Application of Natural Language Processing Techniques to Enhance Web-Based Retrieval of Genealogical Data

Mary D. Taffet[1]

School of Information Studies
Syracuse University
Syracuse, NY 13244-4100
{mdtaffet@syr.edu}

Introduction

Both genealogy-specific and general search engines used for genealogy searches today are fraught with numerous problems. One word queries are extremely common, and often lead to problems of ambiguity, such as:

- distinguishing a name from a geographical location
- distinguishing a name or geographical location from a common term
- distinguishing a name from a title
- distinguishing a surname from a given name

Multiple word queries quite often bring poor results because, in addition to the problems of ambiguity described above that are present for each search term individually, the set of search terms used are often found in different parts of a document; given two search terms, the search engine might find:

- the first name for one person and the last name for another person
- two first names for two different people
- two last names for two different people

These problems are present because most search engines, both general and genealogy-specific, use a keyword-based approach to Information Retrieval (IR). In keyword-based IR, a document is considered relevant for the query if any one of the search terms appears anywhere in the document; case (upper or lower) is generally ignored. The relevant documents might be ranked based on how many search terms appear, the frequency with which those search terms appear or in what part of the document those search terms appear (i.e. title vs. body), but in general this approach is poorly suited to genealogy searching, except in the rare cases where a surname is so unique that any document containing that term is highly relevant.

The solution to these IR problems is obvious; search terms must be disambiguated and documents must be indexed in such a way that that the disambiguated search terms will match to only the relevant terms and phrases.

While general search engines do not have the capability to disambiguate search terms, many of the genealogy-specific search engines already provide search templates, or fielded search interfaces, that are capable of disambiguating query terms. There is usually a slot for first or given name, last name or surname, location, etc. However, it is often not clear that the disambiguation thus provided is in fact used.

---

[1] The author is a Research Analyst at the Center for Natural Language Processing, which is part of the School of Information Studies. The work reported here was sponsored by GENTECH under the 2001 Scholarship program.

Natural Language Processing (NLP) techniques have been successfully used to create "smart" indexes for other domains, such as business, and other text types, such as news feeds. It is only a matter of time before NLP techniques are applied to the domain of genealogy to create "smart" indexes for web-based genealogy searching; these "smart" indexes will provide the ability to match the disambiguated search terms to the relevant terms and phrases.

Natural Language Processing (NLP) Techniques

Natural Language Processing systems that process text documents (typically unstructured text) involve a number of stages of processing as described below in Table 1.

Table 1: Selected Processing Steps in NLP-based Document Processing System

Step	Description
Cleaning	removes unwanted control characters, etc.
Tokenization	adds spaces to separate text at boundary points between words and surrounding punctuation, or between different punctuation marks
End-of-sentence detection	identifies and marks sentence boundaries
Part-of-speech tagging	adds a tag indicating the part of speech for each token
Phrase detection	identifies and marks units that consist of multiple words – typically they are noun phrases of some type, but need not be
Entity detection	identifies and marks entities, which usually consist of person names, place names, organization or company names and other proper nouns
Categorization	identifies and marks what category something belongs to; typically categorization is used primarily for named entities (i.e. proper nouns)
Event detection	identifies and marks events, which generally correspond to verbs
Relation detection	identifies and marks relations, which are connections between two or more entities or between entities and events
XML or SGML tagging	applies the designated tagging scheme used to markup the document for sentences, phrases, entities, categories, events, relations, etc.
Extraction	the identified entities, events, relations, and any other identified concepts (like dates) are extracted from the document and stored externally

The "smart" indexes are typically composed of the extracted entities, events and relations and any other concepts that were also extracted, like dates.

GENTECH Project Experiment and Results

My GENTECH 2001 Scholarship project [1] consisted of obtaining a document collection, marking it up using proven NLP techniques, evaluating the tagging results, and evaluating the improvement in retrieval results gained as a result of the tagging.

