Learning a Radically Lexical Grammar

Danny Solomon **Mary McGee Wood**

Department of Computer Science
University of Manchester
Manchester M13 9PL, UK
{danny,mary}@cs.man.ac.uk

Abstract

We describe a prototype system which induces a categorial grammar from a simple text corpus of children's reading books. Unlike previous attempts at grammar induction, (1) there are no rules of grammar, only a richly structured lexicon; (2) we rely both on an informing linguistic theory and on statistical methods applied to a corpus.

1 Introduction - Learning a 'Grammar'

The work we describe was originally motivated by dissatisfaction with attempts to induce a rule-based grammar from a corpus (eg Berwick 1985), and the suspicion that a word-based grammar - at the logical extreme, a categorial grammar (Wood 1993) - might be easier to 'learn'. With the recent rapid growth of 'empiricism' in NLP, the same work now manifests our dissatisfaction with the view that throwing a large computer at a large corpus to produce a large set of sets of numbers is in itself 'linguistics': we are committed to the honest description of 'real 'data, but only within an independently motivated theoretical framework.

Neither of these are widely held views at present. For the first, even Brent's (1993) 'learning of lexical syntax' takes a distinct component of grammar rules as its starting point in learning a lexicon. For the second, as an example, the tension which pervaded the EACL meeting in Utrecht in April 1993, stoked by invited talks from Church (1993) and Sag (1993), made clear the deep division - even antagonism - between statisticians and theorists on the front line. However the success so far of our implementation gives some evidence for the viability of a totally lexical approach to the automatic learning of grammar, responsive to both principles and facts.

The next section describes further some of the theoretical and practical motivations behind our work. Section

3 briefly sets out the exact categorial grammar used. In Section 4 we describe *L* - a prototype grammar induction system which illustrates the direction we believe will be productive. A discussion and a look towards some prospects complete the paper.

2 Motivation

2.1 Theoretical

It is hard to imagine an automatic (or even semi-automatic) grammar or lexicon induction procedure that could manipulate traditional grammatical entities such that human-meaningful results might obtain. Brute-force methods (ie those that exploit the massive raw computing power currently available cheaply) may well produce some useful results (eg Brown *et al* 1993). However, if any linguistic insight is to appear as a result, we believe an underpinning in linguistic theory is essential - overall success will result from the combination and integration of that at which computers and human linguists excel. We believe this is only possible if both partners work in the same basic framework - and we also believe that training linguists to read disk sectors will probably be unproductive.

Our premise is that:

1) giving grammatical entities structure is useful - for providing a framework to describe them to computers, and, e.g., for generating (automatically) a taxonomy of such entities;
2) such a structure can embody the grammar of a language - all the necessary linguistic knowledge can be incorporated into the structure of the grammatical entity (or category) of each word in a language;
3) given 1 & 2, we can demonstrate that a semi-automatic means for inferring a structural lexicon (equivalent to a grammar) from a corpus of natural language is possible.

To illustrate 1 & 2, we suggest that the lexicon can be unified with grammar, using the theoretical framework of Categorial Grammar (CG). CG is recognized within linguistic theory as the logical ultimate in 'lexical syntax', a model in which all syntactic information is held in the lexical categories of individual words, and there is no separate component of 'grammar rules' - thus Kartunnen's (1989) 'radical lexicalism' (from which our title derives). A CG induction system has been built which provides evidence for 3.

2.2 Practical

Two major challenges of NLP systems are to support wide-ranging and developing vocabularies, and to work with more than strictly syntactically correct 'input. We believe that both can be addressed by (at least semi-) automatic lexicon growth, or induction.

Imagine an application that operates on 'real-world' textual input. For example, in the medical domain, a system that produces a semantic analysis of free-text hospital discharge summaries. A fixed lexicon is clearly

impracticable. Practitioners should not (and indeed will not) accept any artificially imposed limited vocabulary - it is simply not appropriate for their task. There are (at least) two imaginable strategies for handling the situation when a previously unknown word is encountered (either when encountering new domains, or extra complexity in a 'known' domain):

1) Bail out, and ask a linguist/lexicographer to manually augment the lexicon/grammar rules
2) Have the NLP system itself make sensible and useful suggestions as to the new word's syntactic category (and, potentially, its conceptual structure).

