PATTERN RECOGNITION APPLIED TO THE ACQUISITION OF A GRAMMATICAL CLASSIFICATION SYSTEM FROM UNRESTRICTED ENGLISH TEXT

Eric Steven Atwell and Nicos Frixou Drakos
Artificial Intelligence Group
Department of Computer Studies
Leeds University, Leeds LS2 9JT, U.K.
(EARN/BITNET: eric%leeds.ai@ac.uk)

ABSTRACT

Within computational linguistics, the use of statistical pattern matching is generally restricted to speech processing. We have attempted to apply statistical techniques to discover a grammatical classification system from a Corpus of 'raw' English text. A discovery procedure is simpler for a simpler language model; we assume a first-order Markov model, which (surprisingly) is shown elsewhere to be sufficient for practical applications. The extraction of the parameters of a standard Markov model is theoretically straightforward; however, the huge size of the standard model for a Natural Language renders it incomputable in reasonable time. We have explored various constrained models to reduce computation, which have yielded results of varying success.

Pattern recognition and NLP

In the area of language-related computational research, there is a perceived dichotomy between, on the one hand, "Natural Language" research dealing principally with syntactic and other analysis of typed text, and on the other hand, "Speech Processing" research dealing with synthesis, recognition, and understanding of speech signals. This distinction is not based merely on a difference of input and/or output media, but seems also to correlate to noticeable differences in assumptions and techniques used in research. One example is in the use of statistical pattern recognition techniques: these are used in a wide variety of computer-based research areas, and many speech researchers take it for granted that such methods are part of their stock in trade. In contrast, statistical pattern recognition is hardly ever even considered as a technique to be used in "Natural Language" text analysis. One reason for this is that speech researchers deal with "real", "unrestricted" data (speech samples), whereas much NLP research deals with highly restricted language data, such as examples intuited by theoreticians, or simplified English as allowed by a dialogue system, such as a Natural Language Database Query system.

Chomsky (57) did much to discredit the use of representative text samples or Corpora in syntactic research; he dismissed both statistics and semantics as being of no use to syntacticians: "Despite the undeniable interest and importance of semantic and statistical studies of language, they appear to have no direct relevance to the problem of determining or characterizing the set of grammatical utterances" (Chomsky 57 p.17). Subsequent research in Computational Linguistics has shown that Semantics is far more relevant and important than Chomsky gave credit for. Phenomenal advances in computer power and capabilities mean that we can now try statistical pattern recognition techniques which would have been incomputable in Chomsky's early days. Therefore, we felt that the case for Corpus-based statistical Pattern Recognition techniques should be reopened. Specifically, we have investigated the possibility of using Pattern Recognition techniques for the acquisition of a grammatical classification system from Unrestricted English text.

Corpus Linguistics

A Corpus of English text samples can constitute a definitive source of data in the description of linguistic constructs or structures. Computational linguists may use their intuitions about the English language to devise a grammar of English (or of some part of the English language), and then cite example sentences from the Corpus as evidence for their grammar (or counter-evidence against someone else's grammar). Going one stage further, computational linguists may use data from a Corpus as a source of inspiration at the earlier stage of devising the rules of the grammar, relying as little as possible on intuitions about English grammatical structures (see, for example, (Leech, Garside & Atwell 83a)). With appropriate software tools to extract relevant sentences from the computerised Corpus, the process of providing evidence for (or against) a particular grammar might in theory be largely mechanised. Another way to use data from a Corpus for inspiration is to manually draw parse-trees on top of example sentences taken from the Corpus, without explicitly formulating a

corresponding Context-Free or other rewrite-rule grammar. These trees could then be used as a set of examples for a grammar-rule extraction program, since every subtree of mother and immediate daughters corresponds to a phrase-structure rewrite rule; such an experiment is described by Atwell (forthcoming b).

However, the linguists must still use their expertise in theoretical linguistics to devise the rules for the grammar and the grammatical categories used in these rules. To completely automate the process of devising a grammar for English (or some other language), the computer system would have to "know" about theories of grammar, how to choose an appropriate model (e.g. context-free rules, Generalized Phrase Structure Grammar, transition network, or Markov process), and how to go about devising a set of rules in the chosen formalism which actually produces the set of sentences in the Corpus (and doesn't produce (too many) other sentences).

