Machine Learning of Grammatical Knowledge

Pat Langley
Computational Learning Laboratory
Center for the Study of Language and Information
Stanford University, Stanford, CA 94305 USA

The Task of Grammar Acquisition

An important task for computational linguistics and machine learning involves the acquisition of grammatical knowledge.

- *Given:* Partial experience with the sentences for some language L;
- *Given:* Background knowledge about constraints on the target language;
- *Find:* grammatical knowledge that lets one recognize, parse, or generate sentences in L.

We will consider some issues in grammar acquisition and some approaches to this induction task.

This learning problem is also known as *grammar induction* or *grammatical inference*.

Some Rhetorical Distinctions

The literature on grammar acquisition is littered with arguments about supposedly deep differences in research paradigms, such as:

- empiricist vs. nativist theories
  - despite the clear role of both experience and knowledge
- connectionist vs. symbolic vs. probabilistic models
  - despite their very similar inductive behavior

Such rhetorical distinctions are more about politics than about science.

Some Realistic Distinctions

In contrast, some realistic issues that have more scientific relevance include:

- the space of grammars considered
- the nature of the training data
- the nature of the training regimen
- the nature of the performance element

In this talk, I will focus on these issues and downplay rhetorical distinctions.

Each issue concerns *constraints* that make the induction task more or less tractable for learners.

The Space of Grammatical Hypotheses

One approach to making the grammar induction task tractable is to restrict the hypothesis space.

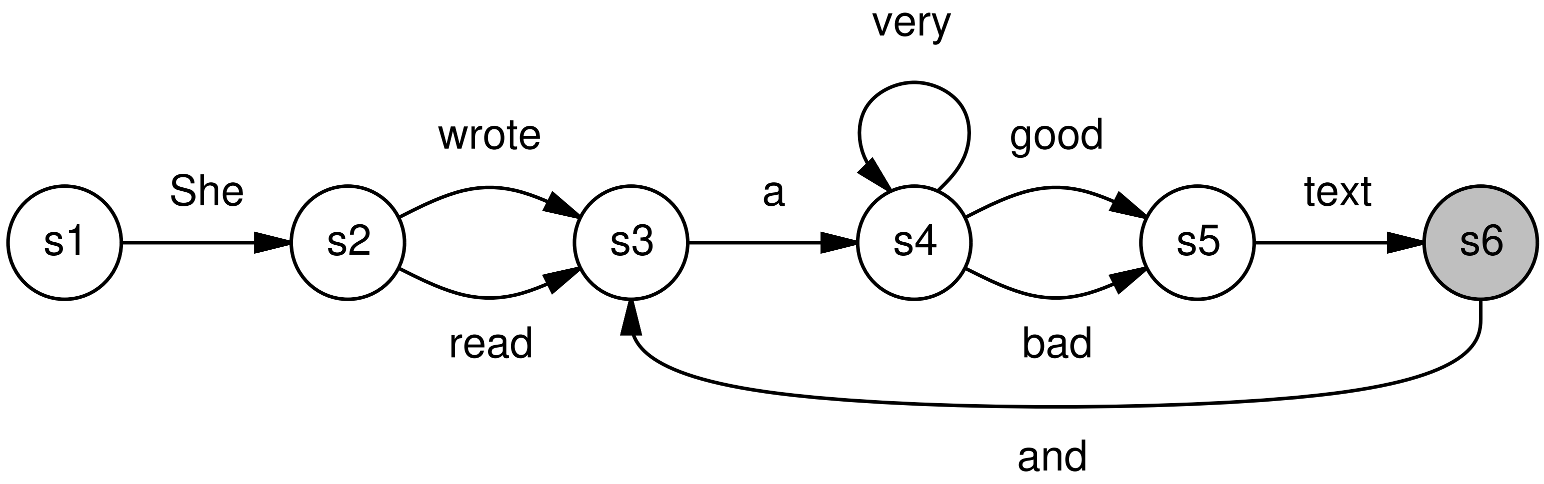
Restricted classes of languages that researchers have studied include:

- k-reversible languages
- finite-state (regular) languages
- context-free languages
- parametric languages