Similarly, there are (at least) three potential strategies for dealing with the situation in which a system encounters 'syntactically incorrect' input - but for which an error message is neither useful or appropriate - the input comes from something that has actually been said. These three strategies are:

1) Throw it out;
2) Invent a new rule to cope with the particular scenario;
3) Augment the lexicon with additional categories.

We believe that augmenting the lexicon is the only realistic approach to this problem.

An experimental prototype of a Categorial Grammar induction system (known as *L*) has been produced which illustrates and encourages our belief that a lexicon in which richly structured entities are the means of encoding syntactic and semantic knowledge can be 'grown' to meet these demands.

3 Categorial Grammar

The CG induced by *L* is a simple one on the scale of categorial calculi. It uses three atomic categories, S, N, and NP (we will return at the end of this paper to discuss the limitations of this atomic notation); two connectives: / (forward combination) and \ (backward combination); and three combination rules: forward application, backward application, and forward composition. The notation used is consistently result-first, regardless of the direction of the connective. Complex categories are implicitly left-bracketed, i.e. S\NP/NP is equivalent to (S\NP)/NP.

Complex categories are functions named by their arguments and outputs; there is no concept of 'verb', but rather of a function from one nominal to a sentence (S\NP, 'intransitive verb'), from two nominals to a sentence (S\NP/NP, 'transitive verb'), and so on. This 'combinatory transparency', their visible information structure, makes CGs well suited to corpus-based induction of a lexicon/grammar.

We rely on the property of 'parametric neutrality' (Pareschi 1986) - not only can we determine the result of combining two known categories, but from one category and the result we can determine the second category. Thus:

Given NP S\NP --> X then X = S;

Given X S\NP --> S then X = NP;

Given NP X --> S then X = S\NP.

Each category assignment we induce gives us more information to help with the next (as will be seen below); unlike traditional categories, these

have a rich information structure which we can query for help in making further decisions. We can therefore approach a text knowing only the identity of nouns and sentences and the principles of function application and composition, and from these we can induce the complex categories of the other words in the sentence; in other words, we can learn the lexicon, and in it the grammar.

4 *L* - a Categorial Grammar Induction System

4.1 Overview

L is a simple Categorial Grammar grammar induction system, based on holding theoretically motivated and empirically demonstrated linguistic information by representing words and their behaviour as complex structured objects in a format readable by both human and machine. The input is a simple text corpus in which the boundaries of sentences and, initially, the identity of a few nouns are known. The output - the result of the induction process - is a set of lexical categories for all the words in the corpus - which, as we have explained, constitutes a grammar for the corpus. This is made possible by the characteristic of 'parametric neutrality' described above. Many words have multiple categories (eg 'toy', as N 'noun' or N/N 'adjective') - this is how ambiguity is handled. *L* successfully infers multiple categories for many words in the corpus, and, critically, the categories it proposes do make cognitive sense to human linguists. This gives us hope that the system could be usefully guided and helped by humans in what is - in the limit - a difficult task: category assignments proposed by the system can be readily evaluated by its user. The lexicon induced by *L* has been used successfully, as a test, for simple text generation.

The system is implemented in CProlog on a SUN 3/50 workstation.

4.2 The Corpus

The corpus used is a selection of books from the Ladybird Key Words Reading Scheme, a series of books designed to help children learn to read, ordered in a graded sequence which was followed by the system. The system is 'bootstrapped' by a few examples of primitive categories - this is an example of where we feel that best results are obtained by not hog-tying the system for its own sake. Sentence boundaries are given by punctuation. A few nouns are defined in the corpus itself: the first books in the series begin with a sort of 'picture dictionary' of their central characters and objects, and we gave this starting point to the system also. Notice that this fits exactly the use of S and N(P) as atomic categories in CG.

