Proceedings Paper:
Haigh, R, Sampson, G and Atwell, E (1988) Project APRIL: a progress report. In: Hobbs, JR, (ed.) ACL. 26th Annual Meeting of the Association for Computational Linguistics, 07-10 Jun 1988, State University of New York at Buffalo Buffalo, New York, USA. ACL , 104 - 112.

PROJECT APRIL — A PROGRESS REPORT

Robin Haigh, Geoffrey Sampson, Eric Atwell

Centre for Computer Analysis of Language and Speech,
University of Leeds,
Leeds LS2 9JT, UK

ABSTRACT

Parsing techniques based on rules defining grammaticality are difficult to use with authentic inputs, which are often grammatically messy. Instead, the APRIL system seeks a labelled tree structure which maximizes a numerical measure of conformity to statistical norms derived from a sample of parsed text. No distinction between legal and illegal trees arises: any labelled tree has a value. Because the search space is large and has an irregular geometry, APRIL seeks the best tree using simulated annealing, a stochastic optimization technique. Beginning with an arbitrary tree, many randomly-generated local modifications are considered and adopted or rejected according to their effect on tree-value: acceptance decisions are made probabilistically, subject to a bias against adverse moves which is very weak at the outset but is made to increase as the random walk through the search space continues. This enables the system to converge on the global optimum without getting trapped in local optima. Performance of an early version of the APRIL system on authentic inputs is yielding analyses with a mean accuracy of 75.3% using a schedule which increases processing linearly with sentence-length; modifications currently being implemented should eliminate a high proportion of the remaining errors.

INTRODUCTION

Project APRIL (Annealing Parser for Realistic Input Language) is constructing a software system that uses the stochastic optimization technique known as "simulated annealing" (Kirkpatrick et al. 1983, van Laarhoven & Aarts 1987) to parse authentic English inputs by seeking labelled tree-structures that maximize a measure of plausibility defined in terms of empirical statistics on parse-tree configurations drawn from a database of manually parsed English text. This approach is a response to the fact that "real-life" English, such as the material in the Lancaster–Oslo/Bergen Corpus on which our research focuses, does not appear to conform to a fixed set of grammatical rules. (On the LOB Corpus and the research background from which Project APRIL emerged, see Garside et al. (1987). A crude pilot version of the APRIL system was described in Sampson (1986).)

Orthodox computational linguistics is heavily influenced by a concept of language according to which the set of all strings over the vocabulary of the language is partitioned into a class of grammatical strings, which possess analyses all parts of which conform to a finite set of rules defining the language, and a class of strings which are ungrammatical and for which the question of their grammatical structure accordingly does not arise. Even systems which set out to handle "deviant" sentences commonly do so by referring them to particular "non-deviant" sentences of which they are deemed to be distortions. In our work with authentic texts, however, we find the "grammaticality" concept unhelpful. It frequently happens that a word-sequence occurs which violates some recognized rule of English grammar, yet any reader can understand the passage without difficulty, and it often seems unlikely that most readers would notice the violation. Furthermore, a problem which is probably even more troublesome for the rule-based approach is that there is an apparently endless diversity of constructions that no-one would be likely to describe as ungrammatical or deviant. Impressionistically it appears that any attempt to state a finite set of rules covering everything that occurs in authentic English text is doomed to go on adding more rules as long as more text is examined; Sampson (1987) adduced objective evidence supporting this impression.

Our approach, therefore, is to define a function which associates a figure of merit with any

possible tree having labels drawn from a recognized alphabet of grammatical category-symbols; any input sentence is parsed by seeking the highest-valued tree possible for that sentence. The analysis process works the same way, whether the input is impeccably grammatical or quite bizarre. No contrast between legal and illegal labelled trees arises: a tree which would ordinarily be described as thoroughly illegal is in our terms just a tree whose figure of merit is relatively very poor.

