Linguistically Motivated Large-Scale NLP with C&C and Boxer

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

Stephen Clark
Computing Laboratory
Oxford University
Wolfson Building, Parks Road
Oxford, OX1 3QD, UK
stephen.clark@comlab.ox.ac.uk

Johan Bos
Dipartimento di Informatica
Università di Roma “La Sapienza”
via Salaria 113
00198 Roma, Italy
bos@di.uniroma1.it

1 Introduction

The statistical modelling of language, together with advances in wide-coverage grammar development, have led to high levels of robustness and efficiency in NLP systems and made linguistically motivated large-scale language processing a possibility (Matsuzaki et al., 2007; Kaplan et al., 2004). This paper describes an NLP system which is based on syntactic and semantic formalisms from theoretical linguistics, and which we have used to analyse the entire Gigaword corpus (1 billion words) in less than 5 days using only 18 processors. This combination of detail and speed of analysis represents a breakthrough in NLP technology.

The system is built around a wide-coverage Combinatory Categorial Grammar (CCG) parser (Clark and Curran, 2004b). The parser not only recovers the local dependencies output by treebank parsers such as Collins (2003), but also the long-range depdendencies inherent in constructions such as extraction and coordination. CCG is a lexicalized grammar formalism, so that each word in a sentence is assigned an elementary syntactic structure, in CCG’s case a lexical category expressing subcategorisation information. Statistical tagging techniques can assign lexical categories with high accuracy and low ambiguity (Curran et al., 2006). The combination of finite-state supertagging and highly engineered C++ leads to a parser which can analyse up to 30 sentences per second on standard hardware (Clark and Curran, 2004a).

The C&C tools also contain a number of Maximum Entropy taggers, including the CCG supertagger, a POS tagger (Curran and Clark, 2003a), chunker, and named entity recogniser (Curran and Clark, 2003b). The taggers are highly efficient, with processing speeds of over 100,000 words per second.

Finally, the various components, including the morphological analyser morpha (Minnen et al., 2001), are combined into a single program. The output from this program — a CCG derivation, POS tags, lemmas, and named entity tags — is used by the module Boxer (Bos, 2005) to produce interpretable structure in the form of Discourse Representation Structures (DRSs).

2 The CCG Parser

The grammar used by the parser is extracted from CCGbank, a CCG version of the Penn Treebank (Hockenmaier, 2003). The grammar consists of 425 lexical categories, expressing subcategorisation information, plus a small number of combinatory rules which combine the categories (Steedman, 2000). A Maximum Entropy supertagger first assigns lexical categories to the words in a sentence (Curran et al., 2006), which are then combined by the parser using the combinatory rules and the CKY algorithm.

Clark and Curran (2004b) describes log-linear parsing models for CCG. The features in the models are defined over local parts of CCG derivations and include word-word dependencies. A disadvantage of the log-linear models is that they require cluster computing resources for practical training (Clark and Curran, 2004b). We have also investigated perceptron training for the parser (Clark and Curran, 2007b), obtaining comparable accuracy scores and similar training times (a few hours) compared with the log-linear models. The significant advantage of

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 33–36,
Prague, June 2007. ©2007 Association for Computational Linguistics

the perceptron training is that it only requires a single processor. The training is online, updating the model parameters one sentence at a time, and it converges in a few passes over the CCGbank data.

A packed chart representation allows efficient decoding, with the same algorithm — the Viterbi algorithm — finding the highest scoring derivation for the log-linear and perceptron models.

2.1 The Supertagger

The supertagger uses Maximum Entropy tagging techniques (Section 3) to assign a set of lexical categories to each word (Curran et al., 2006). Supertagging has been especially successful for CCG: Clark and Curran (2004a) demonstrates the considerable increases in speed that can be obtained through use of a supertagger. The supertagger interacts with the parser in an adaptive fashion: initially it assigns a small number of categories, on average, to each word in the sentence, and the parser attempts to create a spanning analysis. If this is not possible, the supertagger assigns more categories, and this process continues until a spanning analysis is found.

2.2 Parser Output

The parser produces various types of output. Figure 1 shows the dependency output for the example sentence *But Mr. Barnum called that a worst-case scenario.* The CCG dependencies are defined in terms of the arguments within lexical categories; for example, $\langle (S[dcl]\backslash NP_1)/NP_2, 2\rangle$ represents the direct object of a transitive verb. The parser also outputs grammatical relations (GRs) consistent with Briscoe et al. (2006). The GRs are derived through a manually created mapping from the CCG dependencies, together with a python post-processing script which attempts to remove any differences between the two annotation schemes (for example the way in which coordination is analysed).

