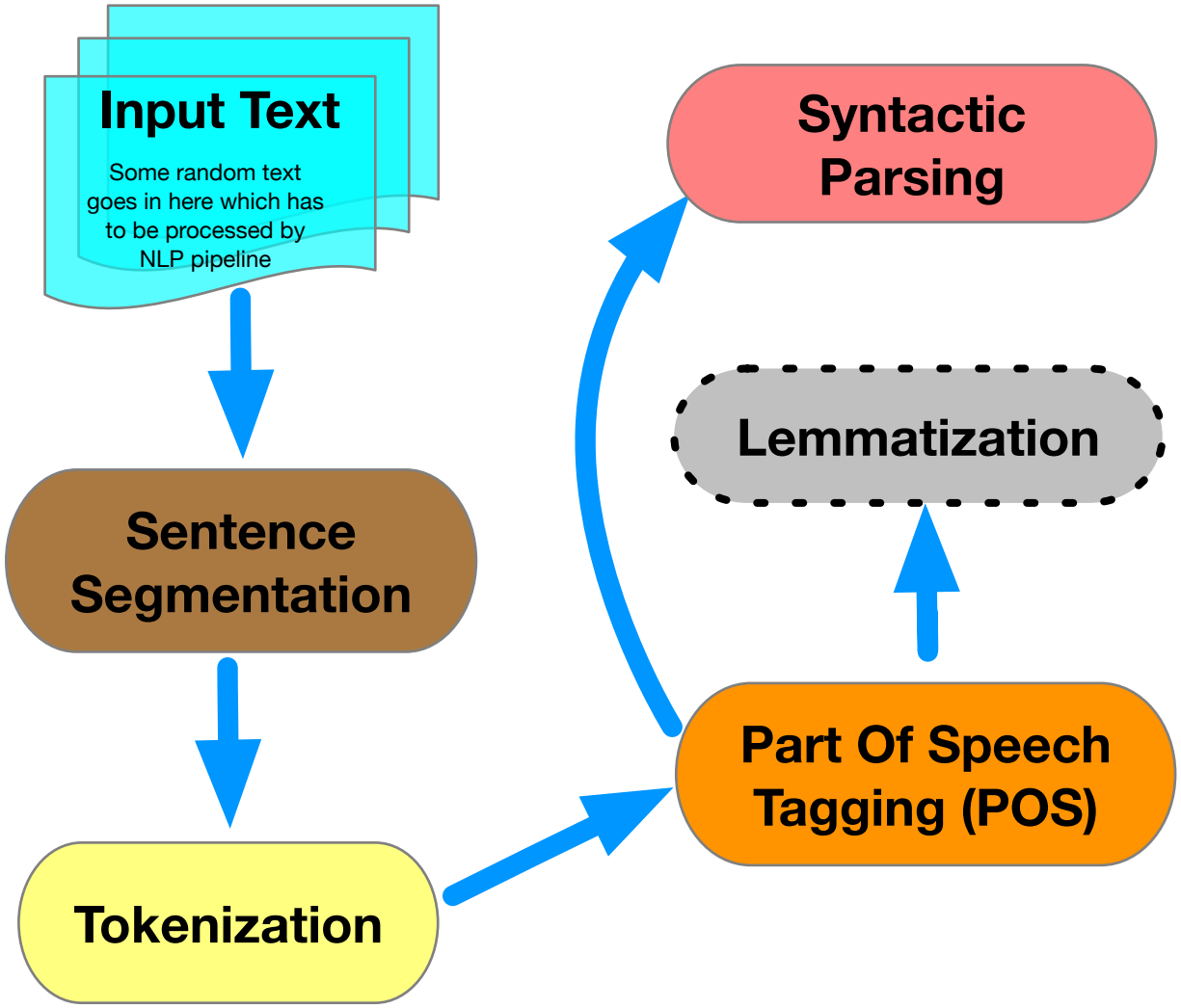
Special Topics in Natural Language Processing

