A Psycholinguistically Motivated version of Tree Adjoining Grammar (TAG)

Sayyed Auwn Muhammad

Introduction

- Language comprehension is incremental
  - Comprehenders build an interpretation of a sentence on word–by–word basis
- Human Sentence Processing Properties
  - Incrementality
  - Connectedness
  - Prediction

Incrementality

- Perception of a word in a sentence leads to integration into a already perceived structure of the sentence
  - Left to right processing
  - word-by-word basis
- Strict Incrementality
  - Fully connectedness [Sturt and Lombardo 2005]
  - Connected under the same syntactic root node

Connectedness

- At any point of incremental sentence processing
  - All words are attached to a single syntactic structure
  - Parser are not allowed to build unconnected tree fragments

Prediction

- To achieve fully connectivity
- Make prediction of upcoming
  - Words
  - Structures
- Prediction about Structure
  - Previous structure
  - Lexicon entries

Notion of Prediction

- Either … or construction [Staub and Clifton 2006]
  - Word *either* triggers prediction of *or* and the second conjunct
  - **Syntactic parallelism** indicates that the second conjunct of a coordinate structure is processed faster if its internal structure is identical to that of the first conjunct .

"Mary is looking for either a maid or a cook"

Notion of Prediction

- Support to perdition
- Linking Parsing with Processing difficulty
- Subject Relative Clause (SRC) occurs more often than Object Relative Clause (ORC) [King and Just,1991; Gibson 1998]
  - *(SRC) The reporter that attacked the senator admitted the error.*
  - *(ORC) The reporter that the senator attacked admitted the error.*
- Higher Processing difficulty occurs when the more probable structure has to be discarded.

Grammar Formalisms

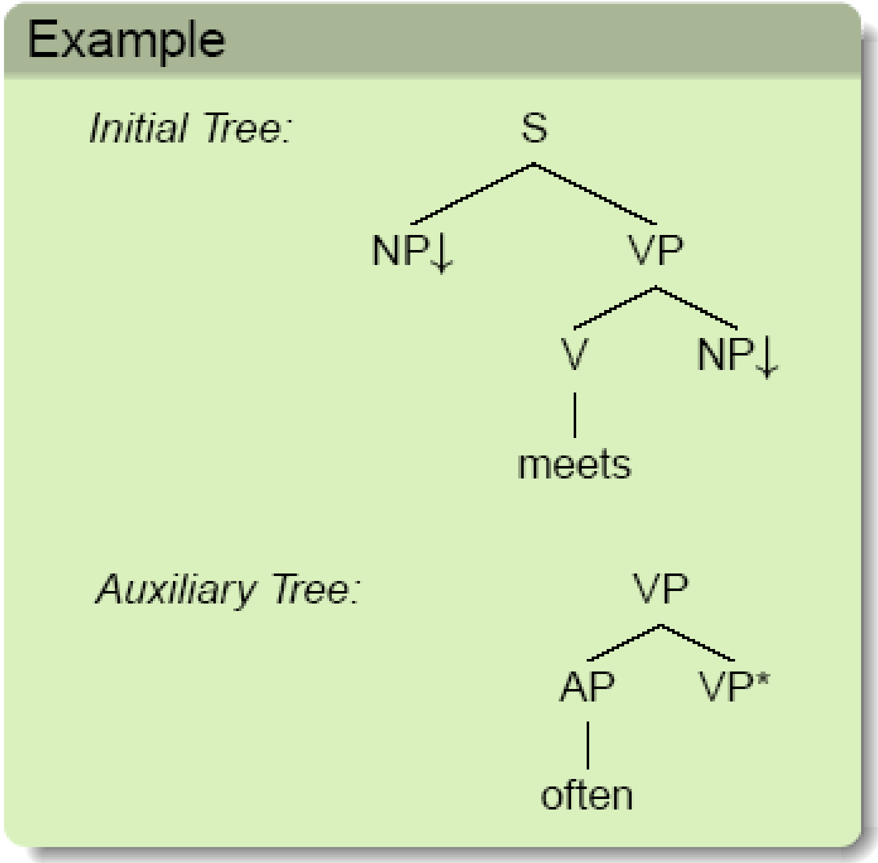
- Context Free Grammar (CFG)
  - Production rules, derivation trees
  - PCFG (Probabilistic CFG)
  - Context Sensitive Grammar (CSG)
- Dependency Grammar (DG)
  - Relation between a word (a head) and its dependents
  - Lack phrasal node
- Combinatory Categorial Grammar (CCG)
- Tree Adjoining Grammar (TAG)
  - More expressive power
  - Richer structural description to sentence [David Chiang 2004]
  - Respect fully connectedness and the prediction task

Tree Adjoining Grammar

- Tree-adjoining grammar (TAG) is defined by [Joshi et al., 1975]
- Rules in a TAG are trees
- Two types of basic trees in TAG:
  - *initial trees* (α)
  - *auxiliary trees* (β)

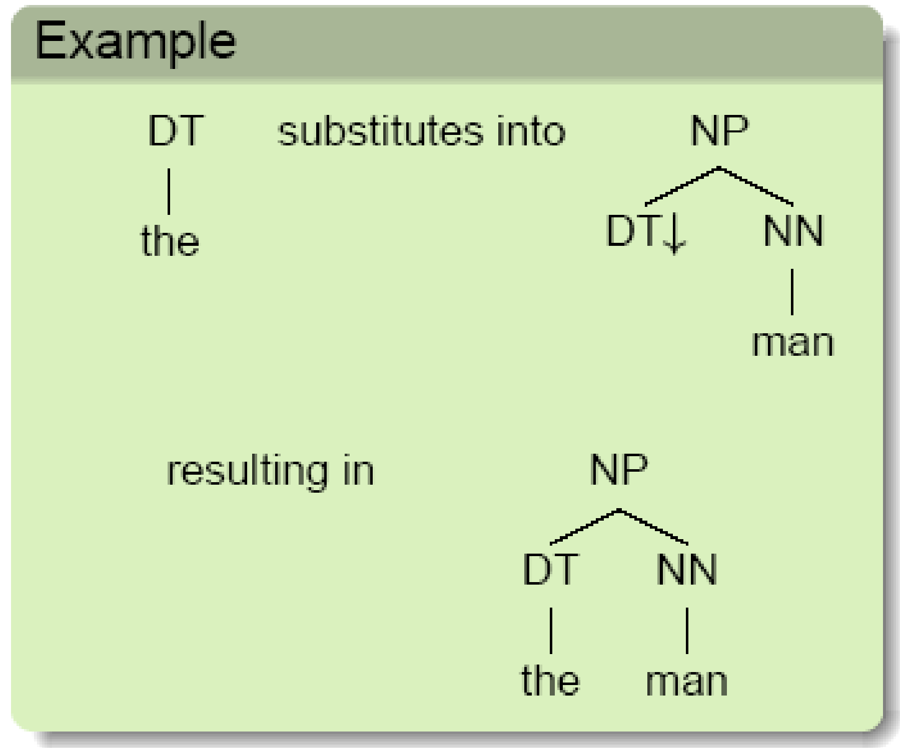
Tree Adjoining Grammar Formalism

- Tree types
  - Initial trees
  - Auxiliary trees
- Operations
  - Substitution
  - Adjunction



