# An Approach to Probabilistic LR Parsing

John Carroll & Ted Briscoe

June 1994

Realistic, wide-coverage NL grammars (such as the ANLT grammar) often produce large numbers of analyses for syntactically legitimate inputs. This paper describes a practical parsing system capable of returning probabilistically highly-ranked analyses for material drawn from a specific corpus on the basis of minimal (supervised) training and manual modification. The system has been tested on definitions from a machine-readable dictionary, and the results of experiments with this corpus are reported[1].

## 1 Probabilistic Approaches to Parsing

### 1.1 Previous Approaches

In the field of speech recognition, statistical techniques based on hidden Markov modelling are well established: sufficiently so to be described in textbooks, for example Holmes (1988:129f). The two main algorithms utilised are the Viterbi (1967) algorithm and the Baum-Welch algorithm (Baum, 1972). The former provides a polynomial solution to the task of finding the most probable derivation for a given input and a stochastic regular grammar, and the latter, a polynomial method of performing iterative re-estimation of the parameters of a (hidden) stochastic regular grammar by considering all possible derivations over a corpus of inputs. Baker (1982) demonstrates that Baum-Welch re-estimation can be extended to context-free grammars (CFGs) in Chomsky Normal Form (CNF).

Typically, in applications of Markov modelling in speech recognition, the derivation used to analyse a given input is not of interest; rather what is sought is the best (most likely) model of the input. In any application of these or similar techniques to parsing though, the derivation selected is of prime interest. Baum (1972) proves that Baum-Welch re-estimation will converge to a local optimum in the sense that the initial probabilities will be modified to increase the likelihood of the corpus given the grammar and 'stabilise' within some threshold after a number of iterations over the training corpus. However, there is no guarantee that the global optimum will be found, and the a priori initial probabilities chosen are critical for convergence on useful probabilities (e.g. Lari & Young, 1990). The main application of these techniques to text has been in the robust, lexical tagging of corpora with part-of-speech labels (e.g. Garside, Leech & Sampson, 1987; de Rose, 1988; Meteer, Schwartz & Weischedel, 1991; Cutting *et al.*, 1992).

Fujisaki *et al.*(1989) describe a corpus analysis experiment using a probabilistic CFG in Greibach Normal Form[2] containing 7,550 rules on a corpus of 4,206 sentences. The (unsupervised) training process involved automatically assigning probabilities to each CF rule on the basis

[1]Most of the material in this paper is adapted from Briscoe & Carroll(1991), Carroll & Briscoe (1992) and Carroll (1993).

[2]Greibach Normal Form is similar to CNF, except that each production contains a terminal category as its leftmost daughter.

of their frequency of occurrence in all possible analyses of each sentence of the corpus. These probabilities were iteratively re-estimated using a variant of the Baum-Welch algorithm, and the Viterbi algorithm was used in conjunction with the CKY parsing algorithm (Kasami, 1965; Younger, 1967) to efficiently select the most probable analysis after training. The model was restricted in that many of the possible parameters (rules) defined over the category set were initially set to zero and training was used only to estimate new probabilities for a set of predefined rules. Fujisaki *et al.* suggest that the stable probabilities will model semantic and pragmatic constraints in the corpus, but this will only be so if these correlate with the frequency of rules in correct analyses, and also if the 'noise' in the training data created by the incorrect parses is effectively factored out. Whether this is so depends on the number of 'false positive' examples with only incorrect analyses, the degree of heterogeneity in the training corpus, and so forth. In experiments reported by Fujisaki *et al.*, when reparsing the corpus that was originally used for training, in 72 out of 84 sentences examined the most probable analysis was also the correct analysis. Six of the remainder were false positives and did not receive a correct parse, whilst the other six did, but it was not marked as the most probable. Thus the success rate (per sentence) is 85%.

Sharman, Jelinek & Mercer (1990) conducted a similar experiment with a grammar in ID/LP format (Gazdar *et al.*, 1985; Sharman, 1989). In this experiment, a supervised training regime was employed. A grammar containing 100 terminals and 16 non-terminals and initial probabilities based on the frequency of ID and LP relations was extracted from a manually parsed corpus of about one million words of text. The resulting probabilistic ID/LP grammar was used to parse sentences drawn from the same corpus. In addition, lexical syntactic probabilities were integrated with the probability of the ID/LP relations to rank parses. In an experiment with 42 sentences of 30 words or less drawn from the training corpus, 18 of the parses were identical to the original manual analyses, whilst a further 19 were 'similar', yielding a success rate of 88%.

Pereira & Schabes (1992) report an experiment using Baum-Welch re-estimation to infer a grammar and associated rule probabilities from a category set containing 15 non-terminals and 48 terminals, corresponding to the Penn Treebank lexical tagset (Santorini, 1990). The training data was 770 sentences, represented as tag sequences, drawn from the treebank. The system was trained in an unsupervised mode and also in a 'semi-supervised' mode in which the manually parsed version of the corpus was used to constrain the set of analyses used during re-estimation. In supervised training, analyses were accepted if they produced bracketings consistent but not necessarily identical with those assigned manually. In supervised mode, training not only converged faster but also resulted in a grammar in which the most probable analysis was compatible with the manually assigned analysis of further test sentences drawn from the treebank in a much greater percentage of cases: 78% as opposed to 35%. This result indicates very clearly the importance of supervised training, particularly in a context where the grammar itself is being inferred in addition to the probability of individual rules.

Sampson, Haigh & Atwell (1989) describe a parser which, in order to assign an analysis, uses a statistically-defined measure of 'closest fit' to the set of hand-assigned analyses contained in training data taken from the LOB treebank (Sampson, 1987). The approach, based on simulated annealing, attempts to ensure that analyses of new data will conform as closely as possible to existing ones. Sampson *et al.* express the results of experiments with this technique in terms of a tree similarity measure with respect to correct analyses assigned by hand. For a 13- sentence

```
1)  S' -> S       (1.0)       6)   NP -> Det N   (.3)
2)  S -> NP VP    (1.0)       7)   NP -> NP PP   (.3)
3)  VP -> Vt NP   (.4)        8)   N -> N N      (.3)
4)  VP -> Vi      (.6)        9)   PP -> P NP    (1.0)
5)  NP -> ProNP   (.4)        10)  N -> N@       (.7)
```

Figure 1: G1—a PCFG.

sample the mean similarity measure was 80%, but only one input received a fully correct analysis. These results suggest that the technique in its current form is not reliable enough for practical NL analysis. Moreover, the analyses assigned, based on the LOB treebank scheme, are not syntactically determinate (for example syntactic relations in unbounded dependency constructions are not represented).