CS698O

Ashutosh Modi
CSE Department, IIT Kanpur

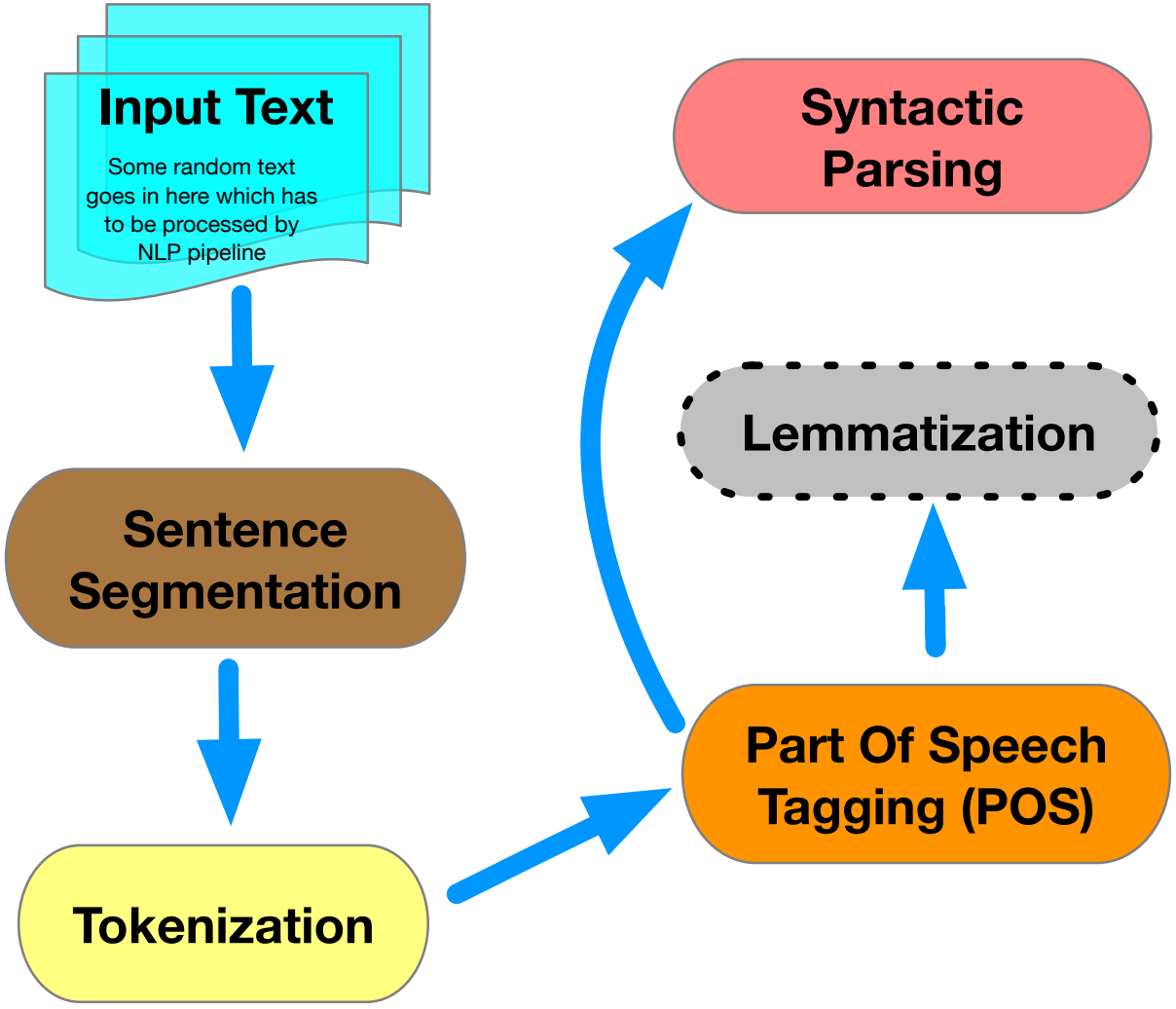
Lecture 19: Parsing 5
Feb 24, 2020

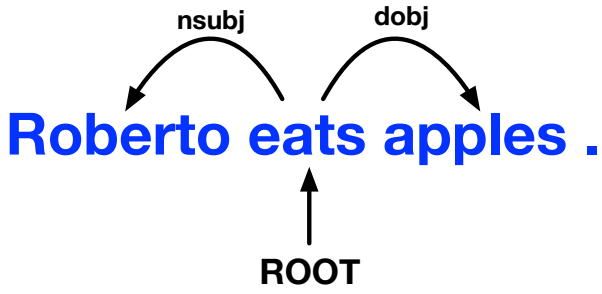


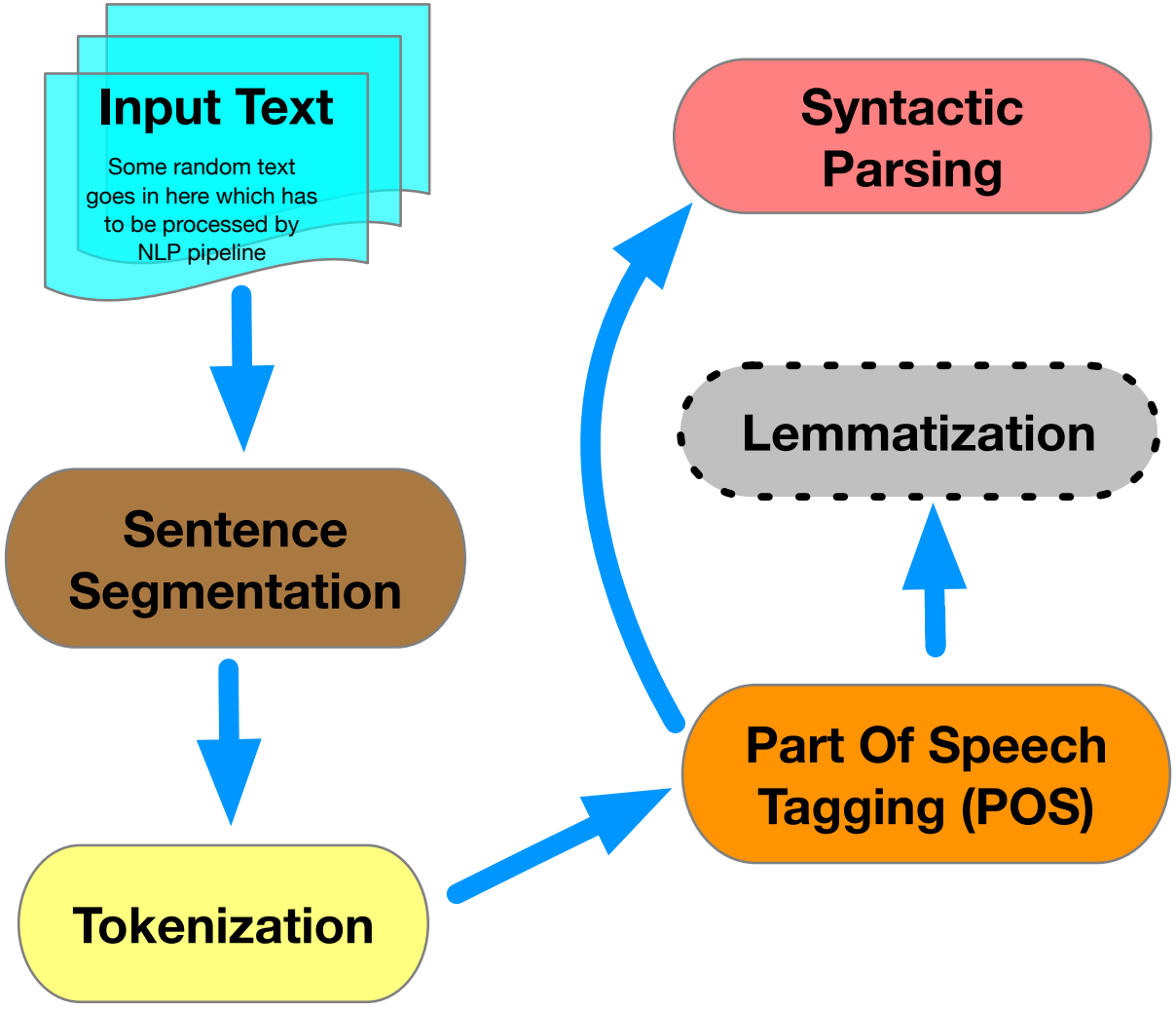
Sentences in a language follow the grammar which is formally specified via the Syntax (rules of the language)

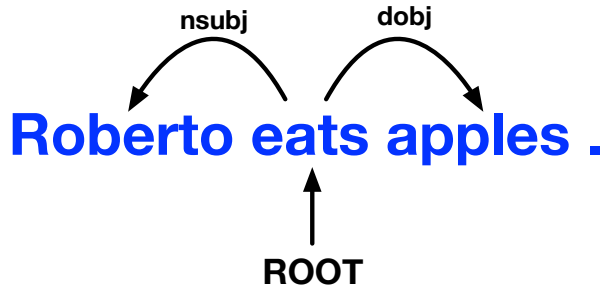
Parsing is the process for syntactically analyzing the sentences