Chomsky (1957), in discussing the goals of linguistic theory, considered the possibility of a *discovery procedure for grammars*, that is, a mechanical method for constructing a grammar, given a corpus of utterances. His conclusion was: "I think it is very questionable that this goal is attainable in any interesting way". Since then, linguists have proposed various different grammatical formalisms or models for the description of natural languages, and there has been no general consensus amongst expert linguists as to the 'best' model. If even human experts can't agree on this issue, Chomsky was probably right in thinking it unreasonable to expect a machine, even an 'intelligent' expert system, to be able to choose which theory or model to start from.

Constrained discovery procedures

However, it may still be possible to devise a discovery procedure if we constrain the computer system to a specific grammatical model. The problem is simplified further if we constrain the input to the discovery procedure, to carefully chosen example sentences (and possibly counter-example non-sentences). This is the approach used, for example, by Berwick (85); his system extracted grammar rules in a formalism based on that of Marcus's PARSIFAL (Marcus 80) from fairly simple example sentences, and managed to acquire "approximately 70% of the parsing rules originally hand-written for [Marcus's] parser". Unfortunately, it is not at all clear that such a system could be generalised to deal with Unrestricted English text, including deviant, idiomatic and even ill-formed sentences found in a Corpus of 'real' language data. This is the kind of problem best suited to statistical pattern matching methods.

The plausibility of a truly general discovery procedure, capable of working with unrestricted input, increases if we can use a very simple model to describe the language in question. Chomsky believed that English could only be described by a phrase structure grammar augmented with transformations, and clearly a discovery procedure for devising Transformational Generative grammars from a Corpus would have to be extremely complex and 'clever'. More recently, (Gazdar et al 85) and others have argued that a less powerful mechanism such as a variant of phrase structure grammar is sufficient to describe English syntax. A discovery procedure for phrase structure grammars would be simpler than one for TG grammars because phrase structure grammars are simpler (more constrained) than TG grammars.

CLAWS

For the more limited task of assigning part-of-speech labels to words, (Leech, Garside & Atwell 83b), (Atwell 83) and (Atwell, Leech & Garside 84) showed that an even simpler model, a first-order Markov model, will suffice. This model was used by CLAWS, the Constituent-Likelihood Automatic Word-tagging System, to assign grammatical wordclass (part-of-speech) markers to words in the LOB Corpus. The LOB Corpus is a collection of 500 British English text samples, each of just over 2000 words, totalling over a million words in all; it is available in several formats (with or without word-tags associated with each word) from the Norwegian Computing Centre for the Humanities, Bergen University (see (Johansson et al 78), (Johansson et al 86)). The Markovian CLAWS was able to assign the correct tag to c96% of words in the LOB Corpus, leaving only a small residual of problematic constructs to be analysed manually (see (Atwell 81, 82)). Although CLAWS does not yield a full grammatical parse of input sentences, this level of analysis is still useful for some applications; for example, Atwell (83, 86c) showed that the first-order Markov model could be used in detecting grammatical errors in ill-formed input English text. The main components of the first order Markov model or grammar used by CLAWS were:

i) a set of 133 grammatical class labels or TAGS, e.g. NN (singular common noun) or JJR (comparative adjective)

ii) a 133*133 tag-pair matrix, giving the frequency of cooccurrence of every possible pair of tags (the rowsums or columnsums giving frequencies of individual tags)

iii) a wordlist associating each word with a list of possible tags (with some indication of relative frequency of each tag where a word has more than one), supplemented by a suffixlist, prefixlist, and other default routines to deal with input words not found in the wordlist

iv) a set of formulae to use in calculating likelihood-in-context, to disambiguate word-tags in tagging new text.

The last item, the formulae underlying the CLAWS system (see (Atwell 83)), constitutes the Markovian mathematical model, and it is too much to ask of any expert system to devise or extract this from data. At least in theory, the first three components could be automatically extracted from sample text WHICH HAS ALREADY BEEN TAGGED, providing there is enough of it (in particular, there should be many examples of each word in the wordlist, to ensure relative tag likelihoods are accurate). However, this is effectively "learning by example": the tagged texts constitute examples of correct analyses, and the program extracting word-tag and tag-pair frequencies could be said to be "learning" the parameters of a Markov model compatible with the example data. Such a learning system is not a truly generalised discovery procedure. Ideally, we would like to be able to extract the parameters of a compatible Markov model from RAW, untagged text.

