

Polynomial Tree Substitution Grammars: Characterization and New Examples

Jean-Cédric Chappelier, Martin Rajman and Antoine Rozenknop

EPFL

I&C–IIF–LIA, IN (Écublens)

CH-1015 Lausanne, Switzerland

{Jean-Cedric.Chappelier, Martin.Rajman, Antoine.Rozenknop}@epfl.ch

Mots-clefs – Keywords

Analyse Syntaxique Probabiliste, Grammaires à Substitution d’Arbres Polynomiales, Grammaires Hors-Contexte Probabilistes, Data-Oriented Parsing.

Stochastic Parsing, Polynomial Tree Substitution Grammars, Stochastic Context-Free Grammars, Data-Oriented Parsing.

Résumé - Abstract

Le but de ce papier est de caractériser (au moins partiellement) les *Grammaires à Substitution d’Arbres Polynomiales* (pSTSG), instances particulières de STSG pour lesquelles la recherche de l’analyse la plus probable peut être effectuée en un temps polynômial. Nous donnons tout d’abord diverses conditions suffisantes, utilisables en pratique, qui garantissent qu’une STSG est polynômiale. Une telle condition suffisante, fondée sur la notion de « tête de syntagme », est ensuite présentée et évaluée.

Polynomial Tree Substitution Grammars, a subclass of STSGs for which finding the most probable parse is no longer NP-hard but polynomial, are defined and characterized in terms of general properties on the elementary trees in the grammar. Various sufficient and easy to compute properties for a STSG to be polynomial are presented. The min-max selection principle is shown to be one such sufficient property. In addition, another, new, instance of a sufficient property, based on lexical heads, is presented. The performances of both models are evaluated on several corpora.

1 Motivations

Stochastic Tree Substitution Grammars (STSG), mainly used in the Data-Oriented Parsing (DOP) framework (Bod, 1998), are grammars the rules of which consist of syntactic trees, called “*elementary trees*”. These elementary trees are combined with the substitution operator¹ to give derivations of complete parse trees. In addition, a probability $p(t)$ is assigned to each elementary tree t of the grammar². These probabilities serve to compute the probabilities of parse trees. Although STSGs are equivalent to Context Free-Grammars (CFG) from a structural

¹denoted by “ \circ ”

²in such a way that the sum of the probabilities of elementary trees that have the same root node is 1.

point of view³, STSGs bring a clear advantage at the probabilistic level. They can capture a much larger set of probabilistic dependencies than the SCFGs, for which the probabilization is restricted to context-free (CF) rules (i.e. depth-1 elementary trees only).

However, STSGs suffer from one major drawback: finding the most probable parse tree (MPP) has been proved to be an NP-hard problem in the most general case (Sima'an, 1996). Various approximated MPP search algorithms have been developed (Bod, 1992; Goodman, 1996; Chappelier & Rajman, 2000), but another alternative, first introduced in (Chappelier & Rajman, 2001), is possible. This approach consists in choosing the set of elementary trees in the STSG in such a way that finding the MPP is no longer NP-hard but polynomial. More precisely, the idea underlying *Polynomial Tree Substitution Grammars* (pSTSG) is to restrict the elementary trees present in the grammar so as to make the MPP search equivalent to the search of a most probable derivation (MPD) with an SCFG that can be derived from the original STSG.

A trivial example of pSTSGs are the SCFGs themselves, which can be seen as STSGs where all the elementary trees are limited to depth-1 trees. A less trivial example of pSTSG, produced according to the “min-max selection principle”, was presented in (Chappelier & Rajman, 2001). In this example, the elementary trees are restricted to all depth-1 trees and all trees, the leaves of which are exclusively terminals (also called “complete trees”).

The goal of this contribution is to generalize this work by providing explicit necessary and sufficient conditions on the elementary trees of the STSG, so that finding the MPP can be achieved in polynomial time. We also present a new type of pSTSGs, which significantly differs from the grammars derived from the min-max selection principle.

The paper first provides a formal definition of pSTSGs and then presents necessary and sufficient conditions for a STSG to be polynomial. Next, various only sufficient but much easier to compute polynomiality conditions are analyzed. The min-max selection principle is shown to fulfill one such condition. Another instance of these sufficient conditions, based on lexical heads, is also presented. Finally, the performances of these two models are evaluated.