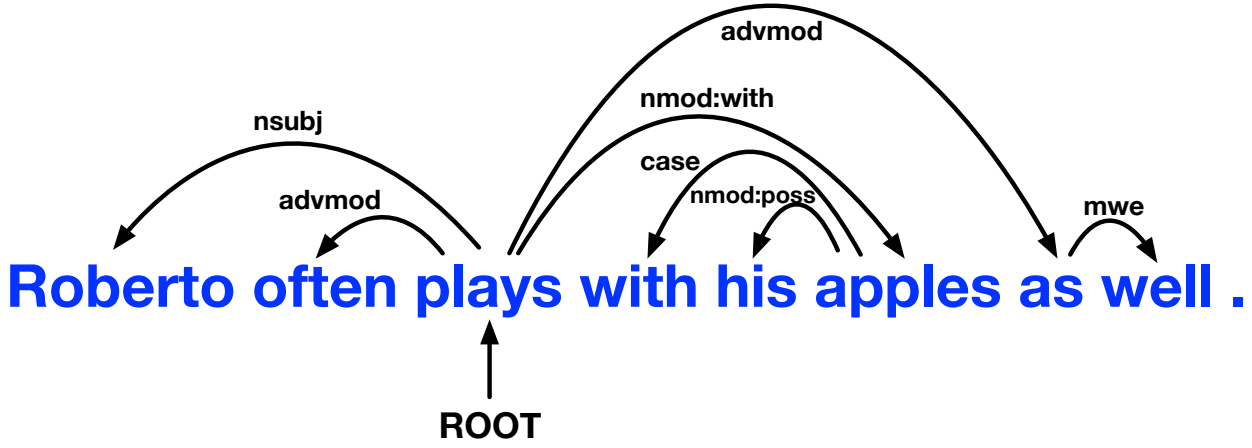
One of the popular Parsing method is Dependency Parsing


Input Text
Some random text goes in here which has to be processed by NLP pipeline
Sentence Segmentation
Tokenization
Part Of Speech Tagging (POS)
Lemmatization
Syntactic Parsing


nsubj
dobj
Roberto eats apples .
ROOT


Input Text
Some random text goes in here which has to be processed by NLP pipeline
Sentence Segmentation
Tokenization
Part Of Speech Tagging (POS)
Lemmatization
Syntactic Parsing


nsubj
dobj
Roberto eats apples .
ROOT


advmod
nmod:with
nsubj
case
advmod
nmod:poss
mwe
Roberto often plays with his apples as well .
ROOT

Syntactic Analysis via Dependency Parsing

- Analyze language in terms of relations between words.
- Earlier versions of Dependency Grammar formulated by Panini in 4 B.C.E.
- Modern work done by French Linguist Tesnière (1959)
- Many different dependency grammar frameworks have been proposed: Prague School's Functional Grammar, Hudson Grammar, etc.

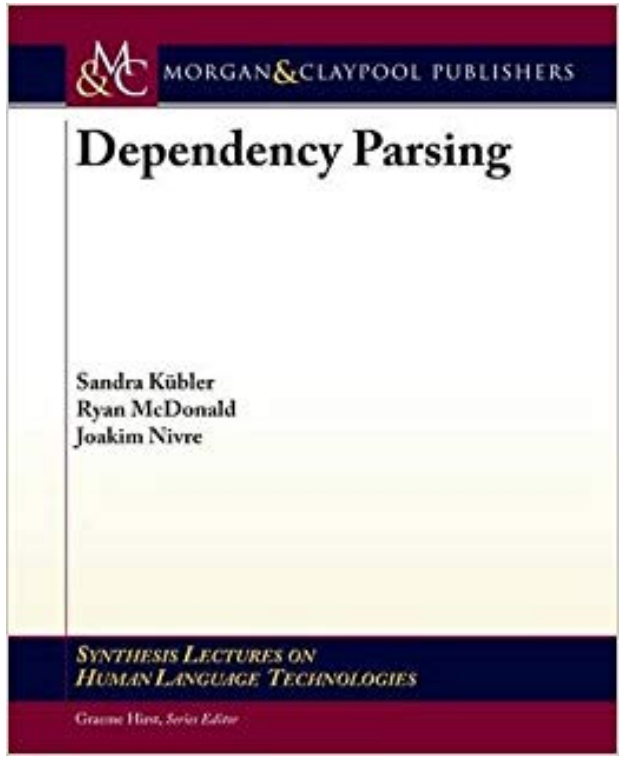
Image: http://www-history.mcs.st-andrews.ac.uk/PictDisplay/Panini.html

Main Reference

Dependency Parsing

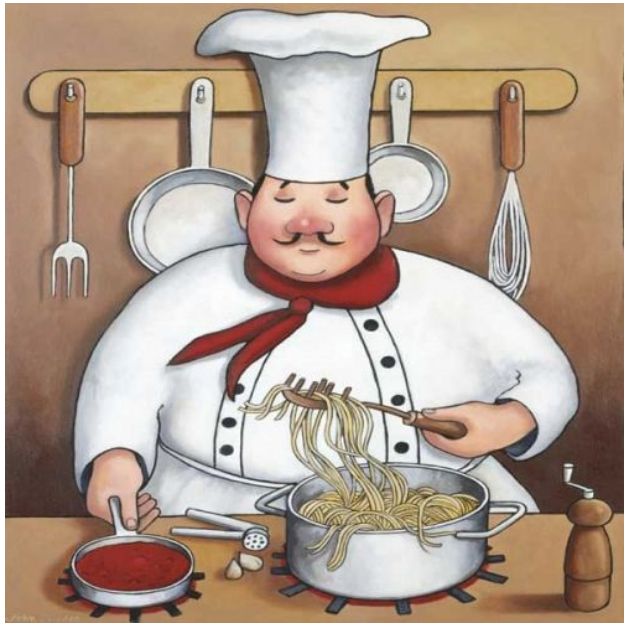
Sandra Kübler, Ryan McDonald, Joakim Nivre

(Synthesis Lectures on Human Language Technologies,
Morgan & Claypool Publishers)



Dependency Grammar

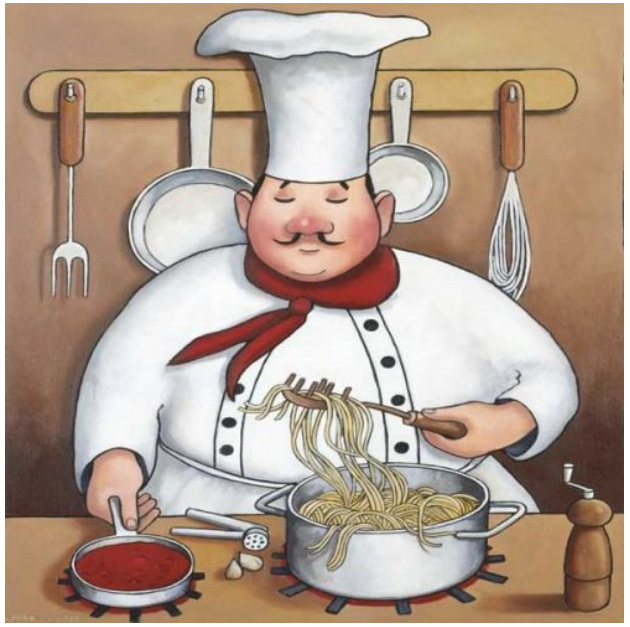
John cooked pasta.



