Mildly Context-Sensitive Grammar Formalisms:

Tree Substitution Grammars

Laura Kallmeyer
Heinrich-Heine-Universität Düsseldorf

Sommersemester 2011

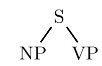
Overview

1. Motivation
2. Tree Substitution Grammars
3. Equivalence of CFG and TSG
4. Applications

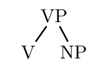
Motivation

- In a CFG, the elements in the grammars represent very small syntactic trees.

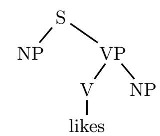
S → NP VP



VP → V NP



- From a linguistic point of view, in particular in a lexicalized grammar, we would like entire constructions to be our elementary building blocks.



Tree Substitution Grammars (1)

This leads to the definition of Tree Substitution Grammars (TSG).

- A TSG consists of a set of syntactic trees.
- From these trees, larger trees can be built by replacing a non-terminal leaf with a new tree whose root node is labeled with the same non-terminal.
- This operation is called substitution.

Tree Substitution Grammars (2)

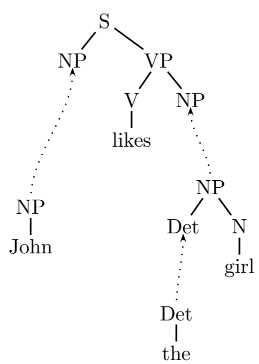
Definition 1 (Substitution)

Let $\gamma = \langle V, E, r\rangle$ and $\gamma' = \langle V', E', r'\rangle$ be syntactic trees with $V \cap V' = \emptyset$ and $v \in V$. $\gamma[v, \gamma']$, the result of substituting γ' into γ at node v is defined as follows:

- *if v is not a leaf or $l(v) \neq l(r')$, then $\gamma[v, \gamma']$ is undefined;*
- *otherwise, $\gamma[v, \gamma'] = \langle V'', E'', r''\rangle$ with $V'' = V \cup V' \setminus \{v\}$ and $E'' = (E \setminus \{\langle v_1, v_2\rangle \mid v_2 = v\}) \cup E' \cup \{\langle v_1, r'\rangle \mid \langle v_1, v\rangle \in E\}$. Furthermore, $v_1 \prec v_2$ in $\gamma[v, \gamma']$ iff either $v_1 \prec v_2$ in γ or $v_1 \prec v_2$ in γ' or $v_1 \in V'$ and $v \prec v_2$ in γ or $v_2 \in V'$ and $v_1 \prec v$ in γ.*

A leaf that has a non-terminal label is called a substitution node.

Tree Substitution Grammars (3)



Tree Substitution Grammars (4)

Definition 2 (Tree Substitution Grammar)

A Tree Substitution Grammar (TSG) is a tuple $G = \langle N, T, S, I\rangle$ where

- *N, T are disjoint alphabets of non-terminal and terminal symbols,*
- *$S \in N$ is the start symbol,*
- *I is a finite set of syntactic trees with labels from N and T.*

Every tree in I is called an elementary tree.

G is called lexicalized if every tree in I has at least one leaf with a label from T.

Tree Substitution Grammars (5)

For a syntactic tree $\gamma = \langle V, E, r\rangle$ with node labeling functions l, we call $\langle V', E', r'\rangle$ with labeling functions l' an instance of γ if there exists a bijective function $h : V \to V'$ such that

- for all $v_1, v_2 \in V$: $\langle v_1, v_2\rangle \in E$ iff $\langle h(v_1), h(v_2)\rangle \in E'$;
- for all $v_1, v_2 \in V$: $v_1 \prec v_2$ in γ iff $h(v_1) \prec h(v_2)$ in γ';
- for all $v \in V$: $l(v) = l'(h(v))$;

In other words, the two trees are isomorphic.

Tree Substitution Grammars (6)

In a derivation step, we select a node with a non-terminal label A, we pick a fresh instance of an elementary tree with root label A from the grammar and we substitute the node for the new tree.

Definition 3 (TSG derivation)

Let $G = \langle N, T, S, I\rangle$ be a TSG.

1. *Let $\gamma = \langle V, E, r\rangle$ and γ' be syntactic trees.*
   γ' can be derived from γ in a single step, $\gamma \Rightarrow \gamma'$ if there is a node $v \in V$ and there is an instance $\gamma_e = \langle V_e, E_e, r_e\rangle$ of a tree from I such that
   - *$V \cap V_e = \emptyset$ (i.e., the node sets are disjoint),*
   - *$\gamma' = \gamma[v, \gamma_e]$ (i.e., γ' is the result of substituting v for γ_e).*
2. *$\overset{*}{\Rightarrow}$ is as usual the reflexive transitive closure of $\Rightarrow$.*

Tree Substitution Grammars (7)

Definition 4 (TSG language)

Let $G = \langle N, T, S, I\rangle$ be a TSG.

1. *We call a tree γ that can be derived from an instance of an elementary tree $\gamma_e \in I$ a derived tree in G.*
2. *The tree language of G is the set of all derived trees $\gamma = \langle V, E, r\rangle$ in G such that*
   - *$l(r) = S$, and*
   - *$l(v) \in T \cup \{\varepsilon\}$ for every leaf $v \in V$.*
3. *For every tree γ with $v_1, \ldots, v_n$ being the leaves in γ ordered form left to right, we define $yield(\gamma) = l(v_1) \cdots l(v_n)$.*
4. *The string language of G is $\{w \mid$ there is a $\gamma \in L_T(G)$ such that $w = yield(\gamma)\}$.*

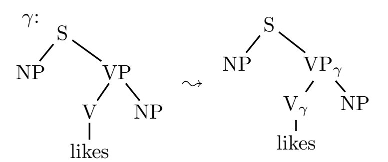
Equivalence of TSGs and CFGs (1)

In spite of the larger domains of locality, the following holds:

Proposition 1 (Equivalence of CFG and TSG) *CFG and TSG are weakly equivalent. Furthermore, except for some relabeling of the nodes, they are even strongly equivalent.*

1. Every CFG can be immediately written as a TSG with every production being understood as a tree with a single root and a daughter for every righthand side symbol
2. In order to construct an equivalent CFG for a given TSG, we have to encode the dependencies between nodes from the same tree within the non-terminal symbols.

Equivalence of TSGs and CFGs (2)



$S \rightarrow NP\ VP_\gamma$

$VP_\gamma \rightarrow V_\gamma\ NP$

$V_\gamma \rightarrow$ likes

Applications

Even though TSGs are almost strongly equivalent to CFGs, they offer an extended domain of locality. This enables them to capture more generalizations than CFGs do.

- TSGs are used in the context of data-oriented parsing (DOP) [Bod et al., 2003].
- Lexicalized TSGs can be extracted from treebanks and used for probabilistic parsing [Post and Gildea, 2009].
- [Cohn et al., 2009] also induce Probabilistic Tree Substitution Grammars from treebanks and use them successfully for parsing.

References

[Bod et al., 2003] Bod, R., Scha, R., and Sima'a, K., editors (2003). *Data-Oriented Parsing*. CSLI Publications, Stanford.

[Cohn et al., 2009] Cohn, T., Goldwater, S., and Blunsom, P. (2009). Inducing compact but accurate tree-substitution grammars. In *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL*, pages 548–556, Boulder, Colorado.

[Post and Gildea, 2009] Post, M. and Gildea, D. (2009). Bayesian learning of a tree substitution grammar. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 45–48, Singapore.