The parser has been evaluated on the predicate-argument dependencies in CCGbank, obtaining labelled precision and recall scores of 84.8% and 84.5% on Section 23. We have also evaluated the parser on DepBank, using the Grammatical Relations output. The parser scores 82.4% labelled precision and 81.2% labelled recall overall. Clark and Curran (2007a) gives precison and recall scores broken down by relation type and also compares the performance of the CCG parser with the RASP parser (Briscoe et al., 2006).

```
Mr._2 N/N_1 1 Barnum_3
called_4 ((S[dcl]\NP_1)/NP_2)/NP_3 3 that_5
worst-case_7 N/N_1 1 scenario_8
a_6 NP[nb]/N_1 1 scenario_8
called_4 ((S[dcl]\NP_1)/NP_2)/NP_3 2 scenario_8
called_4 ((S[dcl]\NP_1)/NP_2)/NP_3 1 Barnum_3
But_1 S[X]/S[X]_1 1 called_4

(ncmod _ Barnum_3 Mr._2)
(obj2 called_4 that_5)
(ncmod _ scenario_8 worst-case_7)
(det scenario_8 a_6)
(dobj called_4 scenario_8)
(ncsubj called_4 Barnum_3 _)
(conj _ called_4 But_1)
```

Figure 1: Dependency output in the form of CCG dependencies and grammatical relations

3 Maximum Entropy Taggers

The taggers are based on Maximum Entropy tagging methods (Ratnaparkhi, 1996), and can all be trained on new annotated data, using either GIS or BFGS training code.

The POS tagger uses the standard set of grammatical categories from the Penn Treebank and, as well as being highly efficient, also has state-of-the-art accuracy on unseen newspaper text: over 97% per-word accuracy on Section 23 of the Penn Treebank (Curran and Clark, 2003a). The chunker recognises the standard set of grammatical “chunks”: NP, VP, PP, ADJP, ADVP, and so on. It has been trained on the CoNLL shared task data.

The named entity recogniser recognises the standard set of named entities in text: person, location, organisation, date, time, monetary amount. It has been trained on the MUC data. The named entity recogniser contains many more features than the other taggers; Curran and Clark (2003b) describes the feature set.

Each tagger can be run as a “multi-tagger”, potentially assigning more than one tag to a word. The multi-tagger uses the forward-backward algorithm to calculate a distribution over tags for each word in the sentence, and a parameter determines how many tags are assigned to each word.

4 Boxer

Boxer is a separate component which takes a CCG derivation output by the C&C parser and generates a semantic representation. Boxer implements a first-order fragment of Discourse Representation Theory, DRT (Kamp and Reyle, 1993), and is capable of generating the box-like structures of DRT known as Discourse Representation Structures (DRSs). DRT is a formal semantic theory backed up with a model theory, and it demonstrates a large coverage of linguistic phenomena. Boxer follows the formal theory closely, introducing discourse referents for noun phrases and events in the domain of a DRS, and their properties in the conditions of a DRS.

One deviation with the standard theory is the adoption of a Neo-Davidsonian analysis of events and roles. Boxer also implements Van der Sandt's theory of presupposition projection treating proper names and defininite descriptions as anaphoric expressions, by binding them to appropriate previously introduced discourse referents, or accommodating on a suitable level of discourse representation.

4.1 Discourse Representation Structures

DRSs are recursive data structures — each DRS comprises a domain (a set of discourse referents) and a set of conditions (possibly introducing new DRSs). DRS-conditions are either basic or complex. The basic DRS-conditions supported by Boxer are: equality, stating that two discourse referents refer to the same entity; one-place relations, expressing properties of discourse referents; two place relations, expressing binary relations between discourse referents; and names and time expressions. Complex DRS-conditions are: negation of a DRS; disjunction of two DRSs; implication (one DRS implying another); and propositional, relating a discourse referent to a DRS.

Nouns, verbs, adjectives and adverbs introduce one-place relations, whose meaning is represented by the corresponding lemma. Verb roles and prepositions introduce two-place relations.

4.2 Input and Output

The input for Boxer is a list of CCG derivations decorated with named entities, POS tags, and lemmas for nouns and verbs. By default, each CCG derivation produces one DRS. However, it is possible for one DRS to span several CCG derivations; this enables Boxer to deal with cross-sentential phenomena such as pronouns and presupposition.

Boxer provides various output formats. The default output is a DRS in Prolog format, with discourse referents represented as Prolog variables. Other output options include: a flat structure, in which the recursive structure of a DRS is unfolded by labelling each DRS and DRS-condition; an XML format; and an easy-to-read box-like structure as found in textbooks and articles on DRT. Figure 2 shows the easy-to-read output for the sentence *But Mr. Barnum called that a worst-case scenario.*

```
 ______________________
| x0 x1 x2 x3          |
|______________________|
| named(x0,barnum,per) |
| named(x0,mr,ttl)     |
| thing(x1)            |
| worst-case(x2)       |
| scenario(x2)         |
| call(x3)             |
| but(x3)              |
| event(x3)            |
| agent(x3,x0)         |
| patient(x3,x1)       |
| theme(x3,x2)         |
|______________________|
```

Figure 2: Easy-to-read output format of Boxer

The semantic representations can also be output as first-order formulas. This is achieved using the standard translation from DRS to first-order logic (Kamp and Reyle, 1993), and allows the output to be pipelined into off-the-shelf theorem provers or model builders for first-order logic, to perform consistency or informativeness checking (Blackburn and Bos, 2005).

