Probabilistic Normalisation and Unpacking of Packed Parse Forests for Unification-based Grammars

John Carroll and **Ted Briscoe**
University of Cambridge, Computer Laboratory
Pembroke Street, Cambridge, CB2 3QG, UK
John.Carroll / Ted.Briscoe @cl.cam.ac.uk

Abstract

The research described below forms part of a wider programme to develop a practical parser for naturally-occurring natural language input which is capable of returning the n-best syntactically-determinate analyses, containing that which is semantically and pragmatically most appropriate (preferably as the highest ranked) from the exponential (in sentence length) syntactically legitimate possibilities (Church & Patil 1983), which can frequently run into the thousands with realistic sentences and grammars. We have opted to develop a domain-independent solution to this problem based on integrating statistical Markov modelling techniques, which offer the potential for rapid tuning to different sublanguages / corpora on the basis of supervised training, with linguistically-adequate grammatical (language) models, capable of returning analyses detailed enough to support semantic interpretation[1].

Probabilistic LR Parsing

Briscoe & Carroll (1991, 1992) describe an approach to probabilistic parse selection using a realistic unification-based grammar of English consisting of approximately 800 phrase structure rules (written in the Alvey Natural Language Tools (ANLT) formalism (Briscoe et al. 1987) which produces rules in a syntactic variant of the Definite Clause Grammar formalism (Pereira & Warren 1980). This is a wide-coverage grammar of English which has been shown, for instance, to be capable of assigning a correct analysis to about 98% of a corpus of 10,000 noun phrases extracted randomly from manually analysed corpora (Taylor, Grover & Briscoe 1989). The ANLT grammar is linked to a lexicon containing about 64,000 entries for 40,000 lexemes, including detailed subcategorisation information appropriate for the grammar, built semi-automatically from the *Longman Dictionary of Contemporary English* (LDOCE, Procter 1978).

The probabilistic model developed by Briscoe & Carroll represents a refinement of probabilistic context-free grammar (PCFG). A maximally informative context-free 'backbone' is derived automatically from the ANLT grammar (in which all categories are represented as feature bundles). This backbone is used to construct a generalised, non-deterministic LR parser (e.g. Tomita 1984, 1987) based on a LALR(1) table. Unification of the 'residue' of features not incorporated into the backbone grammar is performed at parse time in conjunction with reduce operations. Unification failure results in the reduce operation being blocked and the associated derivation being assigned a probability of zero. Probabilities are assigned to transitions in the LALR(1) action table via a process of supervised training based on computing the frequency with which transitions are traversed in a corpus of parse histories constructed using a user-driven, interactive version of the parser. The result is a probabilistic parser which, unlike a PCFG, is capable of probabilistically discriminating derivations which differ only in terms of order of application of the same set of CF backbone rules (within a context defined by the LALR(1) table) but which remains a stochastic first-order Markov model, because the LALR(1) table defines a non-deterministic finite-state machine (FSM) and the total probability of an analysis is computed from the sequence of transitions taken to construct it.

Preliminary experiments parsing noun definitions extracted from LDOCE suggest that this system is able to rank parses in a comparable fashion to systems based on PCFG (Fujisaki et al. 1989), probabilistic ID/LP CFG (Sharman, Jelinek & Mercer 1990) or simulated annealing (Sampson, Haigh & Atwell 1989), whose grammars are couched in a linguistically less adequate formalism and in two cases derived directly from manual analyses of the training and test corpus. On the basis of a training corpus of 150 analysed noun definitions, the parser ranked the correct analysis of

[1] This research is supported by SERC / DTI-IED project 4/1/1261 'Extensions to the Alvey Natural Language Tools' and by ESPRIT BRA 3030 'Acquisition of Lexical Knowledge for Natural Language Processing Systems'.

a sample of 89 sentences from the training corpus as most probable in 76% of cases, and ranked the correct analysis as most probable in 75% of a further 55 test sentences from the same corpus. These results were achieved solely on the basis of statistics concerning the conditional probability of syntactic rules in a syntactically-defined (LR) parse context, therefore a significant number of errors involved incorrect attachment of PPs, analyses of compounds, coordinations, and so forth, where lexical (semantic) information plays a major role. In many of these cases, the correct analysis was in the three highest ranked analyses. Both Sharman et al and Fujisaki et al. achieve slightly better results (about 85% correct parse / sentence), but their grammars integrate information concerning the probability of a lexeme occurring as a specific lexical syntactic category. Using a tree similarity measure, such as that of Sampson et al., the most probable analyses achieve a better than 96% fit to the correct analyses (as opposed to 80% for Sampson et al.'s simulated annealing parser).