Dependency Grammar

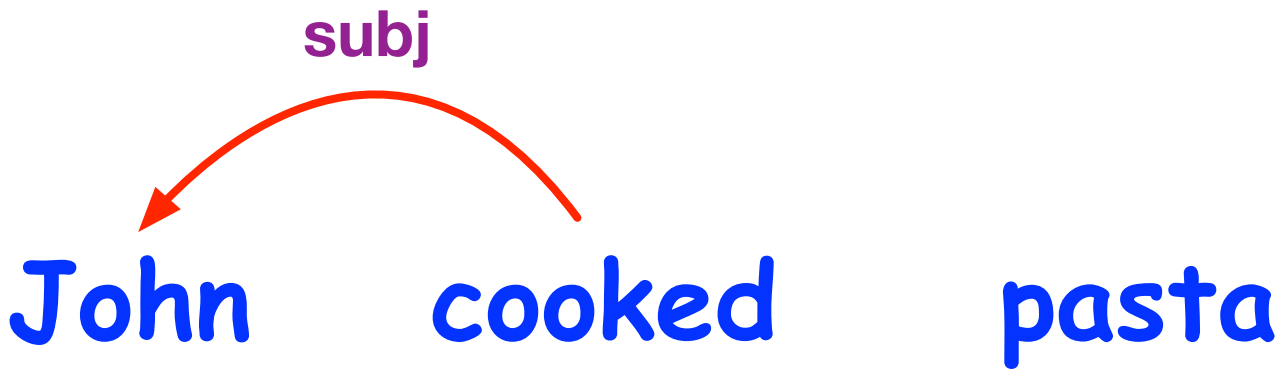
- Words in a sentence are related to each other via binary asymmetric relations called as dependency relations

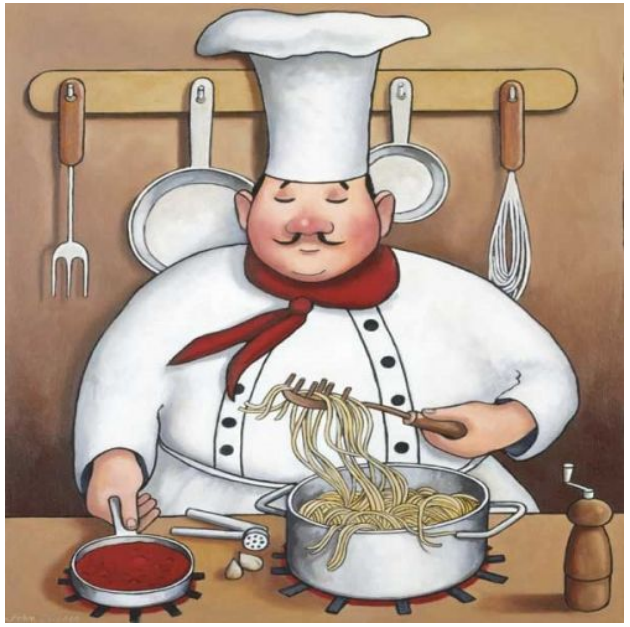
John cooked pasta.



Dependency Grammar

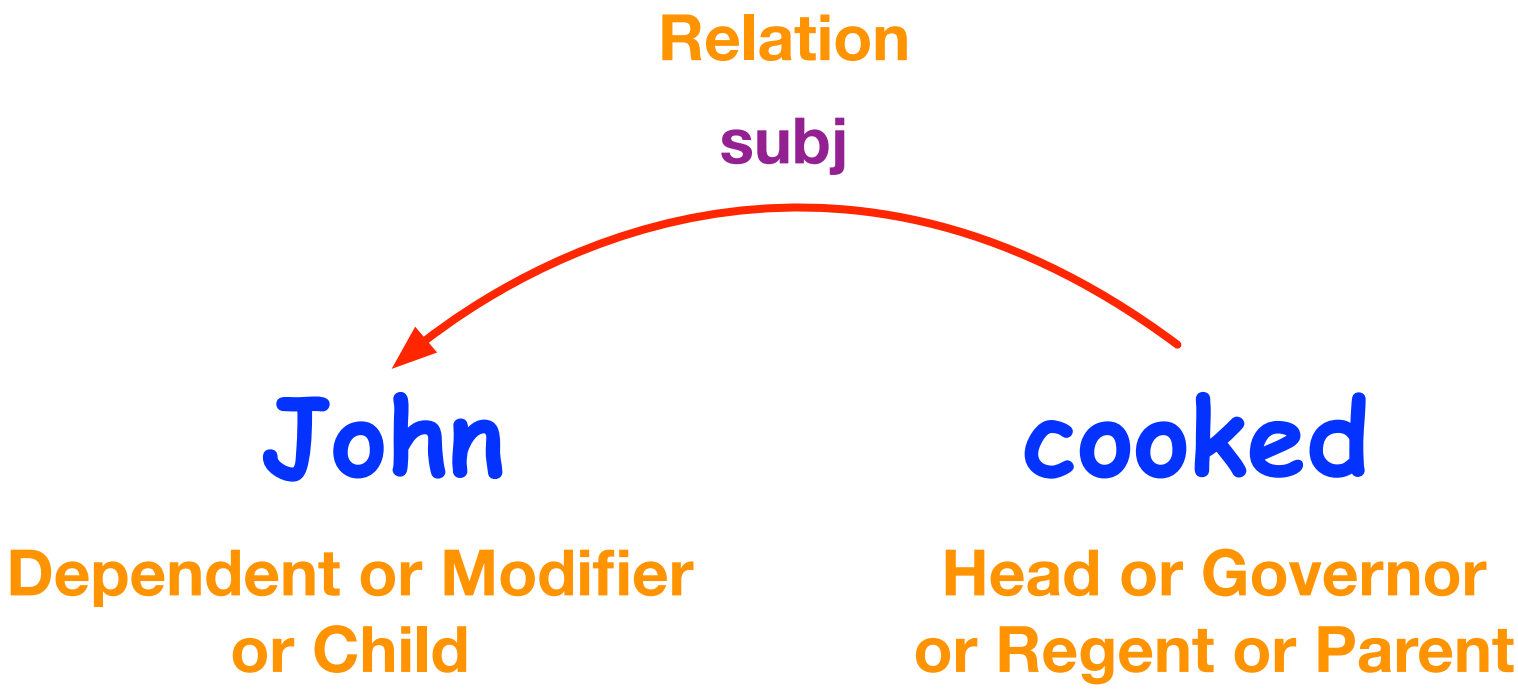
- Words in a sentence are related to each other via binary asymmetric relations called as dependency relations