A collection of 360 unstructured plain text documents was obtained from the Castor Association of America [2]; of these, 348 documents were usable for this project. First the document collection was cleaned and tokenized, followed by end-of-sentence detection. Next Eric Brill's Transformation-Based Part-Of-Speech (POS) tagger [3] was trained on the documents; initially two learning cycles were run. The only phrase detection performed was that for person names and place names (which are named entities) and dates; search patterns were used for this detection (see Table 1). For person names, the first, middle and surnames were identified and marked, as were name suffixes like "Jr." or "Sr." and name

prefixes/titles like “Mr.” or “Mrs.”. Also extracted were names of couples, including the maiden name of the wife. For place names, names of townships (or localities), counties and states were identified and marked. For dates, the day, month and year were identified and marked. Categorization was not done explicitly, but categories were nevertheless captured by the tags used to mark the various concepts. Given the short development and testing time for this project, neither events nor relations were identified or marked. The SGML markup scheme used for this project is shown in Table 2.

Table 2: Search Patterns and SGML Markup Tags

Concept	Sample search patterns	Markup scheme
Person	• First Middle Last born • married First Middle Last • my son, First Middle Last,	<PERSON> <TITLE>**Mr.** </TITLE> <FNAME>**First** </FNAME> <MNAME>**Middle**</MNAME> <LNAME>**Last** </LNAME> <SUFFIX>**Jr.** </SUFFIX> </PERSON>
Couple	• son of First1 Middle1 and First2 Middle2 (Maiden) Last	<COUPLE> <FNAME>**First1** </FNAME> <MNAME>**Middle1** </MNAME> **and** <FNAME>**First2** </FNAME> <MNAME>**Middle2** </MNAME> <MAIDEN>**Maiden**</MAIDEN> <LNAME>**Last** </LNAME> </COUPLE>
Place – Township	• in/of/from/, X Twp • in/of/from/, Y County • in/of/from/, State • in/of/from/, X Township of Y County, State	<PLACE> <TWP>**X** </TWP> <COUNTY>**Y** </COUNTY> <STATE>**State** </STATE> </PLACE>
Place – Locality	• in/of/from/, L, Y County, State	<PLACE> <LOCALE>**L** </LOCALE> <COUNTY>**Y** </COUNTY> <STATE>**State** </STATE> </PLACE>
Date	• dd Month yyyy • Month yyyy • dd Month • Month • about yyyy • probably yyyy • in yyyy	<DATE> <DAY>**dd** </DAY> <MONTH>**Month** </MONTH> <YEAR>**yyyy** </YEAR> </DATE>

Three indexes were created. One was a keyword index created just after the end-of-sentence detection phase. The other two indexes were “smart” indexes consisting only of the SGML-tagged elements shown above (person, couple, place, date). One of the “smart” indexes contained the whole SGML-tagged phrase as a unit and the other “smart” index consisted of each SGML-tagged part as a separate entry (i.e. one entry for FNAME, one for MNAME, one for LNAME, etc.).

Due to time constraints, the evaluation of tagging results was based on one unseen document. The evaluation measures used were recall/coverage (#concepts correctly tagged/#concepts present in the document) and precision/accuracy (#concepts correctly tagged/#concepts tagged); the evaluation results are as shown below in Table 3.

Table 3: Tagging Results Before Corrections

Concept	Recall/Coverage	Precision/Accuracy
Names (person and couple)	13/14 = **92.9%**	13/14 = **92.9%**
Places	1/3 = **33.3%**	1/1 = **100.0%**
Dates	16.66/18 = **92.5%**	16.66/17 = **98.0%**

The errors were due primarily to part-of-speech tagging errors and search pattern errors, though one error was due to the presence of a typographical error in the original document ("1s983" instead of "1983"). After running a third learning cycle for the Brill POS-tagger and correcting the search pattern errors, the tagging results improved as shown below in Table 4.