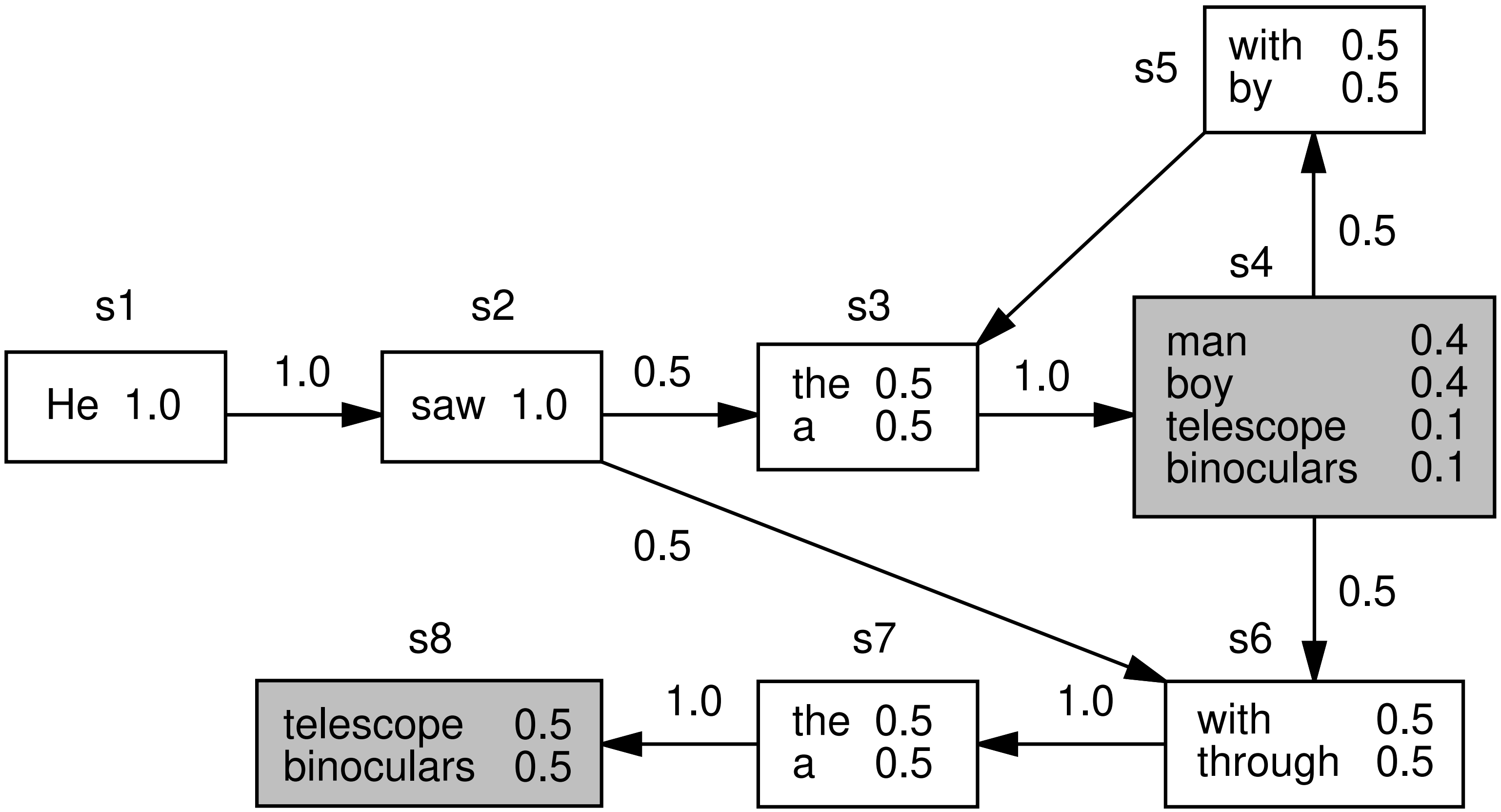
Note that the target language is independent of one's grammatical notation.

The bias-variance tradeoff holds in grammar induction tasks, just as it does in classification domains.

