Lexical and Grammatical Inference

Tom Armstrong and **Tim Oates**
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
1000 Hilltop Circle, Baltimore, Maryland 21250
{tom.armstrong, oates}@umbc.edu

Abstract

Children are facile at both discovering word boundaries and using those words to build higher-level structures in tandem. Current research treats lexical acquisition and grammar induction as two distinct tasks; doing so has led to unreasonable assumptions. State-of-the-art unsupervised results presuppose a perfectly segmented, noise-free lexicon, while largely ignoring how the lexicon is used. This paper combines both tasks in a novel framework for bootstrapping lexical acquisition and grammar induction.

Introduction

The ease with which children learn to discover boundaries in their environments while building grounded high-level structures belies the complexity and computational challenges of the task. We address these two disparate problems by proposing a bootstrap between lexical and grammatical knowledge. We improve lexical acquisition through the addition of a new dimension of information and remove a common assumption to all grammar induction algorithms.

Learning grammars is often an intractable problem unless concessions are made regarding the input, and having complete knowledge of the language's alphabet is a common assumption. Learning lexicons from noise-free data is also a challenge, and determining lexical items largely becomes a problem of finding a set of structure-less substrings. It is unrealistic from a developmental perspective to expect perfect information from noisy environments (e.g., child language acquisition, robot sensor data), but state-of-the-art approaches require it.

This paper explores the utility of including higher-level structural information in the unsupervised learning of a lexicon and removing the requirement that grammar induction algorithms have perfect, segmented input data. We discuss this learning task in terms of what we call the lexical-grammatical interface where the two tasks are bootstrapped together. In this bootstrap, lexical learning is segmenting sequences of categorical data into an inventory of subsequences based upon grammatical information, and grammar induction is taking segmented sequences of categorical data and building generative structures on top of the data. Learning grammars from segmented data is a hard problem, and learning lexicons from noise-free strings is a hard problem. An autonomous learner embedded in an environment must be able to acquire novel words and adapt existing structures. Our ultimate goal is to extend this work to real-valued sensor data where methods must be robust with respect to noise.

Lexical-Grammatical Interface

The lexical-grammatical interface is the interplay between the learning tasks of lexical acquisition and grammar induction (see figure 1). A typical lexicon learning algorithm begins with a stream of categorical data or a set of strings, and its goal is to induce an inventory of lexical items. A typical grammar induction algorithm begins with a set of strings, and its goal is to learn a generative structural model. While lexical learning is done without any regard for structural information, grammar induction assumes a known lexicon and correctly segmented input strings.

Bootstrapping in the Lexical-Grammatical Interface

In this section, we present a high-level sketch of a novel algorithm that operationalizes the bootstrap for lexical acquisition and grammar induction in the domain of regular grammar learning and string segmentation. The learner receives sequences of data from the environment in the form of phonemic transcriptions. The initial lexicon is the total inventory of the phonemes received (a black box segmentation algorithm can create an initial segmentation of the sequences). The segmented sequences serve as input to the grammar induction black box and grammar component. The grammar induction black box returns a grammar given in terms of the current lexicon. Up until this point, the process is a standard grammar-induction pipeline. The question is how to use learned grammatical structures to improve the segmentation and, in turn, improve the lexicon.

First, we use the learned machine to parse each input string (both positive and negative data). Next, we count the number of strings that pass through each pair of adjacent edges. Frequently traversed pairs of edges in an automaton are frequently occurring digrams in the input strings. However, since we have a higher-level grammatical struc-