This conception of parsing as optimization of a function defined for all inputs seems to us not implausible as a model of how people understand language. But that is not our concern; what matters to us is that this model seems very fruitful for automatic language-processing systems. It has a theoretical disadvantage by comparison with rule-based approaches: if an input is perfectly grammatical but contains many out-of-the-way (i.e. low frequency) constructions, the correct analysis may be assigned a low figure of merit relative to some alternative analysis which treats the sentence as an imperfect approximation to a structure composed of high-frequency constructions. However, our experience is that, in authentic English, "trick sentences" of this kind tend to be much rarer than textbooks of theoretical linguistics might lead one to imagine. Against this drawback our approach balances the advantage of robustness. No input, no matter how bizarre, can can cause our system simply to fail to return any analysis. Our sponsors, the Royal Signals and Radar Establishment (an agency of the U.K. Ministry of Defence)[1] are principally interested in speech analysis, and arguably this robustness should be even more advantageous for spoken language, which makes little use of constructions that are legitimate but *recherché*, while it contains a great deal that is sloppy or incorrect.

PARSING SCHEME

Any automatic parser needs some external standard against which its output is judged. Our "target" parses are those given by a scheme previously evolved for analysis of LOB Corpus material, which is sketched in Garside et al. (1987, chap. 7) and laid down in minute detail in unpublished documentation. This scheme was applied in manually parsing sentences totalling *ca* 50,000 words drawn from the various LOB genres: this TreeBank, as we call it, also serves as our source of grammatical statistics. A major objective in the definition of the parsing scheme and the construction of the TreeBank was consistency: wherever alternative analyses of a complex construction might be suggested (as a matter of analytic style as opposed to genuine ambiguity in sense), the scheme aims to stipulate which of the alternatives is to be used. It is this need to ensure the greatest possible consistency which sets a practical limit to the size of the available database; producing the TreeBank took most of one teacher's research time for two years.

The parses yielded by the TreeBank scheme are immediate-constituent analyses of conventional type: they were designed so far as possible to be theoretically uncontroversial. They were not designed to be especially convenient for stochastic parsing, which we had not at that time thought of.

The prior existence of the TreeBank is also the reason why we are working with written language rather than speech: at present we have no equivalent resource for spoken English.

THE PRINCIPLES OF SIMULATED ANNEALING

To explain how APRIL works, two chief issues must be clarified. One is the simulated annealing technique used to locate the highest-valued tree in the set of possible labelled trees; the other is the function used to evaluate any such tree.

We will begin by explaining the technique of simulated annealing. This technique uses stochastic (randomizing) methods to locate good solutions; it is now widely exploited, in domains where combinatorial explosion makes the search space too vast for exhaustive examination, where no algorithm is available which leads systematically to the optimal solution, and where there is a considerable degree of "frustration" in the sense of Toulouse (1977), meaning that a seeming improvement in one feature of a solution often at the same time worsens some other feature of the solution, so that the problem cannot be decomposed into small subproblems which can each be optimized separately. (Com-

[1] Project APRIL has been sponsored since December 1986 under contract MOD2062/0128(RSRE); we are grateful to the Ministry of Defence for permission to publish this paper.

pare how, in parsing, deciding to attach a constituent A as a daughter of a constituent B may be a relatively attractive way of "using up" A, at the cost of making B a less plausible constituent than it would be without A.)

One simple optimization technique, iterative improvement, begins by selecting a solution arbitrarily and then makes a long series of small modifications, drawn from a class of modifications which is defined in such a way that any point in the solution-space can be reached from any other point by a chain of modifications each belonging to the class. At each step the value of the solution obtained by making some such change is compared with the value of the current solution. The change is accepted and the new solution becomes current if it is an improvement; otherwise the change is rejected, the existing solution retained, and an alternative modification is tried. The process terminates on reaching a solution superior to each of its neighbours, i.e. when none of the available modifications is an improvement.

As it stands, such a technique is useless for parsing. It is too easy for the system to become trapped at a point which is better than its immediate neighbours but which is by no means the best solution overall, i.e. at a local but not a global optimum.

