JMLR: Workshop and Conference Proceedings 21:129–132, 2012 The 11th ICGI

Learning Tree Adjoining Grammars from Structures and Strings

Christophe Costa Florêncio C.CostaFlorencio@uva.nl
University of Amsterdam, the Netherlands

Editors: Jeffrey Heinz, Colin de la Higuera and Tim Oates

Abstract

We investigate the learnability of certain subclasses of tree adjoining grammars (TAGs). TAGs are based on two tree-tree operations, and generate structures known as *derived trees*. The corresponding strings form a *mildly context-sensitive* language. We prove that even very constrained subclasses of TAGs are not learnable from structures (derived trees) or strings, demonstrating that this type of problem is far from trivial.[1] We also demonstrate that a large (parameterized) family of classes of TAGs is learnable from strings.

1. Introduction

The increasing availability of annotated corpora (such as the Penn Treebank) over the last two decades has stimulated interest in extraction of linguistic knowledge from such resources. Although pragmatic approaches to this problem have been moderately successful, a lot could be gained by examining it from a more fundamental, theoretic perspective. We show that even very restricted classes of TAGs are not learnable in this setting, and show that a known learnability result for certain graph grammars implies that a large parameterized family of learnable classes of TAGs exists.

Several variants of TAG are in use, and these are often presented in an informal fashion. Since we want to analyze learnability in a formal way (identifiability in the limit), we will use one particular formal definition from Vijay-Shanker and Weir (1994), the expressive power of which is known exactly. It is also a somewhat restricted version, making negative results more general.[2]

2. Negative results

2.1. TAGs with Empty String

Let yield(T) be the function that yields the string obtained by collating the leaves of derived tree T. We slightly abuse notation by defining this function in the natural way over tree languages and TAGs as well.

1. Less general versions of these negative results first appeared in Costa Florêncio (2003).
2. It is restricted in the sense that it is (strongly) lexicalized and does not use the substitution operation. In *lexicalized* TAG (LTAG), at least one terminal symbol (the *anchor*) appears at the frontier of all initial and auxiliary trees, this restricts the weak expressive power to *finitely ambiguous* TALs. The substitution operation can be emulated with adjoining, provided that ε is allowed as leaf. A normal form for TAG is also presented in the same paper, which relies heavily on the use of ε.

Consider the grammars G_n in Figure 1 and G_* in Figure 2. It is clear that $\text{yield}(G_n) = \mathtt{a}^{i+1}, 0 \leq i \leq n$ and $\text{yield}(G_*) = \mathtt{a}^+$. Thus for all $i \in \mathbb{N}, \text{yield}(G_i) \subset \text{yield}(G_{i+1})$, which constitutes an infinite ascending chain, and $\text{yield}(G_*) = \cup_{i\in\mathbb{N}}\text{yield}(G_i)$, which constitutes a limit point for this class. We thus obtain a non-learnability result for the case $\Sigma = \{a, \varepsilon\}$, 1 inital tree and at most 2 auxiliary trees.

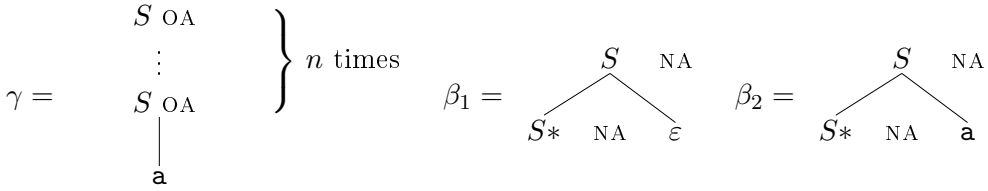

Figure 1: Grammar G_n.

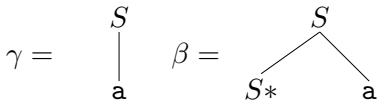

Figure 2: Grammar G_*.

2.2. The class of rigid TAGs is not learnable

Given existing learnability results for categorial grammar (also a lexicalized formalism) it would be reasonable to expectat that analogous results can be obtained for LTAGs. A result from Kanazawa (1998) states that the class of *rigid* CGs, that is, the class of grammars that assign exactly one lexical entry to each symbol in Σ, is learnable from structures.

