0.86	1.00	0.92	0.86	1.00	0.92
BurialDate	3	1545	26	7	0	0	0.27	1.00	0.42	0.27	1.00	0.42
BirthDate	1	9608	21	21	0	0	1.00	1.00	1.00	1.00	1.00	1.00
BirthPlace	4	3090	21	9	0	1	0.43	0.90	0.58	0.43	0.90	0.58
SpouseName	3	3802	15	10	1	1	0.73	0.92	0.81	0.67	0.83	0.74
Father	2	6006	21	13	7	0	0.95	1.00	0.98	0.62	0.65	0.63
Mother	2	5846	17	13	4	0	1.00	1.00	1.00	0.76	0.76	0.76
Field Values	23	68596	214	157	13	2	0.79	0.99	0.88	0.73	0.91	0.81
Record	1	15265	30	29	0	1	0.97	0.97	0.97	0.97	0.97	0.97

In a similar experiment, we programmed GreenQQ with 23 examples all from the first page of death and burial records from the 396-page Miller Funeral Home Records [6]. Figure 2 shows a sample record. Since all information of interest in these records relates directly to the deceased person, only one record type is required. As Table 2 shows, GreenQQ extracted 68,596 field values and organized them into 15,265 records.

Text snippet patterns designating field values vary more in Miller than in Kilbarchan. Checking a random sample of three pages, "Soft" and "Hard" recall scores, respectively 0.79 and 0.73, indicate that more text patterns are needed to cover the variability, especially for burial dates and birth places. GreenQQ can help users find these patterns. Record formation has a respectable F-score of 0.97. Record formation depends on accurately identifying field values for the grouping field. For Miller records, "Name" is the grouping field, and the "Soft" and "Hard" F-scores are 1.00 and 0.97 respectively. These high F-scores mean that almost all field values are properly grouped into records. Indeed, in the three pages checked, only one record contained an extraneous field value.

In Table 2, the "Hard" recall for the Father and Mother classes is 0.62 and 0.72 respectively—both low but only because several patterns did not appear on the first page. In its candidate-rule-generation phase, GreenQQ found additional patterns and suggested rules that would extract and classify them. As examples, after editing, the following rules would be added to the rule set, which would have increased the recall for Father and Mother.

[Father] [f Dr. B. F. Zeller] [Dr. B. F. Zeller]
[Father] [f J.C. MENDENHALL] [J.C. MENDENHALL]
[Mother] [m CATHARINE sp] [CATHARINE]
[Mother] [m ELIZA- $ ^ BETH SHULTZ] [ELIZA- $ ^ BETH SHULTZ]

This last pattern involves an end-of-line hyphen. GreenQQ extracts the name as shown. (Downstream in the processing pipeline in which GreenQQ operates, we resolve end-of-line hyphens so that the name becomes ELIZABETH SHULTZ.)

4. Conclusion

GreenQQ is an effective approach to information extraction. Users program GreenQQ by-example (no expertise beyond general computer literacy is necessary). Given some initial examples, GreenQQ can generate candidate rules for consideration. This feature of GreenQQ is particularly useful for finding text patterns that occur far less frequently than a book's prominent text patterns. GreenQQ facilitates quick development of extraction rules.

In our initial experimentation, precision measures have been particularly good—100% for field values in Table 1 and 99% and 91% respectively for "Soft" and "Hard" field values in Table 2. Recall appears to depend on a potentially long tail of pattern variability. Interestingly (although not surprisingly), record-grouping fields tend to have minimal variability and thus higher recall—all over 90% in Table 1 and all over 95% in Table 2. Together, these precision and recall results indicate the potentially high quality yield of information extracted by GreenQQ.

Further work is needed to make GreenQQ useful in practice and to make it live up to its potential: (1) Assess document applicability. (Documents must have semi-structured record patterns, and record grouping fields must have regular text patterns that are easily discriminated.) (2) Create a user-friendly

interface. (See [7] for a proposal.) (3) Resolve issues (e.g. accommodate OCR errors, identify ambiguous rules, unravel intertwined records of the same type, provide for page crossing patterns, and tailor text abstraction for names, dates, and places).

References

[1] https://media.familysearch.org/company-facts.

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

[3] Stanford Named Entity Recognizer (NER), https://nlp.stanford.edu/software/CRF-NER.shtml.

[4] David W. Embley and George Nagy, Green Interaction for Extracting Family Information from OCR'd Books, *Proceedings of the International Workshop on Document Analysis Systems*, Vienna, Austria, April 24, 2018, to appear.

[5] 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.

[6] *Miller Funeral Home Records*, 1917 – 1950, Greenville, Ohio, 1990.

[7] https://fhtw.byu.edu/archive/2018.