Simulated annealing is a variant which deals with this difficulty by using a more sophisticated rule for deciding whether to accept or reject a modification. In the variant we use, a favourable step is always accepted; but an unfavourable step is rejected only if the loss of merit resulting from the step exceeds a certain threshold. This acceptance threshold is randomly generated at each step from a biassed distribution; it may at any time be very high or very low, but its *mean* value is made to decrease in accordance with some defined schedule as the iteration proceeds, so that initially almost all moves are accepted, good or bad, but moves which are severely detrimental soon start to be rejected, and in the later stages almost all detrimental moves are avoided. This scheme was originally devised as a simulation of the thermodynamic processes involved in the slow cooling of certain materials, hence the name "simulated annealing". Accepting modifications which worsen the current tree is at first sight a surprising idea, but such moves prevent the system getting stuck and instead open up new possibilities; at the same time, there is an inexorable overall trend towards improvement. As a result, the system tends to seek out high-valued areas of the solution space initially in terms of gross features, and later in terms of progressively finer detail. Again, the process terminates at a local optimum, but not before exploring the possibilities so thoroughly that this is in general the global optimum. With certain simplifying assumptions, it has been shown mathematically that the global optimum is always found (Lundy & Mees, 1986): in practice, the procedure appears to work well under rather less stringent conditions than those demanded by mathematical treatments that have so far appeared, and our application does in fact take several liberties with the "pure" algorithm as set out in the literature.

ANNEALING PARSE-TREES

To apply simulated annealing to a given problem, it is necessary to define (a) a space of possible solutions, (b) a class of solution modifications which provides a route from any point in the space to any other, and (c) an annealing schedule (i.e. an initial value for the mean acceptance threshold, a specification of the rate at which this mean is reduced, and a criterion for terminating the process).

Solution space

For us, the solution space for an input sentence n words long is the set of all rooted labelled trees having n leaves, in which the leaf nodes are labelled with the word-class codes corresponding to the words of the sentence (for test inputs drawn from LOB, these are the codes given in the Tagged version of the LOB corpus) and the non-terminal nodes have labels drawn from the set of grammatical-category labels specified in the parsing scheme. The root node of a tree is assigned a fixed label, but any other non-terminal node may bear any category label.

Move set

A set of possible parse-tree modifications allowing any tree to be reached from any other can be defined as follows. To generate a modification, pick a non-terminal node of the current tree at random. Choose at random one of the move-types Merge or Hive. If Merge is chosen, delete the chosen node by replacing it, in its mother's daughter-sequence, with its own daughter-sequence. If the move-type is Hive, choose a random continuous subsequence of the

node's daughter-sequence, and replace that subsequence by a new node having the subsequence as its own daughter-sequence; assign a label drawn from the non-terminal alphabet to the new node. It is easy to see that the class of Merge and Hive moves allows at least one route from any tree to any other tree over the same leaf-sequence: repeated Merging will ultimately turn any tree into the "flat tree" in which every leaf is directly dominated by the root, and since Merge and Hive moves mirror one another, if it is possible to get from any tree to the flat tree it is equally possible to get from the flat tree to any tree. (In reality, there will be numerous alternative routes between a given pair of trees, most of which will not pass through the flat tree.)

New labels for nodes created by Hive moves are chosen randomly, with a bias determined by the labels of the daughter-sequence. This bias attempts to increase the frequency with which correct labels are chosen, without limiting the choice to the label which is best for the daughter-sequence considered in isolation, which may not of course be the best in context.

An early version of APRIL limited itself to just the Merge and Hive moves. However, a good move-set for annealing should not only permit any solution to be reached from any other solution, but should also be such that paths exist between good trees which do not involve passing through much inferior intermediate stages. (See for example the remarks on depth in Lundy & Mees (1986).) To strengthen this tendency in our system it has proved desirable to add a third class of Reattach moves to the move-set. To generate a Reattach move, choose randomly any non-root node in the current tree, eliminate the arc linking the chosen node to its mother, and insert an arc linking it to a node randomly chosen from the set of nodes topologically capable of being its mother. Currently, we are exploring the cost-effectiveness of adding a fourth move-type, which relabels a randomly-chosen node without changing the tree shape; a task for the future is to investigate how best to determine the proportions in which different move-types are generated.