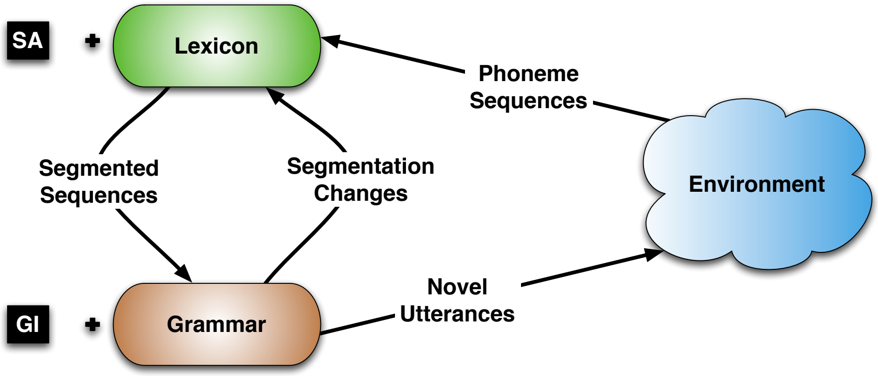

Figure 1: Learning in the Lexical-Grammatical Interface (SA and GI are segmentation and grammar learning black boxes selected based upon the type of sequences and class of languages, respectively)

ture, instead of indiscriminately merging digrams in strings, we have the condition that the same edges must be traversed for a merge to occur. Edge-pair selection proceeds greedily and the most frequent edge pair is used to resegment the positive and negative data. The bootstrap continues with the strings in terms of a new lexicon. This complete approach to grammar learning begins with an overly general grammar and proceeds through the space of possible automata to an overly specific grammar (i.e., one production for each string of positive data). The algorithm returns the grammar with the global minimum lexicon size.

We evaluated the bootstrap using natural language data and the state of the art in grammar induction. The grammar induction community has a series of benchmark languages for comparing learning algorithms: *L1* through *L15*[1] (Tomita 1982; Dupont 1994). Our unsupervised learning algorithm successfully discovers the lexicons and grammars of many test languages. The difficulties with our algorithm and framework result from cases that are challenges for grammar induction algorithms and lexical learning algorithms.

Related and Future Work

The seminal work of Gerry Wolff that has developed into the *compression as computing* paradigm (Wolff 1975) inspires this work. The approaches most similar to this work treat acquiring a lexicon as an unsupervised learning problem with a simplicity bias (Nevill-Manning & Witten 1997; Cohen, Heeringa, & Adams 2002; Brent 1999; Batchelder 2002).

Future work will proceed in three directions. First, we will focus on the theoretical boundaries of the lexical-grammatical bootstrap. That is, we will explore the classes of languages that are lexical- and grammatical-learnable in the Chomsky hierarchy. As the grammar induction component is a black box, learning algorithms for more complex languages can replace RPNI. These results will define a new class of learnable formal languages and will shed light on the learnability of natural languages.

Next, we will harness the generative power of the grammar and lexicon to create novel utterances. A learner embedded in an environment can then be used to experiment with language generation. This information can be used during edge-merge selection. The intuition is that while the automata's structures are different, the surface forms of the utterances are the same. Finally, we will extend our current results using categorical data to time series data and spoken natural language data.

References

Batchelder, E. O. 2002. Bootstrapping the lexicon: A computational model of infant speech segmentation. *Cognition* 83(2):167–202.

Brent, M. 1999. An efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning* 34:71–105.

Cohen, P. R.; Heeringa, B.; and Adams, N. M. 2002. Unsupervised segmentation of categorical time series into episodes. In *ICDM*, 99–106.

Dupont, P. 1994. Regular grammatical inference from positive and negative samples by genetic search: the gig method. In *ICGI '94: Proceedings of the Second International Colloquium on Grammatical Inference and Applications*, 236–245. London, UK: Springer-Verlag.

Nevill-Manning, C. G., and Witten, I. H. 1997. Identifying hierarchical structure in sequences: A linear-time algorithm. *Journal of Artificial Intelligence Research* 7:67.

Tomita, M. 1982. Dynamic construction of finite automata from examples using hill climbing. In *Proceedings of the 4th Annual Cognitive Science Conference*, 105–108.

Wolff, J. G. 1975. An algorithm for the segmentation of an artificial language analogue. *British Journal of Psychology* 66:79–90.

[1] Canonical deterministic finite automata and data are available from `http://www.irisa.fr/symbiose/people/coste/gi_benchs.html`