More for less: learning a wide covering grammar from a small training set

Miles Osborne and Derek Bridge
Department of Computer Science, University of York, Heslington,
York YO1 5DD, U. K.
{miles,dgb}@minster.york.ac.uk

Abstract

This paper describes a grammar learning system which combines model-based and data-driven learning within a single framework. Results from learning grammars with the Spoken English Corpus (SEC) suggest that a combined model-based and data-driven learner can acquire a wide coverage grammar from only a small training corpus.

Keywords: Corpus-based NLP, Statistical NLP, Deductive NLP, Hybrid approaches.

1: Introduction

In this paper, we present some results of our grammar learning system. We show that using unification-based grammars, with a hybrid learning system allows a rapid rate of convergence upon a test corpus with only a modest amount of training material.

In contrast to other researchers (for example (BMMS92; GLS87; Bak79; LY90; VB87)), we try to learn competence grammars and not performance grammars. We also try to learn grammars that assign linguistically plausible parses to sentences. Learning competence grammars that assign plausible parses is achieved by combining model-based and data-driven learning within a single framework (OB93b; OB93a; Osbng). Model-based (deductive) methods are sound (MKKC86) (assuming that the model is consistent), but suffer from incompleteness, whilst data-driven (inductive) methods are unsound (they cannot guarantee that natural languages can be learnt (Gol67)), but complete. Note that 'completeness' here means that the learner is always in a position to make a decision. We let both of the learning styles compensate for each other's weaknesses. A recent result showed that the combined use of induction and deduction produced a grammar that assigned quantitatively more plausible parses to sentences taken from the Spoken English Corpus (SEC) (LG91) than is the case when using either learning style in isolation (OB94).

The system is implemented to make use of the Grammar Development Environment (GDE) (CGBB88) and it augments the GDE with 3300 lines of Common Lisp.

The structure of this paper is as follows. Section 2 gives an overview of the combined model-based and data-driven learner. Section 3 then describes the method used to generate the results, which are then presented in section 4 . Section 5 discusses these results and points the way forward.

2: System overview

Architecture

We assume that the system has some initial grammar fragment, G, from the outset. Presented with an input string, W, an attempt is made to parse W using G. If this fails, the learning system is invoked. Learning takes place through the interleaved operation of a parse completion process and a parse rejection process.

In the parse completion process, the learning system tries to generate rules that, had they been members of G, would have enabled a derivation sequence for W to be found. This is done by trying to extend incomplete derivations using what we call *super rules*. Super rules are the following unification-based grammar rules:

$$\begin{array}{ll} [\,] \rightarrow [\,]\,[\,] & \text{(binary)} \\ [\,] \rightarrow [\,] & \text{(unary)} \end{array}$$

The binary rule says (roughly) that any category rewrites as any two other categories, and the unary rule says (roughly) that any category rewrites as any other category. The categories in unification grammars are expressed by sets of feature-value pairs; as the three categories in the binary super rule and two categories in the

unary super rule specify no values for any of the grammar's features, these rules are the most general (or vacuous) binary and unary rules possible. These rules thus enable constituents found in an incomplete analysis of W to be formed into a larger constituent. In unifying with these constituents, the categories on the right-hand side of the super rules become partially instantiated with feature-value pairs. Hence, these rules ensure that at least one derivation sequence will be found for W.

In fact, in practice, many instantiations of the super rules may be produced by the parse completion process described above. Linguistically implausible instantiations must be rejected and we interleave this rejection process with the parse completion process. Rejection of rules is carried out by the model-driven and data-driven learning processes described below. As an idea of the need to reject super rule instantiations, a short sentence (5 words) was parsed, using just the binary super rule, and without any rejection. This resulted in 15 rules learnt, of which all but one spurious. However, reparsing the same sentence, but this time rejecting implausible super rule instantiations only resulted in the single, desired rule being learnt.

If all instantiation are rejected, then the input string W is deemed ungrammatical. Otherwise, those surviving instantiations of the super rules that were used to create the parse for W are regarded as being linguistically plausible and may be added to G for future use.

Model-driven learning

A grammatical *model* is a high-level theory of syntax. In principle, if the model is complete, an 'object' grammar could be produced by computing the 'deductive closure' of the model (e.g. a 'meta'-rule can be applied to those 'object' rules that account for active sentences to produce 'object' rules for passive sentences). An example of purely model-based language learning is given by Berwick (Ber85). More usually, though, the model is incomplete and this leads us to give it a different rôle in our architecture.

For the purposes of evaluating our architecture, our (incomplete) model currently consists of GPSG Linear Precedence (LP) rules (GKPS85), semantic types (Cas88), a Head Feature Convention (GKPS85) and X-bar syntax (Jac77).