Schedule

The annealing schedule is ultimately a compromise between processing time and quality of results: although the process can be speeded up at will, inevitably speeding up too much will make the system more likely to converge on a false solution when presented with a difficult sentence. Optimizing the schedule is a topic to which much attention has been paid in the literature of simulated annealing, but it seems fair to say that the discussion remains inconclusive. Since it does not in general bear on the specifically linguistic aspects of our project, we have deferred detailed consideration of this issue. We intend however to look at the variation in rate with respect to type of input, exploiting the division of the TreeBank (like its parent LOB Corpus) into genres: we would expect that the simple if sometimes messy sentences of dialogue in fiction, for instance, can be dealt with more quickly than the precise but tortuous grammar of legal prose.

At present, then, we reduce the acceptance threshold at a constant rate which errs on the slow side; we expect that important advances in efficiency will result from improvements in the schedule, but such improvements may be overtaken by other developments to be described in later sections. The rate of decrease of the acceptance threshold is varied inversely with the length of the sentence, with the consequence that the run time varies roughly linearly with sentence length.

EVALUATING PARSE-TREES

The function of the evaluation system is to assign a value to any labelled tree whatsoever, in such a way that the correct parse-tree for any given sentence is the highest-valued tree which can be drawn over the sentence, and the values of other trees over the same sentence reflect their relative merit (though comparisons of values between trees drawn over different sentences are not required to be meaningful).

An advantage of the annealing technique is that in principle it makes no demands on the form of evaluation: in particular, we are not constrained by the nature of the parsing algorithm to assume that the grammar of English is context-free or has any other special property. Nevertheless, we have found it convenient in our early work to start with a context-free assumption and work forward from that.

With this assumption, a tree can be treated as a set of productions $m \rightarrow d_1 d_2 \ldots d_n$ corresponding to the various nodes in the tree, where m is a non-terminal label and each d_i is

either a non-terminal label or a wordtag, and we can assign to any such production a probability representing the frequency of such productions, as a proportion of all productions having m as mother-label; the value assigned to the entire tree will be the product of the probabilities of its productions.

The statistic required for any production, then, is an estimate of its probability of occurrence, and this may be derived from its frequency in the manually-parsed TreeBank. (To avoid circularity, sentences in the TreeBank which are to be used to test the performance of the parser are excluded from the frequency counts.) Clearly, with a database of this size, the figures obtained as production probabilities will be distorted by sampling effects. In general, even quite large sampling errors have little influence on results, since the frequency contrasts between alternative tree-structures tend to be of a higher order of magnitude, but difficulties arise with very low frequency productions: in particular, as an important special case, many quite normal productions will fail to occur at all in the TreeBank, and are thus not distinguished in our raw data from virtually-impossible productions. But it seems reasonable to infer probability estimates for unobserved productions from those of similar, observed productions, and more generally to smooth the raw frequency observations using statistical techniques (see for instance Good (1953)). (One consequence of such smoothing is that no production is ever assigned a probability of zero.) A natural response by linguists would be to say that a relationship of "similarity" between productions needs to be defined in terms of subtle, complex theoretical issues. However, so far we have been impressed by results obtainable in practice using very crude similarity relationships.

Our current evaluation method is only slightly more elaborate than the technique described in Sampson (1986), whereby the probability of a production was derived exclusively from the observed frequencies of the various pairwise transitions between daughter-labels within the production (that is, for any production $m \rightarrow d_0 d_1 \ldots d_n d_{n+1}$, where d_0 and d_{n+1} are boundary symbols, the estimated probability was the product of the observed frequencies of the various transitions $m \rightarrow \ldots d_i d_{i+1} \ldots$ $(0 \leq i \leq n)$ with zeroes replaced by small positive values). This approach was suggested by the success of the CLAWS system for grammatically disambiguating words in context (Garside et al. 1987, chap. 3), which uses an essentially Markovian model, and by the success of Markovian techniques in automatic speech understanding research from the Harpy project onwards (e.g. Lea 1980, Cravero et al. 1984).