2 Formal Definition of pSTSGs

2.1 Tree Decompositions

To formalize the notion of pSTSG, we need to extend the notion of derivation to trees which are not necessarily parse trees. To do so, we introduce the more general notion of *tree decomposition*:

Definition 1 A partition of a tree T into elementary trees t_1, \dots, t_m of a given STSG \mathcal{G} , is called a decomposition of T (with respect to \mathcal{G}) and will be denoted by $\langle t_1, \dots, t_m \rangle_T$.⁴

Consider for instance the tree $T = \begin{array}{c} S \\ / \quad \backslash \\ S \quad S \\ | \\ a \end{array}$. One possible decomposition of T is $\langle t_1, t_2 \rangle_T$, where

$t_1 = \begin{array}{c} S \\ / \quad \backslash \\ S \quad S \end{array}$ and $t_2 = \begin{array}{c} S \\ | \\ a \end{array}$ (which are supposed to be elementary trees of the considered STSG).

³The same language is recognized and the same parse trees are produced for a given sentence.

⁴This notation, which is a sequence and not a set, is not at all ambiguous and has to be understood as the depth-first description of the corresponding partition of T .

Notice that $\langle t_1, t_2 \rangle_T$ is different from the derivation $t_1 \circ t_2 = \begin{array}{c} S \\ / \quad \backslash \\ S \quad S \\ | \quad | \\ a \end{array}$.

For any parse tree T however, there is a clear one-to-one and onto mapping between derivations and decompositions of T . Therefore, we will for parse trees indifferently use either the derivation or the decomposition point of view.

In the STSG framework, the probability of a derivation $t_1 \circ \dots \circ t_n$ is defined as $p(t_1 \circ \dots \circ t_n) = \prod_{t_i} p(t_i)$; and the probability of a full parse tree T , hereafter called “*parse-probability*” of T , is defined as the sum of the probabilities of all its derivations: $P(T) = \sum_{d \Rightarrow T} p(d) = \sum_{d \Rightarrow T} \prod_{t \in d} p(t)$, where the subscript “ $d \Rightarrow T$ ” means “for all derivations d leading to the parse tree T ”.⁵

Similarly, the probability of a decomposition $\langle t_1, \dots, t_m \rangle_T$ of a tree T is defined as $p(\langle t_1, \dots, t_m \rangle_T) = \prod_{t_i} p(t_i)$. The generalization of the parse-probability to a tree T that is not a parse tree (i.e. that does not result from a left-most derivation) is given by: $P(T) = \sum_{\delta \in \Delta(T)} p(\delta)$, where $\Delta(T)$ is the set of all possible decompositions of T into elementary trees of \mathcal{G} .

Finally, for any two decompositions δ and δ' of a tree T , δ is said to be *finer* than δ' if and only if (iff) every elementary tree appearing in δ is itself a subtree⁶ of some elementary tree appearing in δ' . This will be noted $\delta \leq \delta'$.

Notice that “ \leq ” induces a (partial) order on the set of the decompositions of a given tree. This allows us to define the notion of *maximal decomposition*:

Definition 2 A decomposition δ of a tree T is said to be maximal iff any other decomposition of T comparable with δ is finer than δ : $\forall \delta' \in \Delta(T), \delta \leq \delta' \implies \delta' = \delta$

2.2 Definition of pSTSGs

Let us now recall the general framework used for finding pSTSGs as it was originally described in (Chappelier & Rajman, 2001):

For any STSG \mathcal{G} , an SCFG $\mathcal{G}_{\text{equiv}}$ can be constructed in the following way: the rules of $\mathcal{G}_{\text{equiv}}$ are the root-leaves representations of the elementary trees of \mathcal{G} , and of these rules is associated with the **parse**-probability of the corresponding elementary tree.