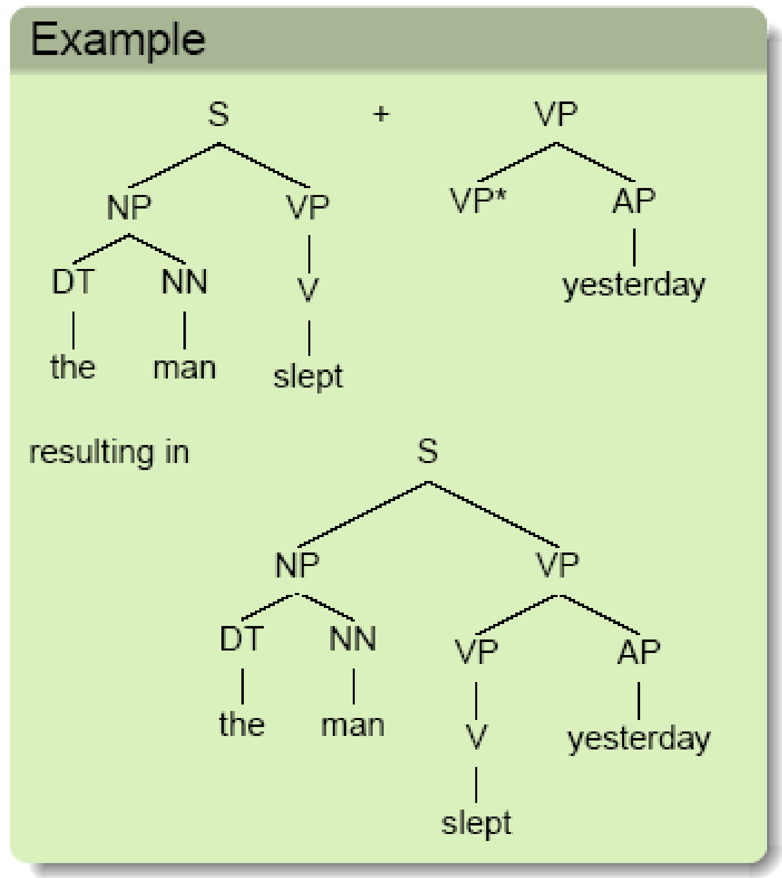
Substitution operation

- Substitution node
- Substitution Symbol ↓
- Derived tree



Adjunction operation

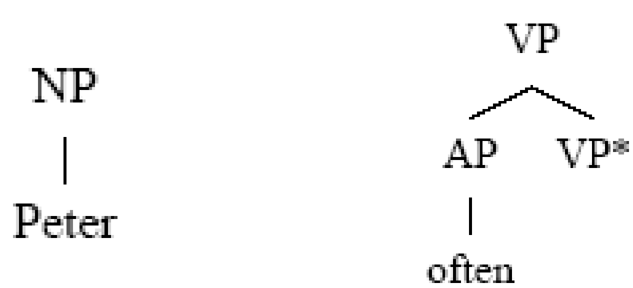
- Same root node as foot node
- Recursive in nature
- Two operations



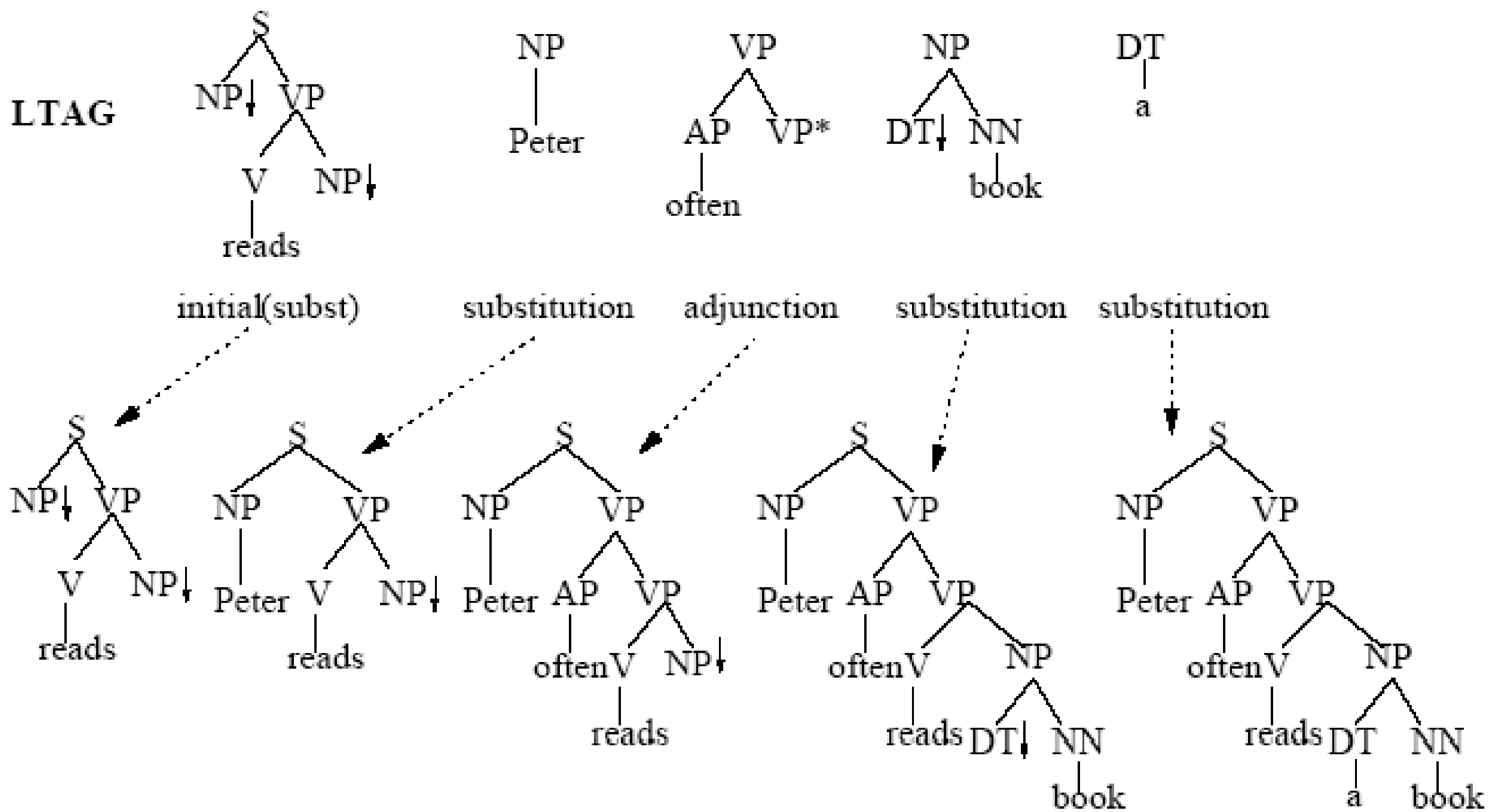
Problem with LTAG

- Does not allow derivation in strictly incremental fashion
- Consider the example of *"Peter often reads a book"*
- The head *reads* which provide the intervening structure has not been encountered yet.

