An integrated system for processing information from genealogical text

Merrill Hutchison, Tim Richards, William Taysom, and Deryle Lonsdale

This presentation introduces an integrated software system that has been designed and implemented to allow processing of difficult text. The genealogical text subdomain often exhibits characteristics that do not occur in more general written language. The system we describe has been developed specifically to address the extraction and analysis of information from such specialized text.

First, we survey the nonstandard or exaggerated linguistic characteristics that English-language genealogical text (and indeed that of other languages) often exhibits. For example, in English genealogical prose frequent repetition of subject pronouns is avoided---they are simply dropped, though this would usually be considered ungrammatical except in diaries. Also, genealogical text frequently mentions names, dates, and places in ways that cause problems for traditional natural language processing (NLP) systems. We briefly illustrate how variation from grammatical norms is also common in other languages for genealogical text, though for this talk we focus on English. We discuss how this type of prose is typically preprocessed and tokenized, and then mention how our approach is implemented as the first stage in our integrated system. The result of our integrated approach, that of preprocessing raw genealogical text, is render it more amenable to subsequent linguistic-based treatment.

We next discuss the central component of the system and the various modules that have been integrated into it. The core system has been built on the Soar intelligent agent architecture. The Soar system has been used in a wide array of data-oriented agent-directed applications including robotic team coordination, complex task simulation, and intelligent tutoring. Whereas a comprehensive natural-language processing system (NL-Soar) has already been built on the Soar architecture, its use for processing genealogical text is not possible given the difficulties mentioned above. Accordingly, a more general and robust approach was needed in order to implement a Soar-based solution. The solution was to integrate into the Soar system a more robust and flexible parser for English. We discuss how the integration was achieved, and how it is advantageous over more traditional methods.

Soar-based processing of genealogical text thus begins by parsing the incoming preprocessed text; this step is performed via a link-based dependency parser. This parser, the Link Grammar engine, has been used for such functions as information extraction, shallow parsing, and machine translation, and particularly suited where grammaticality is an issue. Work on this project involved integrating this component into Soar via an interface built with the Toolkit Command Language (Tcl) and C-based API's. Though the grammar could be run in isolation, its integration with the Soar context is important, because the next stage crucially relies on these parsing results, and is best done within the Soar framework.

After the parse has been performed and the best one selected according to several system-internal criteria, the parser's information is fed into the working memory of Soar agent.

This enables the agent to perform the semantic and pragmatic processing stage, which has been specially implemented via Discourse Representation Theory (DRT). DRT provides a model-theoretic approach to discourse-level operations such as predication, object identity, pronominal referent resolution, and anaphor treatment. All of these functions are crucial to extracting important information from text, and genealogical prose is particularly rich in these areas. We illustrate how DRT-based processing leverages link-parse output, the types of data it generates, and its extraction of pertinent genealogical information. Mention is made of how it represents an improvement over traditional extraction and parsing approaches.

When relevant information has been identified by the system and encoded into appropriate data representations, the agent outputs information into one of a variety of possible formats. We illustrate how the Soar system, after having carried out a DRT analysis, converts and outputs genealogical information to a GEDCOM file. We also mention other possible output formats, and potential future Soar-based applications that involve genealogical information.

This presentation thus gives a survey of the overall processing approach, the system architecture, the data formats, the knowledge sources, and the types of information extracted and output by the system. Functionality and coverage of the system are demonstrated with respect to a popular and particularly valuable, yet linguistically idiosyncratic and problematic resource used for American (specifically New England) genealogy.