RUNNEWTAGSET

Statistical pattern recognition techniques have been used in many fields of scientific computing for data classification and pattern detection. In a typical application, there will be a large number of data records, each of which will have a fairly complex internal structure; the task is to somehow group together sets of data records with 'similar' internal structures, and/or to note types of internal structures which occur frequently in data records. For example, a speech pattern recognition system is 'trained' with repeated examples of each word in its vocabulary to recognise the stereotypical structure of the given speech signal, and then when given a 'new' sound it must classify it in terms of the 'known' patterns. In attempting to devise a grammatical classification system for words in text, a record consists of the word itself, and its grammatical context. A reasonably large sample of text such as the million-word LOB Corpus corresponds to a huge amount of data if the 'grammatical context' considered with each word is very large. The simplest model is to assume that only the single word immediately to the left and/or right of each TARGET word is important in the context; and even this oversimplification of context entails vast amounts of processing.

If we assume that each word can belong to one and only one word-class, then whenever two words tend to occur in the same set of immediate (lexical) contexts, they will probably belong to the same word-class. This idea was tested using a suite of programs called RUNNEWTAGSET to group words in a c200,000-word subsection of the LOB Corpus into word-classes. The system only attempted to classify wordforms which occurred a hundred times or more, the minimum sample size for lexical collocation analysis suggested by Sinclair et al (70). All possible pairings of one wordform with another wordform (w1,w2) were compared: if the immediate lexical contexts in which w1 occurred were significantly similar to the immediate contexts of w2, the two were deemed to belong to the same word-class, and the two context-sets were merged. A threshold was used to test "significant similarity"; initially, only words which occurred very frequently in the same contexts were classified together, but then the threshold was lowered in stages, allowing less and less similar context-sets to be merged at each stage.

Unfortunately, the 200,000-word sample turned out to be far too small for conclusive results: even in a sample of this size, only 175 words occur 100 times or more. However, this program run took several weeks, so it was impractical to try a much larger text sample. There were some promising trends; for example, at the initial threshold level, <will should could must may might>, <in for on by at during>, <is was>, <had has>, <it he there>, <they we>, <but if when while>, <make take>, <end use point question>, and <sense number> were grouped into word-classes on the basis of their immediate lexical contexts, and in subsequent reductions of the threshold these classes were enlarged and new classes were added. However, even if the mammoth computing requirements could be met, this approach to automatic generation of a tagset or word-classification system is unlikely to be wholely successful because it tries to assign every word to one and only one word-class, whereas intuitively many words can have more than one possible tag. For example, this technique will tend to form three separate classes for nouns, verbs, and words which can function in both ways. For further details of the RUNNEWTAGSET experiment, see (Atwell 86a, 86b).

Baker's algorithm

Baker (75, 79) gives a technique which might in theory solve this problem. Baker showed that if we assume that a language is generated by a Markov process, then it is theoretically possible, given a sufficiently large sample of data, to automatically calculate the parameters of a Markov model compatible with the data. Baker's method was proposed as a technique for automatic training of the parameters of a model of an acoustic processor, but it could in theory be applied to the syntactic description of text. In Baker's technique, the principle parameters of the Markov model were two matrices, a(i,j) and b(i,j,k). For the word-tagging application, i and j correspond to tags, while k corresponds to a word; a(i,j) is the probability of tag i being followed by tag j, and b(i,j,k) is the probability of a word with tag i being followed by the word k with tag j. a(i,j) is the direct equivalent of the tag-pair matrix in the CLAWS model above. b(i,j,k) is analogous to the wordlist, except

that the information associated with each word is more detailed: instead of just a relative frequency for each tag that can appear with the word, there is a frequency for every possible pair of <previous tag - this tag>. Baker's model is mathematically equivalent to the one used in CLAWS; and it has the advantage that if the true matrices a(i,j) and b(i,j,k) are not known, then they can be calculated by analysing raw text. We start with initial estimates for each value, and then use an iterative procedure to repeatedly improve on these estimates of a(i,j) and b(i,j,k).