We intend to extend this system in a number of ways to integrate the probability of a lexeme occurring with a specific lexical syntactic (sub)category, the probability of structurally-defined collocational patterns and statistical rule induction techniques to deal with cases of undergeneration (e.g. Briscoe & Waegner 1992). However, in this paper we address two more specific and related issues concerning the probabilistic interpretation of the system: firstly, the appropriate method of combining the probability of (partial) analyses, and secondly, a method for tractably computing the n-best analyses from the complete set licensed by the grammar.

Probabilistic Packed Parse Forests

Ideally, the computation of the most probable analysis or the n-best analyses defined by our probabilistic LR parser should not involve exhaustive search of the space of syntactically legitimate analyses defined by the ANLT grammar for any given input. However, it is not possible to introduce any Viterbi-style optimisation into the computation of local maximal paths through the probabilistic non-deterministic FSM defined by the parse table, because at any point in a derivation a maximal path may receive a probability of zero through unification failure, rendering a hitherto non-maximal local path maximal again. Unfortunately, the effects of feature propagation cannot be localised with respect to the computation of most probable sub-analyses, whilst any attempt to incorporate featural information into the probabilistic component of the grammar would result either in an intractably large grammar, or a model with too many free parameters, or both.

Our parser is based on Kipps' (1989) LR recogniser (a re-formulation of Tomita's (1984, 1987) algorithm), generalised for the case of unification grammars (Alshawi 1992). The parser constructs a packed parse forest representation of the complete set of analyses licensed by the ANLT grammar for a given input. In this representation identical sub-analyses are shared between differing superordinate analyses (as in chart parsing and other tabular parsing techniques) and sub-analyses covering the same portion of input are packed if the subsumption relation defined on unification-based formalisms holds between their root categories. In a probabilistic packed parse forest the probabilities of sub-analyses are associated with each node in the forest and in the case of packed nodes a distinct probability is maintained for each distinct sub-analysis at that node. Although this approach can be exponential in sentence length for some relatively unnatural grammars (Johnson 1989), in practice we have been able to generate packed parse forests for sentences containing over 30 words having many thousands of analyses. Schabes (1991) describes a Earley-like context-free LR parsing algorithm that is guaranteed polynomial in sentence length; however, a unification grammar version of this turns out to be exponential for some types of grammar, since on each reduce action the daughters of the rule involved in the reduction have to be unified with every possible alternative sequence of the sub-analyses that are being consumed by the rule. Our conclusion from experiments we have carried out with an implementation of the algorithm is that for the ANLT it offers no advantages over a Tomita-style parser.

Although we are able to generate packed parse forests for relatively long sentences, our previous technique for unpacking these forests to find the n-best analyses was not optimal since it involved a frequently near exhaustive search of the forest and was, therefore, exponential in the length of the input. Unsurprisingly, this was the major source of computational intractability in our system and for sentences of over 20 words often led to system failure. It is not straightforward to optimise this computation because once again remaining unifications involving the different featural extensions of packed nodes according to the superordinate and subordinate nodes to which they can be linked can lead to failure of a derivation encoded in the parse forest, unlike in the case of unpacking PCFG packed parse forests (e.g. Wright, Wrigley & Sharman 1991) where Viterbi-style optimisation is possible.

In a stochastic model the probability of a (sub-) analysis should be the product of the probability of the analyses combined to construct it. There are, however, problems with using this measure to compare analyses produced by LR parsers because LR parse tables define FSMs which are not ergodic, and non-deterministic LR parsers need not be time synchronous (contrary to conventional practice in (hidden) Markov modelling). In practice, this means that at any point during parsing it is difficult to evaluate all competing analyses for the same portion of input because these may not all be available and may involve differing numbers of state transitions in the LR table (corresponding to the 'depth' of the resultant syntactic tree). In previous work (Briscoe & Carroll 1991, 1992) we attempted to avoid this problem by computing the geometric mean of each (sub-) analysis of a given input and only comparing the geometric means for complete analyses, following Magerman & Marcus (1991). Using geometric means, as opposed to products, achieves a crude form of normalisation with respect to the length of the derivation. More recently we have been experimenting with a method of normalisation with a clearer probabilistic interpretation, described below.