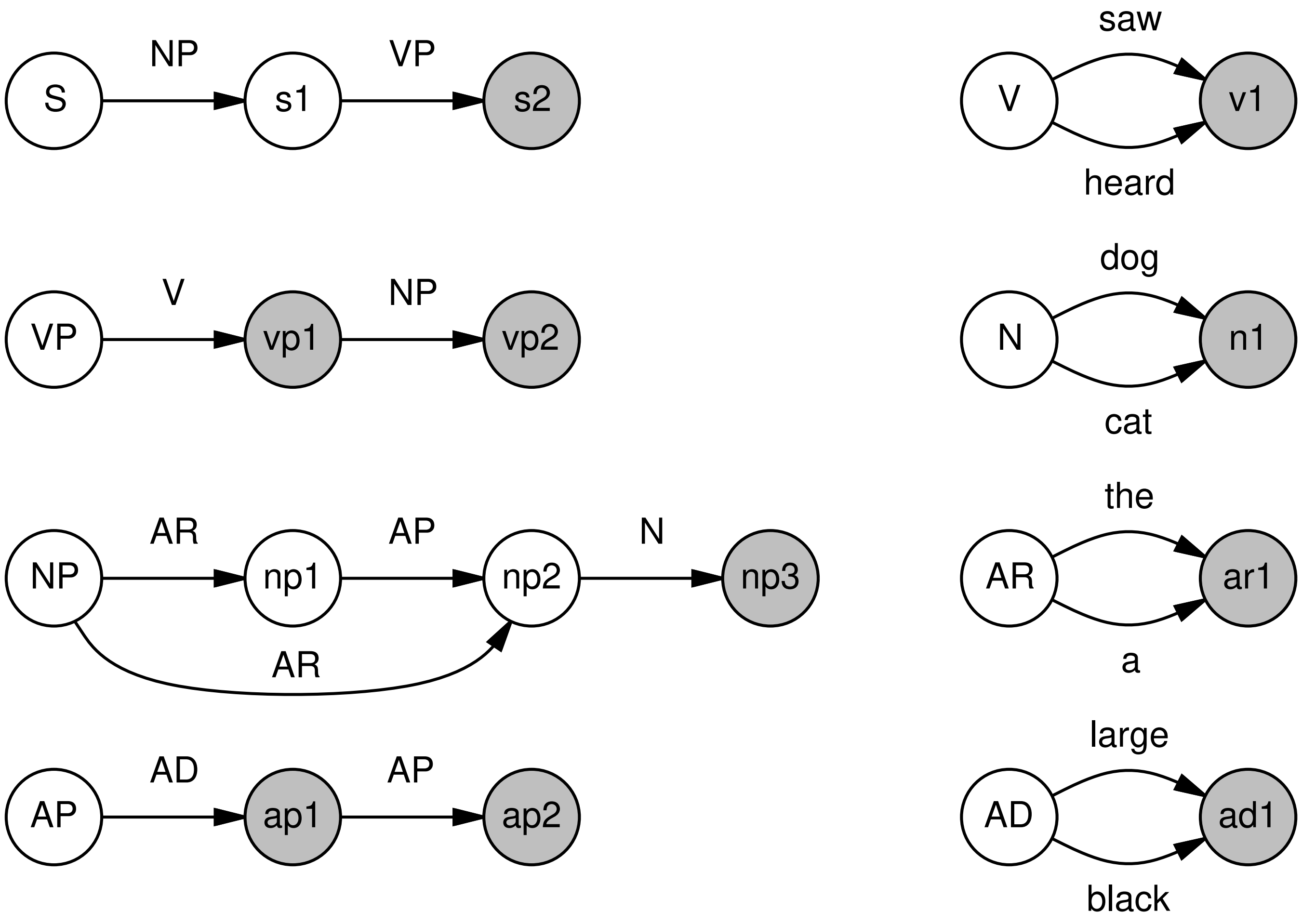
Transition Network for a Finite-State Grammar



Hidden Markov Model for a Finite-State Grammar



Recursive Transition Network for a Context-Free Grammar



Nature of the Grammar Induction Task

Research on grammar induction has explored two main classes of task:

- inducing grammars from sentences paired with meanings
    - or with parse trees for those sentences
- inducing grammars from training sentences
    - with or without negative training cases

These tasks correspond roughly to supervised and unsupervised learning.