Unfortunately, although this grammar discovery procedure might work in theory, the amount of computation in practice turns out to be vast. We must iteratively estimate a likelihood for every <tag-tag> pair for a(i,j), and for every possible <tag-tag-word> triple for b(i,j,k). Work on tagging the LOB Corpus has shown that a tag-set of the order of 133 tags is reasonable for English (if we include separate tags for different inflections, since different inflexions can appear in distinguishable syntactic contexts). Furthermore, the LOB Corpus has roughly 50,000 word-forms in it (counting, for example, "man", "men", "mans", "manned", "manning", etc as separate wordforms). Working from the 'raw' LOB Corpus, we would have to estimate c18,000 values for a(i,j), and 900,000,000 values for b(i,j,k). As the process of estimating each a(i,j) and b(i,j,k) value is in itself computationally expensive, it is impractical to use Baker's formulae unmodified to automatically extract word-classes from the LOB Corpus.

Grouping by suffix

To cut down the number of variables, we tried the simplifying assumption that the last five letters of a word determine which grammatical class(es) it belongs to. In other words, we assumed words ending in the same suffix shared the same wordclass; a not unreasonable assumption, at least for English. CLAWS was able to assign grammatical classes to almost any given word using a wordlist of only c7000 words supplemented by a suffixlist, so the assumption seemed intuitively reasonable for most words. To further reduce the computation, we used tag-pair probabilities from the tagged LOB Corpus to initialise a(i,j): by using 'sensible' starting values rather than completely arbitrary ones, convergence should have been much more rapid. Unfortunately, there were still far too many interdependent variables for computation in a reasonable time: we estimated that even with a single LOB text instead of the complete Corpus, the first iteration alone in Baker's scheme would take c66 hours!

Alternative constraints

An alternative approach was to abandon Baker's algorithm and introduce other constraints into the First Order Markov model. Another intuitively acceptable constraint was to allow each word to belong to only a small number of possible word classes (Baker's algorithm allowed words to belong to many different classes, up to the total number of classes in the system). This allowed us to try entirely different algorithms suggested by (Wolff 76) and (Wolff 78), based on the assumption that the class(es) a word belongs to are determined by the immediate contexts that word appears in in the example texts. Unfortunately, these still involved prohibitive computing times. Wolff's second model was the more successful of the two, coming up with putative classes such as <and at for in of to>, <had was>, <a an it one the>, <at by in not on to with> and <but he i it one there>; yet our implementation took 5 hours CPU time to extract these classes from an 11,000 word sample.

Heuristic constraints

We are beginning to investigate alternative strategies; for instance, Artificial Intelligence techniques such as heuristics to reduce the 'search space' would seem appropriate. However, any heuristics must not be tied too closely to our intuitive knowledge of the English language, or else the resultant grammar discovery procedure will effectively have some of the grammar "built in" to it. For example, one might try constraining the number of tags allowed for each specific word (e.g "the", "of", "sexy" can have only one tag; "to", "her", "book" have two possible tags; "cold", "base", "about" have three tags; "back", "bid", "according" have four tags; "bound", "beat", "round" have five tags; and so on); but this is clearly against the spirit of a truly automatic discovery procedure in the Chomskyan sense. A more 'acceptable' constraint would be a general limit of, say, up to five tags per word. A discovery procedure would start by assuming that the context-set of every word could be partitioned into five subsets, and then it would attempt a Prolog-style 'unification' of pairs of similar context-subsets, using belief revision techniques from Artificial Intelligence (see, for example, (Drakos 86)).

Applications

Overall, we concede that the case for statistical pattern-matching for syntactic classification is not proven. However, there have been some promising results, which deserve further investigation, since there would be useful applications for any successful pattern recognition technique for the acquisition of a grammatical classification system from Unrestricted English text.

