Parsing Beyond Context-Free Grammars:

Tree Adjoining Grammars

Laura Kallmeyer, Patrick Hommers
Heinrich-Heine-Universität Düsseldorf

Sommersemester 2013

Overview

1. Adjunction and substitution
2. Tree Adjoining Grammar
3. Adjunction constraints
4. Derivation trees
5. TAG and natural languages

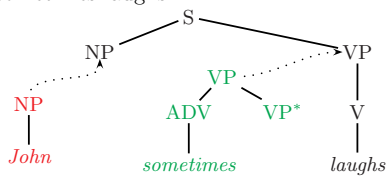
Adjunction and substitution (1)

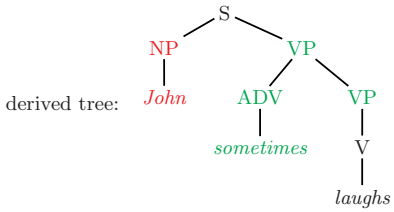
Tree Adjoining Grammars (TAG)
[Joshi et al., 1975, Joshi and Schabes, 1997]: Tree-rewriting system: set of *elementary* trees with two operations:

- adjunction: replacing an internal node with a new tree. The new tree is an auxiliary tree and has a special leaf, the foot node.
- substitution: replacing a leaf with a new tree. The new tree is an initial tree

Adjunction and substitution (2)

(1) John sometimes laughs





Adjunction and substitution (3)

Definition 1 (Auxiliary and initial trees)

1. *An auxiliary tree is a syntactic tree $\langle V, E, r\rangle$ such that there is a unique leaf f marked as foot node with $l(r) = l(f)$. We write this tree as $\langle V, E, r, f\rangle$.*
2. *An initial tree is a non-auxiliary syntactic tree.*

As a convention, the foot node is marked with a "*".

Adjunction and substitution (4)

Definition 2 (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.

Adjunction and substitution (5)

Definition 3 (Adjunction)

Let $\gamma = \langle V, E, r\rangle$ be a syntactic tree, $\gamma' = \langle V', E', r', f\rangle$ an auxiliary tree and $v \in V$. $\gamma[v, \gamma']$, the result of adjoining *γ' into γ at node v is defined as follows:*

- *if $l(v) \neq l(r')$, then $\gamma[v, \gamma']$ is undefined;*
- *otherwise, $\gamma[v, \gamma'] = \langle V'', E'', r''\rangle$ with $V'' = V \cup V' \setminus \{v\}$ and $E'' = (E \setminus \{\langle v_1, v_2\rangle \mid v_1 = v \text{ or } v_2 = v\}) \cup E' \cup \{\langle v_1, r'\rangle \mid \langle v_1, v\rangle \in E\} \cup \{\langle f, v_2\rangle \mid \langle v, v_2\rangle \in E\}$.*

Tree Adjoining Grammar (1)

Definition 4 (Tree Adjoining Grammar) *A Tree Adjoining Grammar (TAG) is a tuple $G = \langle N, T, S, I, A\rangle$ such that*

- *T and N are disjoint alphabets, the terminals and nonterminals,*
- *$S \in N$ is the start symbol,*
- *I is a finite set of initial trees, and*
- *A is a finite set of auxiliary trees.*

The trees in $I \cup A$ are called elementary trees.

G is lexicalized iff each elementary tree has at least one leaf with a terminal label.

Tree Adjoining Grammar (2)

- Every elementary tree is considered a derived tree in a TAG.
- In every derivation step, we pick a fresh instance of an elementary tree from the grammar and we add derived trees (by substitution or adjunction) to certain nodes in this tree.
- The trees in the tree language are the derived initial trees with root label S.

Tree Adjoining Grammar (3)

Definition 5 (Derived tree)

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

1. *Every instance γ of a $\gamma_e \in I \cup A$ is a derived tree in G.*
2. *For pairwise disjoint $\gamma_1, \ldots, \gamma_n, \gamma$ such that $\gamma_1, \ldots, \gamma_n$ are derived trees in G ($1 \leq i \leq n$) and $\gamma = \langle V, E, r\rangle$ is an instance of a $\gamma_e \in I \cup A$ such that $v_1, \ldots, v_n \in V$ are pairwise different: if $\gamma' = \gamma[v_1, \gamma_1] \ldots [v_n, \gamma_n]$ is defined then γ' is a derived tree in G.*
3. *These are all derived trees in G.*

Note that this definition adopts a bottom-up perspective: derived trees are added to elementary trees.

Tree Adjoining Grammar (4)

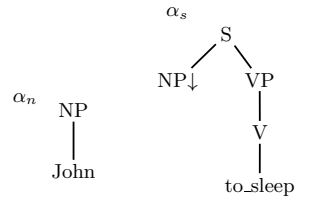
Definition 6 (TAG language)

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

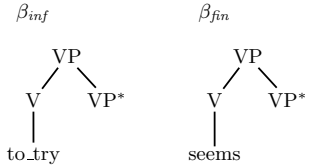
1. *The tree language of G is the set of all derived trees $\gamma = \langle V, E, r\rangle$ in G such that*
   - *$l(r) = S$, and*
   - *$l(v) \in T \cup \{\varepsilon\}$ for every leaf $v \in V$.*
2. *The string language of G is $\{w \mid$ there is a $\gamma \in L_T(G)$ such that $w = yield(\gamma)\}$.*

Tree Adjoining Grammar (5)

Initial trees:



Auxiliary trees:



Adjunction constraints (1)

TAGs as defined above are more powerful than CFG but they cannot generate the copy language.