Subsequent versions of APRIL have begun to incorporate an evaluation measure which makes limited use of non-Markovian relationships. Each label in the non-terminal alphabet is associated with a transition network, each arc of which is assigned a probability as well as a (non-terminal or terminal) label: the probability estimate for a node labelled m is the product of the probabilities of the consecutive arcs in the transition network for m which carry the labels of the node's daughter-sequence. Unlike the FSAs commonly used in computational linguistics, ours are required to accept any label-sequence: a "crazy" sequence will be assigned a low but non-zero value. Indeed our networks make no attempt to reflect subtle nuances of grammaticality; they diverge from Markovian networks only to represent a limited number of fundamental issues that are lost in a pure Markovian system.

APRIL IN ACTION

It is rather difficult to convey non-mathematically a feel for the way in which the system converges from an arbitrary tree to the correct tree by a sequence of random moves. In the earliest stages, labelled nodes are being created, moved and destroyed at a rapid rate in all regions of the tree, but after a while it starts to become apparent that certain local features are tending to persist. These tend to be the most strongly marked features grammatically, such as constituents comprising a single pronoun or an auxiliary verb. While such a feature persists, surrounding developments are constrained by it: other new nodes can be created if they are compatible, but new nodes which would conflict cannot appear. Thus the grammatical words form a skeleton on which the phrases and clauses can start to hang, and we find there is a perceptible gradually increasing tendency for the tree to consist of nodes and substructures which fit together well into a coherent whole. Speaking anthropomorphically, the system tends to make the simplest and most clear-cut decisions first, and the more subtle decisions later. But the strength of the system

lies in the fact that no such decision is final: each is constantly being reappraised in the light of developments in its surroundings.

CURRENT PERFORMANCE

In order to assess APRIL's performance we need an objective way to compare output with target parses, i.e. a measure of similarity between pairs of distinct trees over the same sequence of leaf nodes. We know of no standard measure for this, but we have evolved one that seems natural and fair. For each word of input we compare the chains of node-labels between leaf and root in the two trees, and compute the number of labels which match each other and occur in the same order in the two chains as a proportion of all labels in both chains; then we average over the words. (We omit discussion of a refinement included in order to ensure that only fully-identical tree-pairs receive 100% scores.) With respect to our parsing technique, this performance measure is conservative, since averaging over words means that high-level nodes, dominating many words, contribute more than low-level nodes to overall scores, but APRIL tends to discover structure in a broadly bottom-up fashion.

At the time of writing, our latest results were those of a test run carried out in early February 1988, 14 months into a 36-month project, over 50 LOB sentences drawn from technical prose and fiction, with mean, minimum, and maximum lengths of 22.4, 3, and 140 words respectively. (Note that our parsing scheme, and therefore our word-counts, treat punctuation marks as separate "words".) The alphabet of non-terminal labels from which APRIL chooses when labelling new nodes included virtually all the distinctions required by our scheme in an adequately parsed output; and it included several of the more significant phrase-subcategory distinctions whose role in the scheme is to guide the parser towards the correct output rather than to appear in the output (Garside et al. 1987, p. 89). Altogether the non-terminal alphabet included 113 distinct labels.

For a 22-word sentence, the number of distinct trees with labels drawn from a 113-member alphabet (and obeying the restrictions our scheme places on the occurrence of nodes with only single daughters) is about 5×10^{103}. To put this in perspective, finding a particular labelled tree in a search space of this size is like finding a single atom of gold in a solid cube of gold a thousand million light-years on a side.

Mean score of the 50 output analyses was 75.3%. This is not yet good enough for incorporation into practical language-processing application software, but bearing in mind the preliminary nature of the current version of the system we are heartened by how good the scores already are. Furthermore, above about 15 words there appears to be no correlation between sentence-length and output score, offering a measure of support for our decision to use an annealing schedule which increases processing time roughly linearly with input length. Kirkpatrick et al. (1983) suggest that linear processing is adequate for simulated annealing in other domains, but orthodox deterministic approaches to computational linguistics do not permit linear parsing except for highly artificial well-behaved languages.