## 1.2 Problems with Probabilistic CFG

Although the experimental results reported for the probabilistic parsing systems mentioned above are quite impressive, there are nevertheless several inherent problems with probabilistic CFG (including ID/LP) based systems. Firstly, although CFG is an adequate model of the majority of constructions occurring in natural language (Gazdar & Mellish, 1989), it is clear that wide-coverage CFGs will need to be very large indeed, and this will lead to difficulties of (manual) development of consistent grammars and, possibly, to computational intractability at parse time (particularly during the already computationally expensive training phase). Secondly, without further apparatus, the analyses returned by grammars based on a linguistically relatively unsophisticated formalism such as CFG (even in an ID/LP version), or derived by analogy with manual analyses of the training corpus and therefore not well-formed with respect to any given generative grammar (e.g. Sampson *et al.*'s system), are not likely to be detailed enough to support subsequent semantic interpretation. Thirdly, associating probabilities with CF rules means that information about the probability of a rule applying at a particular point in a parse derivation is lost. This leads to complications distinguishing the probability of different derivations when the same rule can be applied several times in more than one way. Figure 1 gives an example of a probabilistic CFG (PCFG); each production is associated with a probability and the probabilities of all rules expanding a given non-terminal category sum to one[3].

In PCFG, the probability of a particular parse is the product of the probabilities of each rule used in the derivation. Thus the probability of parse a) in figure 2 is 0.09. Parse b) must have an identical probability, though, because the same rule is applied twice in each case. Similarly, the probabilities of c) and d) are also identical (0.09) for essentially the same reason. However, the rules concerned are natural treatments of noun compounding and prepositional phrase attachment in English, and the different derivations correlate with different interpretations. For example, a) would be an appropriate analysis for toy coffee grinder, whilst b) would be appropriate for cat food tin, and each of c) and d) yields one of the two possible interpretations of the noun phrase the man in the park with the telescope. Ideally, these structural configurations should be kept

[3]Booth & Thompson (1973) detail other restrictions which ensure consistency.

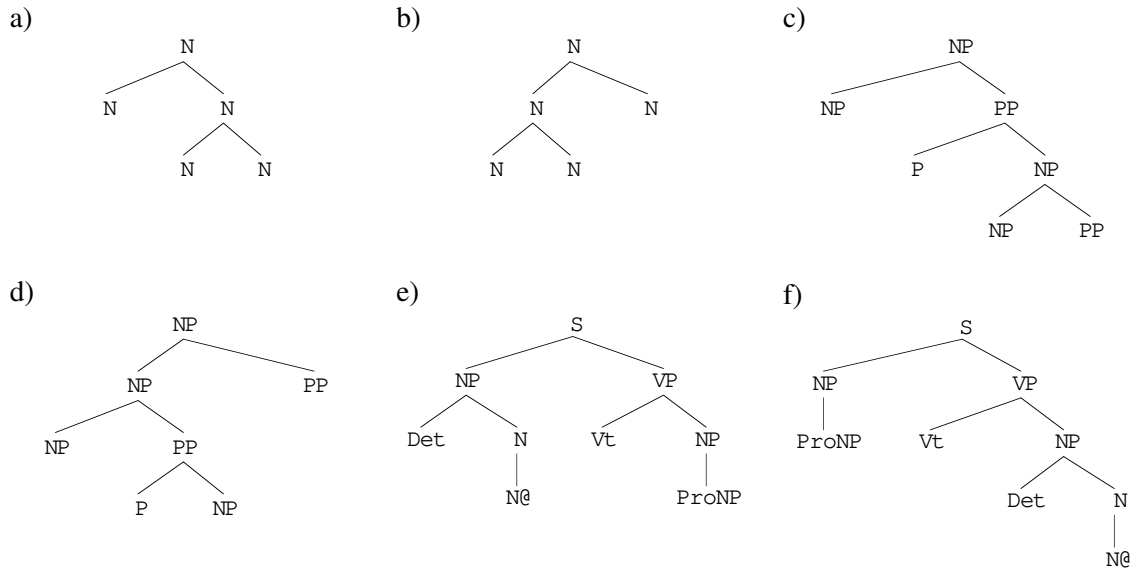


Figure 2: Context-free Derivations.

probabilistically distinct in case there are structurally conditioned differences in their frequency of occurrence, as would be predicted, for example, by the theory of parsing strategies (e.g. Frazier, 1988). Fujisaki *et al.* (1989) propose a rather inelegant solution for the noun compound case which involves creating 5,582 instances of 4 morphosyntactically identical rules for classes of word forms with distinct bracketing behaviour in noun-noun compounds. This technique enlarges the grammar excessively, merges probabilistic lexical information with probabilistic structural information, and also multiplies out the parameters of the probabilistic model.

Moreover, PCFGs cannot model the context dependence of rule use; for example, a noun phrase is more likely to be expanded as a pronoun in subject position than elsewhere (e.g. Magerman & Marcus, 1991) but only one global probability can be associated with the relevant CF production. Thus the PCFG model predicts (incorrectly) that e) and f) will have the same probability of occurrence. These considerations suggests the need for a technique which employs a more adequate grammatical formalism than CFG and a more context dependent probabilistic model.

Several researchers (Wright & Wrigley, 1989; Wright, 1990; Ng & Tomita, 1991; Wright, Wrigley & Sharman, 1991) have proposed using the LR technique as a practical method of parsing with a PCFG. In these approaches, an LR parse table is derived from the CF component of the grammar in the standard way, but in addition, the rule probabilities are distributed amongst sets of actions in the LR parse table. The distribution is performed in such a way as to ensure that the product of the probabilities attached to the LR actions which are taken in the derivation of any given analysis will be identical to the probability which would be assigned to the analysis by the original PCFG.

The following sections describe a novel approach to probabilistic parsing, using the LR parsing technique as a way to obtain a finite-state representation of a non finite-state grammar incorporating information about parse context, and demonstrate that LR parse tables do indeed provide an appropriate amount of contextual information to solve the problems described above. In this approach,

a CF backbone grammar and LR parse table is computed for the ANLT grammar (see Briscoe & Carroll, 1993), and probabilities are associated directly with the parse table, rather than being associated with rules of the grammar as in PCFG (or equivalently, being mapped from a PCFG into an LR parse table). An LR parse state encodes information about the left and right context of the current parse. In the system, probabilities are assigned to transitions in the LALR(1) table via a process of supervised training based on computing the frequency with which transitions are traversed in a disambiguated corpus of parse histories constructed using a user-driven, interactive LR parser. Deriving probabilities relative to the parse context allows a probabilistic parser, unlike a PCFG, to distinguish 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) or situations in which identical rules apply with differing probabilities in different contexts. The probabilistic LR parser remains a stochastic first-order finite-state model, though, because the LALR(1) table defines a non-deterministic finite-state machine and the total probability of an analysis is computed from the sequence of transitions taken to construct it.

### 1.3 Language Models and Decision Procedures

The approach outlined above does not result in a language model (as is PCFG), since the probability of a sentence is not the sum of the probabilities of all its possible derivations. The approach instead constitutes a decision procedure. Language models trained using Baum-Welch re-estimation maximise the sum of the probabilities of the sentences in a corpus given a grammar. However, it is debatable whether this property is what is required when the task is to select a single, plausible derivation as the analysis for each sentence, since there is no clear connection between the probability of a derivation and that of a sentence. For this task, it would be more appropriate to maximise the probability of the correct derivation, given a sentence in the training corpus. Thus, it is not clear that language models are suitable for the task of parse selection.