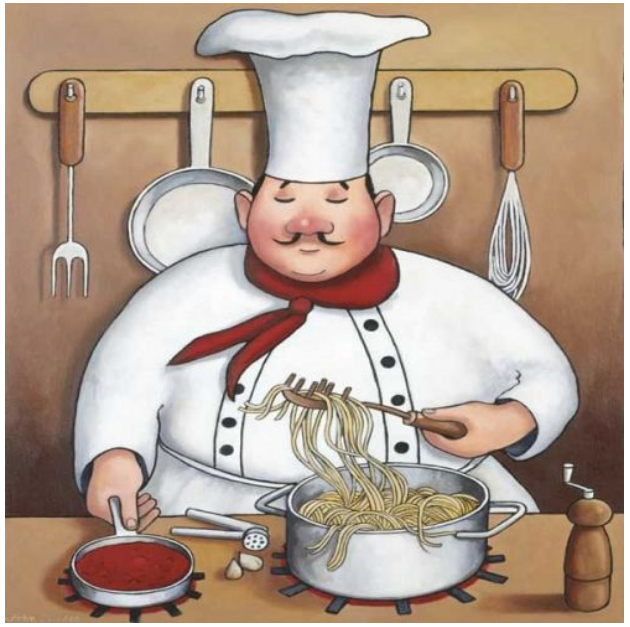




Dependency Grammar

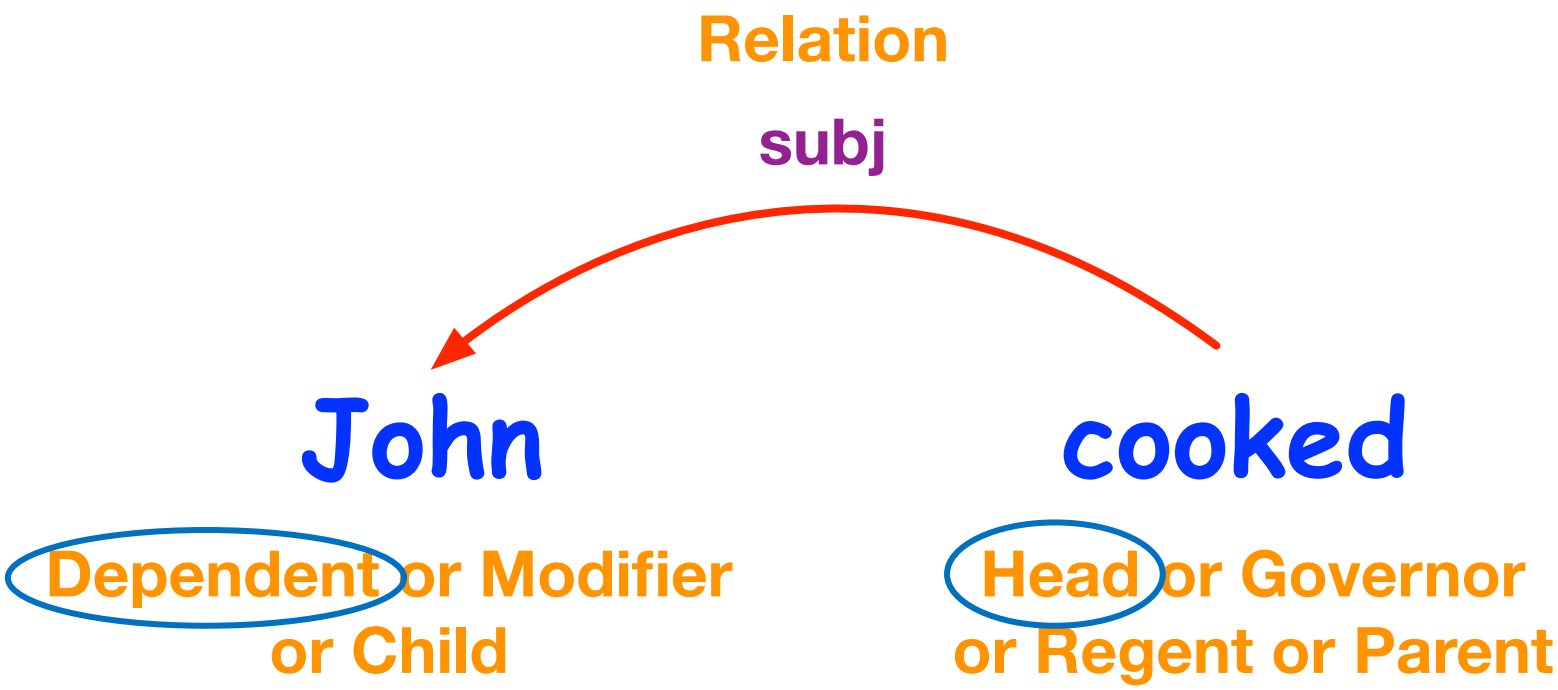
- Words in a sentence are related to each other via binary asymmetric relations called as dependency relations

