Syntactic Theory

Tree-Adjoining Grammar (TAG)

Yi Zhang

Department of Computational Linguistics
Saarland University

November 5th, 2009

What you should have known so far ...

- Phrase structure grammars
  - Context-free grammar (CFG)
- Dependency grammar

What you should have known so far . . .

- Phrase structure grammars
  - Context-free grammar (CFG)
- Dependency grammar

Outline

Outline

Tree-Adjoining Grammar

- Describing natural language syntax in CFG is not aways effective/possible
- Comparing to CFG, TAG is an extended formalism
  - Basic elements in TAG are trees, instead of atomic symbols
  - TAG is a *tree-rewriting* (instead of *strings rewriting*) system
  - TAG is *mildly context-sensitive*
- A lexically-oriented formalism (especially the lexicalized tree adjoining grammar (LTAG))

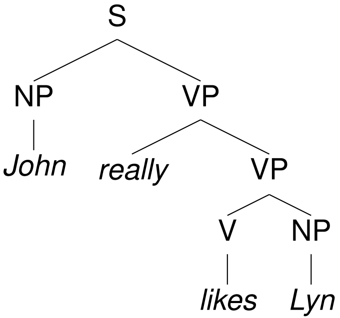
A Brief Review of the History and Variants of TAG

- Originally developed by Aravind Joshi (1975)
- Lexicalized Tree-Adjoining Grammar (LTAG)
- Synchronous TAG (STAG)
- Multi-component TAG (MCTAG)

Outline

Phrase Structure Tree & CFG

1. $S \rightarrow NP\ VP$
2. $VP \rightarrow really\ VP$
3. $VP \rightarrow V\ NP$
4. $V \rightarrow likes$
5. $NP \rightarrow John$
6. $NP \rightarrow Lyn$



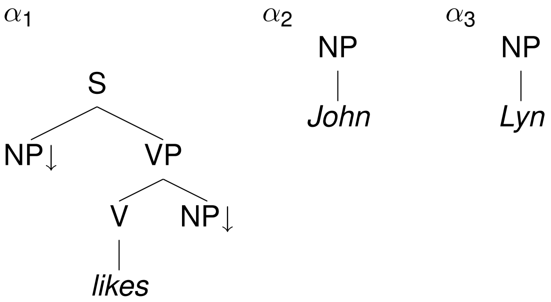
- The locality of each rule is limited to one level of branching in the tree
- PS tree directly reflects the derivation steps of the CFG

Limitations of CFG as Linguistic Formalism

- Limited locality makes it difficult to describe (even slightly) non-local linguistic phenomena
- Although it is possible to extend the CFG with complex categories (e.g. via lexicalization), the grammar soon gets "ugly"

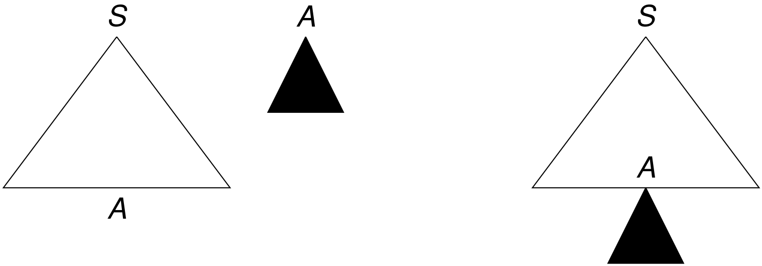
Tree-Substitution Grammar

- Elementary structures are phrase structure trees
- A downward arrow (↓) indicates where a substitution takes place

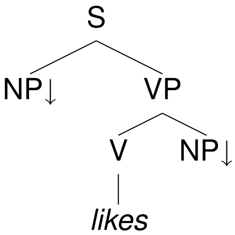


Substitution Operation

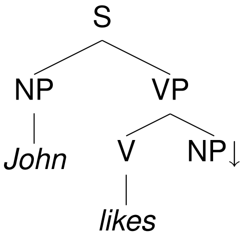
- The substitution operation allows one to insert elementary trees into other elementary trees
- Where there is a node marked for substitution (↓) on the **frontier**, an elementary tree rooted in the same category can be substituted there



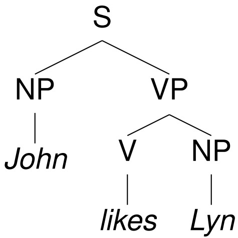
Substitutions & Derived Tree



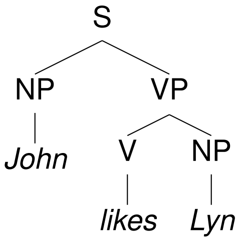
Substitutions & Derived Tree



Substitutions & Derived Tree



Substitutions & Derived Tree



- A (completely) **derived tree** has no more substitution nodes on the frontier
- The order of substitutions is irrelevant

Elementary Trees

Elementary trees are the building blocks of `TSG` and `TAG`
For `TSG`, all the elementary trees are so-called **initial trees**, which are characterized as followings:

- interior nodes labeled by non-terminal symbols
- frontier nodes labeled by terminal and non-terminal symbols
- non-terminal nodes on the frontier of the initial tree are marked for substitution (and conventionally noted with $\downarrow$)

Tree-Substitution Grammar: Formal Definition

- A **Tree-Substitution Grammar (TSG)** is a quadruple (Σ, NT, I, S), where
  1. Σ is a finite set of terminal symbols
  2. NT is a finite set of non-terminal symbols: $\Sigma \cap NT = \Phi$
  3. S is a distinguished non-terminal symbol: $S \in NT$
  4. I is a finite set of initial trees

Lexicalization

- A grammar is **"lexicalized"** if it consists of:
  - a finite set of structures each associated with a lexical item; each lexical item will be called the *anchor* of the corresponding structure
  - an operation or operations for composing the structures

Theorem

Lexicalized grammars are finitely ambiguous

- We say a formalism $\mathscr{F}$ can be **lexicalized** by another formalism $\mathscr{F}'$, if for any *finitely ambiguous* grammar $\mathscr{G}$ in $\mathscr{F}$ there is a grammar $\mathscr{G}'$ in $\mathscr{F}'$ such that $\mathscr{G}'$ is a lexicalized grammar and such that $\mathscr{G}$ and $\mathscr{G}'$ generate the same tree set (and hence the same language).

Lexicalization

- A grammar is **“lexicalized”** if it consists of:
  - a finite set of structures each associated with a lexical item; each lexical item will be called the *anchor* of the corresponding structure
  - an operation or operations for composing the structures

Theorem

Lexicalized grammars are finitely ambiguous

- We say a formalism $\mathscr{F}$ can be **lexicalized** by another formalism $\mathscr{F}'$, if for any *finitely ambiguous* grammar $\mathscr{G}$ in $\mathscr{F}$ there is a grammar $\mathscr{G}'$ in $\mathscr{F}'$ such that $\mathscr{G}'$ is a lexicalized grammar and such that $\mathscr{G}$ and $\mathscr{G}'$ generate the same tree set (and hence the same language).

Lexicalization

- A grammar is “**lexicalized**” if it consists of:
  - a finite set of structures each associated with a lexical item; each lexical item will be called the *anchor* of the corresponding structure
  - an operation or operations for composing the structures

Theorem

Lexicalized grammars are finitely ambiguous

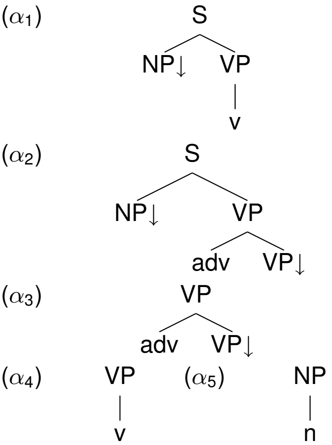
- We say a formalism $\mathscr{F}$ can be **lexicalized** by another formalism $\mathscr{F}'$, if for any *finitely ambiguous* grammar $\mathscr{G}$ in $\mathscr{F}$ there is a grammar $\mathscr{G}'$ in $\mathscr{F}'$ such that $\mathscr{G}'$ is a lexicalized grammar and such that $\mathscr{G}$ and $\mathscr{G}'$ generate the same tree set (and hence the same language).

Problem with Lexicalization in TSG

Consider this CFG

1. S → NP VP
2. VP → adv VP
3. VP → v
4. NP → n

It can be lexicalized in a TSG

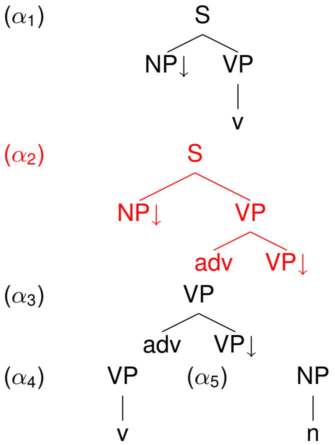


Problem with Lexicalization in TSG

Consider this CFG

1. S $\rightarrow$ NP VP
2. VP $\rightarrow$ adv VP
3. VP $\rightarrow$ v
4. NP $\rightarrow$ n

It can be lexicalized in a TSG



Linguistically motivated???

Is TSG Good Enough?

Theorem

Finitely ambiguous context-free grammars cannot be lexicalized with a tree-substitution grammar

Proof.

1. $S \rightarrow S\ S$
2. $S \rightarrow a$

(Try to prove there is no lexicalzed TSG that generates the same tree language) □

Is TSG Good Enough?

Theorem

Finitely ambiguous context-free grammars cannot be lexicalized with a tree-substitution grammar

Proof.

1. S $\rightarrow$ S S
2. S $\rightarrow$ a

(Try to prove there is no lexicalzed TSG that generates the same tree language) □

References I

Joshi, A. and Schabes, Y. (1997).
Tree-adjoining grammars.