The Second Release of the RASP System

Ted Briscoe[†] **John Carroll**[‡] **Rebecca Watson**[†]

[†]Computer Laboratory, University of Cambridge, Cambridge CB3 OFD, UK
firstname.lastname@cl.cam.ac.uk

[‡]Department of Informatics, University of Sussex, Brighton BN1 9QH, UK
J.A.Carroll@sussex.ac.uk

Abstract

We describe the new release of the RASP (robust accurate statistical parsing) system, designed for syntactic annotation of free text. The new version includes a revised and more semantically-motivated output representation, an enhanced grammar and part-of-speech tagger lexicon, and a more flexible and semi-supervised training method for the structural parse ranking model. We evaluate the released version on the WSJ using a relational evaluation scheme, and describe how the new release allows users to enhance performance using (in-domain) lexical information.

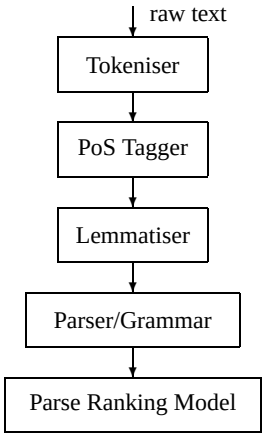

Figure 1: RASP Pipeline

1 Introduction

The first public release of the RASP system (Briscoe & Carroll, 2002) has been downloaded by over 120 sites and used in diverse natural language processing tasks, such as anaphora resolution, word sense disambiguation, identifying rhetorical relations, resolving metonymy, detecting compositionality in phrasal verbs, and diverse applications, such as topic and sentiment classification, text anonymisation, summarisation, information extraction, and open domain question answering. Briscoe & Carroll (2002) give further details about the first release. Briscoe (2006) provides references and more information about extant use of RASP and fully describes the modifications discussed more briefly here.

The new release, which is free for all non-commercial use[1], is designed to address several weaknesses of the extant toolkit. Firstly, all modules have been incrementally improved to cover a greater range of text types. Secondly, the part-of-speech tagger lexicon has been semi-automatically enhanced to better deal with rare or unseen behaviour of known words. Thirdly, better facilities have been provided for user customisation. Fourthly, the grammatical relations output has been redesigned to better support further processing. Finally, the training and tuning of the parse ranking model has been made more flexible.

[1]See http://www.informatics.susx.ac.uk/research/nlp/rasp/ for licence and download details.

2 Components of the System

RASP is implemented as a series of modules written in C and Common Lisp, which are pipelined, working as a series of Unix-style filters. RASP runs on Unix and is compatible with most C compilers and Common Lisp implementations. The public release includes Lisp and C executables for common 32- and 64-bit architectures, shell scripts for running and parameterising the system, documentation, and so forth. An overview of the system is given in Figure 1.

2.1 Sentence Boundary Detection and Tokenisation

The system is designed to take unannotated text or transcribed (and punctuated) speech as input, and not simply to run on pre-tokenised input such as that typically found in corpora produced for NLP purposes. Sentence boundary detection and tokenisation modules, implemented as a set of deterministic finite-state rules in Flex (an open source re-implementation of the original Unix Lex utility)

Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, pages 77–80,
Sydney, July 2006.

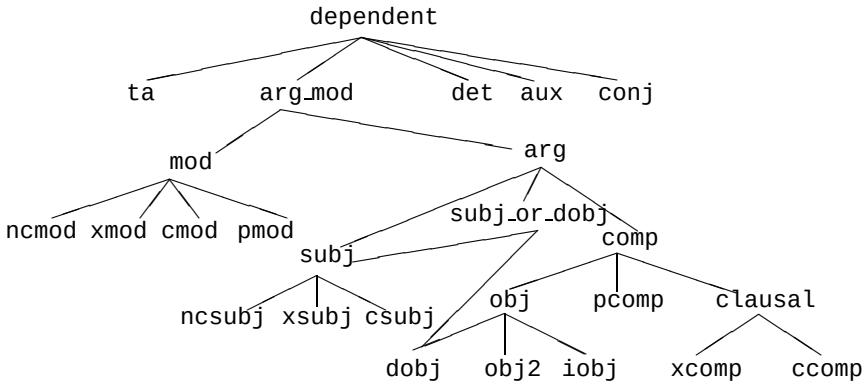

Figure 2: The GR hierarchy

and compiled into C, convert raw ASCII (or Unicode in UTF-8) data into a sequence of sentences in which, for example punctuation tokens are separated from words by spaces, and so forth.