Probabilistic Unpacking and Normalisation

Although no Viterbi-style optimisation is possible during the course of parsing (since a hitherto maximal path may be subject to a unification failure and thus must be abandoned), the technique can justifiably be applied to the probabilistic packed parse forest once parsing is complete. Our approach is first to construct a parse forest representing the full set of analyses licensed by the grammar, and then to identify the m-best analyses in the forest with a probabilistically-guided search, simultaneously normalising the scores for partial analyses of identical portions of the input. The best m complete analyses that have been identified are then unpacked from the forest, this process involving a small number of further unifications. The analyses for which these unifications all succeed are returned. If the number of analyses returned is less than the n wanted, the process could be repeated with a larger value for m.

In practice, this approach leads to a considerable practical improvement in the time taken to produce the n-best analyses. Although the algorithm remains in the worst case exponential in sentence length, in practice it will never need to search more than a fraction of the packed parse forest to recover the best analyses (except in the case of pathological grammars). In addition, by separating the probabilistic computation from the creation of the parse forest, we avoid some of the problems induced by the non-local nature of feature propagation.

Identifying the Best Partial Analyses

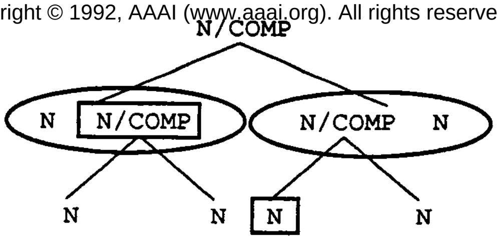
Our technique for identifying the best analyses in a probabilistic packed parse forest involves maximising the (normalised) score at nodes in the parse forest, in a similar manner to the Viterbi algorithm. The method starts at the left edge of the forest (at vertex zero), at each successive step normalising (as described below) the probabilities assigned to all partial analyses from vertices zero to v which end at the same node in the forest, and then extending just the m best partial analyses at each node to reach vertex $v + 1$. The process stops after pruning the lowest scoring analyses when the vertex v is just after the last word in the input, at which point all the analyses will be complete and will end at the root node of the forest. These analyses are returned as the m-best set.

A partial analysis is taken to end at the highest node which dominates the lexical node at the right edge of the analysis but which does not dominate the next lexical node to the right; or alternatively at the root node if there are no further words in the input. For example, the partial analyses from vertex zero to vertex 2 in the packed forest shown in figure 1 end at the boxed nodes labelled **N/COMP** and **N**. Figure 2 shows the two analyses which are represented in the forest. Since our parser uses a shift-reduce strategy, it constructs rightmost-first derivations of the input. A partial analysis thus consists of one or more subtrees whose heights decrease when looked at from left to right.

Each node in the forest contains the number and the product of the LR parse table transitions[2] taken up to that point in the analysis of the subtree dominated by the node. When extending a partial analysis with a new subtree, the transition count and product figures for the new extended analysis depend on the relationship between the heights of nodes at the roots of the subtrees currently making up the partial analysis and the height of the new (extending) subtree. If the root node of the new subtree is closer to the root of the forest than some of the current ones, then it dominates them, and so the figures for the extended analysis are those of the new node combined as appropriate with those of nodes in the partial analysis which preceded the dominated nodes. Otherwise a disjoint subtree is being added to the right edge of the partial analysis, and the figures at the new node are combined with those for all the current nodes. Packed nodes must be dealt with specially since the figures at a node take no account of nodes which are packed below it. In fact, it suffices to keep track of the differences between the figures at a node and those at the root of a subtree packed at that point whenever the subtree is incorporated into a partial analysis, and to factor in these differences whenever the figures at a node are subsequently required.

In the implementation, the parse tree corresponding to a partial analysis is represented implicitly by the set of packed nodes that it contains; when the probabilistic search of the parse forest is finished, the actual tree is unpacked from the forest in an efficient depth-first traversal. Nodes at which there is no packing, or at which none of the packed nodes are in the given set, are incorporated into the tree. Packed nodes which are members of the set are incorporated, after performing the unification required to check that they are consistent with the rest of the analysis built so far, and the traversal continues inside the packed subtree.

[2] In fact what is stored is the sum of the logarithms of the transitions.



Ellipses enclose packed nodes.

Boxes mark end nodes of partial analyses from vertices 0 to 2.

Figure 1: A Packed Parse Forest.

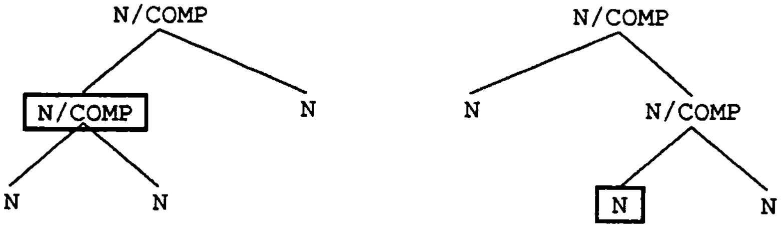


Figure 2: Analyses that are Represented in the Forest.