In such a setup, the probability of a derivation in $\mathcal{G}_{\text{equiv}}$ is always less than or equal to the parse-probability of the corresponding parse tree in \mathcal{G} . Therefore, if for each parse tree T produced by \mathcal{G} there exists at least one derivation in $\mathcal{G}_{\text{equiv}}$, the probability of which is the parse-probability of T in \mathcal{G} , the grammar $\mathcal{G}_{\text{equiv}}$ can be used instead of \mathcal{G} for finding the MPP in polynomial time. In other words, the MPD search with $\mathcal{G}_{\text{equiv}}$ and the MPP search with \mathcal{G} become equivalent.

An STSG for which there exists a polynomial time MPP search algorithm is called a pSTSG. In addition, we call *effective pSTSG* a STSG for which an equivalent SCFG parsing scheme as described above can be build. Notice that this method cannot be applied to all STSGs, as it is

⁵With STSGs, a given parse tree can indeed have several different derivations, even with the “left-most non-terminal first” rewriting convention.

⁶including the tree itself

not always possible, in the most general case, to exhibit in polynomial time a derivation that holds the parse-probability of the parse tree it corresponds to.

Denoting by $\mathcal{E}(\mathcal{G})$ the set of elementary trees of \mathcal{G} and by $\mathcal{D}(\mathcal{G})$ the set of the parse trees generated by \mathcal{G} , let us now provide a more formal definition of effective pSTSGs:

Definition 3 A STSG \mathcal{G} is said to be effectively polynomial iff $\forall T \in \mathcal{D}(\mathcal{G}), \exists t_1, \dots, t_k \in \mathcal{E}(\mathcal{G})$ s.t. $T = t_1 \circ \dots \circ t_k$ and $P(T) = \prod_i P(t_i)$.

In other words, an STSG is effectively polynomial iff, for any parse tree, there exists at least one derivation such that the product of the parse-probabilities of its constituents is the parse-probability of the parse tree.

Notice that the product $\prod_i P(t_i)$ involved in the above definition, which corresponds to the probability of a derivation in $\mathcal{G}_{\text{equiv}}$, is different from $\prod_i p(t_i)$, the probability of the derivation $t_1 \circ \dots \circ t_k$ in \mathcal{G} : the former is the product of the parse-probabilities of the elementary trees, as defined in the former section, whereas the later is the product of the elementary probabilities of the elementary trees.

3 Conditions for Polynomiality

Theorem 1 A STSG is effectively polynomial iff any parse tree it can generate has a unique maximal derivation.

This characterization of effective pSTSG, requires to consider the set of all the parse trees that can be generated by the STSG (which very often is an infinite set). As such, it is difficult to apply in practice. For this reason, more practical (but also less general) conditions for polynomiality are now presented. To do so, we first introduce the notion of *atomic* grammars, for which we then provide a sufficient polynomiality condition, much easier to check in practice.

Definition 4 A grammar is said to be *atomic* iff any depth-1 subtree of any elementary tree is also an elementary tree.

In other words, a grammar is atomic if it contains at least all the CF rules that appear in its elementary trees.

Definition 5 For any proper subtree t' of a tree t , we define the expansion of t' in t as the set of leaves of t' which are not leaves of t .

We can now give a sufficient condition for atomic STSGs to be polynomial:

Theorem 2 An atomic STSG is polynomial if the expansions of any depth-1 elementary tree in any other elementary tree are all the same.

The above theorem can be used to define the following algorithm, aiming at the automated extraction of atomic pSTSGs from a tree-bank:

Algorithm 1

1. Extract the CFG from the tree-bank;
2. For each CF rule, define a unique expansion; and extract from the corpus all subtrees in which CF rules have this expansion and this expansion only.

4 Practical Examples of pSTSG

The min-max selection principle (Chappelier & Rajman, 2001) can now be reinterpreted as an extraction procedure leading to the atomic STSG for which each CF rule is associated with the expansion scheme consisting in always expanding all its (non-terminal) leaves. It is therefore a pSTSG by theorem 2.

Another possible way to automatically extract pSTSG from tree-banks using Algorithm 1 is to choose as systematic expansion scheme for depth-1 trees an expansion restricted to only one given non-terminal leaf. One way to implement this in an automatic way is to choose the lexical head as expansion node. For example, Collins (1999) defined a set of rules to automatically determine the lexical heads of the CF rules. For our experiments with Bod’s version of the ATIS treebank, we adapted Collins’ rules, which were developed for the Wall-Street Journal treebank, to this corpus.