- Gold's results are irrelevant to empirical work on grammar induction.

Nature of the Training Regimen

Research on grammar induction has studied a number of training regimens:

- random selection of training sentences
- simpler sentences before more complex ones
- learner-selected training sentences (active learning)

These constrain learning in different ways, yet each is a legitimate scenario for grammar induction.

But one cannot compare results obtained with one training regimen with those from another.

Nature of the Performance Element

Research in grammar induction has explored at least three broad classes of performance element:

- transition networks (problem-reduction parsers)
- shift-reduce parsers (state-space parsers)
- predictive parsers

Each framework suggests different approaches to grammar acquisition.

The latter decompose the problem into multiple supervised learning tasks.

Each can be implemented within different representational formalisms, such as rules, neural networks, or probabilistic schemes.

Some Methods for Grammar Induction

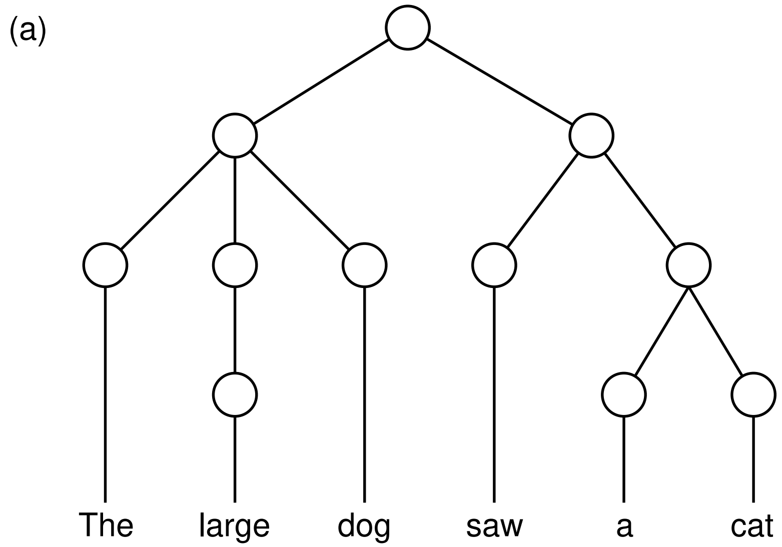
The area of grammar induction has a long history that includes algorithms for:

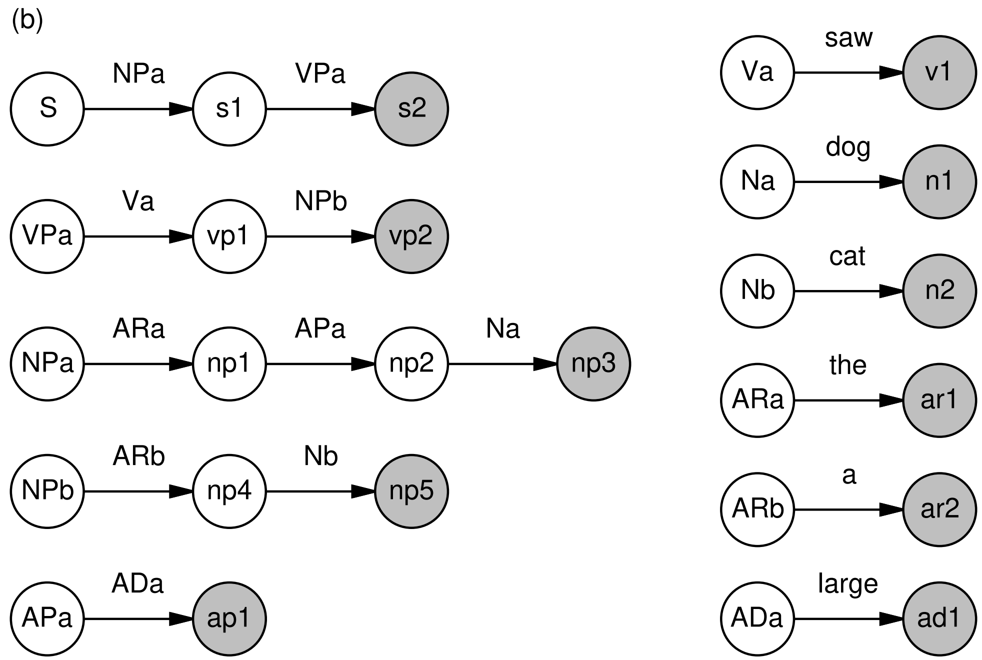
- incremental learning from parse trees (Anderson, 1977; Charniak, 1993)
- forming classes and phrases from distributions (Wolff, 1982; Stolcke, 1994)
- learning search control from successful parses (Berwick, 1980; Zelle, 1994)