Since the first release this part of the system has been incrementally improved to deal with a greater variety of text types, and handle quotation appropriately. Users are able to modify the rules used and recompile the modules. All RASP modules now accept XML mark up (with certain hard-coded assumptions) so that data can be pre-annotated—for example to identify named entities—before being passed to the tokeniser, allowing for more domain-dependent, potentially multiword tokenisation and classification prior to parsing if desired (e.g. Vlachos *et al.*, 2006), as well as, for example, handling of text with sentence boundaries already determined.

2.2 PoS and Punctuation Tagging

The tokenised text is tagged with one of 150 part-of-speech (PoS) and punctuation labels (derived from the CLAWS tagset). This is done using a first-order ('bigram') hidden markov model (HMM) tagger implemented in C (Elworthy, 1994) and trained on the manually-corrected tagged versions of the Susanne, LOB and (subset of) BNC corpora. The tagger has been augmented with an unknown word model which performs well under most circumstances. However, known but rare words often caused problems as tags for all realisations were rarely present. A series of manually developed rules has been semi-automatically applied to the lexicon to ameliorate this problem by adding further tags with low counts to rare words. The new tagger has an accuracy of just over 97% on the DepBank part of section 23 of the Wall Street Journal, suggesting that this modification has resulted in competitive performance on out-of-domain newspaper text. The tagger implements the Forward-Backward algorithm as well as the Viterbi algorithm, so users can opt for tag thresholding rather than forced-choice tagging (giving >99% tag recall on DepBank, at some cost to overall system speed). Recent experiments suggest that this can lead to a small gain in parse accuracy as well as coverage (Watson, 2006).

2.3 Morphological Analysis

The morphological analyser is also implemented in Flex, with about 1400 finite-state rules incorporating a great deal of lexically exceptional data. These rules are compiled into an efficient C program encoding a deterministic finite state transducer. The analyser takes a word form and CLAWS tag and returns a lemma plus any inflectional affixes. The type and token error rate of the current system is less than 0.07% (Minnen, Carroll and Pearce, 2001). The primary system-internal value of morphological analysis is to enable later modules to use lexical information associated with lemmas, and to facilitate further acquisition of such information from lemmas in parses.

2.4 PoS and Punctuation Sequence Parsing

The manually-developed wide-coverage tag sequence grammar utilised in this version of the parser consists of 689 unification-based phrase structure rules (up from 400 in the first release). The preterminals to this grammar are the PoS and punctuation tags[2]. The terminals are featural descriptions of the preterminals, and the non-terminals project information up the tree using an X-bar scheme with 41 attributes with a maximum of 33 atomic values. Many of the original

[2]The relatively high level of detail in the tagset helps the grammar writer to limit overgeneration and overacceptance.

rules have been replaced with multiple more specific variants to increase precision. In addition, coverage has been extended in various ways, notably to cover quotation and word order permutations associated with direct and indirect quotation, as is common in newspaper text. All rules now have a rule-to-rule declarative specification of the grammatical relations they license (see §2.6). Finally, around 20% of the rules have been manually identified as 'marked' in some way; this can be exploited in customisation and in parse ranking. Users can specify that certain rules should not be used and so to some extent tune the parser to different genres without the need for retraining.

The current version of the grammar finds at least one parse rooted in S for about 85% of the Susanne corpus (used for grammar development), and most of the remainder consists of phrasal fragments marked as independent text sentences in passages of dialogue. The coverage of our WSJ test data is 84%. In cases where there is no parse rooted in S, the parser returns a connected sequence of partial parses covering the input. The criteria are partial parse probability and a preference for longer but non-lexical combinations (Kiefer *et al.*, 1999).