Table 4: Tagging Results After Corrections

Concept	Recall/Coverage	Precision/Accuracy
Names (person and couple)	14/14 = **100.0%**	14/14 = **100.0%**
Places	3/3 = **100.0%**	3/3 = **100.0%**
Dates	17.66/18 = **98.1%**	17.66/18 = **98.1%**

To evaluate the improvement in search retrieval results, 5 searches were run – (1) surname only, (2) middle name only, (3) township name only, (4) first and last name, (5) first, middle and last name. These five searches were run against both the keyword index and the "smart" indexes. The results, showing the top 5 candidates for each type of search, are as shown in Table 5. If documents are found, they are listed in decreasing order of search term frequency; if no documents are found, the search returns "Sorry, search term not found".

Table 5: Search Results Improvement After NLP-based SGML Tagging

#	Keyword search query	Keyword search results – top 5	"Smart" search query	"Smart" search results – top 5
1	Marion	10-01-02-01 8 10-01-01-02-14 5 10-01-02-05 5 10-01-02-06 5 10-01-02-07 5	name = lname=Marion	Sorry, search term not found
2	Marion	10-01-02-01 8 10-01-01-02-14 5 10-01-02-05 5 10-01-02-06 5 10-01-02-07 5	name = mname=Marion	10-07-01-05-01-07 2 10-01-07 1 10-01-07-02 1 10-01-07-02-03 1 10-01-07-02-07 1
3	Marion	10-01-02-01 8 10-01-01-02-14 5 10-01-02-05 5 10-01-02-06 5 10-01-02-07 5	place = twp=Marion_Township	10-01-02-09-01-02 2 10-01-02-15 2 10-01-02-01 1 10-01-02-05 1 10-01-02-07 1
4	Benjamin Kaster	10-04-02-01 86 10-04-01 52 10-05 43 10-01-02-01 40 10-04 39	name = fname=Benjamin lname=Kaster	10-05 2 10 1 10-01 1 10-01-02-06 1 10-01-06 1
5	Sarah Jane Kaster	10-04-02-01 90 10-04-01 43 10-01-02-01 39 10-01-02-15 31 10-01-06-07 24	name = fname=Sarah mname=Jane lname=Kaster	10-01-02-02 1

As can be seen from the search results shown in Table 5, the keyword search did not distinguish between the appearance of "Marion" as a surname, middle name or township name; it gave the same results for searches 1 through 3. The "smart" search on the other hand was able to distinguish between these different concepts; in two cases it found documents that matched the desired concept (searches 2 and 3) and in one case it recognized that the concept did not exist in the document collection (search 1). For searches 4 and 5, the keyword-based search results are influenced primarily by the frequency of the surname "Kaster" in the document collection. On the other hand, for searches 4 and 5 the "smart" search was able to identify those documents that actually contained the person concepts "Benjamin Kaster" and "Sarah Jane Kaster". The keyword search tends to return a lot of documents that are in general not relevant to the search query. The "smart" search in general returns fewer documents and they are always relevant to the search query.

Conclusion

Though my GENTECH 2001 Scholarship project was a small-scale study limited by time constraints (development and testing time was about 3 weeks) and resource constraints (I wrote my own limited search function), it nevertheless clearly demonstrates how the use of NLP techniques to automatically tag genealogical documents can greatly improve the quality of search results. If applied to the web search

environment, this technology has the capability of revolutionizing the way we search for genealogy information on the Internet. Perhaps one day the Castor Association of America will be able to offer their vast archives [4] online via the web with a smart search interface.

References

[1] Taffet, Mary D. (2001). GENTECH 2001 Scholarship Proposal: Automatic Tagging of Genealogical Data to Enhance Web-based Retrieval. Available at: <http://web.syr.edu/~mdtaffet/GENTECH_Scholarship_Proposal.htm>.

[2] The Castor Association of America. http://maverik.rootsweb.com/caoa/.

[3] Brill, Eric. (1995). Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging. Computational Linguistics. Available at: <http://www.cs.jhu.edu/~brill/CompLing95.ps>.

[4] CAOA Archives. http://maverik.rootsweb.com/caoa/sr-arch.htm.