In order to increase the expressive power, adjunction constraints are introduced that specify for each node

1. whether adjunction is mandatory and
2. which trees can be adjoined.

For a given node,

1. the function f_{OA} specifies whether adjunction is obligatory (value 1) or not (value 0) and
2. the function f_{SA} gives the set of auxiliary trees that can be adjoined.

Adjunction constraints (2)

Definition 7 (TAG with adjunction constraints) *A TAG with adjunction constraints is a tuple* $G = \langle N, T, S, I, A, f_{OA}, f_{SA}\rangle$ *where*

- $\langle N, T, S, I, A\rangle$ *is a TAG as defined above and*
- $f_{OA} : \{v \mid v \text{ is a node in some } \gamma \in I \cup A\} \to \{0, 1\}$ *and*
  $f_{SA} : \{v \mid v \text{ is a node in some } \gamma \in I \cup A\} \to \mathcal{P}(A)$
  where $\mathcal{P}(A)$ *is the set of subsets of* A *are functions such that* $f_{OA}(v) = 0$ *and* $f_{SA}(v) = \emptyset$ *for every leaf* v.

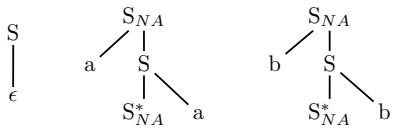
Adjunction constraints (3)

Three types of constraints are distinguished:

- A node v with $f_{OA}(v) = 1$ is said to carry a obligatory adjunction (OA) constraint.
- A node v with $f_{OA}(v) = 0$ and $f_{SA}(v) = \emptyset$ is said to carry a null adjunction (NA) constraint.
- A node v with $f_{OA}(v) = 0$ and $f_{SA}(v) \neq \emptyset$ and $f_{SA}(v) \neq A$ is said to carry a selective adjunction (SA) constraint.

Adjunction constraints (4)

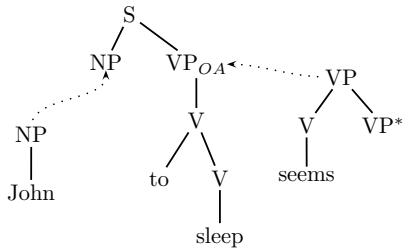
Example: TAG for the copy language:



Adjunction constraints (5)

Example:

(2) John seems to sleep



Adjunction constraints (6)

Definition 8 (Derived tree)

Let $G = \langle N, T, S, I, A, f_{OA}, f_{SA}\rangle$ be a TAG with adjunction constraints.

1. *Every instance of a $\gamma_e \in I \cup A$ is a derived tree obtained from γ_e in G.*
2. *For pairwise disjoint $\gamma_1, \ldots, \gamma_n, \gamma$ such that a) $\gamma_1, \ldots, \gamma_n$ are derived trees obtained from $\gamma_1^e, \ldots, \gamma_n^e$ in G respectively, and b) $\gamma = \langle V, E, r\rangle$ is an instance of a $\gamma_e \in I \cup A$ such that $v_1, \ldots, v_n \in V$ are pairwise different nodes: If*
   - *$\gamma' = \gamma[v_1, \gamma_1] \ldots [v_n, \gamma_n]$ is defined, and*
   - *$\gamma_i^e \in f_{SA}(v_i)$ for all $1 \leq i \leq n$ with γ_i^e an auxiliary tree*

   then γ' is a derived tree obtained from γ_e in G
3. *These are all derived trees in G.*

Adjunction constraints (7)

Definition 9 (Tree language)

Let $G = \langle N, T, S, I, A, f_{OA}, f_{SA}\rangle$ be a TAG with adjunction constraints.

The tree language of G is the set of all derived trees $\gamma = \langle V, E, r\rangle$ in G such that

- *$l(r) = S$,*
- *$f_{OA}(v) = 0$ for all $v \in V$, and*
- *$l(v) \in T \cup \{\varepsilon\}$ for every leaf $v \in V$.*

In the following, whenever we use the term "TAG", this means "TAG with adjunction constraints".

Derivation trees (1)

TAG derivations are described by derivation trees:

For each derivation in a TAG there is a corresponding derivation tree. This tree contains

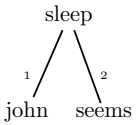
- nodes for all elementary trees used in the derivation, and
- edges for all adjunctions and substitutions performed throughout the derivation.

Whenever an elementary tree γ was attached to the node at address p in the elementary tree γ', there is an edge from γ' to γ labeled with p.