- *LP rules* are restrictions upon *local trees*. A local tree is a (sub)tree of depth one. An example of an LP rule might be (GKPS85, p.50):

  $$[\text{SUBCAT}] \prec \sim [\text{SUBCAT}]$$

  This rule should be read as 'if the SUBCAT feature is instantiated (in a category of a local tree) then the SUBCAT feature of the linearly preceding category should not be instantiated'. The SUBCAT feature is used to help indicate minor lexical categories, and so this rule states that verbs will be initial in VPs, determiners will be initial in NPs, and so on. In our learning system, any putative rule that violates an LP rule is rejected.

- We construct our syntax and semantics in tandem, adhering to the *principle of compositionality*, and pair a semantic rule to each syntactic rule (DWP81). Our semantics uses the typed λ-calculus with extensional typing. For example, the syntactic rule:

  $$S \rightarrow NP\ VP$$

  is paired with the following semantic rule:

  $$\mathbf{VP(NP)}$$

  which should be read as 'the functor **VP** takes the argument **NP**'[1]. The functor **VP** is of type[2]:

  $$<<< e,t >,t >,t >$$

  and the argument **NP** is of type:

  $$<< e,t >,t >$$

  The result of applying **VP(NP)** has the type:

  $$t$$

  For many newly-learnt rules, we are able to check whether the semantic types of the categories can be composed by function application. If they cannot, then the syntactic rule can be rejected. For example, the syntactic rule:

  $$VP \rightarrow\ VP\ VP$$

  has the semantic rule **VP(VP)**, which is ill-formed because the type

  $$<<< e,t >,t >,t >$$

  cannot be applied to itself.

- Head Feature Conventions (HFCs) help instantiate the mother of a local tree with respect to immediately dominated daughters. For example, the verb phrase dominating a third person verb is itself third person.

- X-bar syntax specifies a restriction upon the space of possible grammar rules. Roughly speaking, the RHS of a rule contains a distinguished category called the *head* that characterises the rule. The LHS of the rule is then

[1] Syntactic categories are written in a normal font and semantic functors and arguments are written in a **bold** font.

[2] The exact details of these types is not important to understanding the thrust of this section and so they are not given any detailed justification.

a *projection* of the head. Projecting the head category results in a phrasal category of the same syntactic class as that of the head. For example, the rule $NP \rightarrow Det\ N1$ has a nominal head and a NP projection.

This incomplete model plays two very similar, but different rôles in improving on the hypothetical super rule instantiations resulting from the parse completion process. Firstly, LP rules and semantic types can be used to *reject* instantiations which, through containing rule right-hand sides whose categories are wrongly ordered or do not comprise categories of compatible semantic type, violate these aspects of the model. Note that the rule rejection process can be as restrictive as desired. For a stricter model, the LP rules and types can be made more general. For a liberal model, the LP rules and types can be restricted in scope. Secondly, the HFC and the principles of X-bar syntax can be used to *refine* putative rules by further instantiating their features, thus forming more specific rules. Note that these rule refinement operators may at times prevent idiosyncratic rules, necessary for generating unusual constructs found in a corpus, being learnt.

Data-driven learning

In most other inductive systems (for example (GLS87; LG91; Mag94)), the data-driven component uses a treebank consisting of flat local trees, generated by highly specific rules. Hence, these systems in turn tend to acquire rules that contain many categories in their right-hand side and are also highly specific. As we have already explained, in our system new rules are proposed by the parse completion process, and the data-driven component, like the model-driven component, acts as a filter upon these proposed rules.

Our data-driven component can prefer learnt rules that are 'similar' to rules previously seen by the parser. For this to work well, the system needs some training, prior to any learning taking place. This consists of a small, preparsed corpus. This can then be used to score instantiations of the super rules when learning begins.

The learner is trained by recording the frequencies of mother-daughter pairs found in parses of sentences taken from the preparsed corpus (LG91). For example, the tree (S (NP Sam) (VP (V laughs))) has the following mother-daughter pairs:

<S,NP>
<S,VP>
<VP,V>

The frequencies of mother-daughter pairs in the parse trees that have previously been assigned to sentences of the preparsed corpus are noted. From these frequencies, the score of each distinct mother-daughter pair can be computed: if pair <A, B> occurs with frequency n out of a total number of N mother-daughter pairs, then the mother-daughter pair's score, f, is:

$$f(< A, B >) = n/N$$

This set of mother-daughter frequencies is computed in advance of using our system for learning.

