A Distributed Natural Language Processing Architecture for Interactive Parse Annotation

Baden Hughes
Department of Computer Science
and Software Engineering
The University of Melbourne
Victoria 3010, Australia
badenh@csse.unimelb.edu.au

James R. Curran
School of Information Technologies
University of Sydney
NSW 2006, Australia
james@it.usyd.edu.au

Annotating sentences with syntactic parse trees is perhaps the most complex and effort intensive type of linguistic annotation. The time and expense of developing parsed corpora is almost prohibitive, despite the significant utility of syntactically parsed corpora for a wide range of natural language processing applications. Consequently there are only a small number of such corpora, including the Penn Treebank (Marcus et al., 1994), the German TiGer Corpus (Skut et al., 1997) and more recently the LinGO Redwoods Treebank (Oepen et al., 2002). In addition to the small number, these corpora are also limited in size, typically around one million words of text.

Unfortunately, the statistical approaches to parsing which have been most successful rely heavily on both the quality and quantity of syntactically annotated resources. Such approaches are very sensitive to the statistical properties of the corpus, and so a parser trained on one genre may perform badly on another (Gildea, 2001). Another major problem with parsed corpora is that they must, at least to some extent, follow a particular syntactic theory or formalism. This is a major difficulty for two reasons: firstly, it means we need separate annotated corpora for each formalism; and secondly, it means that comparing parser evaluations across formalisms is difficult. Fully automated conversion of trees between formalisms is difficult because each analyses certain constructs in idiosyncratic ways. An example is CCGbank (Hockenmaier and Steedman, 2002), a treebank of Combinatory Categorial Grammar (Steedman, 2000) derivations which were converted semi-automatically from the Penn Treebank trees. The result still required laborious editing to produce idiomatic CCG derivations (Hockenmaier, 2003).

Our motivating task is to provide the infrastructure by which the creation of a new corpus of CCG derivations can be conducted more efficiently. We face three key problems: 1) selecting sentence to annotate which creates the most *useful* corpus for statistical parsers. 2) maximising the *annotator efficiency* and *minimising error*; 3) allowing distributed annotators to *share expertise*.

In doing so, we demonstrate the considerable advantage of *mixed initiative annotation* Day et al., 1997, (where the division of labour between computational facilities and human effort is coordinated for increased efficiency) which has become an increasingly common methodology for the preparation of large corpora. This contrasts with other mixed initiative applications to date which have largely decoupled human and machine effort.

The selection problem (1) is addressed using *active learning*. Active learning involves computing which training instances provide the most new information to one (or more) machine learners (Cohn et al., 1995; Dagan and Engelson, 1995). The annotators become oracles answering specific queries posed by the learners.

The annotation problem (2) is addressed by *interactive correction* of the output of our statistical CCG parser. Annotators interactively add constraints to the parser which will return the most probable parse satisfying the constraints.

The distributed expertise problem (3) is addressed using a *workflow manager*. Annotators are able to add comments and questions to partial derivations and have them sent to (potentially remote) experienced annotators for verification. The workflow manager handles scheduling for the active learner.

In our system we are using the CCG parser (Clark and Curran, 2004b), which uses a log-linear model over normal-form derivations to select an analysis. The parser takes a Part-of-Speech (POS) tagged sentence as input with a set of one more more categories assigned to each word. A CCG supertagger (Clark and Curran, 2004a) assigns the lexical categories, using a log-

linear model to identify the most probable categories. The same work shows how dynamic use of the supertagger — starting off with a small number of categories assigned to each word and gradually increasing the number until an analysis is found — can lead to a highly efficient and robust parser.

The parser model parameters are estimated using a discriminative method, that is, one which requires statistics across all incorrect parses for a sentence as well as the correct parse. Since an automatically extracted CCG grammar can produce an extremely large number of parses, the use of a supertagger is crucial in limiting the total number of parses for the training data to a computationally manageable number.

The system architecture for distributed annotation and parsing with active learning has three main modules.

The Visualization and Analysis module provides the end user interface by which a human annotator can review and revise the parser output. The visualisation GUI is implemented in wxPython (Dunn, 2005), an extension of the cross-platform GUI toolkit wxWidgets (Smart et al., 2005) for Python. wxWidgets is particularly notable for its use of native graphical components for a given operating system platform, allowing the interface a native look and feel when run on Windows, Mac or Linux environments.

The Workflow Management module has three main roles: first to interact with the Visualization and Analysis, providing parses to be visualised and refined; second to manage the user and tasks in the process of analysis; and third to interact with the Computational Management module by instantiating the active learning framework for incremental parsing of the corpus data, and subsequent grid execution.

The Computational Management module has two sub-modules. The Active Learning submodule allows for incremental application of refined parses as training data for subsequent iterations of the parser. The Grid sub-module handles low level execution including the queuing, dispatch and execution of analysis tasks, and fetching the results from the distributed computation environment.

An overall strength of our architecture is that the annotators can be both geographically and topologically removed from the workflow manager; which in turn can be separated from the computational grid. The messaging framework adopted is SOAP based - all communication between the Visualization and Analysis module, the Workflow Management module and the Computational Management module are implemented using this lightweight messaging protocol.

References

Stephen Clark and James R. Curran. 2004a. The importance of supertagging for wide-coverage CCG parsing. In *Proc. of the 20th COLING*, pages 282–288, Geneva, Switzerland.

Stephen Clark and James R. Curran. 2004b. Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42nd Annual Meeting of the ACL*, pages 103–110.

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. 1995. Active learning with statistical models. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 705–712. MIT Press.

Ido Dagan and Sean P. Engelson. 1995. Committee-based sampling for training probabilistic classifiers. In *Proc. of the ICML*, pages 150–157.

David Day, John Aberdeen, Lynette Hirschman, Robyn Kozierok, Patricia Robinson, and Marc Vilain. 1997. Mixed-initiative development of language processing systems. In *Proc. of the 5th conference on Applied NLP*, pages 348–355.

Robin Dunn. 2005. wxPython toolkit. `http://www.wxpython.org`.

Daniel Gildea. 2001. Corpus variation and parser performance. In *Proceedings of the EMNLP Conference*, pages 167–202, Pittsburgh, PA.

Julia Hockenmaier and Mark Steedman. 2002. Acquiring compact lexicalized grammars from a cleaner treebank. In *Proceedings of the 3rd LREC Conference*, pages 1974–1981, Las Palmas, Spain.

Julia Hockenmaier. 2003. *Data and Models for Statistical Parsing with Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. 2002. The LinGO Redwoods Treebank: Motivation and preliminary applications. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1253–1257, Taipei, Taiwan.

Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. 1997. An annotation scheme for free word order languages. In *Proceedings of the 5th ACL Conference on Applied NLP*, pages 88–95, Washington, DC.

Julian Smart, Kevin Hock, and Stefan Csomor. 2005. *Cross-Platform GUI Programming with wxWidgets*. Prentice Hall.

Mark Steedman. 2000. *The Syntactic Process*. The MIT Press, Cambridge, MA.