ELIJAH, Extracting Genealogy from the Web

David Barney and Rachel Lee
{dbarney, rlee}@whizbang.com
WhizBang! Labs and Brigham Young University

1. Introduction

On-line genealogy is becoming America's latest craze [1]. The LDS church's FamilySearch website contains only a fraction of the information that is available on the Web. People around the world, both members and non-members, have posted family trees and other genealogical information on tens of thousands of websites throughout the Web. Although the genealogical data posted by individuals represents a vast treasure-trove of information, it cannot be searched easily. One has to visit and search each website individually.

When researching one's ancestors, it would be ideal to have a single central index that combines the genealogical information found throughout the Web that, when searched, would point the user to the original source. Ideally, this index would identify and tag the individual pieces of information found on the web pages so that searches could refer to specific fields of information such as name, birthplace, or death-date. For example, one might want to search all genealogical websites for people with a last name of "Jones," born somewhere near London, England, and dying in or about 1850. Furthermore, the individual pieces of information found on each page should be grouped into distinct records corresponding to individuals or families. Searches should return only pages where all search conditions are met by the same record on the page. For example, in the previously mentioned search, the search should avoid returning pages containing one person whose name was "Jones," a different person born in London, and a third person who died around 1850. With so much genealogical information available, a Web-genealogy search engine must allow searches to be as targeted as possible.

Gathering and indexing genealogical information from the Web is not a trivial problem because of the following three reasons: 1) the structure of data on a page 2) the structure of the website 3) the association of the data into records. First, although the basic information such as name, birthplace, and death-date appears on the pages, the structure in which it is presented differs from website to website. These differing structures make it difficult to create global models that determine where the names, dates, and places appear on the page. It is important to understand the structure to solve problem 3, associations. Second, the Web adds another level of difficulty because the data fields corresponding to a single record can appear on different pages, related only by the hyperlink structure, making it hard to group the related data from different pages together. Third, it is important to understand the context of the data fields in order to associate them correctly within an individual record. For example, it might be easy to identify a date, but determining to which person the date belongs and whether the date is associated with a birth, death, or some other event, requires this contextual knowledge.

In this extended abstract we describe ELIJAH (**E**xtracting **L**ineage **I**nformation with **J**ava using **A**utomated **H**euristics), an approach to creating a central index of family trees (pedigree charts) found on the Web. Rather than trying to solve the whole problem of indexing all forms of genealogical information, which includes census data, parish records, land ownership records, etc., we limit ourselves to indexing the family trees that have been published on the Web. Indexing family tree information is a valuable first step toward solving the larger problem of indexing all forms of genealogical data, and solutions to this problem can provide insights into solving the larger problem. We present a brief overview of our approach in Section 2. Results are given in Section 3. Related work is described in Section 4. Section 5 shows how ELIJAH relates to the approaches described in Section 4.

2. Methodology

Our approach takes advantage of an important insight: although there are many different websites containing family tree information, there are a relative few widely-used software packages for publishing family tree information on-line. Thus, much of the family tree information appearing on the Web appears in one of around 100 [2] published formats, each format corresponding to a software package. Our approach to creating a central index of this information is a two-step process. First, we develop a family tree classifier that groups the family trees found on the Web into ***N*** distinct classes – one per published format. Second, we develop extractors to tag the fields and records in the family trees of each format. By first classifying the family trees based upon their published format, we reduce complexity of the extraction task from the very difficult problem of writing extractors to tag fields and records appearing in any published format, to the much more tractable problem of writing ***N*** sets of extractors to tag fields and records, each set of extractors specific to one of the ***N*** published formats.

In order to describe our approach more fully, we present a graphical depiction of the process in figure 1.