The parse-trees produced in this test run typically show a substantially correct overall structure, with isolated local areas of difficulty where some deviant analysis has been preferred, commonly a constituent wrongly labelled or a constituent attached to the surrounding tree at the wrong level. An encouraging point is that a number of these errors relate to debatable grammatical issues and might not be seen as errors at all. In the years when our target parsing scheme was being evolved, we worried about the idiomatic construction *to try and [do something]*: should *try and Verb* be grouped as a constituent equivalent to a single verb? We finally decided not: we chose to analyse such sequences as co-ordinated clauses. But, where the test sentences include the sequence *I want to try and find properties that...*, APRIL has parsed: *I want [Ti to [VB& try and find] properties that...]*—the analysis which we came close to choosing as correct.

A sentence which raises less trivial issues is illustrated (this is from text E23 in the LOB Corpus). We show the manual parse in the TreeBank (Fig.1), and APRIL's current output (Fig. 2), which contains two errors. First, the final phrase *of the human mind* should be attached as a postmodifier of *mysteries*. At this stage no distinction was made in word-tagging between *of* and other prepositions: there is however a strong tendency (though no absolute rule, of course) for an *of* phrase following a noun to be a postmodifier of the noun, and it is correspondingly rare for such a phrase to be an

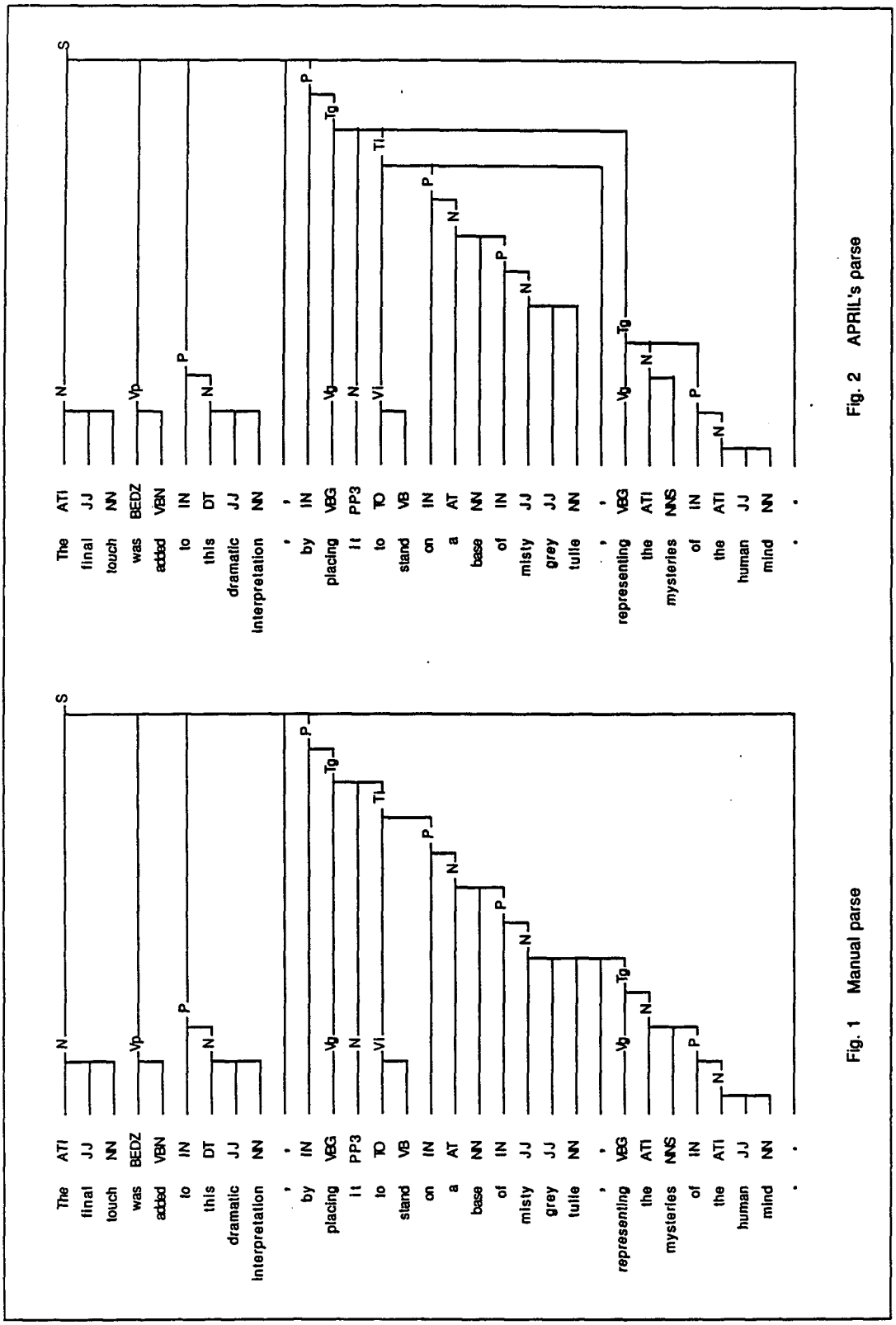

Fig. 1 Manual parse

Fig. 2 APRIL's parse

immediate constituent of a clause. Distinguishing *of* from other prepositions will enable the evaluation system to incorporate a representation of this piece of statistical evidence in its transition probabilities, whereupon this error should be avoided.

Secondly, APRIL has rejected the interpretation of the clause beginning *representing...* as a postmodifier of *tulle*, and has chosen to make this clause appositional to the clause beginning *placing...* (our scheme represents apposition in a manner akin to subordination). This error can be avoided if we note the strong tendency in English (again, not an absolute rule) that postmodifiers of any kind are most often attached to the nearest element that they can logically postmodify, that is, that the chain-structure typified in Fig. 1 is preferred to the embedding-structure in Fig. 2. A preliminary statistical analysis of the TreeBank appears to support the conjecture—developed from the hypothesis formulated by Yngve (1960)—that "the greater the depth of a non-terminal constituent, the greater the probability that either (a) this constituent is the last daughter of its mother, or (b) the next daughter of its mother is a punctuation mark." (We adapt Yngve's notion of depth to non-binary trees.) With this formulation it