Peter often reads a book

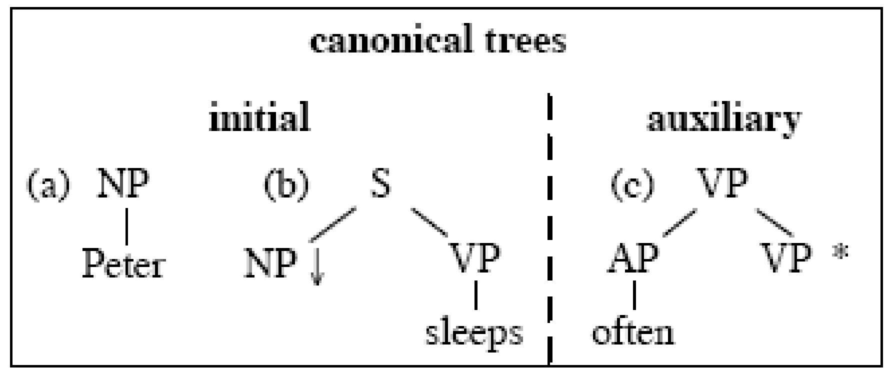


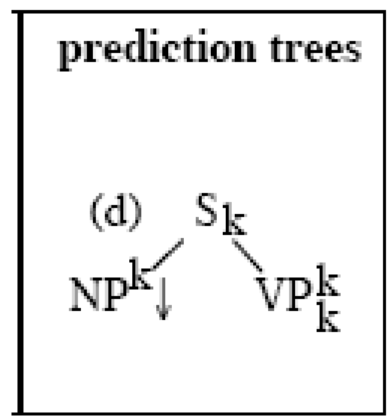
Problem with LTAG



Proposed PLTAG

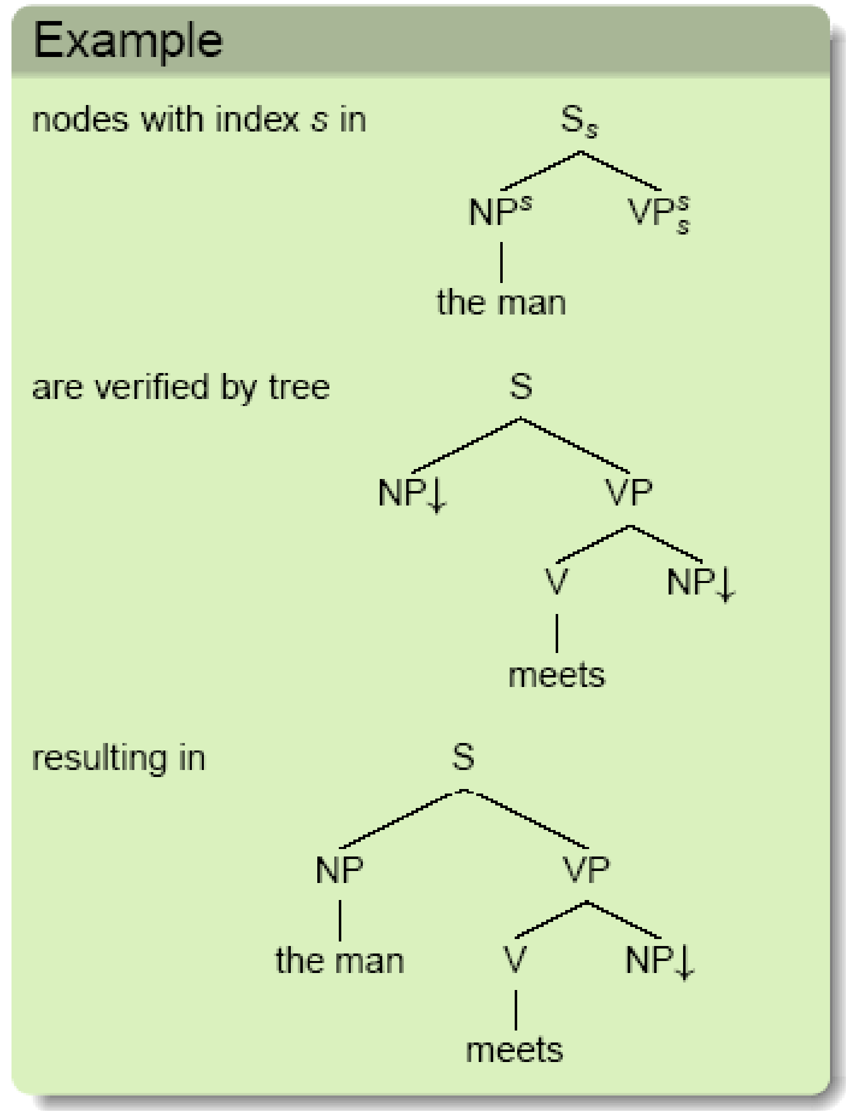
- Lexicon:
  - Canonical LTAG Lexicon
  - Prediction Lexicon
- Operations:
  - Substitution
  - Adjunction
  - Verification





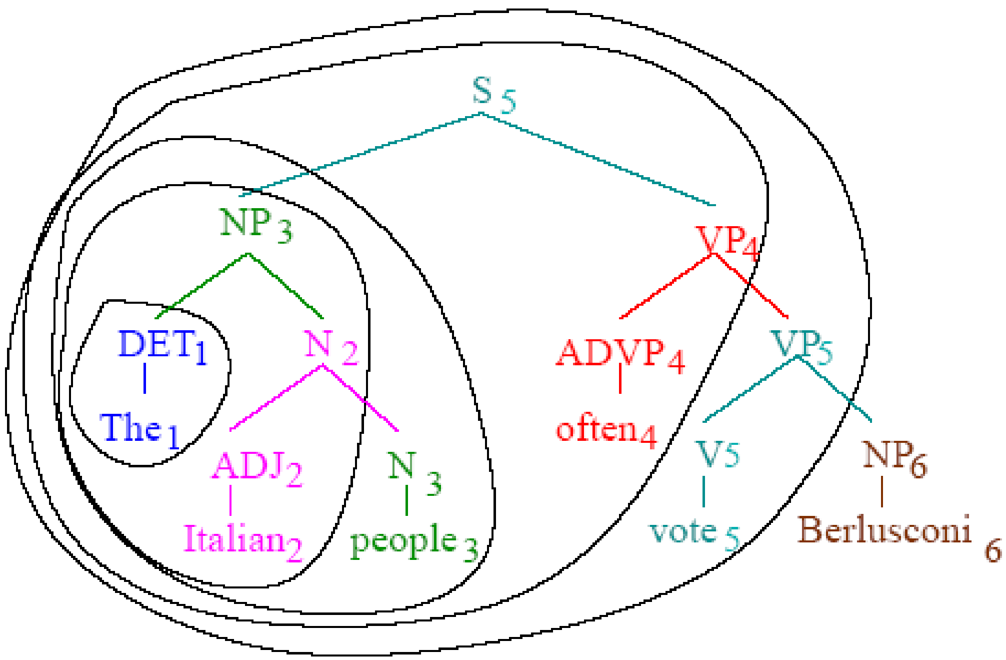
Verification

- Prediction nodes needs to be verified
- Verification is an operation that removes prediction indices



Connection Path Concept

- The minimal amount of structure needed at each word for sentence incremental processing.

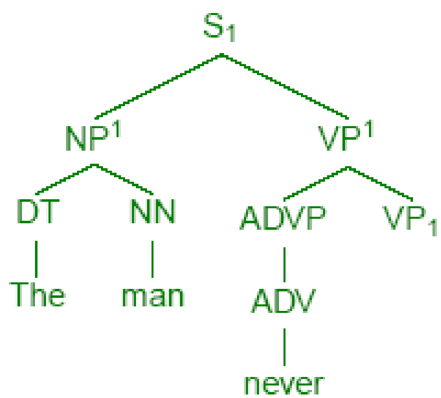


Definition of a PLTAG Derivation:

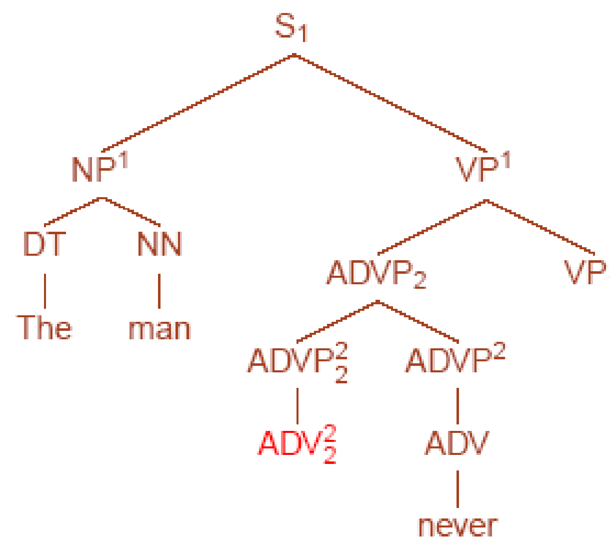
- A PLTAG derivation starts with the tree of the first input word, and then applies **Substitution and adjunction operations** to canonical trees or prediction trees. Every prediction tree has to be **validated** using the **verification operation** later.
- In a partial PLTAG derivation for words $w_1..w_i$, all leaves to the left of w_i must be **fully lexicalized.** A PLTAG derivation is **complete** when every leaf node is labeled with a terminal symbol, none of the nodes in the tree is marked as predictive, and the root symbol of the derived tree is S.