Other researchers have noticed these problems with language models and PCFG, and have described approaches to parsing based on the same idea of using context-sensitive derivation probabilities. For example, Su *et al.* (1989) (see also Chang *et al.*, 1992) define a general probabilistic method of parse ranking for LR parsing with CFG, distinguishing syntactic contextual factors by associating scores with shift actions in the parse table. Pearl (Magerman & Marcus, 1991) and Picky (Magerman & Weir, 1992) use decision procedures which estimate the probability of a derivation as the geometric mean of the product of the conditional probabilities of each CFG rule appearing in the derivation given the (non-terminal trigram) context of its dominating rule and the part-of-speech trigram at that point. Chitrao & Grishman (1990) also describe a context dependent probabilistic model in which the probability of each possible expanding production for each non-terminal appearing on the right hand side of a rule is recorded separately, and report a 23% decrease in incorrect analyses over a previous (standard) PCFG. However, although achieving promising results in a limited domain, all of these systems are based on CFG and thus are unlikely to produce analyses that are detailed enough to support semantic interpretation.

Black *et al.* (1992) describe a probabilistic disambiguation technique called history-based grammar (HBG) which models contextual factors and is also applicable to any grammar formalism which has a derivation process. From a corpus of bracketed sentences, HBG builds decision trees to encode the relevant factors in an analysis that determine the correct parse of a sentence. In

experiments, HBG was found to have an error rate that was 37% less than a comparable PCFG model.

The framework and implemented system described in the rest of this paper is intended to constitute a principled and linguistically adequate approach. The next section describes the techniques used for training the probabilistic LR system, §3 describes the probabilistic parser, and §4 reports the results of a corpus parsing experiment using the system.

# 2 Training the Probabilistic System

The first stage in training the probabilistic system is to construct disambiguated training corpus that is compatible with grammar. Once this is accomplished, probabilities are derived from the disambiguated data.

## 2.1 Constructing a Disambiguated Training Corpus

The major problem with attempting to employ a disambiguated training corpus is to find a way of constructing this corpus in an error-free and resource-efficient fashion. Even manual assignment of lexical categories ('tagging') is slow, labour intensive and error prone. The greater complexity of constructing a complete parse makes the totally manual approach very unattractive, if not impractical. Sampson (1987:83) reports that it took 2 person-years to produce the LOB treebank of 50,000 words. Furthermore, in that project, no attempt was made to ensure the analyses were well-formed with respect to a generative grammar. Attempting to manually construct analyses consistent with a grammar of any size and sophistication would place an enormous additional load on the analyst. Leech & Garside (1991) discuss the problems which arise in manual parsing of corpora concerning accuracy and consistency of analyses across time and analyst, the labour intensive nature of producing detailed analyses, and so forth. They advocate an approach in which simple 'skeleton' parses are produced by hand from previously tagged material, with checking for consistency between analysts. These skeleton analyses can then be augmented automatically with further information implicit in the lexical tags. Whilst this approach may well be the best that can be achieved with fully manual techniques, it is still unsatisfactory. In particular, the quality of any grammar obtained automatically from the parsed corpus is likely to be poor because of the lack of any rigorous checks on the form of the skeleton parses. Such a grammar might, in principle, be trained from the parsed corpus, but there are still likely to be mismatches between the actual analysis assigned manually and any assigned automatically.

These training problems can be overcome by employing a method which produces a training corpus using the same grammar which ultimately is to be trained, thus guaranteeing that any analysis assigned will be a member of the set defined by the grammar. We have experimented with three methods for producing a training corpus.

The first two methods use a similar type of technique to that employed in the construction of a substantial lexicon for the ANLT grammar from unreliable and indeterminate data (Carroll & Grover, 1989). The disambiguated training corpus is created semi-automatically, with manual interaction being restricted to the selection between alternatives defined by the ANLT grammar. Briscoe (1987) demonstrates that the structure of the search space in parse derivations allows a left-to-right, incremental mode of parse selection to be effective. For example, in noun compounds

analysed using a recursive binary-branching rule (N –> N N) the number of analyses correlates with the Catalan series (Church & Patil, 1982). However, Briscoe (1987:154f) shows that with a simple bounded context parser (with one word lookahead) set up to request help whenever a parse indeterminacy arises, it is possible to select any of the 14 analyses of a 5 word compound with a maximum of 5 interactions and any of the 1,430 analyses of a 9 word compound with around 13 interactions. In general, resolution of the first indeterminacy in the input will rule out approximately half the potential analyses, resolution of the next, half of the remaining ones, and so on. For "worst case" CF ambiguities, this approach to parse selection appears empirically to involve numbers of interactions which increase at little more than linear rate with respect to the length of the input.

It is possible to exploit this insight in two different ways. One method is to compute all possible analyses represented as a (packed) parse forest and ask the user to select between competing sub-analyses which have been incorporated into a successful analysis of the input. In this way, only genuine global syntactic ambiguities would need to be considered by the user. However, a disadvantage of this approach is that it relies on a prior (and perhaps CPU-intensive) on-line computation of the full set of analyses. The second method (described in detail in Briscoe & Carroll, 1993; Carroll, 1993) involves a user interacting with an incremental parser to guide it through the search space of possibilities. This has the advantage of being guaranteed to be computationally tractable, but the potential disadvantage of requiring the user to resolve local syntactic ambiguities which will not be incorporated into a successful analysis. Nevertheless, LR and augmented lookahead techniques mitigate this problem to a certain extent. We have experimented with both of these methods, and currently prefer the first, mainly because fewer choices are in general required[4].

The third method for producing a training corpus uses independently-produced bracketings (parses) of the training corpus. Such data have recently become available, produced from manually parsing large numbers of sentences. The method takes a sentence and an unlabelled bracketing for it, and computes the analysis licensed by the grammar that is 'closest' to the bracketing (in fact, the analysis which minimises the number of constituents which are in the bracketing but missing from the analysis[5]).

### 2.1.1 Associating Probabilities with an LR Parse Table

The sequences of shift, reduce and accept actions in the LALR(1) parse table performed to construct the analyses in the disambiguated training corpus are computed, and the frequencies of parse table transitions the actions represent are used to create a probabilistic version of the parse table for a non-deterministic probabilistic parser.

The most straightforward technique for associating probabilities with the parse table is to assign a probability to each action in the action part of the table (e.g. Wright, 1990; Ng & Tomita, 1991). In this approach, probabilities taken from a PCFG are distributed over sets of actions in such a way as to ensure that for every analysis licensed by the grammar, the product of the scores of the LR actions

[4]The amount of interaction required can be decreased still further by counting the number of analyses for each sentence in the training corpus in a pre-processing step, and not attempting to train on those which have many analyses.