We are encouraged here that our approach also has some psychological plausibility. Children learning a language do so by learning the names of things first, then how those names can be fitted together into propositions. That a sequence designed to help human learners should prove suitable for teaching a computer suggests that they may be working along similar lines.

An interesting side-effect of this choice is that the corpus is often syntactically odd - it was designed to help children's reading, rather than to teach grammar. However, the success of *L* on such an 'unsyntactic' (though understandable) corpus gives promise for its application on other 'real-world' corpora - real text and perhaps spoken word. (It must be admitted that *L* was completely baffled, in reading 'The Three Billy Goats Gruff', by the 'sentence' 'Trip trap, trip trap, trip trap!') - but this is surely allowable at this stage as an extreme case.)

4.3 Principles of Operation

L works due to a combination of computer processing power and statistical evidence applied to an underlying linguistic theory - in our view, the 'best of both worlds'.

It is a simple system; this simplicity itself we regard as a significant achievement.

In the description which follows, some example output from a simple text-based interface to *L* is included to illustrate the various processes involved.

The system has a few 'boot-strapped' primitive categories, as explained above. A multi-pass iterative approach is used to analyse and further annotate the corpus - the first pass uses just the few identified categories. On each pass, *L* assigns categories to more words of the corpus. (The strategy bears some resemblance to island-based parsing, which similarly begins with the point(s) of greatest certainty in an input string and works outwards from them.) Each pass has three parts:

1) The system selects which word to try to categorise in this pass. It uses statistical evidence to choose for analysis the word which occurs in the corpus with the most consistent, already categorised, (immediate) nearest neighbours. Clearly a precondition for this approach is to have at least some categorised words in the corpus - having some boot-strapped categories enables *L* to embark in a sensible direction.

2) Assign a category. If the word to be assigned is the last remaining uncategorised word in a sentence, then the principle of parametric neutrality is applied. Due to the compositional, recursive nature of Categorial Grammar categories, *L* can always find a category to fit. For example:

```
Pass 4 ...
missing link completion -
  assigning 's\np/np' to 'likes'
 with a confidence of 14/16
EXAMPLE:
  original sentence : Peter likes
 the ball.
  before this pass  : np likes np
  now reduced to    : s
```

Note that the assignment found is only applied to those instances in the corpus which are both sanctioned by CG, and 'deemed appropriate' by *L* itself - in this case, 14 out of 16 occurrences. This mechanism allows *L* the opportunity of assigning multiple different categories to a word, to cope with ambiguity.

Note also that the information given by the 'confidence' measure is more than the probability which would be expressed by reducing it to 'one in ...'. 16/16 indicates a common (in this

small corpus) and unambiguous word, while a word given 1/1 has been found a category on its one appearance; a word with a rating of 10/15 is established as regularly ambiguous, but 2/3 could prove on further exposure to be mainly regular, with only one occurence of an alternative category. These degrees of probability are taken into account by the algorithm which assigns categories in new text.

If the word is not the last in the sentence to be categorised, a more complex approach is required. The argument and direction of the resulting category is obtained as a by-product of the previous stage - the result is determined by an examination of the 'behaviour' of the resulting category in the corpus. In this context, 'behaviour' is defined as the pattern of neighbours' categories in the corpus so far. For example:

```
Pass 5 ...
  assigning 'np/n' to 'jane's'
 with a confidence of 5/6
EXAMPLE:
  original sentence : Here is
 jane's shop.
  before this pass : Here is
 jane's n
  now reduced to   : Here is np
```

In this case, the word jane's is chosen because, informally, it frequently occurs before a N. This tells us that the category to assign must have an argument of N, and the direction must be forward - in other words a x/N. This category is completed with a NP because this is a reasonable behavioural match with the rest of the corpus - NP often appears after is. Note that statistical evidence is again essential.

3) Having obtained a putative category, this is then applied to the occurrences of the word in the corpus as long as it is sanctioned by the semantics of CG. In this way, ambiguity is captured - only those occurrences which 'fit' are categorised at each pass.