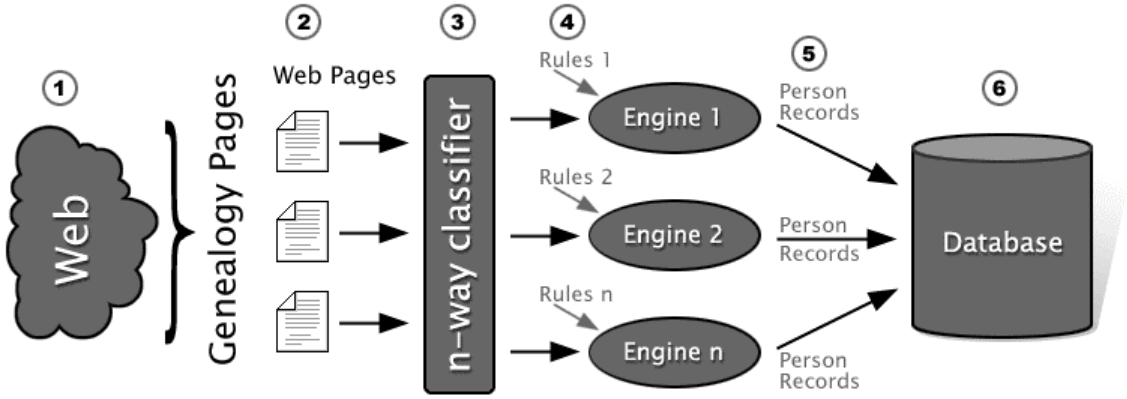

Figure 1: Description of the ELIJAH Process

1. Web pages are gathered from the Internet. In our case we found the pages by hand. Alternatively, a spider with a classifier that distinguishes family tree pages from non-family tree pages could perform this task.
2. The family tree web pages are fed into an ***N***-way classifier.
3. This classifier has a set of hand coded rules to identify each of the formats for which we have developed extraction engines. We developed classification rules to identify fifteen different publishing formats, corresponding to fifteen of the software programs for publishing family tree information including: Family Origin, GedPage, Master Genealogist, Roots Web, HTML Genie, Family Tree Maker, and Ged2HTML. The rule conditions are regular expressions over the HTML of the page. For example, if the regular expression "(?:<hr[^>]*>.*?(?:husband:|wife:|children).*?<hr)" matches, the page is determined to belong to the GedPage format. If a page does not match any of the classification rules, it is not passed on to any extraction engine.
4. Next, the extraction rules specific to the format determined by the classifier are run over the page. We developed a set of extraction rules for each of the fifteen publishing formats identified above. The rules were implemented using regular expressions in Java.
5. The records are extracted. We used simple validation, such as checking that the dates that are extracted match a regular expression that describes dates.
6. The extracted records are put into a searchable database. In our system, the database was an XML file.

3. Results

We ran ELIJAH over 51 websites chosen at random that contained family tree information. Of those, 15 had information inside of paragraphs of text or were written by hand. Our simple extraction-rule approach is unable to extract information from prose, limiting the upper-bound of the percentage of sites from which ELIJAH can extract information to 80%. Of the 36 family-tree-structured (non-prose) websites, we were able to extract data records from 41% of them using our fifteen sets of extraction rules. Looking at the sites we did not have rule sets for, there were two main characteristics: 1) the data was in text or images and 2) the information was contained in an applet (InterneTree). The free text information is not well suited to our techniques since we are relying strongly on html formatting to give us informational clues. The images and applets we are unable to process at all. If we also remove those types of sites we can extract data from 55% of the sites. The other 45% represent cases where 1) the data is in a format for which we have not developed a parser or 2) the parsers are not tuned to extract all the information with high accuracy. The 15 parsers we developed represent a small percentage of the possible formats, yet this 55% represents a significant portion of the extractable data. This percentage will increase as new parsers are developed and existing parsers are improved.

Websites with information in prose or written by hand	15
Websites with family-tree-structured information extracted by ELIJAH	16
Websites with family-tree-structured information not extracted by ELIJAH	13
Websites with insufficient html structure	7
Total	51

Table 1: Breakdown of ELIJAH's Results

4. Related Work

There are two general approaches used to extract information from the Web: global models and site-specific wrappers. At WhizBang! Labs, both approaches are used. Our approach represents a third, middle-ground alternative.

The first common method is a global model. Several information extraction systems powered by WhizBang! Labs such as Flipdog.com (http://www.flipdog.com) – a job search site, and Cora [3] (http://www.cora.whizbang.com) – a search engine for computer science research papers, use a global model. To develop a global model, a large amount of training data is gathered from many different websites and used to train classifiers and extractors based upon various machine learning techniques. These techniques identify patterns for extracting data that can be applied to any website. Because it is general, it is able to effectively find and use similarities. The main disadvantage of this approach is the large amount of time it takes to gather a training set and tune the classifiers and extractors. Also, although it is good in general, on specific sites it may do poorly.

The second common method is a site-specific wrapper. This is a method preferred by companies such as Junglee and discussed in papers by Ashish and Knoblock[4]. A set of extraction rules is generated on a site-by-site basis either by hand or by some form of automatic rule learning. This has the advantage of being highly accurate on the specific websites for which the wrappers have been generated. However, it can be time consuming to construct a wrapper for each site and maintain it if the site's HTML structure changes.

5. Conclusion

Table 2 describes the relative advantages of the approaches discussed above. The ELIJAH method we propose takes advantage of the characteristics of online genealogical information and the strengths of both approaches. While there are a large number of websites with family tree information on them, the information is formatted in a relatively small number of ways. We can use this information to produce a set of wrappers that extracts data from specific formats with high accuracy. This amounts to

creating wrappers that work across many sites. We use a global model to identify which format a specific family tree belongs to. The main advantage of ELIJAH is that it takes relatively little time to build these reusable wrappers that can extract information with high accuracy from a large number of webpages. ELIJAH is able to combine the two common methods to produce a system able to effectively extract information from on-line genealogical data.

	Small # of websites	**Large # of websites**
Small # of distinct page formats	Site-Specific Wrapping	ELIJAH
Large # of distinct page formats	N/A	Global Model

Table 2: Relative advantages of Site-specific Wrapping, Global Models, and ELIJAH

Bibliography

[1] *Time*, April 19, 1999.

[2] Cyndi Howells. Cyndi's List - Software & Computers. http://www.cyndislist.com/software.htm, March 3, 2001.

[3] Andrew McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. A Machine Learning Approach to Building Domain-Specific Search Engines. In *The Sixteenth International Joint Conference on Artificial Intelligence*, 1999.

[4] Naveen Ashish and Craig A. Knoblock. Semi-automatic Wrapper Induction for Internet Information Sources. In *Proceedings of the Second IFCIS International Conference on Cooperative Information Systems*, 1997.