Current researchers often ignore the wealth of earlier work on the topic.

As a result, they can miss important ideas for constraining the learning process.

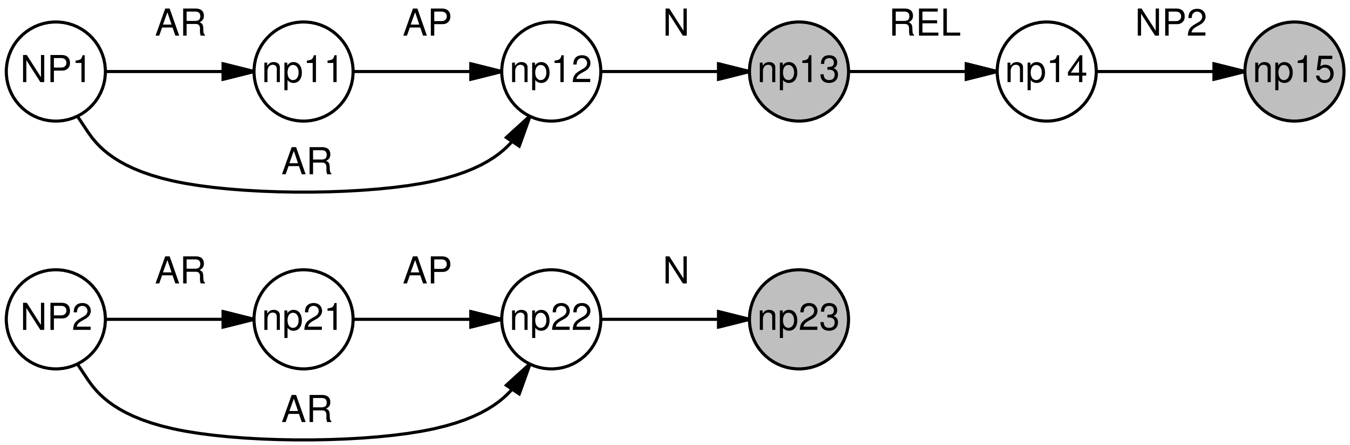
Parse Tree and Learned Recursive Transition Network



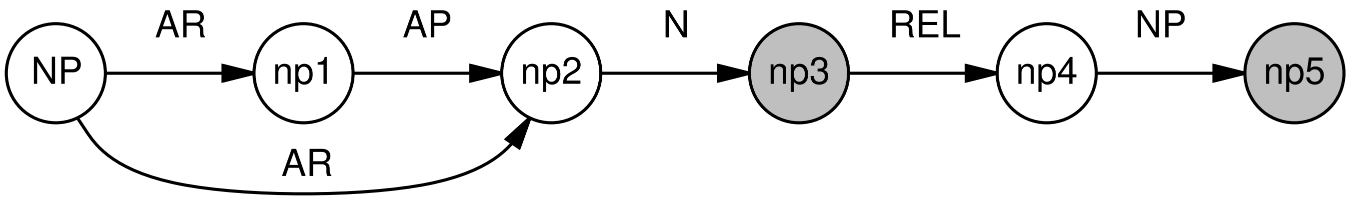


Two Transition Networks and a Merged Network

(a)



(b)



Learning from Distributional Information

One method for inducing context-free grammars from sample sentences using distributional information involves:

- *Start state:* one rewrite rule for each training sentence
- *Learning operators:* create a new symbol/rule, merge existing symbols
- *Search control:* greedy search alternating between create and merge
- *Evaluation metric:* minimum description length or a related measure
- *Halting condition:* no candidates lead to improvement

This approach starts with more complex, specific grammars and moves toward simpler, more general ones.

