Robust Accurate Statistical Annotation of General Text

Ted Briscoe[*] and **John Carroll**[†]

[*]Computer Laboratory
University of Cambridge
Ted.Briscoe@cl.cam.ac.uk

[†]Cognitive and Computing Sciences
University of Sussex
John.Carroll@cogs.susx.ac.uk

Abstract

We describe a robust accurate domain-independent approach to statistical parsing incorporated into the new release of the ANLT toolkit, and publicly available as a research tool. The system has been used to parse many well known corpora in order to produce data for lexical acquisition efforts; it has also been used as a component in an open-domain question answering project. The performance of the system is competitive with that of statistical parsers using highly lexicalised parse selection models. However, we plan to extend the system to improve parse coverage, depth and accuracy.

1. Introduction

In recent years, considerable progress has been made in accurate statistical parsing of realistic texts. However, a great deal of this progress has been achieved with systems based on lexicalised probabilistic models of parse selection optimised on the Wall Street Journal treebank (e.g. Collins, 1999; Charniak, 2000). Evaluation of such systems has been primarily in terms of the PARSEVAL scheme tree similarity measures of (labelled) precision and recall and crossing bracket rate. The alternative approach to robust parsing, favoured by most commercial and academic information extraction systems, is to use (cascaded) finite-state transducers, often augmented with heuristics such as the longest match preference, to construct partial phrasal-level parses (e.g. Appelt *et al.*, 1995; Abney, 1996). This approach has the advantage of being much less domain-specific and does not require large quantities of manually annotated training data. However, the output is neither as complete nor as accurate as state-of-the-art statistical parsers.

There are several reasons to believe that finite-state methods of this latter kind will not be able to achieve the same level of accuracy as a well-designed statistical parser. The first is that heuristics like longest match interact in complex ways with the large number of manually coded rules required in a wide-coverage system, making effective development of further rules increasingly difficult and requiring increasingly painstaking manual specification of the contexts of legitimate application for each rule. The second is that modular cascaded systems must inevitably resolve some ambiguities earlier than is optimal because of the requirement that the output from each phase of processing is deterministic. A third is that many such systems achieve much of their domain independence by basing rules as much as possible on part-of-speech (PoS) tags, rather than specific lexical items, in order to limit the number of rules required. Evaluation has been sporadic, but suggests that such systems are significantly less accurate at finding both phrase boundaries and grammatical relations.

We have developed an approach to robust accurate, but domain-independent, statistical parsing (RADISP) which attempts to combine the strengths of both approaches. This is a pipelined modular system, in which a beam search for the most probable overall analysis is done on the thresholded output of each phase—see Figure 1. First, text is tokenised using a deterministic finite-state transducer. Second, tokens are PoS and punctuation tagged using a HMM with a large lexicon and well-developed unknown word handling module. However, only very improbable tags are removed at this phase. Next deterministic morphological analysis and lemmatisation is performed on the PoS tagged tokens. Third, the lattice of tags is parsed using a manually-developed wide-coverage grammar of such PoS and punctuation tags. Finally, the *n*-best parses are selected from the parse forest using a probabilistic parse selection model conditioned on the structural parse context, degree of support for a subanalysis in the parse forest, and lexical information when available. The output of the parser can be displayed as syntactic trees, and/or factored into a sequence of (weighted) grammatical relations between lexical heads, or into a sequence of elementary predications and possibly underspecified equational constraints in a minimal recursion semantic representation. Figure 2 shows an intermediate stage of processing of the simple (TREC8QA) question *What debts did Qintex leave?* along with three of the various output representations.