4.4. Results and Evaluation

L's initial corpus was books 1a and 1b of the Ladybird Key Words Reading Scheme. They contain 351 word tokens, using a vocabulary of about 20 different words. This corpus was completely processed in 17 passes. Processing later books in the series has brought *L*'s current vocabulary up to some 55 words. This is still small, of course, but the nature of the induction process means that growth should 'snowball' as each known word helps in the categorization of further new words. (And see Shieber quoted in Ritchie (1987) for a revealing discussion of the vocabulary sizes of most NLP research prototypes).

Within this limited vocabulary, *L* has 'correctly' induced examples of the following categories: determiners, adjectives, prepositions, conjunctions, intransitive, transitive and di-transitive verbs, imperatives, and some auxiliaries. Furthermore, *L* discovers and represents ambiguity of the following types: adjective vs noun; sentence co-ordination vs noun-phrase co-ordination; prepositional form; noun-phrase vs determiner, and verbs of quotation. An example of the latter are four structural forms that *L* induces for says (as in 'Rhubarb

rhubarb says Jane' or 'Jane says rhubarb rhubarb').

L has also inadvertently re-invented type-raising, assigning the category (S/(S\NP))/N to a sentence-initial determiner: the system's exact method of exploiting parametric neutrality told it that this word needed a following noun to form a function into a sentence from a following 'verb phrase'. A slightly different algorithm would have given the more standard NP/N, and reduction mechanisms could easily be implemented to find simpler equivalents, where possible, of highly complex proposed categories. Indeed they will almost certainly be needed, as witness *L*'s assignment of the category:

S\NP/(S\NP\S)/(S\NP\NP)

to you as the last uncategorized word in the sentence 'Here you are Jane says Peter'.)

Evaluation of the results took two forms:

1) Use of the lexicon for generation. Many different lexicons could have been produced which would account only for the training corpus. Using the lexicon for generation of new text provided evidence that it was more general. The text generated was in character with the corpus - for example, 'Peter you are in it says I'. This is an important result; we have evidence that the grammar created is general, but does not over-generate.
2) Inspection. L produced cognitively plausible results - i.e. as well as producing categories that enable the entire corpus to reduce to a sequence of Ss, the results reflect what are traditionally (manually) assigned to each word - for a wide range of syntactic constructions, providing further evidence that the lexicon produced is not just corpus-specific.

5 Discussion and Prospects

L, as an experimental prototype, has demonstrated the practicability of a radically lexical approach for managing some of the major challenges for NLP. It learns the grammar of (the words of) new text and represents what it has learned in structured linguistic entities which are readable equally by computer and computational linguist. Its only starting point is the general rule of function application and the identity of a few nouns and sentences. We firmly believe that this minimalist methodology - assume as little as possible, and use principles which are as general as possible - is sound.

L copes with that bugbear of NLP - ambiguity; it successfully infers multiple categories for many words in the corpus, and, critically, the categories it has so far proposed have been hand-checked and do make cognitive sense to human linguists. This makes us optimistic as to the ease with which such systems and their users will be able to co-operate. The lexicon induced by *L* has been used successfully, as a test, for simple text generation.

Obviously there is a great deal of work still to be done. The simple atomic S, N, NP category notation used by *L* cannot represent finer-grained morphological information such as number, case, gender, and tense. This is clearly shown by the first

sentence *L* generated: 'I likes I'. Our first priority is therefore to introduce a more complex notation, probably using bundles of attribute/ value pairs in the style of Unification Categorial Grammar (Zeevat 1988). (Clearly our comments at the beginning of this paper about the value of structured representations apply *a fortiori* here also.) We expect that a minimal explicit seeding of the corpus with these values will allow the system to 'learn' them also.

Secondly, lexical semantics has not yet been addressed. The move to a feature/value notation will also provide a framework in which this is possible. The seeding and learning will be a more difficult task, which awaits investigation. A log of which particular words regularly co-occur should at least help in automatically establishing broad semantic fields - again, we expect a judicious balance of statistical and theoretical information to be appropriate here.