PLTAG

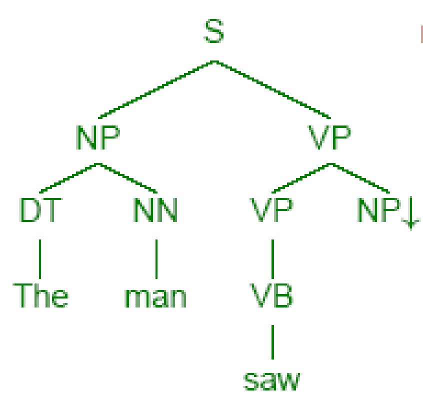
OK:



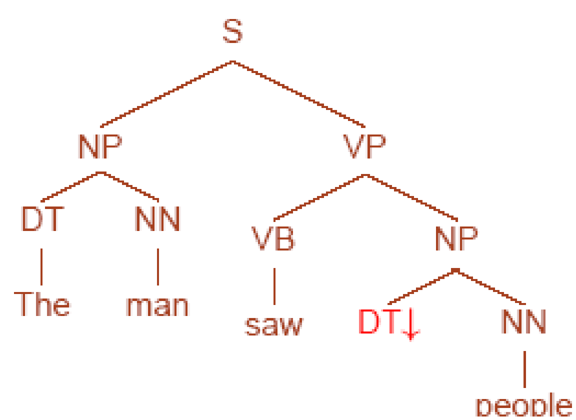
not OK:



OK:

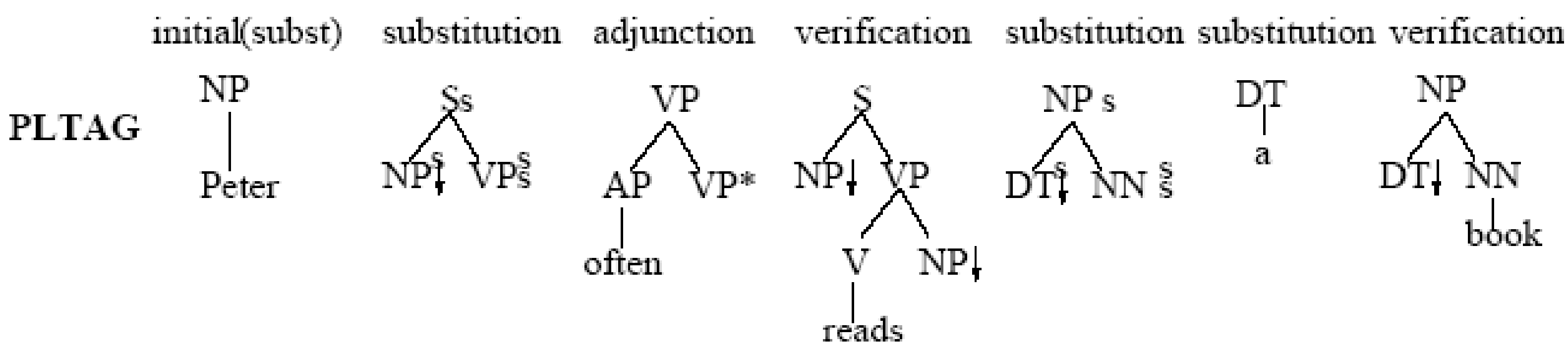


not OK:



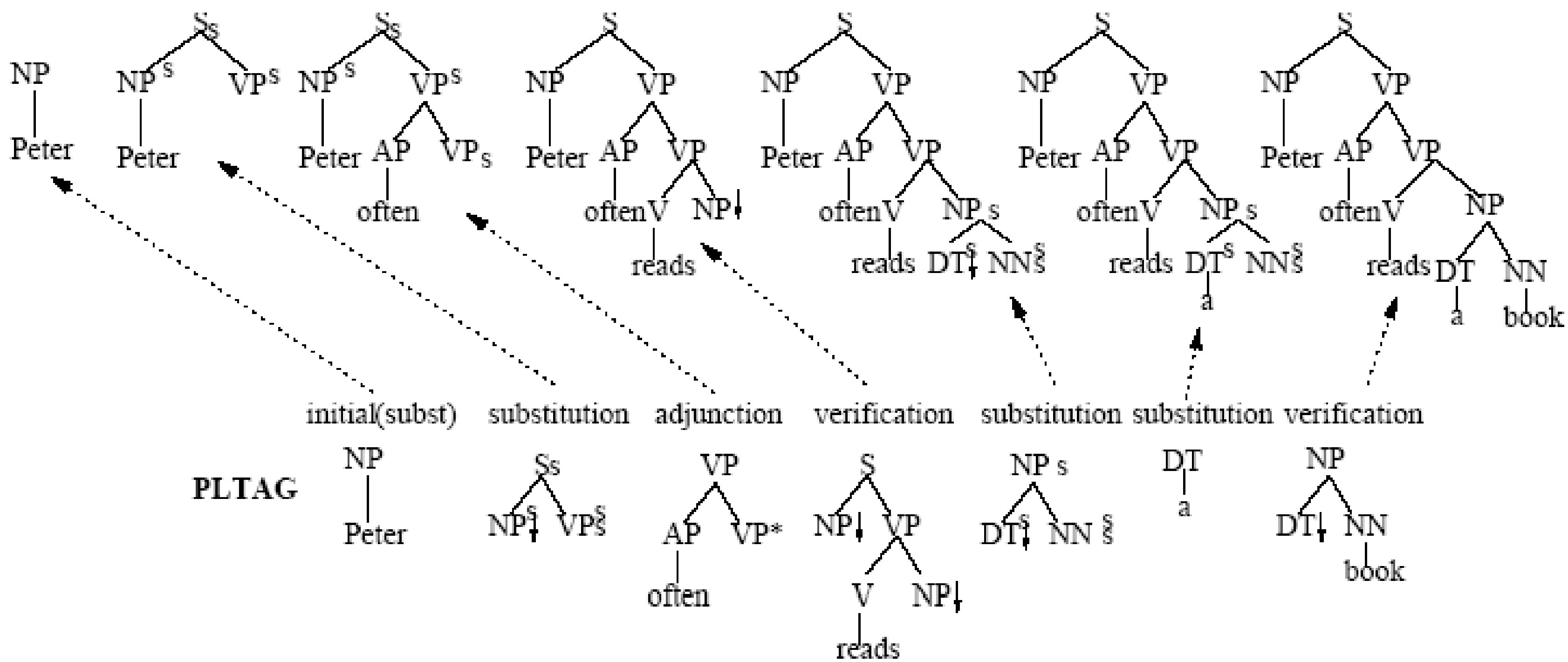
PLTAG

- PLTAG derivations are always strictly incremental.