5 Usage of the Tools

The taggers (and therefore the parser) can accept many different input formats and produce many different output formats. These are described using a "little language" similar to C printf format strings. For example, the input format %w|%p \n indicates that the program expects word (%w) and POS tag (%p) pairs as input, where the words and POS tags are separated by pipe characters, and each word-POS tag pair is separated by a single space, and whole sentences are separated by newlines (\n). Another feature of the input/output is that other fields can be read in which are not used in the tagging process, and also form part of the output.

The C&C tools use a configuration management system which allows the user to override all of the default parameters for training and running the taggers and parser. All of the tools can be used as standalone components. Alternatively, a pipeline of the tools is provided which supports two modes: local file reading/writing or SOAP server mode.

6 Applications

We have developed an open-domain QA system built around the C&C tools and Boxer (Ahn et al., 2005). The parser is well suited to analysing large amounts of text containing a potential answer, because of its efficiency. The grammar is also well suited to analysing questions, because of CCG's treatment of long-range dependencies. However, since the CCG parser is based on the Penn Treebank, which contains few examples of questions, the parser trained on CCGbank is a poor analyser of questions. Clark et al. (2004) describes a porting method we have developed which exploits the lexicalized nature of CCG by relying on rapid manual annotation at the lexical category level. We have successfully applied this method to questions.

The robustness and efficiency of the parser; its ability to analyses questions; and the detailed output provided by Boxer make it ideal for large-scale open-domain QA.

7 Conclusion

Linguistically motivated NLP can now be used for large-scale language processing applications. The C&C tools plus Boxer are freely available for research use and can be downloaded from http://svn.ask.it.usyd.edu.au/trac/candc/wiki.

Acknowledgements

James Curran was funded under ARC Discovery grants DP0453131 and DP0665973. Johan Bos is supported by a "Rientro dei Cervelli" grant (Italian Ministry for Research).

References

Kisuh Ahn, Johan Bos, James R. Curran, Dave Kor, Malvina Nissim, and Bonnie Webber. 2005. Question answering with QED at TREC-2005. In *Proceedings of TREC-2005*.

Patrick Blackburn and Johan Bos. 2005. *Representation and Inference for Natural Language. A First Course in Computational Semantics*. CSLI.

Johan Bos. 2005. Towards wide-coverage semantic interpretation. In *Proceedings of IWCS-6*, pages 42–53, Tilburg, The Netherlands.

Ted Briscoe, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of the Interactive Demo Session of COLING/ACL-06*, Sydney.

Stephen Clark and James R. Curran. 2004a. The importance of supertagging for wide-coverage CCG parsing. In *Proceedings of COLING-04*, pages 282–288, Geneva, Switzerland.

Stephen Clark and James R. Curran. 2004b. Parsing the WSJ using CCG and log-linear models. In *Proceedings of ACL-04*, pages 104–111, Barcelona, Spain.

Stephen Clark and James R. Curran. 2007a. Formalism-independent parser evaluation with CCG and DepBank. In *Proceedings of the 45th Annual Meeting of the ACL*, Prague, Czech Republic.

Stephen Clark and James R. Curran. 2007b. Perceptron training for a wide-coverage lexicalized-grammar parser. In *Proceedings of the ACL Workshop on Deep Linguistic Processing*, Prague, Czech Republic.

Stephen Clark, Mark Steedman, and James R. Curran. 2004. Object-extraction and question-parsing using CCG. In *Proceedings of the EMNLP Conference*, pages 111–118, Barcelona, Spain.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.

James R. Curran and Stephen Clark. 2003a. Investigating GIS and smoothing for maximum entropy taggers. In *Proceedings of the 10th Meeting of the EACL*, pages 91–98, Budapest, Hungary.

James R. Curran and Stephen Clark. 2003b. Language independent NER using a maximum entropy tagger. In *Proceedings of CoNLL-03*, pages 164–167, Edmonton, Canada.

James R. Curran, Stephen Clark, and David Vadas. 2006. Multi-tagging for lexicalized-grammar parsing. In *Proceedings of COLING/ACL-06*, pages 697–704, Sydney.

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

H. Kamp and U. Reyle. 1993. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Kluwer, Dordrecht.

Ron Kaplan, Stefan Riezler, Tracy H. King, John T. Maxwell III, Alexander Vasserman, and Richard Crouch. 2004. Speed and accuracy in shallow and deep stochastic parsing. In *Proceedings of HLT and the 4th Meeting of NAACL*, Boston, MA.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2007. Efficient HPSG parsing with supertagging and CFG-filtering. In *Proceedings of IJCAI-07*, Hyderabad, India.

Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

Adwait Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the EMNLP Conference*, pages 133–142, Philadelphia, PA.

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