During learning, after parse completion by the super rules, local trees in completed parses can be scored. The score is computed recursively, as follows:

- For local trees of the form (A (B C)) whose daughters are leaves, the score of the local tree is:

$$score(A) = gm(f(< A, B >), f(< A, C >))$$

  where gm is the geometric mean. We take the geometric mean, rather than the product, to avoid penalising local trees that have more daughters over local trees that have fewer daughters (MM91).

- For interior trees of the form (B (C D)), the score of the local tree is:

$$score(B) = gm(score(C) \times f(< B, C >), score(D) \times f(< B, D >))$$

(This does leave the problem of dealing with mother-daughter pairs that arise in completed parses but which did not arise in the preparsed corpus. These can be given a low score. Giving them a score ensures that all trees can be scored, and thus the data-driven learner is 'complete'.)

After scoring, instantiations of the super rule that have daughters whose scores exceed some threshold can be accepted. Other instantiations can be rejected.

The approach we have described is a generalisation of the work of Leech, who uses a simple phrase structure grammar, whereas we use a unification-based grammar (Lee87). Note that there are also mechanisms to retract rules, or to specialise rules, at some later stage in the learning process. These mechanisms have not been fully tested and so are not reported in this paper.

3: Method

In this section we will show how the previously described system can be used to create a wide covering grammar using only a small training set. We will also show how the system converges upon this grammar. *Coverage* is defined as the proportion of sentences of the test set that are generated by the grammar, whilst *convergence* is defined as how many sentences in the training set are required to achieve this coverage. Clearly, a

learning approach that converges using a smaller size of training material is preferable to one that converges more slowly, other things being equal.

We use the SEC as a source of training material. The SEC is a collection of monologues for public broadcast and is small (*circa* 50,000 words) in comparison to other corpora, such as the Lancaster-Oslo-Bergen Corpus (JLG78), but sufficiently large to demonstrate the capabilities of the learning system. Furthermore, the SEC is tagged and parsed, thus side-stepping the problems of constructing a suitable lexicon and of creating an evaluation corpus to determine the plausibility of the learnt grammars.

To determine the convergence of our system, the following steps were taken:

- Three disjoint sets of sentences were arbitrarily selected from the SEC. These were *pretrain* (less than 20 sentences), *train* (60 sentences) and *test* (60 sentences).
- A grammar, G, was used as the initial grammar. This was manually constructed and consisted of 97 unification-based rules with a terminal set of the CLAWS2 tagset (BGL93).
- The Model was configured to consist of 4 LP rules, 32 semantic types, X-bar syntax and a Head Feature Convention.
- The sentences in *pretrain* were used to provide the initial estimate of mother-daughter frequencies later used by the data-driven learner.
- Each sentence in *train* was then parsed. For those that failed to parse, interleaved parsing and learning, using both the data-driven component and the model-driven components, was invoked. Notionally, *train* was partitioned into groups of 10 sentences which were processed incrementally. After dealing with each group, the resulting grammar was saved for subsequent inspection. This gave 6 grammars, $G_1 \ldots G_6$, where $L(G_i) \subseteq L(G_{i+1})$.
- *Test* was then parsed, without learning, using each of the grammars $G, G_1, \ldots, G_6$. The number of sentences successfully parsed in *test* using each grammar was recorded.

4: Results

The following table shows the growth in grammar size (measured in number of rules) and the coverage of that grammar (as a percentage of *test*):

Grammar	Size	Coverage
G	97	26.6
G_1	101	38.3
G_2	107	43.3
G_3	111	43.3
G_4	118	46.6
G_5	122	51.6
G_6	128	75.0

In other words, after learning from the first 60 sentences, the grammar G_6 had 128 rules and parsed 75.0% of *test*. This compares with the original grammar G's 97 rules covering 26.6% of *test*.

5: Discussion

With only a small training corpus, we increased the coverage of a manually constructed grammar by approximately 50%. By comparison with other researchers who use purely inductive methods, we seem to get good results with only a small *train*. Below we compare the size of our *train* with that of others reported in the literature:

Approach	*Train*(no. of words)
Sharman *et al.* (SJM88)	50,000
Black *et al.* (BGL93)	470,000
Brill *et al.* (BMMS92)	900,000
Osborne and Bridge	837

We make no formal claims in this paper that these systems produce 'equivalent' grammars, given the difficulty in making such a comparison, but instead argue that expressing grammaticality concisely within a model reduces the need to infer grammaticality from exposure to large amounts of text. Support for this view can be found in the human language acquisition literature, which suggests that children do not acquire languages purely inductively. A child does not encounter enough training material to account for the acquisition of a full adult competence grammar (Cho75; Whi89; Cho81). Instead, children are thought to learn language largely deductively. Our results reflect this view: languages can be identified with only a modest amount of training material if deduction is used. Approaches that use induction alone will need ever increasing amounts of data to attempt to identify in the limit any given natural language.

