References:

Aravind Joshi & Yves Schabes, "Tree-Adjoining Grammars"

Anthony Kroch & Aravind Joshi, "The Linguistic Relevance of TAG"

Aravind Joshi & Yves Schabes, "Tree-Adjoining Grammars"

References:

February 3, 2004

Alon Lavie

Tree-Adjoining Grammars

I1-72 Grammar Formalisms
Tree-Adjoining Grammars (TAGs)

Originally proposed by Joshi around 1975, TAGs are a linguistically-motivated extension of CFGs. They extend the domain of locality from a single rule to a tree structure, separate recursion from domain of dependencies, and allow lexicalization: useful for linguistic and practical reasons. These two properties of TAGs can simplify linguistic description by localizing important linguistic constraints and dependencies. The elementary objects manipulated by TAGs are trees rather than rules.

Conceived as a linguistically-motivated extension of CFGs, TAGs are seen as an extension of the syntactic tree formalism, originally proposed by Joshi around 1975.
Extended Domain of Locality and Recursion

\[ S \rightarrow NP \rightarrow VP \rightarrow V \rightarrow NP \rightarrow \text{likes} \]

\[ NP \rightarrow NP \rightarrow VP \rightarrow ADV \rightarrow \text{passionately} \]

\[ \text{Harry peanuts passionately} \]

Fig. 8.2. Elementary Trees for a TAG, G1.
A TAG consists of a quintuple of elements:

\[
(S, \mathcal{I}, \mathcal{N}, \mathcal{T}, \mathcal{A})
\]

- \(S\): a designated start symbol
- \(\mathcal{N}\): a finite set of non-terminal symbols
- \(\mathcal{T}\): a finite set of terminal symbols
- \(\mathcal{I}\): a finite set of initial trees
- \(\mathcal{A}\): a finite set of auxiliary trees

Trees in \(\mathcal{I} \cup \mathcal{A}\) are called elementary trees.

A tree built by combining two other trees is called a derived tree.

A tree whose root is labeled by \(X\) is called an \(X\)-type tree.

Derived trees are built using two composition operations: 

- Adjoining
- Substitution

Formal Definition of TAGs
Properties

- Initial Trees
Initial Trees

\[
\begin{align*}
\text{Ik}\text{ik} & \quad \text{likes} \\
\text{N} & \quad \text{N} \\
\text{NP} & \quad \text{NP}
\end{align*}
\]

\[
\text{S} \quad \text{VP} \quad \text{NP}
\]
The label of the root must be the same as the label of the root node.

(annotated with an asterisk (*))

Nodes on the frontier labeled by non-terminal symbols are marked for substitution except one, called the foot node.

- Non-terminals from $\mathcal{L}$

- Nodes on the frontier can be labeled with terminal symbols from $\mathcal{Z}$

- Interior nodes are labeled with non-terminal symbols from $\mathcal{L}$

Properties •

Auxiliary Trees
Elementary Trees

Auxiliary Tree

Initial Trees

Grammar Formalisms
Adjoining constructs a new tree from an auxiliary tree and another.

Let $\mathcal{G}$ be an auxiliary tree whose root node is labeled by $X$.

Let $u$ be a tree that contains a node $n$ labeled by $X$ that is not marked for substitution.

Adjoining is not allowed on nodes marked for substitution.

Adjoining constructs a new tree from an auxiliary tree $\mathcal{G}$ and another.

For substitution:

$\exists \mathcal{G}$ is attached to the root node of $\mathcal{G}$.

$\exists \mathcal{G}$ is attached to $\mathcal{G}$ in place of $\mathcal{G}$.

The subtree of $\mathcal{G}$ rooted at node $n$ is removed from $\mathcal{G}$.

$\exists \mathcal{G}$ is attached to $\mathcal{G}$ at node $u$ to create a new tree as follows:
Auxiliary Trees and Adjunction
Constraint on Adjunction

- Selective Adjunction

- Obligatory Adjunction

- Null Adjunction

The following constraints can be specified on nodes in an elementary tree:

- Selective Adjunction

- Null Adjunction

- Obligatory Adjunction

- Selective Adjunction
Substitution constructs a new tree from an initial tree $1$ and another tree $2$. $\Rightarrow$ substitution is not necessary in the "pure" TAG formalism but is useful for linguistic reasons: a natural operation for attaching mandatory arguments.

- A new tree $\gamma$ is created by replacing $u$ with $c_1$.
- Let $c_2$ be a tree with frontier node $u$ which is marked for substitution and labeled by $X$.
- Let $c_1$ be an initial tree whose root is labeled by $X$ and marked for substitution.
- Substitution is only allowed on the frontier of a tree at non-terminal nodes.
- $c_2$ can be any elementary or derived tree.
- $c_1$ must be an initial tree or a tree derived from an initial tree.
- Substitution constructs a new tree from an initial tree $c_1$ and another tree $c_2$. 

Substitution
Substitution

\[ \alpha \]

\[
S \rightarrow NP \rightarrow N \rightarrow \text{likes} \rightarrow VP \rightarrow NP \rightarrow N \rightarrow \text{likes} \rightarrow NP \rightarrow N \rightarrow S'
\]
Adjoining and Substitution
Derivation Trees

- In a TAG, a derived tree does not contain enough information to determine how it was constructed.
- A derivation tree is necessary to uniquely specify how a derived tree was constructed.
- The root of a derivation tree is labeled by an S-type initial tree.
- All other nodes in a derivation tree are labeled by an auxiliary tree (for adjunction) or an initial tree (for substitution).
- Each node (except the root) is also associated with the address of the node in the parent tree where the composition operation took place.
- Trees adjoined to their parent tree are linked by a solid line.
- Trees substituted into their parent tree are linked by a dashed line.
Derivation Example

Initial Trees

Auxiliary Tree
Derivation Example

\[
\alpha \text{NXN} \{\text{pasta}\} (2.2) \quad \beta \text{vXAR} \{\text{quickly}\} (1) \quad \gamma \text{NXN} \{\text{Maya}\} (1)
\]

Tree: derived-tree-130118

---

Derivation Tree
Derivation Example
Derivation Example: Compositional
Derivation Example: Non-compositional