Thirdly - once we have established a notation adequate to these demands - we will grow the lexicon/grammar by processing further texts of increasing complexity - first within the Ladybird series, but going on to real-world text, possibly in a medical domain. We do not anticipate serious difficulty in progressing through the Ladybird series, but we are well aware that there is a significant step from there to 'adult' text, at which point scaling up may well not be trivial.

We have mentioned in passing that it is encouraging from a psychological perspective to find that a text corpus designed to aid human learning should prove well suited to machine learning. Of course learning to read a known language is a different task from learning a language. However we hope eventually to explore the potential of our approach for modelling children's learning, and perhaps the use of its text generation ability in producing teaching material.

The success of our approach - at least at prototype level - should be contrasted with other attempts at grammar induction. Some, typically, use traditional atomic 'grammatical categories' with no inherent information content, mapped in complex ways (which must also be learned) onto a large set of 'grammar rules'. Others 'learn' columns of numbers which could equally well describe the co-occurrence of bird-tracks in snow with various garden shrubs. To quote Pustejovsky *et al* (1993:354), 'statistical results themselves reveal nothing, and require careful and systematic interpretation by the investigator to become linguistic data.'

L is inspired and informed by an independently motivated and respected theory of natural language, and depends for its realization on a corpus of real-world text. Our theoretical understanding of how words combine gives us a principled way into a text corpus; statistical evidence suggests and confirms the behaviour of words and therefore their lexical/grammatical categories.

NLP appears currently to be split by civil war between theorists with sound principles but no real data and statisticians with volumes of data but no linguistic principles. There will only be significant progress with real prospects in NLP when the theory-driven and empiricist approaches respect each other and work together.

We hope we have shown one way in which this can be done.

References

Berwick, Robert. 1985. **The Acquistion of Syntactic Knowledge**. MIT Press, Cambridge, Mass.

Brent, Michael. 1993. *From Grammar to Lexicon: Unsupervised Learning of Lexical Syntax*. **Computational Linguistics** 19.2 pp 243-262.

Brown, Peter F., Stephen A. Della Pietra, Vincent J Della Pietra, and Robert L. Mercer. 1993. *The Mathematics of Machine Translation: Parameter Estimation*. **Computational Linguistics** 19.2 pp 263-311.

Church, Ken. 1993. *Termworks: Tools for Human Translators*. Invited talk, Sixth Conference of the European Chapter of the Association for Computational Linguistics, Utrecht, 21 April 1993.

Kartunnen, Lauri. 1989. *Radical Lexicalism*. In Baltin & Kroch (eds). **Alternative Conceptions of Phrase Structure**. University of Chicago Press.

Pareschi, Remo. 1986. *Combinatory grammar, logic programming and natural language processing*. Ms, Dept of Artificial Intelligence, University of Edinburgh.

Pustejovsky, James, Sabine Bergler, and Peter Anick. 1993. *Lexical Semantic Techniques for Corpus Analysis*. **Computational Linguistics** 19.2 pp 331-358.

Ritchie, Graeme. 1987. *The Lexicon*. In Whitelock, Wood, Somers, Johnson, and Bennett (eds). **Linguistic Theory and Computer Applications**. Academic Press, London.

Sag, Ivan. *Extraction without traces, empty COMPs or function composition*. Invited talk, Sixth Conference of the European Chapter of the Association for Computational Linguistics, Utrecht, 22 April 1993.

Solomon, W.D. 1991 *Learning a Grammar*. University of Manchester Dept of Computer Science Technical Report UMCS-AI-91-12-1.

Wood, Mary McGee. 1993. **Categorial Grammars**. Routledge, London.

Zeevat, Henk. 1988. *Combining categorial grammar and unification grammar*. In Reyle & Rohrer (eds). **Natural Language Parsing and Linguistic Theories**. Reidel, Dordrecht.