We have argued elsewhere that an evaluation scheme measuring recovery of grammatical relations between lexical heads has a number of advantages over one measuring tree similarity (Carroll, Briscoe and Sanfilippo, 1998). Similar relation-based schemes have been employed by others (e.g. Lin, 1998; Collins, 1999; Srinivas, 2000). Measured in this way our results appear broadly competitive with those produced by state-of-the-art statistical parsers (Briscoe *et al.*, 2002a). Objective comparison across systems is hampered by the fact that systems extract differing sets of relations and have been evaluated on different test suites. Our system achieves a F_1-score of 76.5% on a manually constructed test suite of 500 sentences from the Susanne corpus (see Carroll *et al.*, 1998; Briscoe *et al.*,

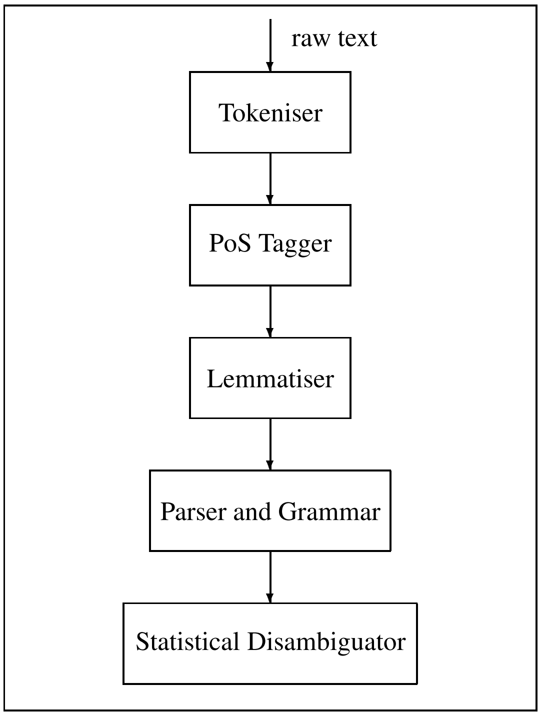

Figure 1: System architecture.

```
^ ^_^:1
What What_DDQ:1
debts debt+s_NN2:1
did do+ed_VDD:1
Qintex Qintex_NP1:1
group NNJ1:0.007 VV0:0.0007 NN1:0.99
leave VV0:0.644 NN1:0.355
? ?_?:1

(T/txt-sc1/-+-
 (S/whnp_s (NP/det_n What_DDQ
            (N1/n debt+s_NN2))
  (S/sai/- do+ed_VDD
   (S/np_vp
    (NP/name_n1
      (NP/n1_name/-
       (N1/n Qintex_NP1))
       (N1/n group_NN1))
    (V/0 leave_VV0))))
 ?_?)

(ncsubj leave:6_VV0 group:5_NN1 _)
(detmod _ debt+s:2_NN2 What:1_DDQ)
(ncmod _ group:5_NN1 Qintex:4_NP1)
(aux _ leave:6_VV0 do+ed:3_VDD)
(dobj leave:6_VV0 debt+s:2_NN2 _)

ARGN u3 x1
What_rel x1
debt_rel x1
do_rel u3
ARG1 u3 x4
Qintex_rel x4
group_rel x4
leave_rel u3
?_rel u11
```

Figure 2: Example of text input to the parser/grammar, and three different types of output: syntactic tree, grammatical relations, and minimal recursion semantic representation.

2002a for further details). Other published results report F_1-scores in the region of 80–84% using a comparable evaluation with coarser grained sets of grammatical relations (not including, for example, control relations) and differing test sets. There is evidence from the results reported by Srinivas (2000) that Susanne data, drawn from a variety of genres, constitutes a harder test than the more homogeneous Wall Street Journal. The RADISP system can output probabilistically weighted competing grammatical relations, allowing subsequent processing modules to make principled trade offs between precision and recall. At 90% precision, the system achieves 45% recall on the same test data (Carroll and Briscoe, 2001).