[5]This approach is not based on penalising crossings (see e.g. Grishman, Macleod & Sterling, 1992), since it seems plausible that, for example, with the training bracketing *((a b) ((c) d))*, the analysis *((a b) ((c d)))* with one span not in the training bracketing but one crossing is better than *(a b c d)* with three spans missing but zero crossings. So additional bracketing in grammar analyses (which is usually the case) is not penalised.

in its derivation is identical to the probability that would be assigned to the analysis by the PCFG. However, if this constraint is relaxed, and probabilities are associated directly with individual actions in the parse table (rather than being derived from a PCFG), then the resulting probabilistic model can be more sensitive to parse context. For example, in a derivation for the sentence he loves her using the grammar G1, the steps of reducing the first pronoun and second pronoun in the sentence to NP—using the same rule NP –> ProNP in both cases—can be distinguished by the different lookahead items paired with the reduce actions relating to this rule); in the first case, the lookahead item will be Vi, and in the second, $. Although it is sensitive to parse context to a certain extent, this approach still does not make maximal use of the context encoded in the parse table, and it is possible to devise situations in which the reduction of a pronoun in subject position and elsewhere would be indistinguishable in terms of lookahead alone; for example, in this framework, if appropriate rules for adverbs are added to G1, then Adv would be the lookahead for reductions of pronouns in both subject and object positions in inputs like he probably loves her passionately.

A slightly more complex approach is to further subdivide reduce actions according to the state reached after the reduce action has applied. This state is used together with the resultant non-terminal to define the state transition in the goto part of the parse table. Thus, this technique corresponds to associating probabilities with transitions in the automaton defined by the parse table, rather than with actions in the action part of the table. For example, a reduction of pronoun to NP in subject position in the parse table for G1 always results in the parser returning to state zero (from which the goto table deterministically prescribes a transition to state 7 with non-terminal NP). Reduction to NP of a pronoun in object position always results in the parser returning to state 11. Thus training on a corpus with more subject than non-subject pronominal NPs will now result in a probabilistic preference for reductions which return to 'pre-subject' states with 'post-subject' lookaheads. Of course, this does not mean that it will be impossible to devise grammars in which reductions cannot be kept distinct which might, in principle, have different frequencies of occurrence. However, this approach appears to be the natural stochastic, probabilistic model which emerges when using an LR table. Any further sensitivity to context would require sensitivity to patterns in larger sections of a parse derivation than can be defined in terms of such a table.

Figure 3 shows a probabilistic LALR(1) parse table for G1 which was derived from a simple, partial (and artificial) training phase. Since probabilities are associated with transitions rather than with actions, there can be more than one transition corresponding to a single reduce action, with a separate probability for each depending on the state reached after the action; for example, in state 4 with lookahead N@, the probability of reducing with rule ten is 0.17 if the state reached is 3 and 0.22 if the state reached is 5. The actions which have no associated probabilities correspond to transitions which have not been utilised during the training phase; each is assigned a smoothed probability which is the reciprocal of the result of adding one to the total number of observations of transitions actually taken from that state. For this reason, the explicit probabilities in each row of the table sum to less than one. Unseen events are thus assigned differential probabilities in a manner analogous to the type of smoothing performed by the Good-Turing technique (Good, 1953). The goto part of the table is not shown because it is always deterministic and, therefore, probabilities are not associated with the entries in it.

| State | $ | Det | N@ | P | ProNP | Vi | Vt |
|---|---|---|---|---|---|---|---|
| 0 | | s3<br>(.5) | | | s2<br>(.5) | | |
| 1 | r1<br>(0 .83) | | | | | | |
| 2 | r5 | | | r5<br>(8 .33) | | r5 | r5<br>(0 .50) |
| 3 | | | s4<br>(1.00) | | | | |
| 4 | r10<br>(3 .11 6 .11) | | r10<br>(3 .17 5 .22) | r10<br>(3 .11) | | r10<br>(3 .11 5 .11) | r10 |
| 5 | r6<br>(8 .13 11 .13) | | s4<br>(.33) | r6<br>(11 .13) | | r6<br>(0 .20) | r6 |
| 6 | r8<br>(3 .17 5 .17) | | r8 / s4<br>(3 .25) / (.17) | r8 | | r8<br>(3 .17) | r8 |
| 7 | | | | s8 | | s13<br>(.43) | s11<br>(.43) |
| 8 | | s3<br>(.5) | | | s2<br>(.5) | | |
| 9 | r9<br>(12 .40) | | | r9 / s8<br>(12 .40) / | | r9 | r9 |
| 10 | r7<br>(11 .40) | | | r7<br>(11 .40) | | r7 | r7 |
| 11 | | s3<br>(.75) | | | s2 | | |
| 12 | r3<br>(7 .43) | | | s8<br>(.43) | | | |
| 13 | r4<br>(7 .75) | | | | | | |
| 14 | r2<br>(0 .84) | | | | | | |
| 15 | acc<br>(1.00) | | | | | | |

Figure 3: A probabilistic version of the parse table for G1.

### 2.1.2 Using LR Transition Probabilities

The difference between the approach described above and one based on PCFG can be brought out by considering parse derivations using the probabilistic parse table for G1. Assuming that probabilities are being used simply to rank parses, the total probability of an analysis can be computed by taking the product of the probabilities of each transition taken during its derivation. Figure 4 shows the two possible complete derivations for a sentence like the winter holiday camp closed consisting of a determiner, three nouns and an intransitive verb. The ambiguity concerns whether the noun compound is left- or right-branching, and, as discussed in §1, a PCFG cannot distinguish these two derivations. The probability of each step can be read off the action table (figure 3), and is shown after the lookahead item in the derivations.

In step 8 a shift-reduce conflict occurs so the stack 'splits' while the left- and right-branching analyses of the noun compound are constructed. The a) branch corresponds to the right-branching derivation and the product of the probabilities is $4.6 \times 10^{-8}$, whilst the product for the left-branching b) derivation is $5.1 \times 10^{-7}$. Since the table was constructed from parse histories with a preponderance of left-branching structures this is the desired result. In practice, this technique is able to distinguish and train accurately on three of the five possible structures for a four-word noun-noun compound; but it inaccurately prefers a completely left-branching analysis over structures of the form *((n n) (n n))* and *((n (n n)) n)*. Moving on to five-word noun-noun compounds, performance degrades further. However, this level of performance on such structural configurations is likely to be adequate, because correct resolution of most ambiguity in such constructions is likely to be dominated by the actual lexical items which occur in individual texts. Nevertheless, if there are systematic structural tendencies evident in corpora (for example, Frazier's (1988) parsing strategies predict a preference for left-branching analyses of such compounds), then the probabilistic model is sensitive enough to discriminate between them.

# 3 Probabilistic LR Parsing

The probabilistic parser uses the LR table transition probabilities derived in the training phase (as described in the previous section), to return the set of highest ranked analyses of an input. The parser consists of two processing stages: the first takes the input and computes a probabilistic packed parse forest, and the second extracts the n highest ranked analyses from the forest.