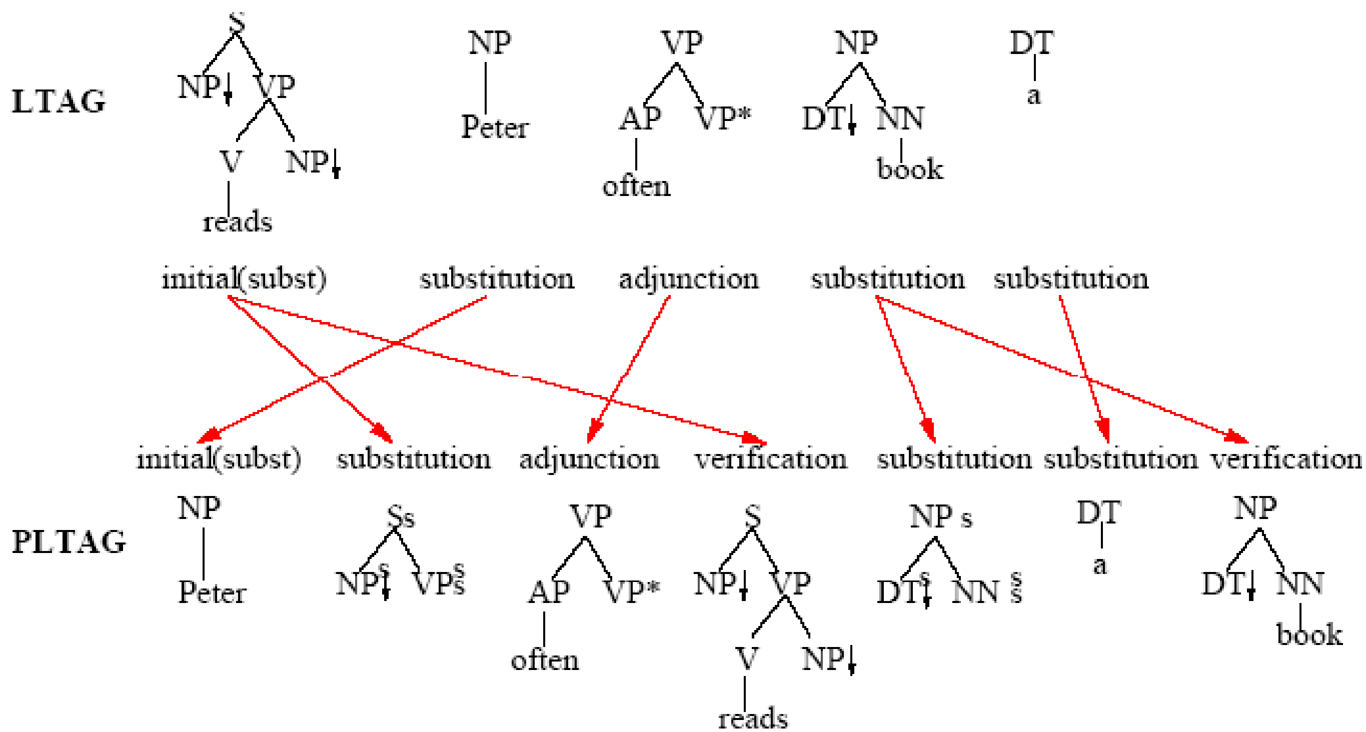
PLTAG

- PLTAG derivations are always strictly incremental.



Relationship between LTAG and PLTAG

- Every LTAG derivation can be translated into an equivalent PLTAG derivation.



Steps in Constructing the Parser

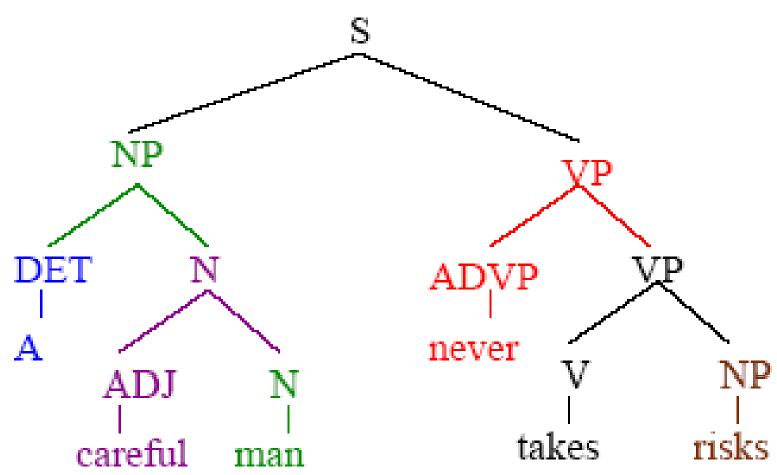
1. Conversion of the Penn Treebank into PLTAG format
2. Lexicon Induction
3. The Incremental Parsing Algorithm
4. The Probability Model
5. Parser Evaluation

Conversion of the Penn Treebank into PLTAG format

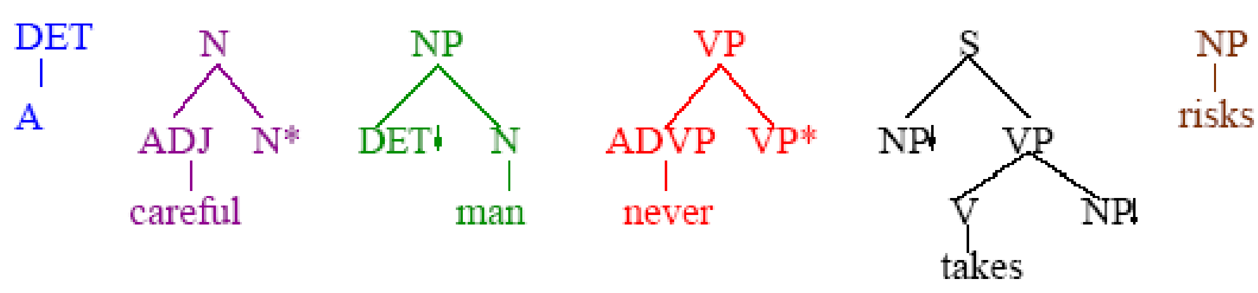
- Induce the lexicon (both canonical trees and prediction trees) needed for PLTAG from Penn Treebank

Lexicon Induction: creating 1

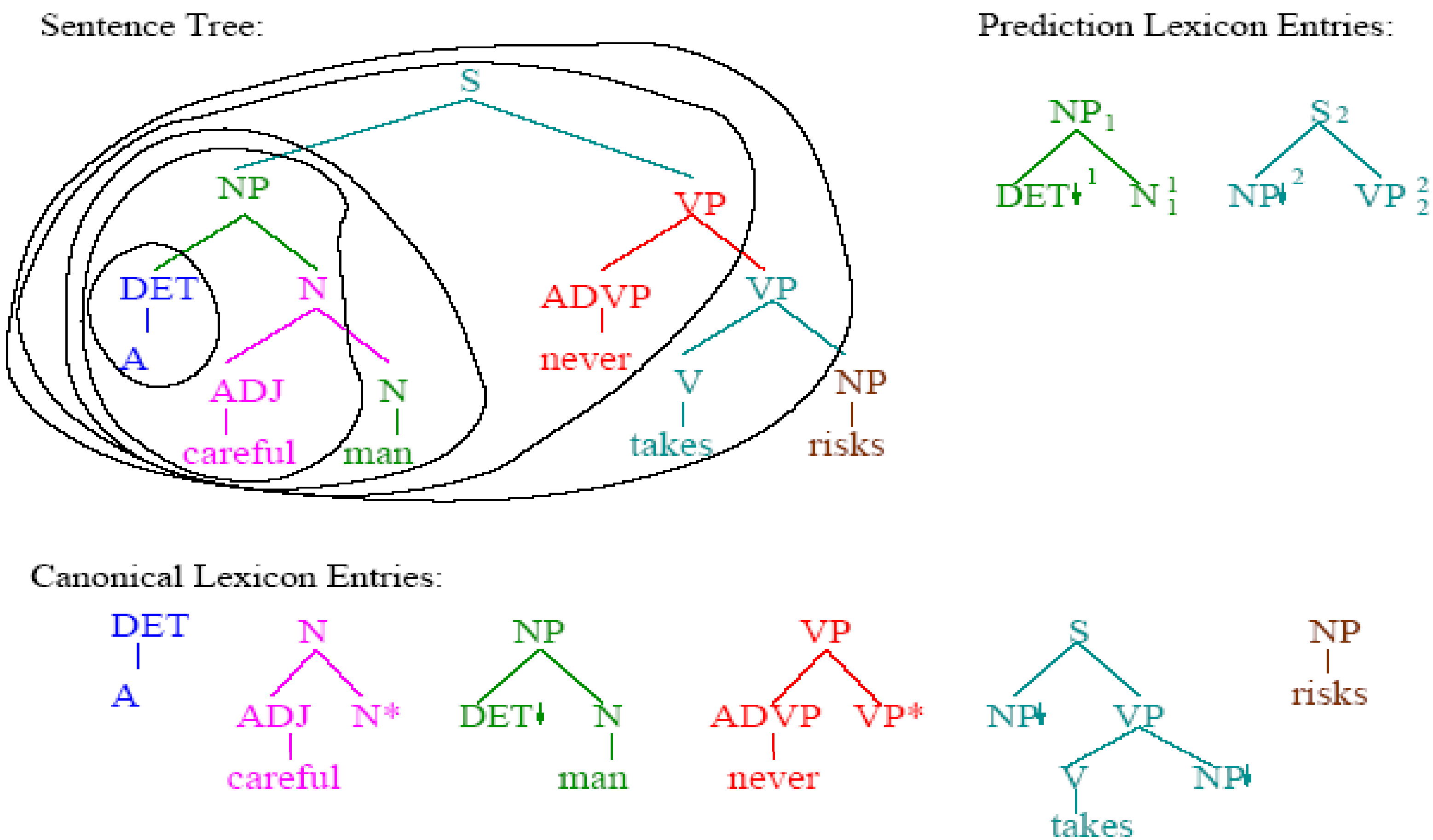
Sentence Tree:



Canonical Lexicon Entries:



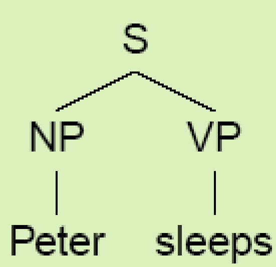
Lexicon Induction: creating 2



The Parsing Algorithm

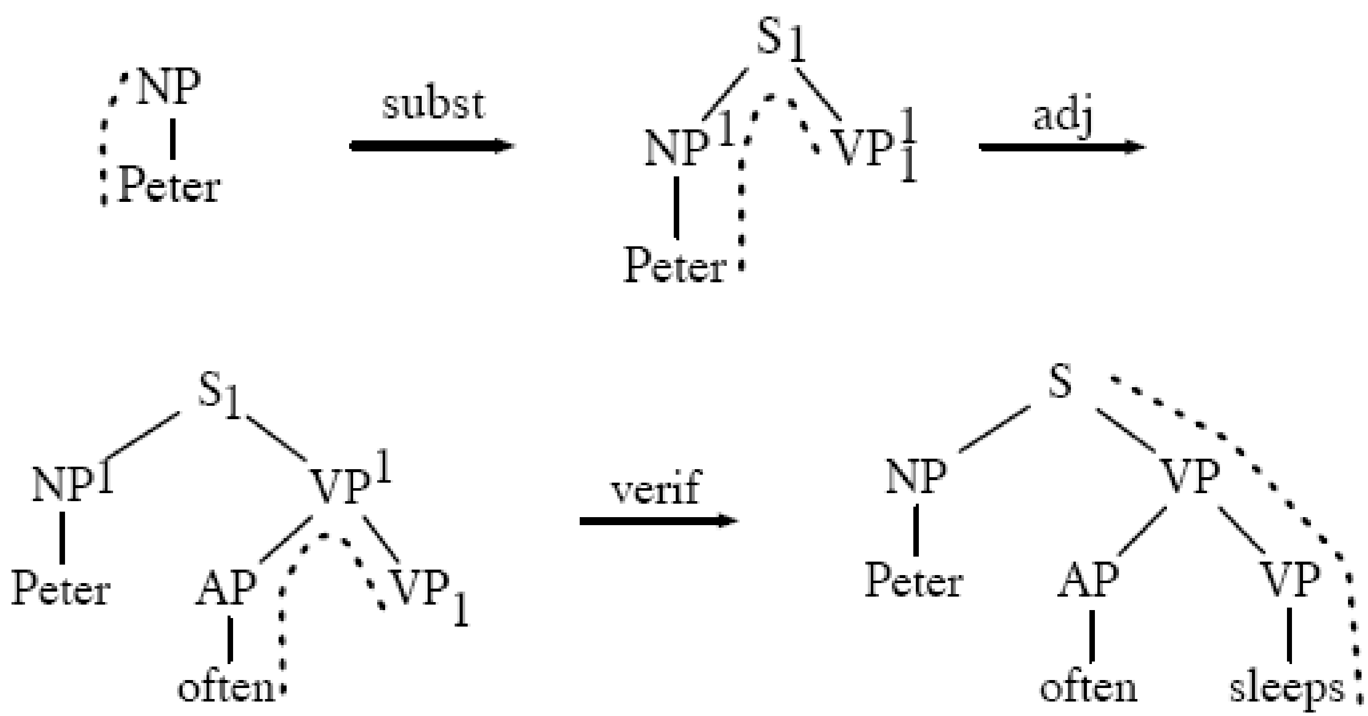
- Requirements:
  - Produce incremental and fully connected structures at every point in time
  - Only produce valid PLTAG trees
- Helpful Concept: Fringes
  - tree can be described by its depth-first traversal
  - only part of incremental tree is relevant at each step

Example



(S, NP, Peter, Peter, NP, VP, sleeps, sleeps, VP, S)

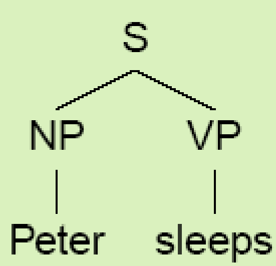
Verification: Fringes



The Parsing Algorithm

- Requirements:
  - Produce incremental and fully connected structures at every point in time
  - Only produce valid PLTAG trees
- Helpful Concept: Fringes
  - tree can be described by its depth-first traversal
  - only part of incremental tree is relevant at each step

Example

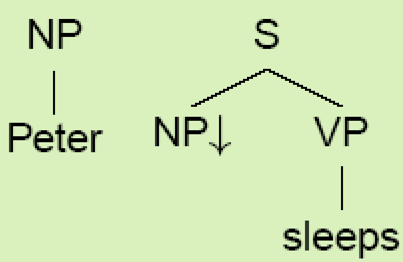


$(S^+, NP^+, \text{Peter}^+, \text{Peter}^-, NP^-, VP^+, \text{sleeps}^+, \text{sleeps}^-, VP^-, S^-)$

The Parsing Algorithm

- Requirements:
  - Produce incremental and fully connected structures at every point in time
  - Only produce valid PLTAG trees
- Helpful Concept: Fringes
  - tree can be described by its depth-first traversal
  - only part of incremental tree is relevant at each step

Example



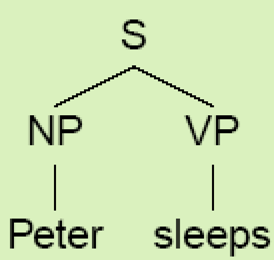
$(NP^+, Peter^+, Peter^-, NP^-)$

$(S^+, NP{\downarrow}^+, NP{\downarrow}^-, VP^+, sleep^+, sleep^-, VP^-, S^-)$

The Parsing Algorithm

- Requirements:
  - Produce incremental and fully connected structures at every point in time
  - Only produce valid PLTAG trees
- Helpful Concept: Fringes
  - tree can be described by its depth-first traversal
  - only part of incremental tree is relevant at each step

Example



(NP^+, $Peter^+$, $Peter^-$, NP^-)

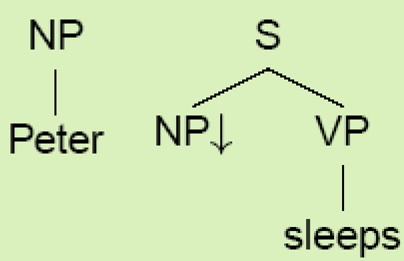
(S^+, $NP\downarrow^+$, $NP\downarrow^-$, VP^+, $sleep^+$, $sleep^-$, VP^-, S^-)