Acknowledgements

We would like to thank Eric Atwell (Leeds University) for allowing access to the SEC, Ted Briscoe (Cambridge University) for supplying the grammar G, Tony Griffiths for proof reading, and the two anonymous referees for suggestions. The first author is supported by a Science and Engineering Research Council grant.

References

J. K. Baker. Trainable grammars for speech recognition. In D. H. Klatt and J. J. Wolf, editors, *Speech Communication Papers for the* 97^{th} *Meeting of the Acoustical Society of America*, pages 547–550. 1979.

Robert C. Berwick. *The acquisition of syntactic knowledge*. MIT Press, 1985.

Ezra Black, Roger Garside, and Geoffrey Leech, editors. *Statistically driven computer grammars of English the IBM-Lancaster approach*. Rodopi, 1993.

Eric Brill, David Magerman, Mitchell Marcus, and Beatrice Santorini. Deducing Linguistic Structure from the Statistics of Large Corpora. In *AAAI-92 Workshop Program: Statistically-Based NLP Techniques, San Jose, California*, 1992.

Claudia Casadio. Semantic Categories and the Development of Categorial Grammars. In Richard T. Oehrle, editor, *Categorial Grammars and Natural Language Structures*, pages 95–123. D. Reidel, 1988.

John Carroll, Claire Grover, Ted Briscoe, and Bran Boguraev. A Development Environment for Large Natural Language Grammars. Technical report number 127, University of Cambridge Computer Laboratory, 1988.

Noam Chomsky. *Reflections on Language*. Pantheon, 1975.

Noam Chomsky. *Lectures on Government and Binding*. Dordrecht: Foris, 1981.

D.R. Dowty, R.E. Wall, and S. Peters. *Introduction to Montague Semantics*. D. Reidel Publishing Company, 1981.

G. Gadzar, E. Klein, G.K. Pullum, and I.A. Sag. *Generalized Phrase Structure Grammar*. Harvard University Press, 1985.

R. Garside, G. Leech, and G. Sampson, editors. *The Computational Analysis of English: A Corpus-based Approach*. Longman, 1987.

E. M. Gold. Language Identification to the Limit. *Information and Control*, 10:447–474, 1967.

Ray S. Jackendoff. *X-Bar Syntax: A Study of Phrase Structure*. The M.I.T Press, 1977.

S. Johansson, G. Leech, and H. Goodluck. Manual of Information to Accompany the Lancaster-Oslo/Bergen Corpus of British English, for Use with Digital Computers. Technical report, Department of English, University of Oslo, 1978.

Fanny Leech. *An approach to probabilistic parsing*. MPhil Dissertation, 1987. University of Lancaster.

Geoffrey Leech and Roger Garside. Running a grammar factory: The production of syntactically analysed corpora or "treebanks". In Stig Johansson and Anna-Brita Stenström, editors, *English Computer Corpora: Selected Papers and Research Guide*. Mouten de Gruyter, 1991.

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

David M. Magerman. *Natural Language Parsing as Statistical Pattern Recognition*. PhD thesis, Stanford University, February 1994.

T. Mitchell, R. Keller, and S. Kedar-Cabelli. Explanation-based generalization: A unifying view. *Machine Learning*, 1.1:47–80, 1986.

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

Miles Osborne and Derek Bridge. Inductive and deductive grammar learning: dealing with incomplete theories. In *Grammatical Inference Colloquium, Essex University*, 1993.

Miles Osborne and Derek Bridge. Learning unification-based grammars and the treatment of undergeneration. In *Workshop on Machine Learning Techniques and Text Analysis, Vienna, Austria*, 1993.

Miles Osborne and Derek Bridge. Learning unification-based grammars using the Spoken English Corpus. In *The Second International Grammatical Inference Colloquim, Alicante, Spain*, 1994.

Miles Osborne. *Learning Unification-based Natural Language Grammars*. PhD thesis, University of York, forthcoming.

R. A. Sharman, F. Jelinek, and R. L. Mercer. Generating a grammar for statistical training. In *Proceedings of the IBM Conference on Natural Language Processing*, 1988.

Kurt Vanlehn and William Ball. A Version Space Approach to Learning Context-free Grammars. *Machine Learning*, 2.1:39–74, 1987.

Lydia White. *Universal Grammar and second language acquisition*. John Benjamins Publishing Company, 1989.