The system has been used to parse over 98% of the 90 million word written section of the British National Corpus (BNC) as well as the Lancaster-Oslo-Bergen Corpus, Spoken English Corpus, the Susanne Corpus and the TREC8 QA track top-ranked document collections. To date, the resulting annotated corpora have been used to develop systems for word sense disambiguation (Carroll and McCarthy, 2000; Lambeau, 2001), anaphora resolution (Preiss, 2002), acquiring verb subcategorisations (Korhonen, 2002) and acquiring selectional preferences (McCarthy and Korhonen, 1998; Clark and Weir, 2001). The system has also been used in experiments in information extraction (Yeh, 2000) and as a component of an open-domain question-answering (Briscoe, Copestake and Teufel, 2002b). The RADISP system is distributed freely for non-commercial use (see <http://www.cogs.susx.ac.uk/lab/nlp/rasp/>). Section 2 describes the components of the RADISP system in more detail. Section 3 discusses ongoing extensions to the baseline system.

2. Components of the RADISP System

The RADISP system is implemented as a series of modules written in C or Common Lisp, which can be pipelined analogously to a series of Unix-style filters. It will run under Unix, Linux, or SUNOS with most C compilers and most Common Lisp implementations.

2.1. Tokenisation

The system is designed to take unannotated text or transcribed (and punctuated) speech as input and not simply to run on pretokenised input such as the Brown, LOB, Susanne or WSJ corpora. A tokenisation program, implemented as a set of deterministic finite-state rules in Flex (an open source reimplementation of the original Unix Lex utility Levine, Mason and Brown, 1992) and compiled into C, converts raw ASCII data into a sequence of tokens in which punctuation is separated from words by spaces and sentence boundaries are marked.

This component of the system requires further development as it does not annotate or preserve other document structure such as paragraph boundaries. It can be easily replaced with other more developed tools (e.g. the Edinburgh TTT system, Grover *et al.*, 2000). However, it is simple to add additional rules when new corpora reveal specific problems.

2.2. PoS and Punctuation Tagging

The tokenised text is tagged with one of the 155 CLAWS-2 part-of-speech (PoS) and punctuation labels. 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 a statistical unknown word model (Piano, 1996; Weischedel *et al.*, 1993) and achieves around 97% per word accuracy when tested on similar data. However, the accuracy of the unknown word component is around 80%, so that performance on less similar data can quickly degrade.

As the Forward-Backward algorithm (FBA) has been implemented in addition to the Viterbi algorithm (Elworthy, 1994), the tagger can trade-off precision against recall by returning all but the most improbable tags up to some relative threshold ranked according to the posterior probabilities found using the FBA. Returning a mean 1.3 tags per word has been claimed to improve recall by an order of magnitude (de Marcken, 1990), and turns out to have little impact on speed of tagging. It is possible that a more accurate tagger might allow us to dispense with thresholding and increase system throughput without loss of accuracy. However developing such a parser is non-trivial since extant approaches all tend to degrade quickly on lexically dissimilar text.

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%, and the system is able to process more than 200K words per second on standard hardware (Minnen, Carroll and Pearce, 2001).

When the analyser is applied to the thresholded output of the tagger a distinct analysis is returned for each CLAWS tag returned by the tagger. The primary value of morphological analysis is to enable later modules to make use of lexical information associated with lemma forms and to facilitate further acquisition of such information from lemmas in parses.

2.4. PoS and Punctuation Sequence Parsing

The lattice of lemma+affix_tag forms is passed to a modified version of the probabilistic generalised LR parser (Briscoe and Carroll, 1993; Inui *et al.*, 1997), augmented with limited lexical information encoding the probability of some phrasal verb combinations (i.e. verb plus preposition/particle) and the conditional probability of high to mid frequency verbs appearing with any one of 23 subcategorisation frames.

The manually-developed wide-coverage tag sequence grammar utilised in this version of the parser consists of about 400 unification-based phrase structure rules (see Briscoe and Carroll, 1995 for further details). It is designed to enumerate possible valencies for predicates (verbs, adjectives and nouns) by including separate rules for each pattern of possible complementation in English. The distinction between arguments and adjuncts is expressed by adjunction of adjuncts to maximal projections ($XP \rightarrow XP\ Adjunct$) as opposed to government of arguments (i.e. arguments are sisters within $X1$ projections; $X1 \rightarrow X0\ Arg1 \ldots ArgN$).