Here we describe Langley and Stromsten's (2001) GRIDS, but Wolff (1982), Stolcke (1994), and Petasis et al. (2002) report similar systems.

Creating a New Symbol

One of GRIDS' learning operators involves creating a new symbol that becomes the left-hand side of a rewrite rule.

$$\begin{aligned} NP &\rightarrow ART\ ADJ\ NOUN \\ NP &\rightarrow ART\ ADJ\ ADJ\ NOUN \end{aligned}$$

$$\Downarrow$$

$$\begin{aligned} NP &\rightarrow ART\ AP1 \\ NP &\rightarrow ART\ ADJ\ AP1 \\ AP1 &\rightarrow ADJ\ NOUN \end{aligned}$$

This operator produces simpler grammars but does not affect their generality.

Merging Two Symbols

The second GRIDS learning operator involves merging two symbols that occur in the the left-hand sides of rewrite rules.

$$NP \rightarrow ART\ AP1$$
$$NP \rightarrow ART\ AP2$$
$$AP1 \rightarrow ADJ\ NOUN$$
$$AP2 \rightarrow ADJ\ AP1$$

$$\Downarrow$$

$$NP \rightarrow ART\ AP1$$
$$AP1 \rightarrow ADJ\ NOUN$$
$$AP1 \rightarrow ADJ\ AP1$$

This operator produces grammars that are simpler and more general; it can also lead to recursive grammars, as shown above.

GRIDS' Evaluation Function

To direct its search, GRIDS uses an evaluation function that maximizes the posterior probability of the grammar M_i given the training sentences D:

$$\arg\min_i \left[-\log P(D|M_i) - \log P(M_i)\right] ,$$

which minimizes the description length of grammar M_i plus that for training sentences D, encoded as derivations in M_i.

This metric is biased against large grammars with overly specific rules and also against small but overgeneral grammars.

Evaluation of Grammar Induction Methods

Many ideas on experimental evaluation carry over to grammar induction from other areas of machine learning:

- the need for well-defined performance measures
- the need for separate training and test sets
- the need to average over different training sets
- the usefulness of learning curves
- advantages of natural *and* synthetic domains

Grammar induction would benefit from a well-formed experimental method.

- But the goals of science are understanding and insight.
- Competitions and bake-offs will not help us reach these goals.

An Example Study of Grammar Induction

To understand GRIDS' learning behavior, we examined four different grammars:

- English sentences with adjective phrases
- English sentences with relative clauses
- strings of a's followed by an equal number of b's
- strings of balanced parentheses

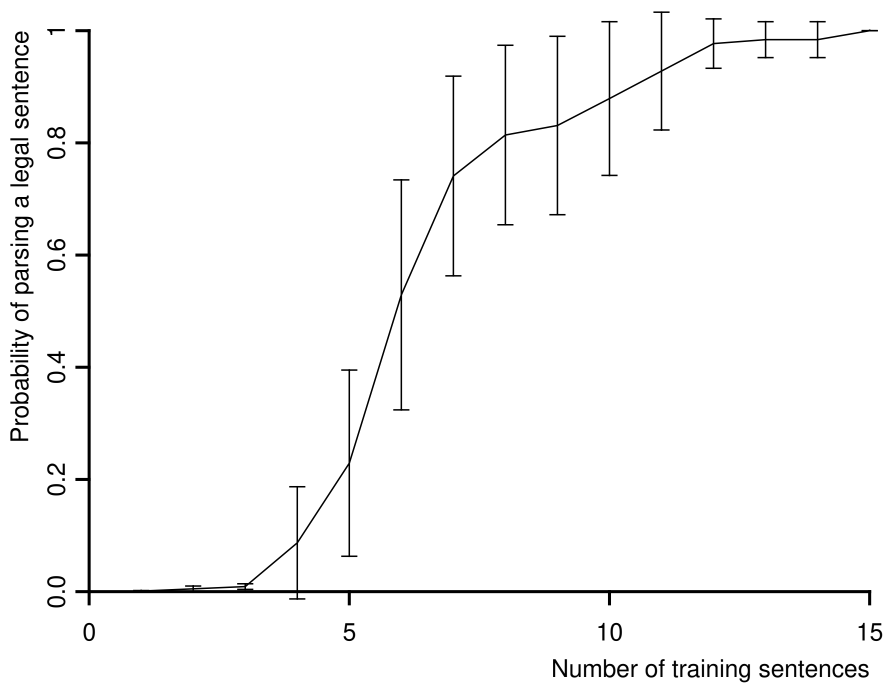
We used each of these target grammars T to generate 20 random training sets.