Note that variables in formulae mentioned above such as i and j are not tag names (NN, VB, etc), but just integers denoting positions in a tag-pair matrix. In a Markov model,

a tag is defined entirely by its cooccurrence likelihoods with other tags, and with words: labels like NN, VB will not be generated by a pattern recognition technique. However, if we assumed initially that there are 133 tags, e.g. if we initialised a(i,j) to a 133*133 matrix, then hopefully there should be some correlation between distributions of tags in the LOB tagset and the automatically generated tagset. If there is poor correlation for some tags (e.g. if the automatically-derived tagset includes some tags whose collocational distributions are unlike those of any of the tags used in the LOB Corpus), then this constitutes empirical, objective evidence that the LOB tagset could be improved upon.

In general, any alternative wordclass system could be empirically assessed in an analogous way. The Longman Dictionary of Contemporary English (LDOCE; Procter 78) and the Oxford Advanced Learner's Dictionary of Current English (OALD; Hornby 74) give detailed grammatical codes with each entry, but the two classification systems are quite different; if samples of text tagged according to the LDOCE and OALD tagsets were available, a pattern recognition technique might give us an empirical, objective way to compare and assess the classification systems, and suggest particular areas for improvement in forthcoming revised editions of LDOCE and OALD. This would be particularly useful for Machine Readable versions of such dictionaries, for use in Natural Language Processing systems (see, for example, (Akkerman et al 85), (Alshawi et al 85), (Atwell forthcoming a)); these could be tailored to a given application domain (semi-)automatically.

Even though the experiments mentioned achieved only limited success in discovering a complete grammatical classification system, a more restricted (and hence more achievable) aim is to concentrate on specific word classes which are traditionally recognised as difficult to define. For example, the techniques were particularly successful at finding groups of words corresponding to invariant function word classes, such as particles; Atwell (forthcoming c) explores this further.

A bottleneck in commercial exploitation of current research ideas in NLP is the problem of tailoring systems to specialised linguistic registers, that is, application-specific variations in lexicon and grammar. This research, we hope, points the way to (semi-)automating the solution for a wide range of applications (such as described, for example, by Atwell (86d)). Particularly appropriate to the approach outlined in this paper are applications systems based on statistical models of grammar, such as (Atwell 86c). If *grammar discovery* can be made to work not just for variant registers of English, but for completely different languages as well, then it may be possible to automate (or at least greatly simplify) the transfer of systems such as that described by Atwell (86c) to a wide variety of natural languages.

Conclusion

Automatic grammar discovery procedures are a tantalising possibility, but the techniques we have tried so far are far from perfect. It is worth continuing the search because of the enormous potential benefits: a discovery procedure would provide a solution to a major bottleneck in commercial exploitation of NLP technology. We are keen to find collaborators and sponsors for further research.

REFERENCES

Akkerman, Erik, Pieter Masereeuw, and Willem Meijs 1985 *Designing a computerized lexicon for linguistic purposes* Rodopi, Amsterdam

Alshawi, Hiyan, Branimir Boguraev, and Ted Briscoe 1985, "Towards a lexicon support environment for real time parsing" in Proceedings of the Second Conference of the European Chapter of the Association for Computational Linguistics, Geneva

Atwell, Eric Steven 1981 *LOB Corpus Tagging Project: Manual Pre-edit Handbook.* Departments of Computer Studies and Linguistics, University of Lancaster

Atwell, Eric Steven 1982 *LOB Corpus Tagging Project: Manual Postedit Handbook (A mini-grammar of LOB Corpus English, examining the types of error commonly made during automatic (computational) analysis of ordinary written English.)* Departments of Computer Studies and Linguistics, University of Lancaster

Atwell, Eric Steven 1983 "Constituent-Likelihood Grammar" in *Newsletter of the International Computer Archive of Modern English (ICAME NEWS)* 7: 34-67, Norwegian Computing Centre for the Humanities, Bergen University

Atwell, Eric Steven 1986a *Extracting a Natural Language grammar from raw text* Department of Computer Studies Research Report no.208, University of Leeds

Atwell, Eric Steven 1986b, "A parsing expert system which

learns from corpus analysis" in Willem Meijs (ed) *Corpus Linguistics and Beyond: Proceedings of the Seventh International Conference on English Language Research on Computerised Corpora, Amsterdam, Netherlands* Rodopi, Amsterdam

Atwell, Eric Steven 1986c, "How to detect grammatical errors in a text without parsing it" Department of Computer Studies Research Report no.212, University of Leeds; to appear in *Proceedings of the Association for Computational Linguistics Third European Chapter Conference, Copenhagen, Denmark* (elsewhere in this book).