Although the grammar enumerates complementation possibilities and checks for global sentential well-formedness, it does not attempt to associate most 'displaced' constituents with their canonical position / grammatical role. Therefore, the resulting parser is 'intermediate' in the sense that it extends a purely phrasal analysis but not to the point where a complete logical form can be recovered deterministically in all cases. The current version of the grammar finds at least one parse rooted in S for about 80% of the Susanne corpus, and a significant proportion of the remainder consists of phrasal fragments marked as independent text sentences in passages of dialogue. For other corpora the proportion of parses rooted in S recovered can be lower; in the case of the BNC it falls to 67%, primarily probably because of poorer tokenisation and sentence boundary detection. In cases where there is no parse rooted in S, the parser returns an optimal connected sequence of partial parses which covers the input. The criteria are partial parse probability and a preference for longer but non-lexical partial parse combinations (Kiefer *et al.*, 1999). The parser takes average time roughly quadratic in the length of the input (Carroll, 1994). With respect to the Susanne corpus the grammar has an average parse base of 1.28, meaning that it assigns an average of 1.28^n parses to a sentence of n tokens. Sentences for which a parse forest cannot be constructed within 15 seconds are timed out resulting, for example, in less than 2% of timeouts on the BNC corpus. The average throughput is 40 words per CPU second on standard hardware.

2.5. *N*-Best Parse Tree Output

The parse forest packs subanalyses in a graph structured stack using a subsumption rather than identity check, as is standard with unification-based formalisms. This entails that some features must be unified when packed subanalyses are unpacked. Probabilities are associated with subanalyses via those associated with specific reduce actions in the probabilistic LR table. The *n*-best (i.e. most probable) parses can be efficiently extracted by unpacking subanalyses and unifying the remaining features, following pointers to contained subanalyses and choosing alternatives in order of probabilistic ranking. This process backtracks occasionally when unification fails during the unpacking process (Oepen and Carroll, 2000).

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. The most common output format is one which replaces the full featural description of each node in a derivation with the rule name used to construct the local tree. These rule names are manually encoded in the grammar to retain essential features of the local tree and provide the information required for the subsequent output transformations we currently utilise. This is the format displayed in Figure 2 above.

2.6. Weighted GR Output

We originally proposed transforming trees to sets of named grammatical relations (GRs) of the type illustrated in Figure 2 above as a technique for facilitating fine-grained cross-system evaluation (see Carroll *et al.*, 1998 where a detailed specification of the representation is also given). However, because this representation of the grammatical information in a derivation is factored into a set of 'atomic' components which can be typed via their names, it can also be a useful output representation for other tasks and for subsequent processing. For instance, the argument relations from derivations can be used for acquiring predicate subcategorisation or selectional preference information. Factoring 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 the proportion of such derivations in the set produced by the parser (Carroll and Briscoe, 2001). Factoring of derivations into sets of bilexical dependencies can also, in principle, support reranking of derivations using a lexicalised discriminative model (Hektoen, 1997; Collins, 2000).

The GR set for a derivation is computed from the derivation tree labelled with rule names. Some relations are based on information not accessible in a single local tree, for example, whether a non-clausal subject is logically the direct object of a passive participle. Use of non-local information also allows the GR representation to extend what is directly encoded in the derivation. For example, the grammar does not attempt to relate preposed wh-phrases to their canonical position, but appropriate GRs can be recovered reliably in the many cases where there is only a single candidate verb. The example in Figure 2 represents such a case.