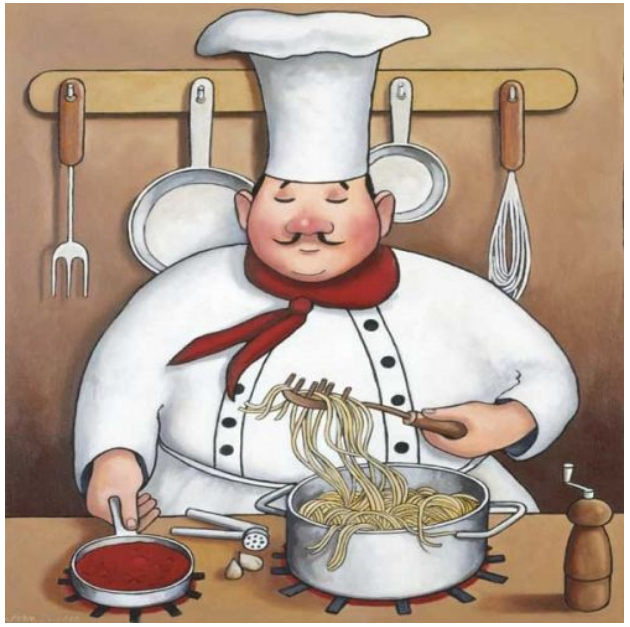




Dependency Grammar

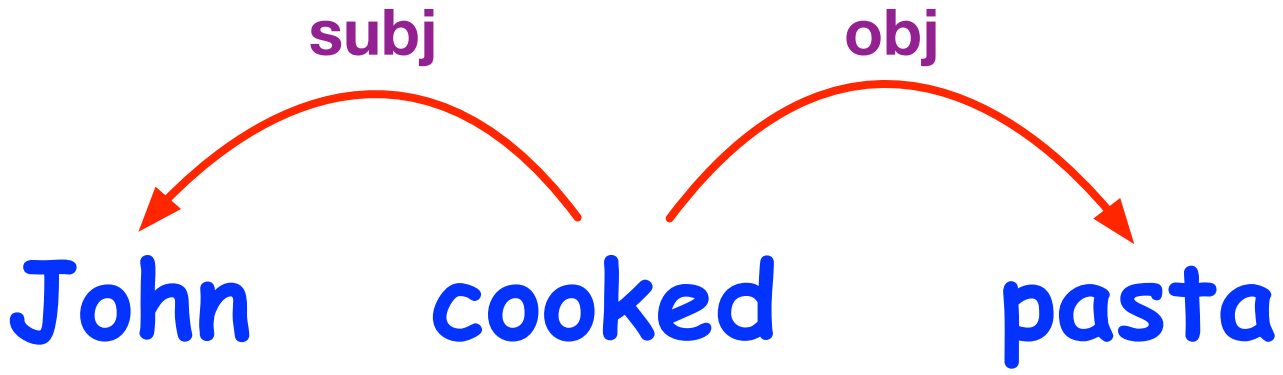
- Words in a sentence are related to each other via binary asymmetric relations called as dependency relations





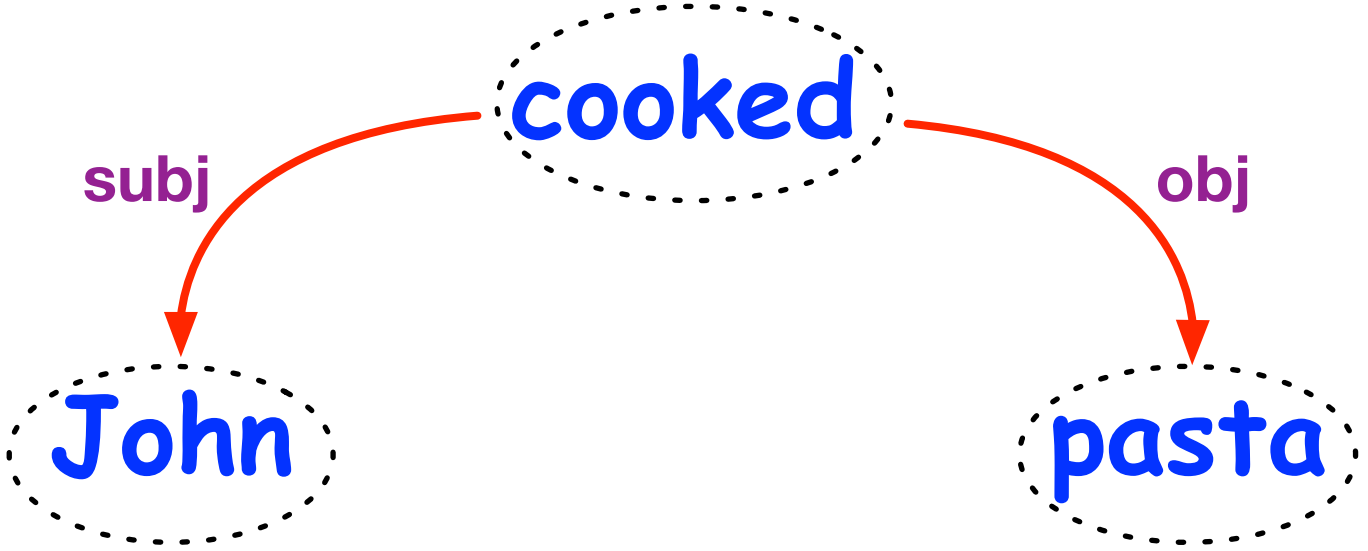
Dependency Grammar

- Words in a sentence are related to each other via binary asymmetric relations called as dependency relations