Normalising Partial Analyses

Since partial analyses spanning the same vertices may well have been derived via differing numbers of LR state transitions, scores for competing analyses must be normalised before they can be compared and the best ones identified. We have experimented with two alternative normalisation methods.

The first method is to normalise the scores of all the partial analyses which end at a given node in the forest to the length of the longest derivation in any of those analyses; for each derivation that is shorter by say d transitions, its score is multiplied by the geometric mean of the score raised to the power of d. So, for example, if at one stage the longest derivation was of length 5, two other derivations of length 3 and 4, with transition scores

a) 0.9 0.6 0.05
b) 0.5 0.2 0.2 0.08

would be multiplied by factors of 0.09 ($= 0.3^2$) for a), and 0.2 for b) to give normalised scores. This computation can be performed quickly since transition scores are held as logarithms.

The second, more complex, normalisation method is to compute the products of the transitions in all the alternative sequences of transitions which extend partial analyses ending at a given node, to normalise the products, as above, to the length of the longest sequence, and then to normalise the resulting scores so that they sum to 1. Next, we assign to each extended analysis the product of the score for the original partial analysis and the score for the extending transition sequence. Finally, the scores for all the extended analyses are normalised so they sum to 1. The rationale behind this method is that once the parse forest has been constructed we know that all sub-analyses that it contains spanning vertices 0 to v contribute to some successful, complete analysis (modulo unification failure in the final unpacking). Therefore, we can normalise the probabilities of all competing sub-analyses for successively increasing portions of the input so that the sum of their scores is 1. With the additional normalisation that is carried out to take account of differing derivation lengths, it should be the case that we can take the product of the probabilities for a partial analysis and one that extends it to obtain the probability of the extended analysis without introducing unwanted biases into the resultant ranking.

Empirical Results

We have re-run the experiment parsing LDOCE noun definitions from Briscoe & Carroll (1992) using the new approach to probabilistically ranking parses and unpacking the packed parse forest. Using the first normalisation scheme, the results we have obtained are marginally better than the original ones. Reparsing the training corpus and automatically comparing the most highly ranked analysis with the original parse, for the 89 definitions between two and ten words in length inclusive (mean length 6.2), in 69 cases the correct analysis (as defined by the training corpus) was also the most highly ranked. Taking correct parse / sentence as our measure then the result is 78%, as compared with 76% originally. Reparsing the further set of 55 LDOCE noun definitions not drawn from the training corpus, each containing up to ten words (mean length 5.7), in 41 cases the correct parse was the most highly ranked, giving a correct parse / sentence measure of 75%, the same as before. The second normalisation scheme unexpectedly produced much worse results: only 60% of the correct analyses were returned as the highest ranked when reparsing the training corpus.

Despite the fact that our current implementation of probabilistic unpacking is quite crude and could be speeded up significantly, for a sample of ten-word definitions with up to 150 analyses, the time the implementation takes to return the three highest ranked analyses from the parse forests is of the order of a factor of 3 less than in the original near-exhaustive search of the forest. In the course of searching for the best analyses, on average only about 20% of the nodes in the forests were visited. For longer definitions the difference is even more marked. Our current implementation has returned analyses for every definition for which the parser managed to successfully create a parse forest; the longest such definition is 31 words in length. Previously, the longest definition that we managed to return an analysis for was only 22 words long. We have been unable to compute the full set of analyses for these definitions, due to space and time constraints, but have calculated that they have at least 2000 analyses.

In the experiments that we have carried out so far, once the forest has been searched for the highest scoring analyses, it appears that the great majority of unifications that take place when unpacking them are successful. In the tests outlined above, setting up pruning of partial analyses to discard all but the best three at each stage, the number of final analyses returned for each definition was only less than the three required in the few cases where there were actually less than three analyses in total.

Conclusions and Future Work

In this paper we have presented a technique for probabilistically unpacking a packed parse forest with interleaved normalisation of the scores of partial analyses covering identical portions of the input. We have successfully applied the technique to packed parse forests created by a unification-based non-deterministic LR parser, but the technique is generally applicable to any Tomita-style parse forest created by a shift-reduce parser in which each node contains sufficient information to be able to compute a normalised probability for the subtree dominated by the node.

The technique allows our system to return the highest ranked analyses for sentences that are significantly longer than could be coped with using a near-exhaustive search of parse forests; in fact in our experiments the system has been able to return analyses for every sentence for which a parse forest could be computed. The normalisation methods described in this paper are also better-founded than the method that was used previously. However, we are still attempting to improve the system in this area.