(S^+, NP^+, $Peter^+$, $Peter^-$, NP^-, VP^+, $sleeps^+$, $sleeps^-$, VP^-, S^-)

The Parsing Algorithm

- Requirements:
  - Produce incremental and fully connected structures at every point in time
  - Only produce valid PLTAG trees
- Helpful Concept: Fringes
  - tree can be described by its depth-first traversal
  - only part of incremental tree is relevant at each step

Example



(NP^+, $Peter^+$, • **$Peter^-$**, **NP^-**)

(• **S^+**, **$NP{\downarrow}^+$**, $NP{\downarrow}^-$, VP^+, $sleep^+$, $sleep^-$, VP^-, S^-)

(S^+, NP^+, $Peter^+$, $Peter^-$, NP^-, VP^+, $sleeps^+$, • **$sleeps^-$**, **VP^-**, **S^-**)

Probability Model

Substitution: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) = 1$

Adjunction: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) + P(NONE|\eta_\beta) = 1$

Verification: $\sum_{\varepsilon} P(\varepsilon|\pi_\beta) = 1$

$$
\begin{aligned}
P(\varepsilon|\eta_\beta) &= P(\tau_\varepsilon|\eta_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_\eta) \\
P(\varepsilon|\pi_\beta) &= P(\tau_\varepsilon|\pi_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_{\pi_\eta}) \\
P(\eta_\beta) &= P(\tau_\eta, \lambda_\eta, c_\eta, n_\eta, b_\eta, a_f, tm)
\end{aligned}
$$

based on [Chiang, 2000]

Explanation

Probabilities are normalized with respect to other elementary trees ε that can attach at node η in prefix tree β with the same operation.

elementary trees	ε	prefix tree	β	prediction trees	π
tree structures	τ	integration point node	η	a tree's head leaf	λ

Probability Model

Substitution: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) = 1$

Adjunction: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) + P(NONE|\eta_\beta) = 1$

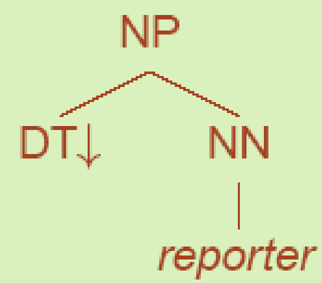
Verification: $\sum_{\varepsilon} P(\varepsilon|\pi_\beta) = 1$

$$P(\varepsilon|\eta_\beta) = P(\tau_\varepsilon|\eta_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_\eta)$$
$$P(\varepsilon|\pi_\beta) = P(\tau_\varepsilon|\pi_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_{\pi_\eta})$$
$$P(\eta_\beta) = P(\tau_\eta, \lambda_\eta, c_\eta, n_\eta, b_\eta, a_f, tm)$$

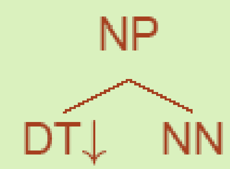
based on [Chiang, 2000]

Explanation

elementary tree ε:



is estimated as template τ_ε:



and lexeme λ_ε: *reporter*

elementary trees	ε	prefix tree	β	prediction trees	π
tree structures	τ	integration point node	η	a tree's head leaf	λ

Probability Model

Substitution: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) = 1$

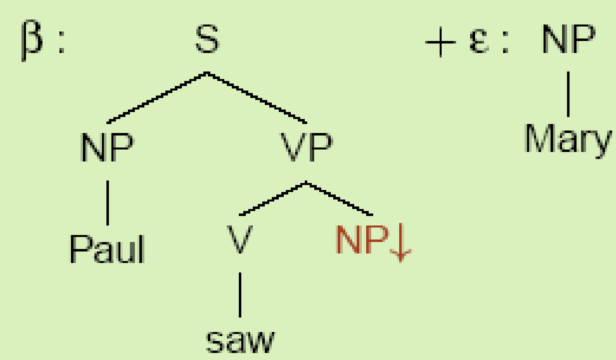
Adjunction: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) + P(NONE|\eta_\beta) = 1$

Verification: $\sum_{\varepsilon} P(\varepsilon|\pi_\beta) = 1$

$$
\begin{aligned}
P(\varepsilon|\eta_\beta) &= P(\tau_\varepsilon|\eta_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_\eta) \\
P(\varepsilon|\pi_\beta) &= P(\tau_\varepsilon|\pi_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_{\pi_\eta}) \\
P(\eta_\beta) &= P(\tau_\eta, \lambda_\eta, c_\eta, n_\eta, b_\eta, a_f, tm)
\end{aligned}
$$

based on [Chiang, 2000]

Explanation



integration node η_β

elementary trees	ε	prefix tree	β	prediction trees	π
tree structures	τ	integration point node	η	a tree's head leaf	λ

Probability Model

Substitution: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) = 1$

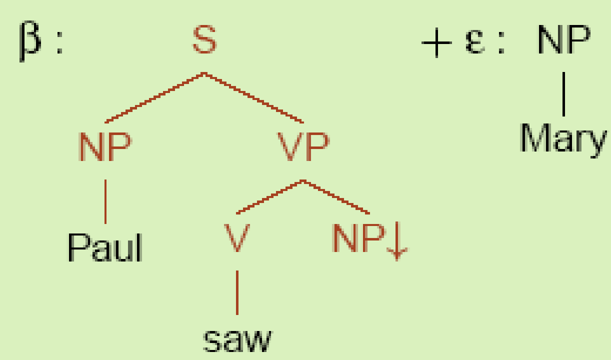
Adjunction: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) + P(NONE|\eta_\beta) = 1$

Verification: $\sum_{\varepsilon} P(\varepsilon|\pi_\beta) = 1$

$$P(\varepsilon|\eta_\beta) = P(\tau_\varepsilon|\eta_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_\eta)$$
$$P(\varepsilon|\pi_\beta) = P(\tau_\varepsilon|\pi_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_{\pi_\eta})$$
$$P(\eta_\beta) = P(\tau_\eta, \lambda_\eta, c_\eta, n_\eta, b_\eta, a_f, tm)$$

based on [Chiang, 2000]

Explanation



integration tree template τ_η

elementary trees	ε	prefix tree	β	prediction trees	π
tree structures	τ	integration point node	η	a tree's head leaf	λ

Probability Model

Substitution: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) = 1$

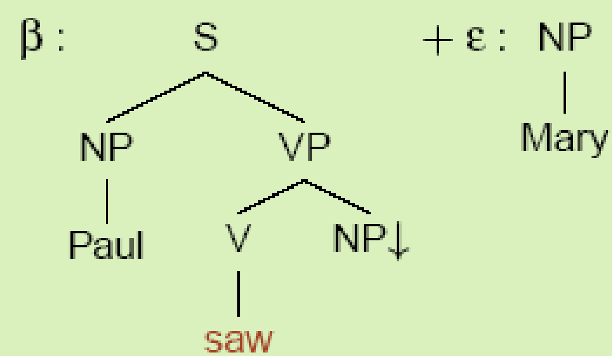
Adjunction: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) + P(NONE|\eta_\beta) = 1$

Verification: $\sum_{\varepsilon} P(\varepsilon|\pi_\beta) = 1$

$$P(\varepsilon|\eta_\beta) = P(\tau_\varepsilon|\eta_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_\eta)$$
$$P(\varepsilon|\pi_\beta) = P(\tau_\varepsilon|\pi_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_{\pi_\eta})$$
$$P(\eta_\beta) = P(\tau_\eta, \lambda_\eta, c_\eta, n_\eta, b_\eta, a_f, tm)$$

based on [Chiang, 2000]

Explanation



integration tree lexeme λ_η

elementary trees	ε	prefix tree	β	prediction trees	π
tree structures	τ	integration point node	η	a tree's head leaf	λ

