TOOLS AND TECHNIQUES FOR RAPID PORTING

Joe McCarthy
University of Massachusetts
Computer Science Department
Box 34610
Amherst, MA 01003-4610
jmccarthy@cs.umass.edu

Each of the four presentations in this special topic session focused on issues that arise in porting an information extraction system to a new domain or on specific tools that are used to accomplish this task.

Charlie Dolan, from Hughes Research Laboratories, discussed some of the difficulties in using trainable components in an information extraction system. The UMass/Hughes system used six different trainable components in their MUC5 system; portability between the EJV and EME domains was achieved partly through retraining these components. One of these components, the Trainable Template Generator (TTG), contained 33 different decision trees, each used to establish either a string-fill or set-fill slot in a template object or a relational link between template objects (a pointer slot). One of the issues that came up in the design of TTG was how to configure and manage a "multi-classifier" containing a forest of decision trees.

Another issue that arose in the context of the UMass/Hughes system was what constitutes the "corpus". While the training of every trainable component was based on the texts and, in some cases, the key templates, from either the EJV or EME corpus, each one had a different view of the corpus. All components used some processed form of the raw texts and/or templates for training and most used very particular segments of processed text as their training material.

The last issue highlighted in this presentation was the difficulty of defining the criteria used by humans to make classifications in their preparation of training materials. There was considerable debate among the development team members as to what constitutes an appositive construction for the trainable appositive classifier, or how to distinguish various verb form part-of-speech tags for the trainable part-of-speech tagger (OTB). The debate usually could not be resolved until some material had already been prepared and examples of the difficult cases had been seen, which often entailed a revision of some of the training material once criteria had been refined.

Barry Friedson, of Martin Marrietta Corporation, described a set of tools used by the the GE/MMC-CMU team for adapting their information extraction system, SHOGUN, to new domains. They have developed their own version of the scoring program, which provides a more focused, interactive, evaluation of their system during processing of a text. It is also more flexible than the official scoring program used in MUC5 in that it can work with either key templates or annotated text.

A collated keyword-in-context (KWIC) browser allows inspection of the contexts in which important words are used in the text. Lexical patterns that are associated with relevant information can be identified based on the output of the browser. A future extension of this tool will permit automatic induction of such patterns. The Term Generator was another tool that made use of the corpus. This tool used a statistical analysis of both the texts and the answer keys to make a selection of the best product/service slot-fill in the EJV and JJV response templates, which improved system performance on this slot in both languages.

NL Grep takes a potential pattern used by the GE/MMC-CMU information extraction system and returns all instantiations of that pattern in the texts. This provides system developers with feedback on how effective these patterns are at extracting relevant information from the texts, identifying patterns that may need further refinement.

The Workbench was one of the tools shown at the demonstration session of MUC5. Intended for use

by information analysts, this tool allows an analyst to trace the execution of the extraction system, tune the configuration of the system to maximise either recall or precision, and permit analyst intervention in order to correct mistakes made by the system.

The PAKTUS system, presented by Bruce Loatman of PRC, Inc., uses a network of case-frames to represent the relevant information extracted from a document. In order to generate task-specific output (such as MUC5 templates) from this generic internal representation, PAKTUS contains a graphical user interface (GUI) that permits a user to map the nodes in a network of case-frames into template objects and slot-fills.

The user provides a sample sentence from the corpus for PAKTUS to parse, creating its case-frame representation of the information in the sentence. The user then identifies which fragments in the case-frame are relevant to a specific template object and which of these fragments are optional to the instantiation of such a template object. For each relevant fragment in the case-frame, the user maps the fragment to a slot in the template object. The pattern for the template object is displayed for confirmation, then applied to the original sentence, so that the resulting template can be displayed.

PRC now uses this tool for the generation of all mapping rules, a task that was once done manually. For the EJV domain in MUC5, 147 rules were used to map case-frame fragments into templates.

Ralph Weishedel presented a tool used by BBN's PLUM system that uses a quick, high-level categorization of nouns and verbs to improve the accuracy of the patterns used to extract information from texts. The categories are based on the information requirements of the domain and task. Porting to a new domain may involve redefining the categories and recategorising nouns and verbs found in a corpus of texts from the domain.

The PLUM system creates a word co-occurence frequency matrix, based on a finite state pattern matcher applied to segmented sentences, with part-of-speech labels, taken from the corpus. The accuracy of the resulting patterns is nearly doubled when a mutual information statistical model employing the categorisation of nouns and verbs is used to collapse the rows and columns in this matrix.

In an experiment from the JJV domain, randomly selected subsets of patterns generated both with and without the model were evaluated. 44% of the patterns generated without the aid of the model were judged to be accurate, while 87% of the patterns generated with the model were deemed accurate.