2.7. Robust MRS Output

The GR representation stays deliberately close to surface syntax, although it does encode some logical / underlying relations via extra parameters on specific named relations (see Carroll *et al.*, 1998 for details). However, it does not map easily to a logical form or predicate-argument structure, and the tag sequence grammar, unlike the ANLT full grammar (Grover, Carroll and Briscoe, 1993) is not able to construct a logical form deterministically from the derivation because of the intermediate level of analysis achieved.

Nevertheless, for some tasks that we want to use the RADISP system for, such as parsing of highly-ranked documents relevant to queries in open-domain question-answering (Briscoe *et al.*, 2002b), it is useful to be able to output an underspecified semantic representation, in our case robust minimal recursion semantics (MRS, Copestake *et al.*, 1999). In robust MRS the arguments of predicates are represented using Parsons' (1990) event-based scheme: instead of, for example, *give(e,x,y,z)*, Robust MRS uses *give(e), arg1(e,x), arg2(e,y), arg3(e,z)*. This is done because the arity of predicates is not known in advance when parsing with the tag sequence grammar, so arguments can be added incrementally. It also allows for underspecification of argument positions: *argN* is used to indicate that some argument relationship holds, but that it might be *arg1*, *arg2*, or so on.

Composition of semantics is done according to a very simple algebra. In composition, semantic structures consist of a hook, currently with a single element, which is the index, a list of elementary predications, and a list of equalities. Each word of the input is associated with an elementary predication with the single argument being an event or object depending only on its PoS tag. These predications are accumulated as the semantics is composed, along with sets of variable equalities, insofar as these can be reliably inferred from the derivation tree labelled with rule names (as above). Thus, the mechanism for computing robust MRSs from trees is very similar to that used to compute GRs, but the output format is that of a factored and underspecified MRS. Since there is a semantics for this formalism (Copestake, Lascarides and Flickinger, 2001), it is possible to define proof-theoretically relationships between (underspecified) MRSs, and these can be exploited in, for example, matching fully-specified MRSs for questions, recovered using a deeper but slower and more fragile parser, with the often underspecified MRSs extracted from document collections using the RADISP system.

3. Conclusions and Further Work

The baseline RADISP system described here has usable performance for a number of tasks. Its relative domain-independence coupled with competitive levels of accuracy make it especially practical for easy deployment on new or varied data from differing domains. However, there are a number of ways in which the system might be improved which we are currently exploring.

It is likely that further improvement of parse selection accuracy will require a lexicalised model. Our current approach to lexicalisation is to use a model based on the discriminative technique of Hektoen (1997) applied to the sets of grammatical relations extracted from competing derivations. In this approach, the model is trained to discriminate between the correct and incorrect derivations for a given sentence in terms only of the differing grammatical relations between them. Thus lexical dependencies are utilised in an efficient manner ameliorating data sparsity. It is clear that this model results in more accurate parse selection when trained and tested on treebanks containing similar material. The challenge is to make it work as a domain-specific reranking technique which can be effectively trained from noisy automatic structurally-selected parses for a given domain.

There are many other lexical issues affecting overall accuracy. It is likely that improved tokenisation including better handling of idiomatic and semi-idiomatic phrases would both ameliorate some parse failures and guide parse selection. However, reliance on lexical information leads to domain dependence so our broad approach is to attempt to seed the RADISP system with such information acquired from automatic parses obtained without it, as with our approach to subcategorisation (Carroll, Minnen and Briscoe, 1998; Korhonen, 2002).

The current manually-developed tag sequence grammar was originally developed for the subcategorisation acquisition task. However, it has since been extended to yield parses for a higher proportion of data and to recover more informative representations of constructions largely irrelevant to subcategorisation, such as names and dates, for example. One way to improve system accuracy on specific tasks would be to develop different grammars or subgrammatical components tuned to specific tasks. To some extent, this is already the situation as the grammar is quite modular, and subparts such as detailed rules for names, dates or punctuation can be removed or added. However, no systematic investigation of the performance effects has been undertaken.