Atwell, Eric Steven 1986d "Beyond the micro: advanced software for research and teaching from computer science and artificial intelligence" in Leech, Geoffrey and Candlin, Christopher (eds.) *Computers in English language teaching and research: selected papers from the British Council Symposium on computers in English language education and research, Lancaster, England* 167-183, Longman

Atwell, Eric Steven (forthcoming a) "A lexical database for English learners and users: the Oxford Advanced Learner's Dictionary" to appear in *Proceedings of ICDBHSS87, the 1987 International Conference on DataBases in the Humanities and Social Sciences, Montgomery, Alabama, USA*

Atwell, Eric Steven (forthcoming b) "Transforming a Parsed Corpus into a Corpus Parser", to appear in *Proceedings of the 1987 ICAME 8th International Conference on English Language Research on Computerised Corpora, Helsinki, Finland*

Atwell, Eric Steven (forthcoming c) "An Expert System for the Automatic Discovery of Particles" to appear in *Proceedings of the 1987 International Conference on the Study of Particles, Berlin, East Germany*

Atwell, Eric Steven, Geoffrey Leech and Roger Garside 1984, "Analysis of the LOB Corpus: progress and prospects", in Jan Aarts and Willem Meijs (ed), *Corpus Linguistics; Proceedings of the ICAME Conference on the use of computer corpora in English Language Research, Nijmegen, Netherlands* Rodopi.

Baker, J K 1975 "Stochastic modeling for automatic speech understanding" in D R Reddy (ed) *Speech recognition* Academic Press

Baker, J K 1979 "Trainable grammars for speech recognition" in Klatt, D H and Wolf J J (eds.) *Speech communication papers for the 97th meeting of the acoustical society of America:* 547-550

Berwick, R 1985 *The acquisition of syntactic knowledge* MIT Press, Cambridge (MA) and London

Chomsky, Noam 1957 *Syntactic Structures* Mouton, The Hague

Drakos, Nicos Frixou 1986 *Electrical circuit analysis using algebraic manipulation and belief revision* Department of Computer Studies, Leeds University

Leech, Geoffrey, Roger Garside, and Eric Steven Atwell 1983a, "Recent developments in the use of computer corpora in English language research" in *Transactions of the Philological Society* 1983: 23-40.

Leech, Geoffrey, Garside, Roger and Atwell, Eric Steven 1983b "The Automatic Grammatical Tagging of the LOB Corpus" in *Newsletter of the International Computer Archive of Modern English (ICAME NEWS)* 7: 13-33, Norwegian Computing Centre for the Humanities, Bergen University

Gazdar, Gerald, Ewan Klein, Geoffrey Pullum, and Ivan Sag 1985 *Generalized Phrase Structure Grammar* Blackwell, Oxford

Hornby, A S, with Cowie, A P (eds.) 1974 *Oxford Advanced Learner's Dictionary of Current English (third edition)* Oxford University Press

Johansson, Stig, Geoffrey Leech and Helen Goodluck 1978 *Manual of information to accompany the Lancaster-Oslo/Bergen Corpus of British English, for use with digital computers* Department of English, Oslo University

Johansson, Stig, Eric Atwell, Roger Garside, and Geoffrey Leech 1986 *The Tagged LOB Corpus* Norwegian Computing

Centre for the Humanities, University of Bergen, Norway.

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

Procter, Paul (editor-in-chief) 1978 *Longman Dictionary of Contemporary English* Longman

Sinclair, J, Jones, S, and Daley, R 1970 *English lexical studies*, Report to OSTI on project C/LP/08; Dept of English, Birmingham University

Wolff, J G 1976 "Frequency, Conceptual Structure and Pattern Recognition" in *British Journal of Psychology* 67:377-390

Wolff, J G 1978 "The Discovery of Syntagmatic and Paradigmatic Classes" in *ALLC Bulletin* 6(1):141