At first glance, there seem to be two natural options for defining rigidity for TAGs; any $s \in \Sigma$ occurs at most once as leaf in $elem(\mathcal{G}_{\text{rigid}})$, or alternatively, any $s \in \Sigma$ occurs at most once as leaf in $init(\mathcal{G}^{\flat}_{\text{rigid}})$. Note that the latter is more permissive than the former, i.e., $\mathcal{L}_{\text{rigid}} \subset \mathcal{L}^{\flat}_{\text{rigid}}$.

Consider the set of grammars G_n defined in Figure 3. Any of the internal nodes of initial tree γ allows for adjunction with just the auxiliary tree β. Since foot and header of this tree are labeled NA, such an adjunction can only take place once for every internal node A, such an adjunction is not obligatory. Thus, for any grammar G_n, the number of adjunctions i for any derivation is bounded by $0 \leq i \leq n$. It is easy to see that this implies $\text{yield}(G_n) = \cup_{0\leq i\leq n}\mathtt{a}^i\mathtt{b}$, thus for all $i \in \mathbb{N}, \text{yield}(G_i) \subset \text{yield}(G_{i+1})$, therefore this set of grammars constitutes an infinite ascending chain.

To show existence of a limit point we need a grammar G_* such that $\text{yield}(G_*) = \cup_{i\in\mathbb{N}}G_i$. Such a grammar is shown in Figure 4, and differs from any G_n in that its one initial tree has no internal nodes, and its single auxiliary tree β has no NA-restrictions on its nodes. The number of adjunctions in a derivation is unbounded, and any of these (optional) adjunctions adds a symbol $\mathtt{a}$ as leaf of the derived tree, somewhere to the left of the rightmost leaf $\mathtt{b}$. Thus $\text{yield}(G_*) = \cup_{i\in\mathbb{N}}\mathtt{a}^i\mathtt{b} = \cup_{i\in\mathbb{N}}\text{yield}(G_i)$, and is therefore a limit point for the class.

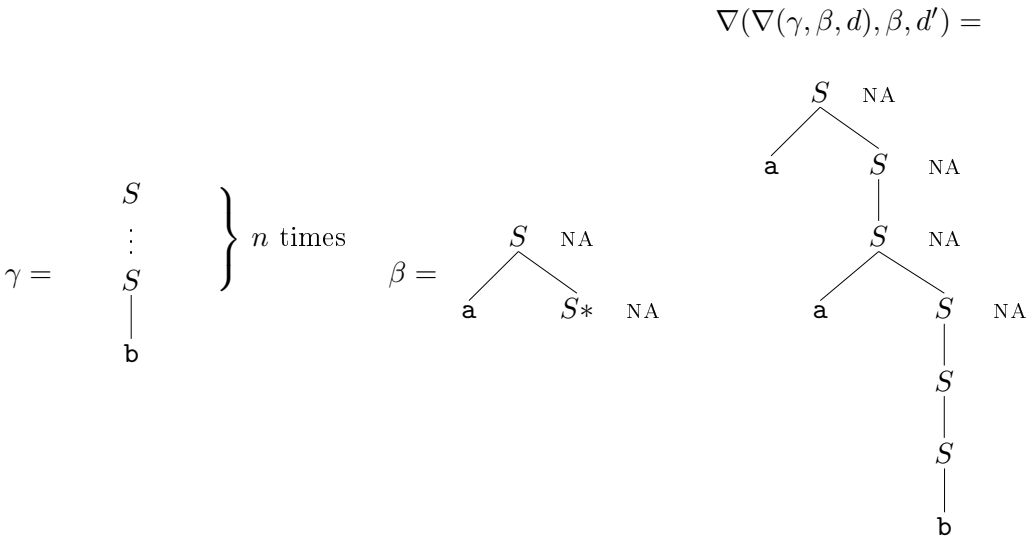

Figure 3: Grammar G_n with example derived tree for aab using grammar G_4.