## 3.1 Computing a Probabilistic Parse Forest

The first stage of the probabilistic LR parser is based on a non-deterministic unification-based LR parser (briefly, a generalisation for unification grammar of Kipps' (1989) CF LR recogniser). This stage constructs a packed parse forest representation of the set of analyses licensed by the grammar for a given input. In this representation identical sub-analyses are shared between differing superordinate analyses, and sub-analyses covering the same portion of input are packed if the root categories are related by subsumption.

Ng & Tomita (1991) describe an LR parser for PCFG which associates probabilities with nodes in the graph-structured stack. When branches of the stack are merged (either due to a shift to the same state, or a goto action to an already existing vertex), then the vertex probability must be

```
                    0) 0                  (Det)
                    1) 0 Det 3            (N@1)  .50
                    2) 0 Det 3 N@ 4       (N@2) 1.00
                    3) 0 Det 3 N          (N@2)
                    4) 0 Det 3 N 5        (N@2)  .17
                    5) 0 Det 3 N 5 N@ 4   (N@3)  .33
                    6) 0 Det 3 N 5 N      (N@3)
                    7) 0 Det 3 N 5 N 6    (N@3)  .22

 8a) 0 Det 3 N 5 N 6 N@ 4  (Vi)  .17      8b) 0 Det 3 N          (N@3)
 9a) 0 Det 3 N 5 N 6 N     (Vi)           9b) 0 Det 3 N 5        (N@3)  .25
10a) 0 Det 3 N 5 N 6 N 6   (Vi)  .06     10b) 0 Det 3 N 5 N@ 4  (Vi)   .33
11a) 0 Det 3 N 5 N         (Vi)          11b) 0 Det 3 N 5 N     (Vi)
12a) 0 Det 3 N 5 N  6      (Vi)  .08     12b) 0 Det 3 N 5 N 6   (Vi)   .11
13a) 0 Det 3 N             (Vi)          13b) 0 Det 3 N         (Vi)

                   14) 0 Det 3 N 5        (Vi)   .17
                   15) 0 NP               (Vi)
                   16) 0 NP 7             (Vi)   .20
                   17) 0 NP 7 Vi 13       ($)    .43
                   18) 0 NP 7 VP          ($)
                   19) 0 NP 7 VP 14       ($)    .75
                   20) 0 S                ($)
                   21) 0 S 1              ($)    .83
                   22) 0 S'               ($)
                   23) 0 S' 15            ($)   1.00
```

Figure 4: Parse derivations for *the winter holiday camp closed*.

updated to be the sum of the products of the action and parent vertex probabilities. Because an addition operation is involved, this calculation is inconvenient to perform when probabilities are represented as logarithms (which make multiplication fast), so Ng & Tomita propose an approximate 'maximum likelihood approach' in which addition is replaced by maximisation. When splitting the stack on the reduction (popping) of a merged vertex, the probabilities of the parent vertices have to be recovered, requiring quite a complicated arithmetical manipulation. In general there can be more than one splitting in a single reduce action, and the arithmetic involved "can be tricky" (Ng & Tomita, 1991:157), leading them to propose reverting to a simpler type of (tree-structured) stack.

The probabilistic LR parser overcomes these problems by associating probabilities with nodes in the *parse forest* being built, rather than with nodes in the *graph-structured stack*. Each node in the forest (including packed nodes) holds the probability of the LR transition—shift or reduce—which created that node. The total probability for any analysis in the forest can be computed simply by combining the probabilities of the nodes it contains, without regard for any splitting and merging of the stack.

Ideally, for any given input, the computation of the most probable analysis (or n-best analyses) defined by the probabilistic LR parser should not involve the construction of the complete parse forest, and thus an exhaustive search of the space of syntactically legitimate analyses defined by the grammar. 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. The parser therefore computes the complete parse forest.

### 3.2 Probabilistic Unpacking and Normalisation

Wright, Wrigley & Sharman (1991) describe a Viterbi-like algorithm to find the n-best analyses from a probabilistic parse forest, in which sub-analysis probabilities are the product of the probabilities of the lexical items spanned and of the PCFG rules involved in the sub-analysis derivation. The algorithm involves evaluating all the competing analyses for the same portion of input and pruning the lowest scoring ones. However, the competing (sub-)analyses at any stage might contain differing numbers of rule applications (since Wright *et al.* do not restrict the grammar to CNF, or some other notation which would guarantee that all analyses covering the same input would contain the same number of rules), and so the system would be likely to develop a bias in favour of analyses involving fewer rules, or equivalently 'smaller' trees, almost regardless of the training material[6]. Magerman & Marcus (1991) observe this problem in the experiments they perform, and propose that analysis probabilities be normalised by computing the geometric mean of the rule probabilities before they are compared. Using geometric means, as opposed to products, achieves a crude form of normalisation with respect to the number of rules in the analysis. This form of normalisation, though, cannot be integrated into Wright *et al.*'s unpacking algorithm without compromising the algorithm's soundness, since if all sub-analyses are normalised before pruning, then it is possible that a sub-analysis that is lower scoring after normalisation but which contains more rules than a competing one might be pruned, whereas the position could be reversed after the sub-analyses were integrated into a higher-level analysis. On the other hand, if sub-analyses were not normalised, then analyses involving smaller numbers of rules might be pruned even if ultimately they would form high-scoring complete analyses after the final normalisation step.

In addition, Wright *et al.*'s approach cannot be applied as it stands to unification-based systems because, in unpacking the forest, the 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 CFG parse forests.

The most straightforward way of extracting the n-best (normalised) analyses from a unification-based probabilistic parse forest would be to exhaustively unpack the forest to retrieve each analysis, normalise its probability and return the highest scoring analyses. This method is impractical for large forests, and indeed, with the ANLT grammar, the longest string for which the parse forest has successfully been fully unpacked is only 22 words long. However, a heuristic (but nevertheless correct) technique for a best-first non-exhaustive search of the parse forest has been developed, which, in practice, makes the recovery of the most probable analyses much more efficient (allowing

[6] An approach to training based on a version of Baum-Welch re-estimation applied to non-CNF grammars (e.g. the trellis algorithm of Kupiec (1991)), although maximising the likelihood of the corpus given the grammar, would also be likely to incorrectly rank the analyses for individual sentences due to an inappropriate bias towards smaller trees.

analysis of input strings containing over 30 words).

The technique is to identify the m-best analyses in the forest (where $m \geq n$) 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, 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, some of the problems induced by the non-local nature of feature propagation are avoided.

### 3.2.1 Identifying the Best Partial Analyses

The probabilistically-guided search to find the best analyses in a probabilistic packed parse forest involves maximising the (normalised) score at nodes in the parse forest at successive positions in the input, in a similar manner to the Viterbi algorithm. The method starts at the left edge of the forest—at string position zero, just before the first word—extending the current set of partial analyses to the next position in the input. (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). The score of a partial analysis is the product of the scores of the nodes it contains. At each successive position $p$, the scores of all partial analyses ending at the same node in the forest are normalised, and all but $m$ highest scoring ones at each node are pruned. The process halts after pruning the lowest scoring analyses when $p$ is equal to the length of 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.

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.