The work we have undertaken on probabilistic techniques is fully compatible with any grammar developed in the ANLT formalism. However, very little work has been undertaken with the ANLT full grammar since the initial experiment parsing dictionary definitions (Briscoe and Carroll, 1993) because this grammar requires accurate subcategorisation information for all lexical items to function effectively. Recently, though, Grover and Lascarides (2001) have demonstrated that a useful system can be built recovering full logical forms for around 30% of sentences from a sample of the Medline corpus, using a modified version of the ANLT full grammar but backing off to PoS tags for unknown words. If we want to accurately recover full logical forms in a practical way, the full grammar is a valuable wide-coverage resource to achieve this. The challenge is to develop a method of deployment which does not critically rely on detailed and accurate lexical information for every domain. Using the output of the probabilistic parser with the tag sequence grammar to constrain application of the full grammar is one avenue we intend to explore in the ongoing quest for a practical full parser.

Acknowledgements

The development of the new release of the ANLT system incorporating the RADISP system has been supported by the EPSRC RASP project (grants GR/N36462 and GR/N36493) and greatly facilitated by Anna Korhonen, Diana McCarthy, Mark McLauchlan, and Judita Preiss. Ann Copestake developed the idea of robust MRS and wrote the code that extracts such representations from grammar derivations. Much of the new system rests on earlier work developing the ANLT toolkit or tools integrated with it by Bran Boguraev, David Elworthy, Claire Grover, Kevin Humphries, Guido Minnen, Miles Osborne and Larry Piano.

References

Abney, S. (1996) 'Part-of-speech tagging and partial parsing' in Church, K. *et al.* (ed.), *Corpus-based Methods in Language and Speech,* Kluwer, Dordrecht.

Appelt, D., J. Hobbs, J. Bear, D. Israel, M. Kameyama, A. Kehler, D. Martin, K. Myers and M. Tyson (1995) 'The SRI FASTUS system, MUC-6 test results and analysis', *Proceedings of the 6th Message Understanding Conference,* Morgan Kaufmann, San Mateo, CA, pp. 237–248.

Briscoe E. and J. Carroll (1993) 'Generalised probabilistic LR parsing of natural language (corpora) with unification-based grammars', *Computational Linguistics, vol.19.1,* 25–60.

Briscoe, E. and J. Carroll (1995) 'Developing and Evaluating a Probabilistic LR Parser of Part-of-Speech and Punctuation Labels', *Proceedings of the 4th International Workshop on Parsing Technologies (IWPT'95),* Prague / Karlovy Vary, Czech Republic, pp. 48–58.

Briscoe, E., J. Carroll, J. Graham and A. Copestake (2002a) 'Relational evaluation schemes', *Proceedings of the Workshop at LREC'02 on Beyond PARSEVAL: Towards Improved Evaluation Measures for Parsing Systems,* Gran Canaria.

Briscoe, E., A. Copestake and S. Teufel (2002b) *CSTIT MPhil, Language Practical 2,* Computer Laboratory, University of Cambridge.

Carroll, J. (1994) 'Relating complexity to practical performance in parsing with wide-coverage unification grammars', *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics,* Las Cruces, NM, pp. 287-294.

Carroll, J. and E. Briscoe (2001) 'High precision extraction of grammatical relations', *Proceedings of the 7th ACL/SIGPARSE International Workshop on Parsing Technologies (IWPT'01),* Beijing, China.

Carroll, J., E. Briscoe and A. Sanfilippo (1998) 'Parser evaluation: a survey and a new proposal', *Proceedings of the International Conference on Language Resources and Evaluation,* Granada, pp. 447–454.

Carroll, J. and D. McCarthy (2000) 'Word sense disambiguation using automatically acquired verbal preferences', *Computers and the Humanities, vol.31.1–2,* 109–111.