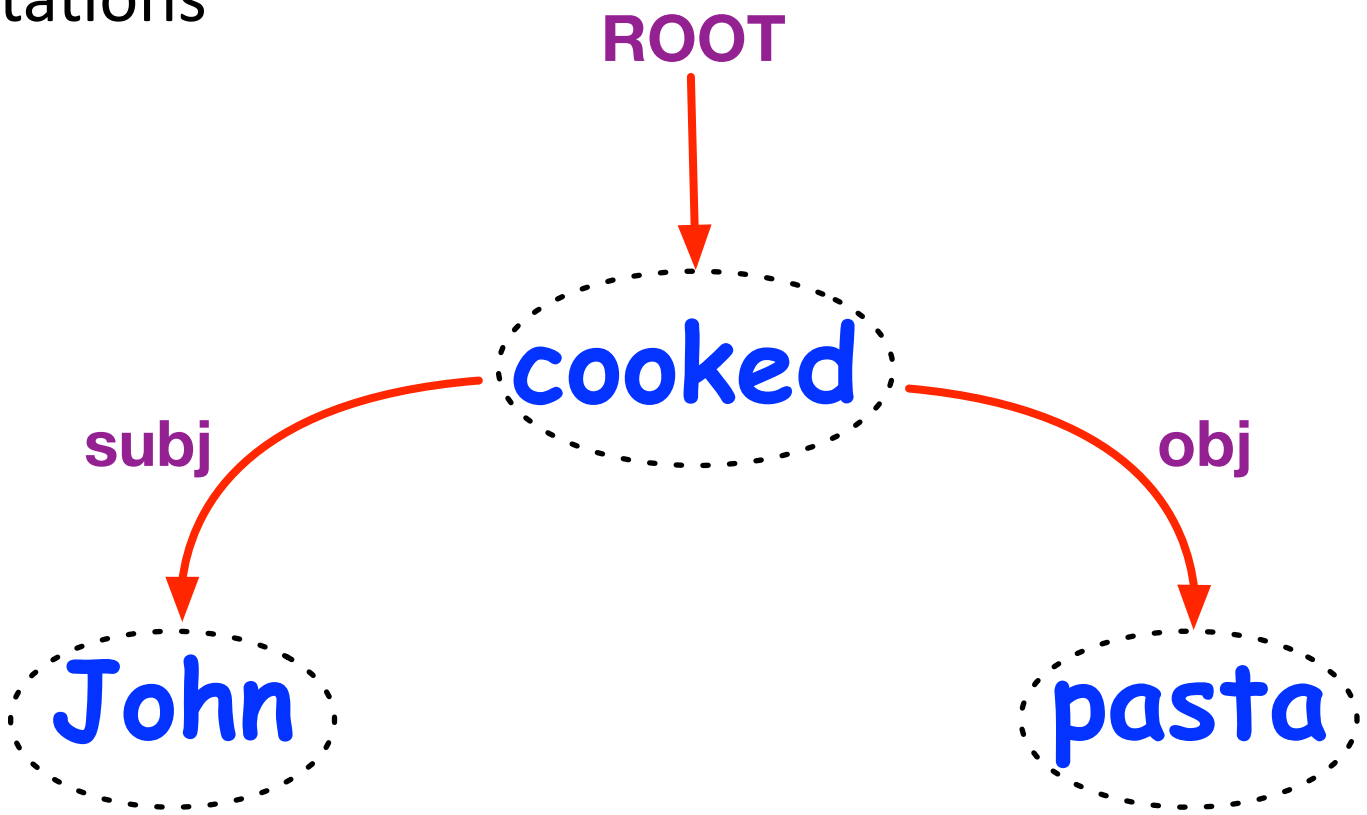
Dependency Graph

- Labeled Directed Graph
- Nodes → Words
- Labelled Edges → Dependency Relations



Dependency Graph

- Every dependency graph has a **ROOT**
- It is for technical convenience which helps in computational implementations

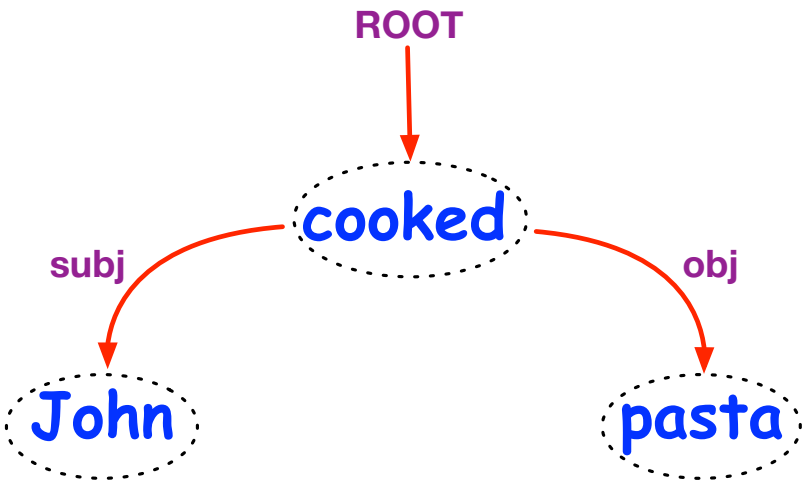


Dependency Parsing (DP)

- The task of automatically analyzing dependency structure of a given input sentence.

John cooked pasta.

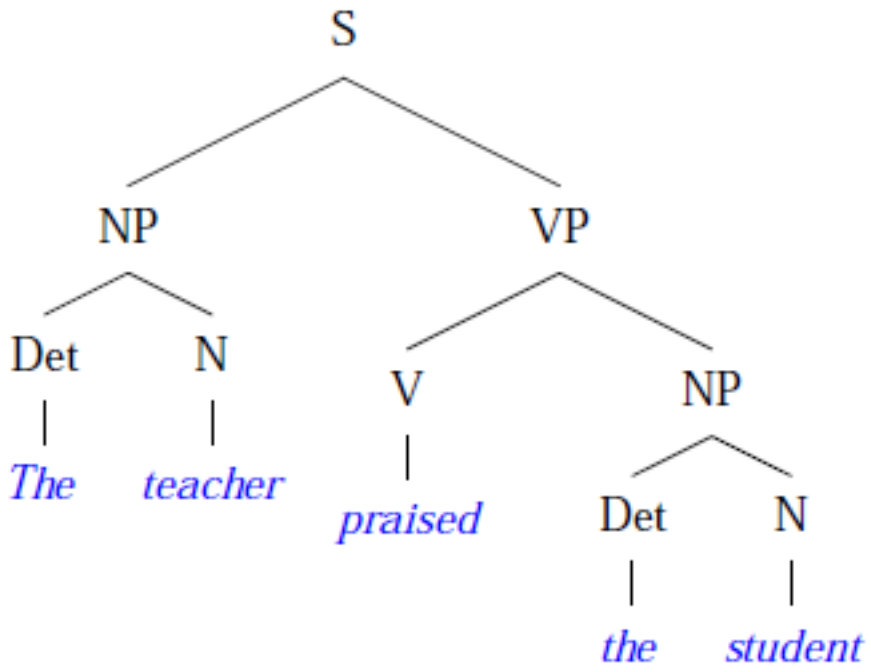
$$\xrightarrow[\text{Parsing}]{\text{Dependency}}$$



Why Dependency Parsing (DP)?

- Helpful in resolving ambiguities.
- Useful for extracting information due to relationship between words
- Provides input to many NLP tasks, like Semantic Role Labeling, Question Answering, etc.
- Useful for analyzing languages with free word order

Constituent Parsing (CFG Parsing)



- Sentences divided into constituent phrases
- Relationship between words is not binary as in Dependency parsing
- Defined by context free grammar rules

S -> NP VP
NP -> N Det
N -> dog, teacher
VP -> V NP

Dependency Parsing (DP)

$$S(w_0, w_1, ..., w_n) \xrightarrow[\text{Parsing}]{\text{Dependency}} G(V, A)$$

Dependency Parsing (DP)

$$\underbrace{S\,(\,\underbrace{w_0}_{\text{ROOT}}\;,\;\underbrace{w_1\,,\ldots,\,w_n}_{\text{words in the sentence}}\,)}_{\text{Sentence}} \xrightarrow[\text{Parsing}]{\text{Dependency}} \underbrace{G(\;\underbrace{V}_{\text{nodes/vertices}}\;,\;\underbrace{A}_{\text{edges/arcs}}\;)}_{\text{Labeled Directed Graph}}$$

Terminology

$$\underbrace{S\,(\,\underbrace{w_0}_{\text{ROOT}}\,,\,\underbrace{w_1\,,\ldots,\,w_n}_{\text{words in the sentence}}\,)}_{\text{Sentence}} \xrightarrow[\text{Parsing}]{\text{Dependency}} \underbrace{G(\,\underbrace{V}_{\text{nodes/vertices}}\,,\,\underbrace{A}_{\text{edges/arcs}}\,)}_{\text{Labeled Directed Graph}}$$

Dependency Relations

$$R = \{r_1, r_2, ..., r_m\}$$

Nodes

$$V \subseteq \{w_0, w_1, ..., w_n\}$$

Arcs

$$A \subseteq V \times R \times V$$

Terminology

$$\underbrace{S\,(\,\underbrace{w_0}_{\text{ROOT}}\,,\,\underbrace{w_1\,,\ldots,\,w_n}_{\text{words in the sentence}}\,)}_{\text{Sentence}} \xrightarrow[\text{Parsing}]{\text{Dependency}} \underbrace{G(\,\underbrace{V}_{\text{nodes/vertices}}\,,\,\underbrace{A}_{\text{edges/arcs}}\,)}_{\text{Labeled Directed Graph}}$$