Despite the fact that the current implementation of probabilistic unpacking is quite crude and could be speeded up significantly, for a sample of ten-word input strings with up to 150 analyses (from the experiment reported in §4 below), the time the implementation takes to return the three highest ranked analyses from the parse forests is of the order of a factor of three less than in an exhaustive unpacking 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 inputs the difference is even more marked. The current implementation is able to return n-best analyses for every string for which the parser manages to create a parse forest; the longest such string is 31 words in length.

Due to space and time constraints, the parser is unable to compute the full set of analyses for this string, but calculations indicate that it has at least two thousand analyses.

In the experiments 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 experiment reported in section §4, setting up pruning of partial analyses to discard all but the best three at each stage (m), the number of final analyses returned for each input was only less than the three required (n) in the few cases where there were actually less than three analyses in total.

### 3.2.2 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 with respect to derivation length before they can be compared and the best ones identified. The scores of all the competing partial analyses which end at a given node in the forest are normalised 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, for a) the geometric mean of the three figures, squared (0.32 = 0.09), and for b), the geometric mean of the three figures (0.2), to give normalised scores. This computation can be performed quickly since transition scores are held as logarithms.

The need for the normalisation step reflects the fact that, although the model is more sensitive to context than PCFG, it is by no means a perfect probabilistic model of NL[7]. For example, the stochastic nature of the model and the fact that the entire left context of a parse derivation is not encoded in LR state information means that the probabilistic model cannot take account of, say, the pattern of resolution of earlier conflicts in the current derivation. Another respect in which the model is approximate is that probabilities are associated with the context-free backbone rather than with the unification grammar itself. Successful unification of features at parse time does not affect the probability of a (partial) analysis, whilst unification failure, in effect, sets the probability of any such analysis to zero. As long as the probabilistic model is only used to rank successful analyses, this is not particularly problematic. However, as mentioned above, parser control regimes attempting some form of best-first search using probabilistic information associated with transitions might return incorrect results, given this property.

[7]Magerman & Marcus (1991) argue that it is reasonable to use the geometric mean when computing the probability of sub-analyses because the independence assumptions which motivate using products do not hold for such an approximate model.

# 4 Experiments Parsing LDOCE Definitions

## 4.1 The Experiments

In order to test the techniques and ideas described in the sections above, an experiment was carried out using as the test corpus a subset of noun definitions taken from the Longman Dictionary of Contemporary English (LDOCE; Procter, 1978)[8]. This corpus was chosen because the vocabulary employed is restricted (to approximately 2,000 morphemes), the average definition length is about 10 words (with a maximum of around 30), and each definition is independent, allowing phenomena such as ellipsis to be ignored. In addition, the language of definitions represents a recognisable sublanguage, allowing the task of tuning a general purpose grammar to be investigated. For the purposes of the experiment, a corpus of approximately 32,000 noun definitions was created from LDOCE by extracting the definition fields and normalising the definitions to remove punctuation, font control information, and so forth. A lexicon was created for this corpus by extracting the appropriate lemmas and matching these against entries in the ANLT lexicon. The 10,600 resultant entries were loaded into the ANLT morphological system (Ritchie *et al.*, 1987) and this sublexicon and the full ANLT grammar formed the starting point for the training process.

A total of 246 definitions, selected without regard for their syntactic form, were parsed semi-automatically using the interactive LR parsing system described above[9]. During this process, further rules and lexical entries were created for some definitions which failed to parse. 150 were successfully parsed and 63 lexical entries and 14 rules were added. Some of the rules required reflected general inadequacies in the ANLT grammar; for example, rules were added to deal with new partitives and prepositional phrase and verb complementation. However, seven of these rules cover relatively idiosyncratic properties of the definition sublanguage; for example, the postmodification of pronouns by relative clause and prepositional phrase in definitions beginning something that..., that of..., parenthetical phrases headed by adverbs, such as the period... esp the period..., and coordinations without explicit conjunctions ending with etc, and so forth. Further special rules would be required to deal with parentheses to cover lexicographic conventions such as a man (monk) or woman (nun) who lives in a monastery which were ignored for this test. Nevertheless, the number of new rules required is not great and the need for most was identified very early in the training process. Lexical entries are more problematic, since there is little evidence that the number of new entries required would tail off as more training material was encountered. However, many of the entries required reflect systematic inadequacies in the ANLT lexicon rather than idiosyncrasies of the corpus. It took approximately one person-month to produce this training corpus. As a rough guide, a single interaction with the parser takes an average of 15 seconds. However, the time a parse takes can often be lengthened by incorrect choices (and the consequent need to back up manually) and by the process of adding lexical entries and occasional rules.

The resultant parse histories were used to construct the probabilistic parser (as described in the previous section). This parser was then used to reparse the training corpus and the most highly ranked analyses were automatically compared to the original parse histories[10]. All but 3 of the 150

[8]Use of the machine readable version of LDOCE is by kind permission of Longman Group Ltd.

[9]Ann Copestake created the corpus and extracted the relevant lexical entries from the full ANLT lexicon. Antonio Sanfilippo performed some of the training.

[10]The results reported here are from using the latest versions of the ANLT grammar and LR parsing system. Briscoe

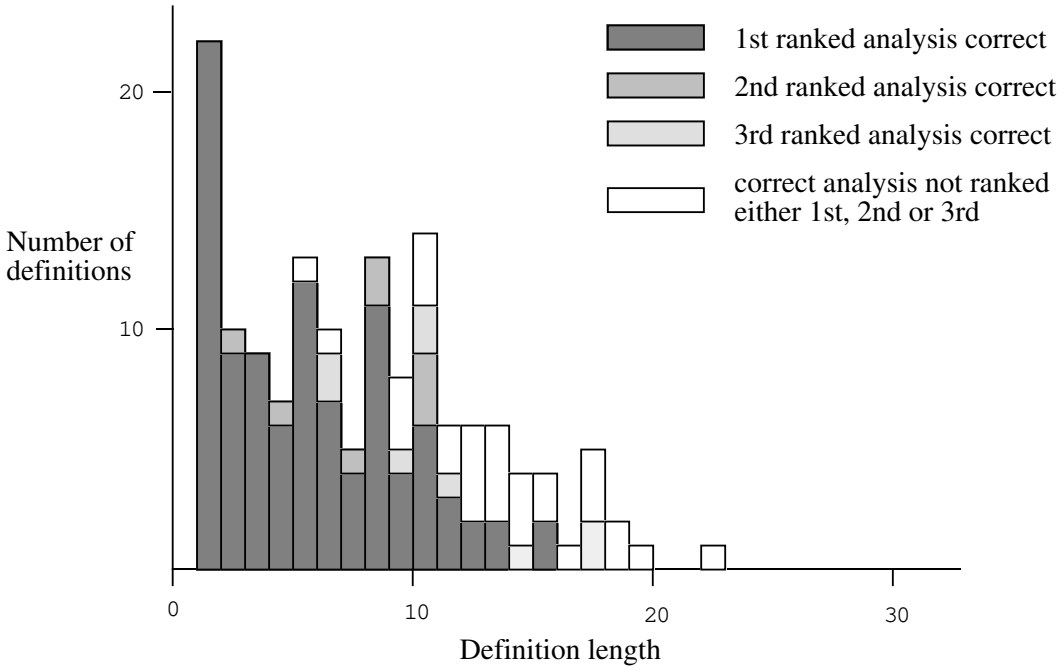


Figure 5: Correctness of results for reparsed definitions with respect to length.