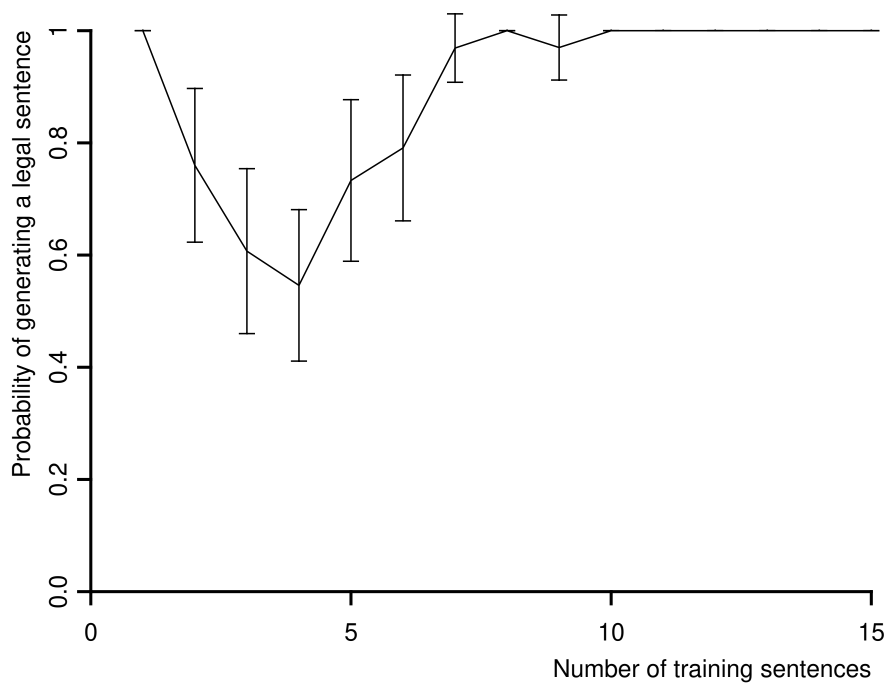
For each training set, GRIDS learned a grammar L, for which we measured:

- the percentage of sentences generated by T that were parsed by L
- the percentage of sentences generated by L that were parsed by T

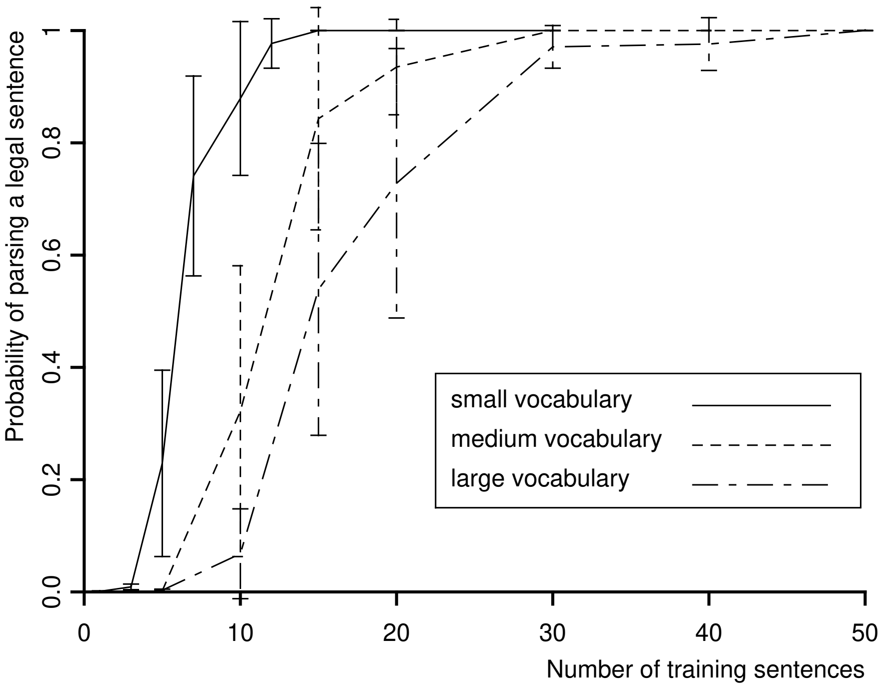
Averages of these scores produced measures of undergeneralization and overgeneralization, respectively.