Choosing a value for m in order to take the m-best partial analyses at each stage appears to be rather arbitrary. We are currently investigating changing our control strategy to take the maximum probability partial analysis after normalisation and then include all others in the set of competing sub-analyses which fall within a given threshold. Then if several analyses had very similar, high probabilities they would all be returned, and pruning out partial analyses would be dependent on a function of probability likelihood rather than an arbitrary number.

Acknowledgements

We would like to thank Longman Group Ltd. for allowing us access to the machine-readable version of LDOCE, and Ann Copestake and Antonio Sanfilippo for considerable help in the analysis of the LDOCE noun definition corpus.

References

Alshawi, H. ed. 1992. *The Core Language Engine.* MIT Press, Cambridge, Massachusetts.

Briscoe, E. and Carroll, J. 1991. *Generalised Probabilistic LR Parsing of Natural Language (Corpora) with Unification-based Grammars.* Cambridge University, Computer Laboratory, TR-224.

Briscoe, E. and Carroll, J. 1992. Generalised Probabilistic LR Parsing for Unification-based Grammars. *Computational Linguistics.* Forthcoming.

Briscoe, E. and Waegner, N. 1992. Robust stochastic parsing using the inside-outside algorithm. In Proceedings of the AAAI Workshop on Statistically-based Techniques in Natural Language Processing. San Jose, California.

Briscoe, E., Grover, C., Boguraev, B. and Carroll, J. 1987. A Formalism and Environment for the Development of a Large Grammar of English. In Proceedings of the 10th International Joint Conference on Artificial Intelligence, 703–708. Milan, Italy.

Carroll, J. and Grover, C. 1989. The derivation of a large computational lexicon for English from LDOCE. In Boguraev, B. and Briscoe, E. eds. *Computational Lexicography for Natural Language Processing.* Longman, London: 117–134.

Church, K. and Patil, R. 1982. Coping with syntactic ambiguity or how to put the block in the box on the table. *Computational Linguistics* 8: 139–149.

Fujisaki, T., Jelinek, F., Cocke, J., Black, E. and Nishino, T. 1989. A probabilistic method for sentence disambiguation. In Proceedings of the 1st International Workshop on Parsing Technologies, 105-114. Carnegie-Mellon University, Pittsburgh.

Johnson, M. 1989. The Computational Complexity of Tomita's Algorithm. In Proceedings of the 1st International Workshop on Parsing Technologies, 203–208. Carnegie-Mellon University, Pittsburgh.

Kipps, J. 1989. Analysis of Tomita's algorithm for general context-free parsing. In Proceedings of the 1st International Workshop on Parsing Technologies, 193–202. Carnegie-Mellon University, Pittsburgh.

Magerman, D. and Marcus, M. 1991. Pearl: a probabilistic chart parser. In Proceedings of the 2nd International Workshop on Parsing Technologies, 193–199. Cancun, Mexico.

Pereira, F. and Warren, D. 1980. Definite clause grammars for language analysis – a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence* 13.3: 231–278.

Procter, P. ed. 1978. *The Longman Dictionary of Contemporary English.* Longman, London.

Sampson, G., Haigh, R. and Atwell, E. 1989. Natural language analysis by stochastic optimization: a progress report on Project APRIL. *Journal of Experimental and Theoretical Artificial Intelligence* 1: 271–287.

Schabes, Y. 1991. Polynomial Time and Space Shift-Reduce Parsing of Arbitrary Context-free Grammars. In Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, 106–113. Berkeley, Ca..

Sharman, R., Jelinek, F. and Mercer, R. 1990. Generating a grammar for statistical training. In Proceedings of the DARPA Speech and Natural Language Workshop, 267–274. Hidden Valley, Pennsylvania.

Taylor, L., Grover, C. and Briscoe, E. 1989. The syntactic regularity of English noun phrases. In Proceedings of the 4th European Meeting of the Association for Computational Linguistics, 256–263. UMIST, Manchester.

Tomita, M. 1984. Parsers for Natural Languages. In Proceedings of the 10th International Conference on Computational Linguistics, 354–357. Stanford, California.

Tomita, M. 1987. An efficient augmented-context-free parsing algorithm. *Computational Linguistics* 13.1: 31–46.

Wright, J., Wrigley, E. and Sharman, R. 1991. Adaptive probabilistic generalized LR parsing. In Proceedings of the 2nd International Workshop on Parsing Technologies, 154–163. Cancun, Mexico.