definitions that were parsed manually were able to be reparsed in a breadth-first fashion. (These three are each over 25 words long). There are 22 definitions one word in length: all of these trivially receive correct analyses. There are 89 definitions between two and ten words in length inclusive (mean length 6.2). Of these, in 68 cases the correct analysis (as defined by the training corpus) is also the most highly ranked. In 13 of the 21 remaining cases the correct analysis is the second or third most-highly ranked analysis. Looking at these 21 cases in more detail, in 8 there is an inappropriate structural preference for 'low' or 'local' attachment (see Kimball, 1973), in 4, an inappropriate preference for compounds, and in 6 of the remaining 9 cases the highest ranked result contains a misanalysis of a single constituent two or three words in length. If these results are interpreted in terms of a goodness of fit measure such as that of Sampson, Haigh & Atwell (1989), the measure would be better than 96%. Taking correct parse per sentence as the measure then the result is 76%. For definitions longer than 10 words this latter figure tails off, mainly due to misapplication of such statistically-induced, but nevertheless structural, attachment preferences. The correct analysis was not returned for any definition of more than 15 words in length. Figure 5 summarises the results.

A further 55 LDOCE noun definitions not drawn from the training corpus were parsed, each containing up to 10 words (mean length 5.7). Of these, in 41 cases the correct parse is the most highly ranked, in 6 cases it is the second or third most highly ranked, and in the remaining 8 cases it is not in the first three analyses. This yields a correct parse per sentence measure of 75%. Examination of the failures again reveals that a preference for local attachment of postmodifiers accounts for 5 cases, a preference for compounds for 1, and the misanalysis of a single constituent

& Carroll (1991; 1993) report earlier versions of this experiment using different versions of the grammar and parser in which the results differed in minor ways.

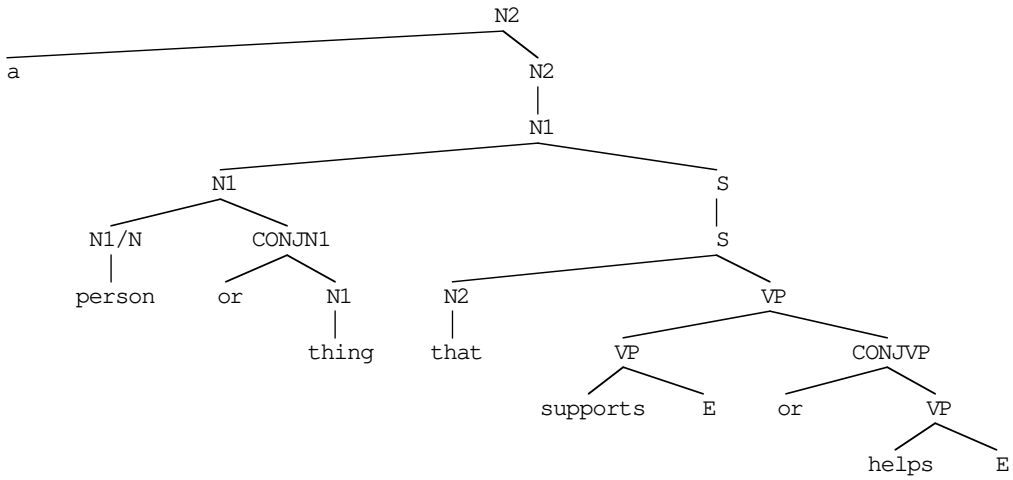


Figure 6: Parse Tree for *a person or thing that supports or helps*.

for 2. The others are mostly caused by the lack of lexical entries with appropriate SUBCAT features.

Figure 6 shows the highest ranked analysis assigned to one definition of aid. This is an example of a false positive which, in this case, is caused by the lack of a lexical entry for support as an intransitive verb. Consequently, the parser finds, and ranks highest, an analysis in which supports and helps are treated as transitive verbs forming verb phrases with object noun phrase gaps, and that supports or helps as a zero relative clause with that analysed as a prenominal subject (compare a person or thing that that supports or helps). It is difficult to fault this analysis and the same is true for the other false positives observed to date. Such false positives present probably the biggest challenge to this type of system.

## 4.2 Discussion of Experimental Results

Fujisaki *et al.* (1989) report a success rate per sentence (with an average length of 11 words) of 85%, when reparsing the training corpus; this figure is impressive, but it is difficult to evaluate properly in the absence of full details concerning the nature of the corpus. For example, if the corpus contains many simple and similar constructions, the unsupervised training that they perform is more likely to converge quickly on a useful set of probabilities. Also, Fujisaki *et al.* appear to treat lexemes as terminal categories, so since their grammar is in Greibach Normal Form, it represents a substantial amount of lexical information. For example, two possible productions might be VP –> believe Sfin and VP –> believe NP; associating differential probabilities with these rules would allow the probabilistic model to represent the relative probability of believe subcategorising for a sentential complement or object noun phrase.

Sharman, Jelinek & Mercer's (1990) experiments yield a success rate (based on a tree similarity metric) of 88% for sentences drawn from the training corpus. In Pereira & Schabes' (1992) experiments, on a test corpus, 78% of the most probable analyses are consistent (with respect to bracketing) with manually assigned analyses. What is noticeable about these experiments is that the results appear to be no better than those of Fujisaki *et al.*, despite the use of supervised training and more sophisticated grammatical models. It is likely that these differences derive from the

corpus material used for training and testing and the lack of such detailed lexical information, and that the results reported by Fujisaki *et al.* will not be achieved with all corpora.

In the preliminary experiments with the ANLT grammar and the LDOCE definitions sublanguage reported above, results comparable to these systems were achieved on the basis of a quite modest amount of manual effort and a very much smaller training corpus. This is because the manually disambiguated parse histories contain little 'noise' and usefully reflect the semantically and pragmatically appropriate analysis in the training corpus, and because the number of failures of coverage were reduced to some extent by the ability to add rules specifically motivated by the corpus during supervised training. Unlike Fujisaki *et al.* or Sharman *et al.*, the system does not integrate any information about lexemes into the rule probabilities, or make any use of information concerning the probability of a lexeme occurring as a specific lexical syntactic category. The results were achieved solely on the basis of statistics concerning the conditional probability of syntactic rules in a syntactically-defined (LR) parse context, and therefore a significant number of errors involve incorrect attachment of prepositional phrases, analyses of compounds, coordinations, and so forth, where lexical (semantic) information plays a major role. In many of these cases, the correct analysis is in the three highest ranked analyses, and moreover the structural preference for local attachment exhibited by the system might be overruled in appropriate contexts if lexeme (or better, word sense) information were taken into account. The slightly worse results (relative to mean definition length) obtained for the unseen data appear to be caused more by the non-existence of a correct analysis in a number of cases, rather than by any significant decline in the usefulness of the rule probabilities.