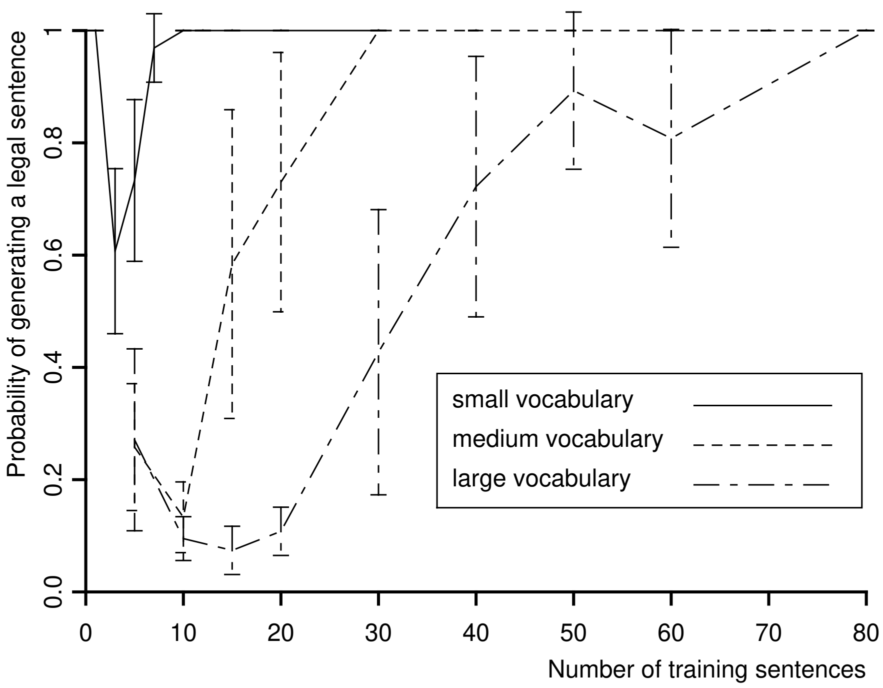
Learning Curves for Relative Clause Grammar





The Effect of Vocabulary Size





GRIDS' learning rate appears to scale linearly with grammar's vocabulary size.

Concluding Remarks

To summarize, the induction of grammatical knowledge is an important research task that:

- involves a number of quite distinct subproblems;
- is influenced by many factors besides the learning method;
- would benefit from fewer rhetorical arguments; and
- lends itself to systematic experimental evaluation.

There remains considerable room for additional research on this interesting and challenging topic.

Selected References on Grammar Induction

Anderson, J. R. (1977). Induction of augmented transition networks. *Cognitive Science*, *1*, 125–157. San Francisco, CA: Morgan Kaufmann.

Berwick, R. C., & Pilato, S. (1987). Learning syntax by automata induction. *Machine Learning*, *2*, 9–38.

Charniak, E. (1993). *Statistical language learning*. Cambridge: MIT Press.

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, *7*, 195–225.

Langley, P. (1982). Language acquisition through error recovery. *Cognition and Brain Theory*, *5*, 211–255.

Langley, P., & Stromsten, S. (2000). Learning context-free grammars with a simplicity bias. *Proceedings of the Eleventh European Conference on Machine Learning* (pp. 220–228). Barcelona: Springer-Verlag.

Zelle, J. M., Mooney, R. J., & Konvisser, J. B. (1994). Combining top-down and bottom-up methods in inductive logic programming. *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 343–351).