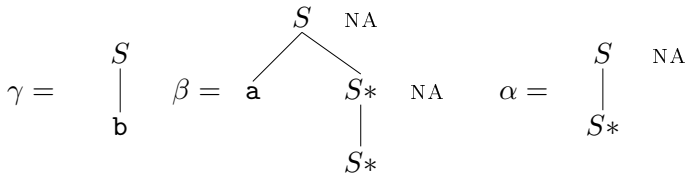

Figure 4: Grammar G_*. The derived tree from Figure 3 is generated by this grammar with $\nabla(\nabla(\nabla(\gamma,\beta,d),\beta,d'),\alpha,d'')$.

Thus, the class of rigid grammars with $|\Sigma| = 2$, $\varepsilon \notin \Sigma$, $|\gamma| = |\beta| = 1$, allowing unary and binary branchings, and allowing labeling with NA, has a limit point and is therefore not (non-effectively) learnable from strings. it is easy to see that we also obtain an infinite ascending chain and a limit point for the derived tree languages, so it follows that this class of grammars also is not (non-effectively) learnable from derived trees.

3. Positive result

In Costa Florêncio (2009), it was shown that certain restricted classes of Hyperedge-Replacement (HR) grammars are learnable from generated hypergraphs. It was shown that, for every constant k, the class of all (hyper)graph languages generated by HR grammars with k as an upper bound on the number of occurrences of any of the terminal labels[3] in the right hand sides has finite elasticity and is thus learnable. It is easy to see that TAG can be seen as a special case of HR grammar, and it is also obvious that every LTAG can

3. Note that, in HR grammars, these labels are applied to the edges instead of the nodes.

be expressed as an HR grammar with at least one terminal label in the right hand side of every rule.

We thus have that the class of derived tree languages generated by k-valued, ε-free LTAGs, without unary branching or control on adjunction, has finite elasticity. Since these languages are recursive, and this class is r.e., it follows from a theorem from Wright (1989) that this class is identifiable in the limit from structures.

Furthermore, a theorem from Kanazawa (1998) states that, given a class $\mathcal{M}$ of languages over Υ that has finite elasticity, and given a finite-valued relation $R \subseteq \Sigma^* \times \Upsilon^*$, $\mathcal{L} = \{R^{-1}[M] \mid M \in \mathcal{M}\}$ also has finite elasticity. Given that, in the case of ε-free LTAGs, the relation between a string language and the derived tree languages that have it as yield is finite-valued for our class (since unary branching is disallowed), finite elasticity of the corresponding class of string languages follows immediately. Again, from Wright's theorem it follows that this class is identifiable in the limit from strings.

4. Conclusion

Our results indicate that, although lexicalized formalisms tend to yield learnable classes when their descriptive complexity is bounded in a suitable way, positive results do not readily carry over from one formalism to the other. The inclusion of ε in the alphabet is especially problematic, since it complicates the relationship between a string and its possible derivations.

The proof of our positive result implies the existence of an enumerative learning algorithm, but given that such algorithms are generally inefficient, complexity issues remain open. Given our earlier complexity results for learnable classes of CG, we conjecture that a learner with polynomial update time exists for the subclass of our learnable class that is subject to the additional restriction of rigidity.

References

Christophe Costa Florêncio. *Learning Categorial Grammars*. PhD thesis, UiL OTS, Utrecht University, 2003.

Christophe Costa Florêncio. Identification of hyperedge-replacement graph grammars. In H. Blockeel, K. Borgwardt, and X. Yan, editors, *7th International Workshop on Mining and Learning with Graphs, Extended Abstracts*, pages 19–21, Leuven, Belgium, July 2–4 2009.

Makoto Kanazawa. *Learnable Classes of Categorial Grammars*. CSLI Publications, Stanford University, distributed by Cambridge University Press, 1998.

Krishnamurti Vijay-Shanker and David J. Weir. The equivalence of four extensions of context-free grammars. *Mathematical Systems Theory*, 27:511–546, 1994.

Keith Wright. Identification of unions of languages drawn from an identifiable class. In *The 1989 Workshop on Computational Learning Theory*, pages 328–333. San Mateo, Calif.: Morgan Kaufmann, 1989.