2.5 Generalised LR Parser

A non-deterministic LALR(1) table is constructed automatically from a CF 'backbone' compiled from the feature-based grammar. The parser builds a packed parse forest using this table to guide the actions it performs. Probabilities are associated with subanalyses in the forest via those associated with specific actions in cells of the LR table (Inui *et al.*, 1997). The *n*-best (i.e. most probable) parses can be efficiently extracted by unpacking subanalyses, following pointers to contained subanalyses and choosing alternatives in order of probabilistic ranking. This process backtracks occasionally since unifications are required during the unpacking process and they occasionally fail (see Oepen and Carroll, 2000).

The probabilities of actions in the LR table are computed using bootstrapping methods which utilise an unlabelled bracketing of the Susanne Treebank (Watson *et al.*, 2006). This makes the system more easily retrainable after changes in the grammar and opens up the possibility of quicker tuning to in-domain data. In addition, the structural ranking induced by the parser can be reranked using (in-domain) lexical data which provides conditional probability distributions for the SUBCATegorisation attributes of the major lexical categories. Some generic data is supplied for common verbs, but this can be augmented by user supplied, possibly domain specific files.

2.6 Grammatical Relations Output

The resulting set of ranked parses can be displayed, or passed on for further processing, in a variety of formats which retain varying degrees of information from the full derivations. We originally proposed transforming derivation trees into a set of named grammatical relations (GRs), illustrated as a subsumption hierarchy in Figure 2, as a way of facilitating cross-system evaluation. The revised GR scheme captures those aspects of predicate-argument structure that the system is able to recover and is the most stable and grammar independent representation available. Revisions include a treatment of coordination in which the coordinator is the head in subsuming relations to enable appropriate semantic inferences, and addition of a text adjunct (punctuation) relation to the scheme.

Factoring rooted, directed graphs of GRs into a set of bilexical dependencies makes it possible to compute the transderivational support for a particular relation and thus compute a weighting which takes account both of the probability of derivations yielding a specific relation and of the proportion of such derivations in the forest produced by the parser. A weighted set of GRs from the parse forest is now computed efficiently using a variant of the inside-outside algorithm (Watson *et al.*, 2005).

3 Evaluation

The new system has been evaluated using our reannotation of the PARC dependency bank (DepBank; King *et al.*, 2003)—consisting of 560 sentences chosen randomly from section 23 of the Wall Street Journal—with grammatical relations compatible with our system. Briscoe and Carroll (2006) discuss issues raised by this reannotation.

Relations take the following form: **(relation subtype head dependent initial)** where **relation** specifies the type of relationship between the **head** and **dependent**. The remaining **subtype** and **initial** slots encode additional specifications of the relation type for some relations and the initial or underlying logical relation of the grammatical subject in constructions such as passive. We deter-

mine for each sentence the relations in the test set which are correct at each level of the relational hierarchy. A relation is correct if the head and dependent slots are equal and if the other slots are equal (if specified). If a relation is incorrect at a given level in the hierarchy it may still match for a subsuming relation (if the remaining slots all match); for example, if a **ncmod** relation is mislabelled with **xmod**, it will be correct for all relations which subsume both **ncmod** and **xmod**, e.g. **mod**. Similarly, the GR will be considered incorrect for **xmod** and all relations that subsume **xmod** but not **ncmod**. Thus, the evaluation scheme calculates unlabelled dependency accuracy at the **dependency** (most general) level in the hierarchy. The micro-averaged precision, recall and F_1 score are calculated from the counts for all relations in the hierarchy. The macroaveraged scores are the mean of the individual scores for each relation.

On the reannotated DepBank, the system achieves a microaveraged F_1 score of 76.3% across all relations, using our new training method (Watson *et al.*, 2006). Briscoe and Carroll (2006) show that the system has equivalent accuracy to the PARC XLE parser when the morphosyntactic features in the original DepBank gold standard are taken into account. Figure 3 shows a breakdown of the new system's results by individual relation.