Probability Model

Substitution: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) = 1$

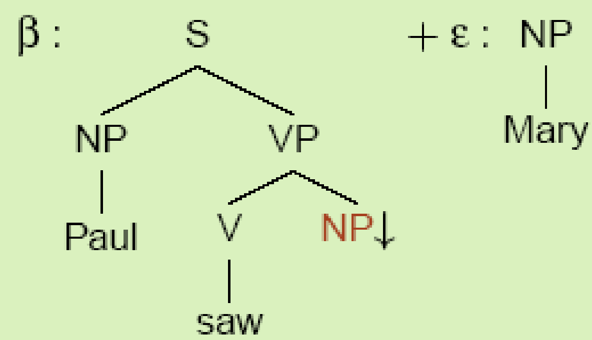
Adjunction: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) + P(NONE|\eta_\beta) = 1$

Verification: $\sum_{\varepsilon} P(\varepsilon|\pi_\beta) = 1$

$$\begin{aligned} P(\varepsilon|\eta_\beta) &= P(\tau_\varepsilon|\eta_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_\eta) \\ P(\varepsilon|\pi_\beta) &= P(\tau_\varepsilon|\pi_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_{\pi_\eta}) \\ P(\eta_\beta) &= P(\tau_\eta, \lambda_\eta, c_\eta, n_\eta, b_\eta, a_f, tm) \end{aligned}$$

based on [Chiang, 2000]

Explanation



integration node category c_η

elementary trees	ε	prefix tree	β	prediction trees	π
tree structures	τ	integration point node	η	a tree's head leaf	λ

Probability Model

Substitution: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) = 1$

Adjunction: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) + P(NONE|\eta_\beta) = 1$

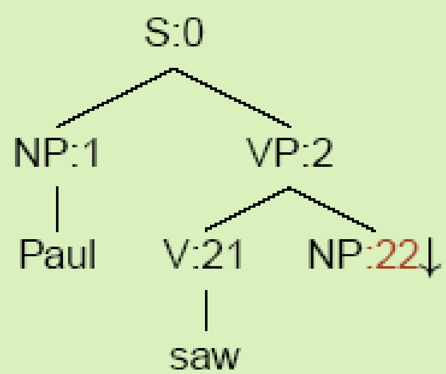
Verification: $\sum_{\varepsilon} P(\varepsilon|\pi_\beta) = 1$

$$
\begin{aligned}
P(\varepsilon|\eta_\beta) &= P(\tau_\varepsilon|\eta_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_\eta) \\
P(\varepsilon|\pi_\beta) &= P(\tau_\varepsilon|\pi_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_{\pi_\eta}) \\
P(\eta_\beta) &= P(\tau_\eta, \lambda_\eta, c_\eta, n_\eta, b_\eta, a_f, tm)
\end{aligned}
$$

based on [Chiang, 2000]

Explanation

β :



integration node position n_η

elementary trees	ε	prefix tree	β	prediction trees	π
tree structures	τ	integration point node	η	a tree's head leaf	λ

Probability Model

Substitution: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) = 1$

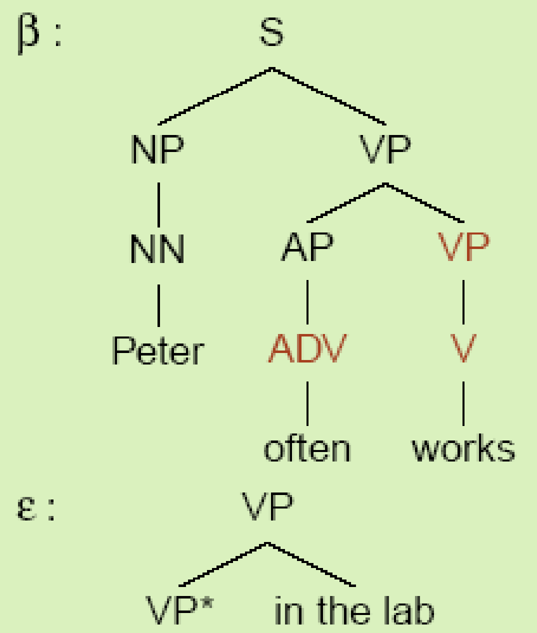
Adjunction: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) + P(NONE|\eta_\beta) = 1$

Verification: $\sum_{\varepsilon} P(\varepsilon|\pi_\beta) = 1$

$$P(\varepsilon|\eta_\beta) = P(\tau_\varepsilon|\eta_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_\eta)$$
$$P(\varepsilon|\pi_\beta) = P(\tau_\varepsilon|\pi_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_{\pi_\eta})$$
$$P(\eta_\beta) = P(\tau_\eta, \lambda_\eta, c_\eta, n_\eta, b_\eta, a_f, tm)$$

based on [Chiang, 2000]

Explanation



attachment context b_η & level a_f

elementary trees	ε	prefix tree	β	prediction trees	π
tree structures	τ	integration point node	η	a tree's head leaf	λ

Probability Model

Substitution: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) = 1$

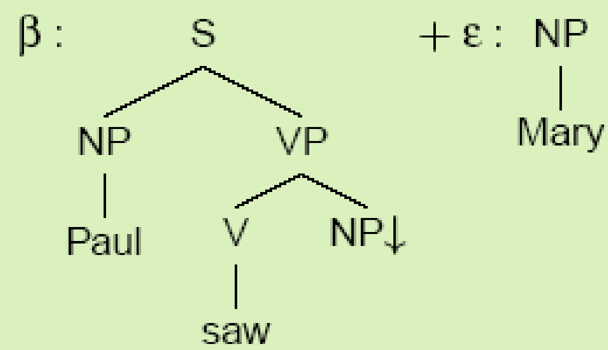
Adjunction: $\sum_{\varepsilon} P(\varepsilon|\eta_\beta) + P(NONE|\eta_\beta) = 1$

Verification: $\sum_{\varepsilon} P(\varepsilon|\pi_\beta) = 1$

$$P(\varepsilon|\eta_\beta) = P(\tau_\varepsilon|\eta_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_\eta)$$
$$P(\varepsilon|\pi_\beta) = P(\tau_\varepsilon|\pi_\beta) \times P(\lambda_\varepsilon|\tau_\varepsilon, \lambda_{\pi_\eta})$$
$$P(\eta_\beta) = P(\tau_\eta, \lambda_\eta, c_\eta, n_\eta, b_\eta, a_f, tm)$$

based on [Chiang, 2000]

Explanation



a trace mark *tm* which marks whether there is a trace at the beginning or end of the fringe

elementary trees	ε	prefix tree	β	prediction trees	π
tree structures	τ	integration point node	η	a tree's head leaf	λ

Parser Evaluation

- Parser Performance

Model	Prec	Recall	F-score	Cov
PLTAG parser	79.43	79.39	79.41	98.09
Pred tree oracle	81.15	81.13	81.14	96.18
No gold POS	77.57	77.24	77.41	98.09

Parser Evaluation

- Comparison with other TAG Parser

Model	incr	con	pred	impl	F
Mazzei et al. (2007)	+	+	+	–	n/a
This work (gold POS)	+	+	+	+	79.4
Kato et al. (2004)	+	+	–	+	79.7
Shen and Joshi (2005)	(+)	–	–	+	(87.4)
Chiang (2000)	–	–	–	+	86.7

References:

1) Vera Demberg and Frank Keller, “A Psycholinguistically motivated version of TAG” 2008, In Proceedings of the 9th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+9 2008), Tuebingen, Germany, June 2008.
2) Vera Demberg , “A Broad-Coverage Model of Prediction in Human Sentence Processing”, 2010 PhD Thesis, The University of Edinburgh.
3) Vera Demberg, “Incremental, Predictive Parsing with Psycholinguistically motivated Tree Adjoining Grammar”

The End

- Thanks for your attention