Carroll, J., G. Minnen, and E. Briscoe (1998) 'Can subcategorisation probabilities help a statistical parser?', *Proceedings of the ACL SIGDAT 6th Workshop on Very Large Corpora (WVLC'98),* Montreal, pp. 118–126.

Charniak, E. (2000) 'A maximum entropy inspired parser', *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics,* Seattle, WA, pp. 132–139.

Clark, S. and D. Weir (2001) 'Class-based probability estimation using a semantic hierarchy', *Proceedings of the 2nd Conference of the North American Chapter of the Association of Computational Linguistics,* Carnegie Mellon University, pp. 95–102.

Collins, M. (1999) *Head-driven statistical models for natural language parsing,* PhD Dissertation, Computer and Information Science, University of Pennsylvania.

Collins, M. (2000) 'Discriminative reranking for natural language parsing', *Proceedings of the 17th International Conference on Machine Learning (ICML2000),* Morgan Kaufmann, San Mateo, CA.

Copestake, A., D. Flickinger, I. Sag and C. Pollard (1999) *Minimal Recursion Semantics: An introduction,* <http://www-csli.stanford.edu/˜aac/papers/newmrs.pdf>.

Copestake, A., A. Lascarides and D. Flickinger (2001) 'An algebra for semantic construction in constraint-based grammars', *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics,* Toulouse, France, pp. 132–139.

de Marcken, C. (1990) 'Parsing the LOB corpus', *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics,* Pittsburgh, PA, pp. 243–251.

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

Grover, C., J. Carroll and E. Briscoe (1993) *The Alvey natural language tools grammar (4th release),* Computer Laboratory, Cambridge University, UK, Technical Report 284.

Grover, C. and A. Lascarides (2001) 'XML-based data preparation for robust deep parsing', *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics,* Toulouse, France, pp. 252–259.

Grover, C., C. Matheson, A. Mikheev and M. Moens (2000) 'LT TTT – A flexible tokenisation tool', *Proceedings of the 2nd International Conference on Language Resources and Evaluation,* Athens, Greece.

Hektoen, E. (1997) 'Probabilistic parse selection based on semantic cooccurrences', *Proceedings of the 5th International Workshop on Parsing Technologies (IWPT'97),* MIT, pp. 113–122.

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.

Korhonen, A. (2002) *Subcategorisation acquisition,* PhD Dissertation, Computer Laboratory, University of Cambridge.

Lambeau, F. (2001) *Verb sense disambiguation from argument relations,* MPhil dissertation, Dept. of Engineering, University of Cambridge..

Levine, J., T. Mason and D. Brown (1992) *Lex & Yacc,* 2nd Edition, O'Reilly, Sebastopol, CA.

Lin, D. (1998) 'Dependency-based evaluation of MINIPAR', *Proceedings of the Workshop at LREC'98 on The Evaluation of Parsing Systems,* Granada, Spain.

McCarthy, D. and A. Korhonen (1998) 'Detecting verbal participation in diathesis alternations', *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics, vol. 2,* Montreal, Canada, pp. 1493–1495.

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.

Parsons, T. (1990) *Events in the semantics of English,* MIT Press: Cambridge, MA.

Piano, L. (1996) *Adaptation of Acquilex tagger to unknown words — release 2,* University of Cambridge Computer Laboratory, unpublished memo.

Preiss, J. (2002) 'Anaphora resolution with memory-based learning', *Proceedings of the 5th Annual CLUK Research Colloquium,* Sheffield.

Srinivas, B. (2000) 'A lightweight dependency analyzer', *Natural Language Engineering, vol.6.2,* 113–138.

Weischedel, R., M. Meteer, R. Schwartz, L. Ramshaw and J. Palmucci (1993) 'Coping with ambiguity and unknown words through probabilistic models', *Computational Linguistics, vol.19.2,* 359–382.

Yeh, A. (2000) 'Using existing systems to supplement small amounts of annotated grammatical relations training data', *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics,* Hong Kong, pp. 126–132.