Relation	Precision	Recall	F_1	std GRs
dependent	79.76	77.49	78.61	10696
aux	93.33	91.00	92.15	400
conj	72.39	72.27	72.33	595
ta	42.61	51.37	46.58	292
det	87.73	90.48	89.09	1114
arg_mod	79.18	75.47	77.28	8295
mod	74.43	67.78	70.95	3908
ncmod	75.72	69.94	72.72	3550
xmod	53.21	46.63	49.70	178
cmod	45.95	30.36	36.56	168
pmod	30.77	33.33	32.00	12
arg	77.42	76.45	76.94	4387
subj_or_dobj	82.36	74.51	78.24	3127
subj	78.55	66.91	72.27	1363
ncsubj	79.16	67.06	72.61	1354
xsubj	33.33	28.57	30.77	7
csubj	12.50	50.00	20.00	2
comp	75.89	79.53	77.67	3024
obj	79.49	79.42	79.46	2328
dobj	83.63	79.08	81.29	1764
obj2	23.08	30.00	26.09	20
iobj	70.77	76.10	73.34	544
clausal	60.98	74.40	67.02	672
xcomp	76.88	77.69	77.28	381
ccomp	46.44	69.42	55.55	291
pcomp	72.73	66.67	69.57	24
macroaverage	62.12	63.77	62.94	
microaverage	77.66	74.98	76.29	

Figure 3: Accuracy on DepBank

Acknowledgements

Development has been partially funded by the EPSRC RASP project (GR/N36462 and GR/N36493) and greatly facilitated by Anna Korhonen, Diana McCarthy, Judita Preiss and Andreas Vlachos. Much of the system rests on earlier work on the ANLT or associated tools by Bran Boguraev, David Elworthy, Claire Grover, Kevin Humphries, Guido Minnen, and Larry Piano.

References

Briscoe, E.J. (2006) *An Introduction to Tag Sequence Grammars and the RASP System Parser,* University of Cambridge, Computer Laboratory Technical Report 662.

Briscoe, E.J. and J. Carroll (2002) 'Robust accurate statistical annotation of general text', *Proceedings of the 3rd Int. Conf. on Language Resources and Evaluation (LREC'02),* Las Palmas, Gran Canaria, pp. 1499–1504.

Briscoe, E.J. and J. Carroll (2006) 'Evaluating the Accuracy of an Unlexicalized Statistical Parser on the PARC DepBank', *Proceedings of the COLING/ACL Conference,* Sydney, Australia.

Elworthy, D. (1994) 'Does Baum-Welch re-estimation help taggers?', *Proceedings of the 4th ACL Conference on Applied NLP,* Stuttgart, Germany, pp. 53–58.

Inui, K., V. Sornlertlamvanich, H. Tanaka and T. Tokunaga (1997) 'A new formalization of probabilistic GLR parsing', *Proceedings of the 5th International Workshop on Parsing Technologies (IWPT'97),* MIT, pp. 123–134.

Kiefer, B., H-U. Krieger, J. Carroll and R. Malouf (1999) 'A bag of useful techniques for efficient and robust parsing', *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics,* University of Maryland, pp. 473–480.

King, T.H., R. Crouch, S. Riezler, M. Dalrymple and R. Kaplan (2003) 'The PARC700 Dependency Bank', *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC-03),* Budapest, Hungary.

Minnen, G., J. Carroll and D. Pearce (2001) 'Applied morphological processing of English', *Natural Language Engineering, vol.7.3,* 225–250.

Oepen, S. and J. Carroll (2000) 'Ambiguity packing in constraint-based parsing — practical results', *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics,* Seattle, WA, pp. 162–169.

Watson, R. (2006) 'Part-of-speech tagging models for parsing', *Proceedings of the 9th Annual CLUK Colloquium,* Open University, Milton Keynes, UK.

Watson, R., E.J. Briscoe and J. Carroll (2006) *Semi-supervised Training of a Statistical Parser from Unlabeled Partially-bracketed Data,* forthcoming.

Watson, R., J. Carroll and E.J. Briscoe (2005) 'Efficient extraction of grammatical relations', *Proceedings of the 9th Int. Workshop on Parsing Technologies (IWPT'05),* Vancouver, Canada.