is relatively easy to incorporate into our evaluation system the necessary adjustments to our transition probabilities, so that trees of the more common type will tend to be preferred; but note that nothing prevents an overriding local consideration from leading the parser to prefer, in any given case, an analysis that departs from this general principle. When this is done, the initial context-free assumption will have been abandoned, to the extent that depths of constituents are taken into account as well as their labels, but no change is needed in the parsing algorithm.

The erroneous parsings in this example flout no rules of syntax that we can formulate and seem to involve no impossible productions, so they could be regarded as valid alternatives in a syntactically ambiguous sentence: a generative grammar could be expected to generate this sentence in several different ways, of which APRIL's would be one. However, as our methods improve we find that more and more sentences which are in principle ambiguous have the same reading selected by purely statistical-syntactic considerations as is preferred by human readers, who also have access to semantic and pragmatic considerations.

FUTURE DEVELOPMENTS

Apart from improving the evaluation system as already discussed, we plan in the near future to adapt APRIL so that it accepts raw text rather than sequences of word-class codes as input, choosing tags for grammatically ambiguous words as part of the same optimization process by which higher structure is discovered. The availability of the (probabilistic but deterministic) CLAWS word-tagging system meant that this was not seen as an initial priority. Raw text input involves a number of problems relating to orthographic matters such as capitalization and hyphenated words, but these problems have essentially been solved by our Lancaster colleagues (Garside et al., chap. 8). We also intend soon to move from the current static system whose inputs are isolated sentences to a dynamic system within which annealing will take place in a window that scans across continuous text, with the system discovering sentence-boundaries for itself along with lower-level structure. (If our system is in due course adapted to parse spoken rather than written input, it is clear that all constituent boundaries including those of sentences would need to be discovered rather than given, and a corollary appears to be that the processing time needed for any length of input must increase only linearly with input length.) As adumbrated in Sampson (1986), we expect to make the dynamic annealing parser more efficient by exploiting the insight of Marcus (1980) that backtracking is rarely needed in natural language parsing: a gradient of processing intensity will be imposed on the annealing window, with most processing occurring in the "newest" parts of the current tree where valuable moves are most likely to be found.