The evaluation protocol consisted in computing an average performance on 25 runs of independent training/test splittings of the corpus. To have an upper bound on the results, the performance on the full corpus was also computed.

The results obtained are summarized in table 1. The head-driven expansion model appears to outperform the basic CF model, but is less performant than the min-max model (Chappelier & Rajman, 2001). One possible explanation could come from the number of parameters. For each of the three models (CF, min-max, head-driven) the number of elementary trees extracted from the whole ATIS corpus is given in table 2. This number is precisely the number of probabilistic parameters for each model. The fact that the min-max selection principle performs better than the head-driven approach on this corpus tends to show that there are enough training data for this model to accurately capture the probabilistic dependencies present in the corpus.

Another aspect to keep in mind is also that the corpus used is rather flat (trees are wide and not very deep) and that the lexical-head of a CF rule is, most of the time, a terminal node. This characteristic of the ATIS corpus implies that the head-driven expansion method does not produce so many new elementary trees (in addition to the CFG) and that most of the new elementary trees are of low depth.

Therefore, head-driven expansion approach to pSTSG should be explored further, on corpora where the notion of “lexical head” is more pertinent than in the ATIS corpus.

	coverage	precision of CF	precision of head-driven	precision of min-max
test 25%	98.5	38.5	39.7	45.5
test 10%	98.6	41.7	42.4	49.7
test 5%	98.2	44.0	45.2	49.8
self-test	100.0	51.0	54.9	88.5

TAB. 1 – Experimental results for CFG, head-driven expansion and min-max selection principle on Bod’s version of the ATIS corpus: percentage of exact match sentences in several test conditions is given. Notice that the coverage of the three models is, by construction, the same.

	CF	head-driven	min-max
nb. of el. trees	381	930	2434

TAB. 2 – Size of the grammar extracted from the whole Bod’s version of the ATIS corpus. For the three grammars, there are 12 non-terminals and 31 Part-of-Speech tags.

5 Conclusion

A complete characterization of pSTSGs has been provided and other, only sufficient but more effective, polynomiality conditions have been presented. These results extend the work previously done on the min-max selection principle for constructing pSTSG and open promising perspectives concerning the production of more general pSTSGs.

The head-driven expansion approach, one new example of a pSTSG has been presented and compared to the min-max selection principle. This new expansion scheme consists in systematically expand (at each level) the lexical head of the all subtrees of a treebank. Although its performance on the ATIS corpus is not as good as the one obtained with the min-max selection principle, this approach should be tested further, for instance on corpora where the notion of “lexical head” is more pertinent.

Another research direction opened by the theoretical results presented here consists in finding other expansion schema in order to construct other, more performant, instances of pSTSGs.

Acknowledgments The authors would like to thank Rens Bod for providing his cleaned-up version of the ATIS corpus, Soham Mazumdar and Evi Commins for their help in the experiments.

Références

- BOD R. (1992). Applying Monte Carlo techniques to Data Oriented Parsing. In *Proceedings Computational Linguistics in the Netherlands*, Tilburg (The Netherlands).
- BOD R. (1998). *Beyond Grammar, An Experience-Based Theory of Language*. Number 88 in CSLI Lecture Notes. Stanford (CA): CSLI Publications.
- CHAPPELIER J.-C. & RAJMAN M. (2000). Monte-Carlo sampling for NP-hard maximization problems in the framework of weighted parsing. In D. CHRISTODOULAKIS, Ed., *Natural Language Processing – NLP 2000*, number 1835 in Lecture Notes in Artificial Intelligence, p. 106–117. Springer.
- CHAPPELIER J.-C. & RAJMAN M. (2001). Polynomial tree-substitution grammars: an efficient framework for Data-Oriented Parsing. In *Proc. of Recent Advances in Natural Language Processing (RANLP 2001)*, p. 65–71.
- COLLINS M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania.
- GOODMAN J. (1996). Efficient algorithms for parsing the DOP model. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, p. 143–152.
- SIMA’AN K. (1996). Computational complexity of probabilistic disambiguation by means of tree grammars. In *Proceedings of COLING’96*, Copenhagen (Denmark). cmp-lg/9606019.