Derivation trees (2)

Example:

derivation tree for the derivation of (2) *John seems to sleep*



TAG and natural languages (1)

Important features of TAG when used for natural languages:

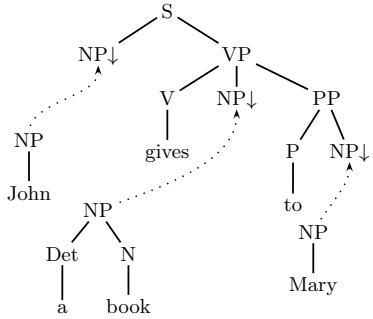
- Grammar is lexicalized
- Recursive parts are put into separate elementary trees that can be adjoined (Factoring of recursion, FR)
- The elementary tree of a lexical predicate contains slots for all arguments of the predicate, for nothing more (Elementary tree minimality).
- Elementary trees can be arbitrarily large, in particular (because of FR) they can contain elements that are far apart in the final derived tree (Extended domain of locality)

For natural language syntax and TAG see
[Kroch, 1987, Abeillé, 1988, Abeillé, 2002, Frank, 2002, XTAG Research Group, 2001].

TAG and natural languages (2)

Example:

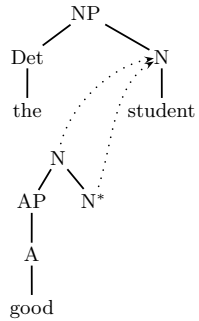
(3) John gives a book to Mary



TAG and natural languages (3)

Example for modification:

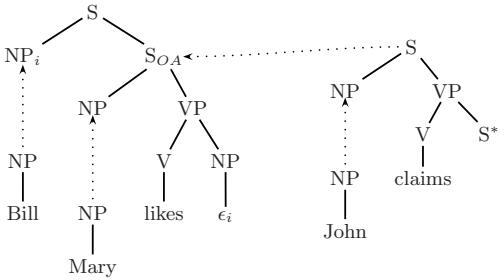
(4) the good student



TAG and natural languages (4)

Example of a long-distance dependency:

(5) a. Bill_i John claims Mary likes t_i
b. Bill_i John claims Susan thinks Mary likes t_i



TAG and natural languages (5)

For natural languages, we usually use Feature-structure based TAG (FTAG), [Vijay-Shanker and Joshi, 1988].

Each node has a top and a bottom feature structure. Nodes in the same elementary tree can share features (extended domain of locality).

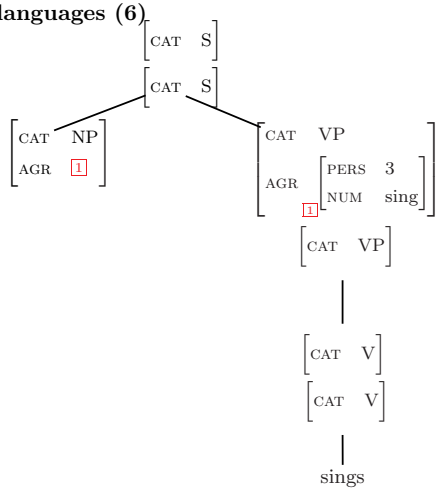
Intuition:

- The top feature structure tells us something about what the node presents within the surrounding structure, and
- the bottom feature structure tells us something about what the tree below the node represents.

In the final derived tree, both must be the same.

TAG and natural languages (6)

Example:



References

[Abeillé, 1988] Abeillé, A. (1988). Parsing French with tree adjoining grammar: some linguistic accounts. In *Proceedings of COLING*, pages 7–12, Budapest.

[Abeillé, 2002] Abeillé, A. (2002). *Une Grammaire Électronique du Français*. CNRS Editions, Paris.

[Frank, 2002] Frank, R. (2002). *Phrase Structure Composition and Syntactic Dependencies*. MIT Press, Cambridge, Mass.

[Joshi et al., 1975] Joshi, A. K., Levy, L. S., and Takahashi, M. (1975). Tree Adjunct Grammars. *Journal of Computer and System Science*, 10:136–163.

[Joshi and Schabes, 1997] Joshi, A. K. and Schabes, Y. (1997). Tree-Adjoining Grammars. In Rozenberg, G. and Salomaa, A., editors, *Handbook of Formal Languages*, pages 69–123. Springer,

Berlin.

[Kroch, 1987] Kroch, A. S. (1987). Unbounded dependencies and subjacency in a Tree Adjoining Grammar. In Manaster-Ramer, A., editor, *Mathematics of Language*, pages 143–172. John Benjamins, Amsterdam.

[Vijay-Shanker and Joshi, 1988] Vijay-Shanker, K. and Joshi, A. K. (1988). Feature structures based tree adjoining grammar. In *Proceedings of COLING*, pages 714–719, Budapest.

[XTAG Research Group, 2001] XTAG Research Group (2001). A Lexicalized Tree Adjoining Grammar for English. Technical report, Institute for Research in Cognitive Science, Philadelphia. Available from `ftp://ftp.cis.upenn.edu/pub/xtag/release-2.24.2001/tech-report.pdf`.