## References


Baker, J. (1982) "Trainable Grammars for Speech Recognition." In *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*, edited by D. Klatt & J. Wolf, 547–550. ASA.

Baum, L. (1972) "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes." *Inequalities,* III: 1–8.

Black, E., F. Jelinek, J. Lafferty, D. Magerman, R. Mercer & S. Roukos (1992) "Towards history-based grammars: using richer models for probabilistic parsing." In *Proceedings of the DARPA Speech and Language Workshop*, .

Booth, T. & R. Thompson (1973) "Applying probability measures to abstract languages." *IEEE Transactions on Computers,* C-22(5): 442–450.

Briscoe, E. (1987) *Modelling Human Speech Comprehension: A Computational Approach*. Chichester, UK: Ellis Horwood; New York: Wiley.

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

Briscoe, E. & J. Carroll (1993) "Generalised probabilistic LR parsing of natural language (corpora) with unification-based grammars." *Computational Linguistics,* 19(1): 25–59.

Carroll, J. (1993) *Practical unification-based parsing of natural language*. Computer Laboratory, Cambridge University, UK, Technical Report 314.

Carroll, J. & E. Briscoe (1992) "Probabilistic normalisation and unpacking of packed parse forests for unification-based grammars." In *Proceedings of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, Cambridge, MA. 33–38.

Carroll, J. & C. Grover (1989) "The derivation of a large computational lexicon for English from LDOCE." In *Computational Lexicography for Natural Language Processing*, edited by B. Boguraev & E. Briscoe, 117–134. London, UK: Longman.

Chang, J-S., Y-F. Luo & K-Y. Su (1992) "GPSM: a generalized probabilistic semantic model for ambiguity resolution." In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, Newark, Delaware. 177–184.

Chitrao, M. & R. Grishman (1990) "Statistical parsing of messages." In *Proceedings of the DARPA Speech and Natural Language Workshop*, Hidden Valley, PA. 263–266.

Church, K. & R. Patil (1982) "Coping with syntactic ambiguity or how to put the block in the box on the table." *Computational Linguistics,* 8: 139–49.

Cutting, D., J. Kupiec, J. Pedersen, & P. Sibun (1992) "A practical part-of-speech tagger." In *Proceedings of the 3rd Applied ACL*, Trento, Italy. 133–140.

de Rose, S. (1988) "Grammatical category disambiguation by statistical optimization." *Computational Linguistics,* 14(1): 31–39.

Frazier, L. (1988) "Grammar and language processing." In *Linguistics: The Cambridge Survey, 2*, edited by F. Newmeyer, 14–45. Cambridge, UK: Cambridge University Press.

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

Gazdar, G., E. Klein, G. Pullum & I. Sag (1985) *Generalized Phrase Structure Grammar*. Oxford, UK: Blackwell.

Gazdar, G. & C. Mellish (1989) *Natural Language Processing in Lisp*. London, UK: Addison-Wesley.

Good, I. (1953) "The population frequencies of species and the estimation of population parameters." *Biometrica,* 40: 237–264.

Grishman, R., C. Macleod & J. Sterling (1992) "Evaluating parsing strategies using standardized parse files." In *Proceedings of the 3rd ACL Conference on Applied Natural Language Processing*, Trento, Italy. 156–161.

Holmes, J. (1988) *Speech synthesis and recognition*. Wokingham, UK: Van Nostrand Reinhold.

Kasami, J. (1965) *An efficient recognition and syntax analysis algorithm for context-free languages*. Air Force Cambridge Research Laboratory, Bedford, MA, Report AFCRL-65-758.

Kimball, J. (1973) "Seven principles of surface structure parsing in natural language." *Cognition,* 2: 15–47.

Kipps, J. (1989) "Analysis of Tomita's Algorithm for General Context-Free Parsing." In *Proceedings of the 1st International Workshop on Parsing Technologies*, Carnegie-Mellon University, Pittsburgh. 193–202.

Kupiec, J. (1991) "A trellis-based algorithm for estimating the parameters of a hidden stochastic context-free grammar." In *Proceedings of the DARPA Speech and Natural Language Workshop*, Hidden Valley, Pennsylvania.

Lari, K. & S. Young (1990) "The estimation of stochastic context-free grammars using the Inside-Outside Algorithm." *Computer Speech and Language Processing,* 4: 35-56.

Leech, G. & R. Garside (1991) "Running a grammar factory: the production of syntactically analysed corpora or 'treebanks'." In *English Computer Corpora: Selected Papers and Bibliography*, edited by S. Johansson & A. Stenstrom. Berlin, Germany: Mouton de Gruyter.

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

Magerman, D. & C. Weir (1992) "Efficiency, robustness and accuracy in picky chart parsing." In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, Newark, Delaware. 40–47.

Meteer, M., R. Schwartz & R. Weischedel (1991) "POST: Using probabilities in language processing." In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Sydney, Australia. 960–965.

Ng, S-K. & M. Tomita (1991) "Probabilistic parsing for general context-free grammars." In *Proceedings of the 2nd International Workshop on Parsing Technologies*, Cancun, Mexico. 154–163.

Pereira, F. & Y. Schabes (1992) "Inside-outside re-estimation for partially bracketed corpora." In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, . 128–135.

Procter, P. (ed.) (1978) *The Longman Dictionary of Contemporary English*. London, UK: Longman.

Ritchie, G., S. Pulman, G. Russell & A. Black (1987) "A computational framework for lexical description." *Computational Linguistics,* 13: 290–307.

Sampson, G. (1987) "The grammatical database and parsing scheme." In *The Computational Analysis of English: A Corpus-based Approach*, edited by R. Garside, G. Leech & G. Sampson, 82–96. London, UK: Longman.

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

Santorini, B. (1990) *Penn Treebank Tagging and Parsing Manual*. Univ of Pennsylvania, CIS Dept, Ms.

Sharman, R. (1989) *Probabilistic ID/LP Grammars for English*. Report 217, IBM UK Scientific Centre, Winchester, England.

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

Su, K-Y., J-N. Wang, M-H. Su & J-S. Chang (1989) "GLR parsing with scoring." In *Proceedings of the 1st International Workshop on Parsing Technologies*, Carnegie-Mellon University, Pittsburgh.

Viterbi, A. (1967) "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm." *IEEE Transactions on Information Theory,* IT-13: 260–269.

Wright, J. (1990) "LR parsing of probabilistic grammars with input uncertainty for speech recognition." *Computer Speech and Language,* 4: 297–323.

Wright, J. & E. Wrigley (1989) "Probabilistic LR parsing for speech recognition." In *Proceedings of the 1st International Workshop on Parsing Technologies*, Carnegie-Mellon University, Pittsburgh. 193–202.

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

Younger, D. (1967) "Recognition and parsing of context-free languages in time $n^3$." *Information and Control,* 10(2): 189–208.