However, simulated annealing is necessarily costly in terms of amount of processing needed. (The schedule used for the run discussed above involved on the order of 30,000 steps generated per input word.) Particularly with a view to applications such as real-time speech analysis, it would be desirable to find a way of exploiting parallel processing in order to minimize the time needed for parse-tree optimization.

Parallelizing our approach to parsing is not a straightforward matter: one cannot, for instance, simply associate a process with each node of a tree, since there is no natural identity relation-

ship between nodes in different trees within the solution space for an input. However, we have evolved an algorithm for concurrent tree annealing which we believe should be efficient, and a research proposal currently under consideration will implement this algorithm, using a transputer array which is about to be installed by a consortium of Leeds departments. In view of the widespread occurrence of hierarchical structures in cognitive science, we hope that a successful solution to the problem of parallel tree-optimization should be of interest to workers in other areas, such as image processing, as well as to linguists.

Lastly, a reasonable criticism of our work so far is that our target parses are those defined by a purely "surfacy" parsing scheme. For some speech-processing applications surface parsing is adequate, but for many purposes deeper language analyses are needed. We see no issue of principle hindering the extension of our methods to deep parsing, but at present there is a serious practical hindrance: our techniques can only be applied after a target parsing scheme has been specified in sufficient detail to prescribe unambiguous analyses for all phenomena occurring in authentic English, and then applied manually to a large enough quantity of text to yield usable statistics. A second currently-pending research proposal plans to convert the Gothenburg Corpus (Ellegård 1978), which consists of relatively deep manual parsings of 128,000 words of the Brown Corpus of American English, into a database usable for this purpose.

REFERENCES

Cravero, M., et al. 1984. "Syntax driven recognition of connected words by Markov models". *Proceedings of the 1984 IEEE International Conference on Acoustics, Speech and Signal Processing.*

Ellegård, A. 1978. *The Syntactic Structure of English Texts.* Gothenburg Studies in English, 43.

Garside, R. G., et al., eds. 1987. *The Computational Analysis of English.* Longman.

Good, I. J. 1953. "The population frequencies of species and the estimation of population parameters". *Biometrika* 40.237-64.

Kirkpatrick, S. E., et al. 1983. "Optimization by Simulated Annealing". *Science* 220.671-80.

van Laarhoven, P. J. M., & E. H. L. Aarts. 1987. *Simulated Annealing: Theory and Applications.* D. Reidel.

Lea, R. G., ed. 1980. *Trends in Speech Recognition.* Prentice-Hall.

Lundy, M. and A. Mees. 1986. "Convergence of an annealing algorithm". *Mathematical Programming* 34.111-24.

Marcus, M. P. 1980. *A Theory of Syntactic Recognition for Natural Language.* MIT Press.

Sampson, G. R. 1986. "A stochastic approach to parsing". *Proceedings of the 11th International Conference on Computational Linguistics (COLING '86)*, pp. 151-5. [GRS wishes to take this opportunity to apologize for the inadvertent near-coincidence of title between this paper and an important 1984 paper by T. Fujisaki.]

Sampson, G. R. 1987. "Evidence against the 'grammatical'/'ungrammatical' distinction". In W. Meijs, ed., *Corpus Linguistics and Beyond.* Rodopi.

Toulouse, G. 1977. "Theory of the frustration effect in spin glasses. I." *Communications on Physics*, 2.115-119.

Yngve, V. 1960. "A model and an hypothesis for language structure". *Proceedings of the American Philosophical Society*, 104.444-66.