Dependency Relations

$$R = \{r_1, r_2, ..., r_m\}$$

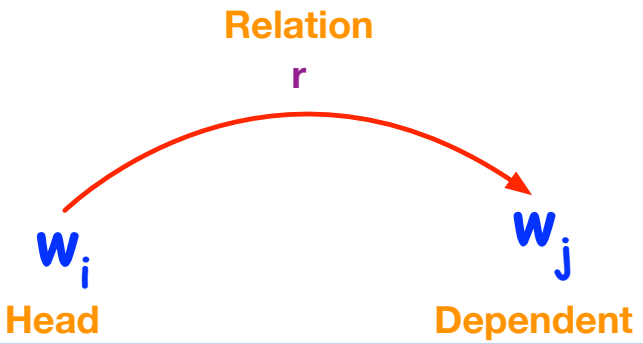
Nodes

$$V \subseteq \{w_0, w_1, ..., w_n\}$$

Arcs

$$A \subseteq V \times R \times V$$

$$(w_i, r, w_j)$$



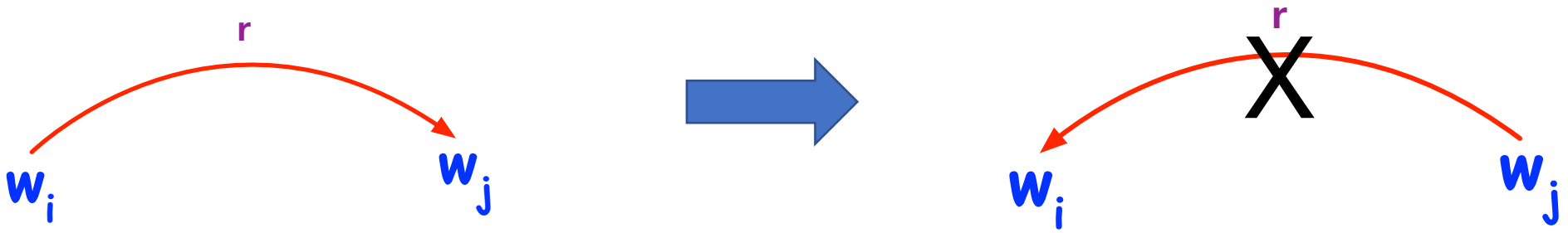
Constraints

Constraints

$$if\ (w_i, r, w_j) \in A\ then\ (w_j, r, w_i) \notin A$$

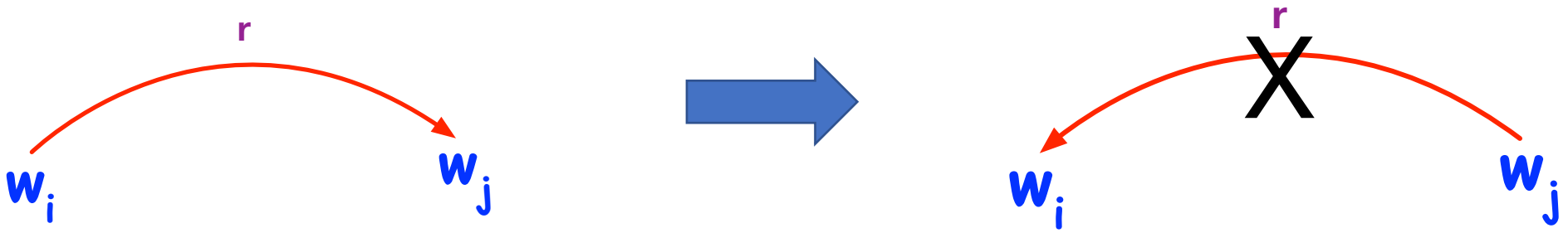
Constraints

$$if\ (w_i, r, w_j) \in A\ then\ (w_j, r, w_i) \notin A$$



Constraints

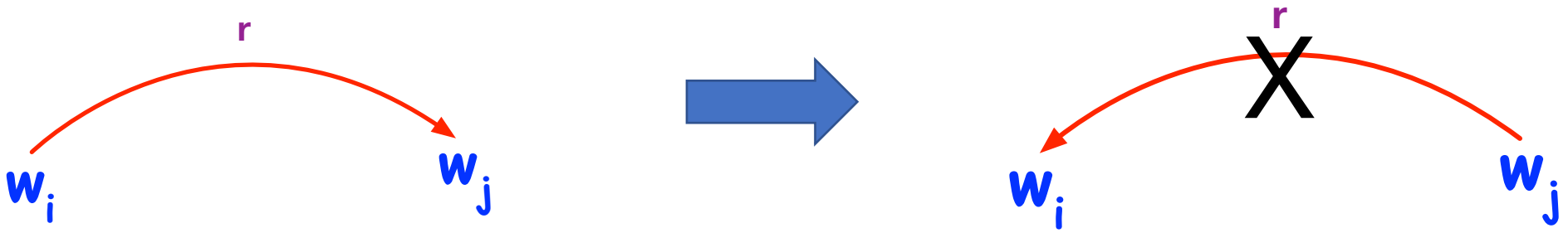
$$if\ (w_i, r, w_j) \in A\ then\ (w_j, r, w_i) \notin A$$



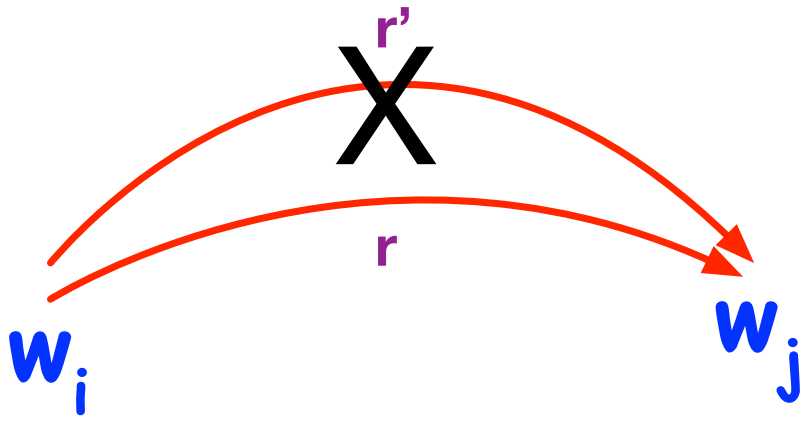
$$if\ (w_i, r, w_j) \in A\ then\ (w_i, r', w_j) \notin A\ \forall\ r' \neq r$$

Constraints

$$if\ (w_i, r, w_j) \in A\ then\ (w_j, r, w_i) \notin A$$

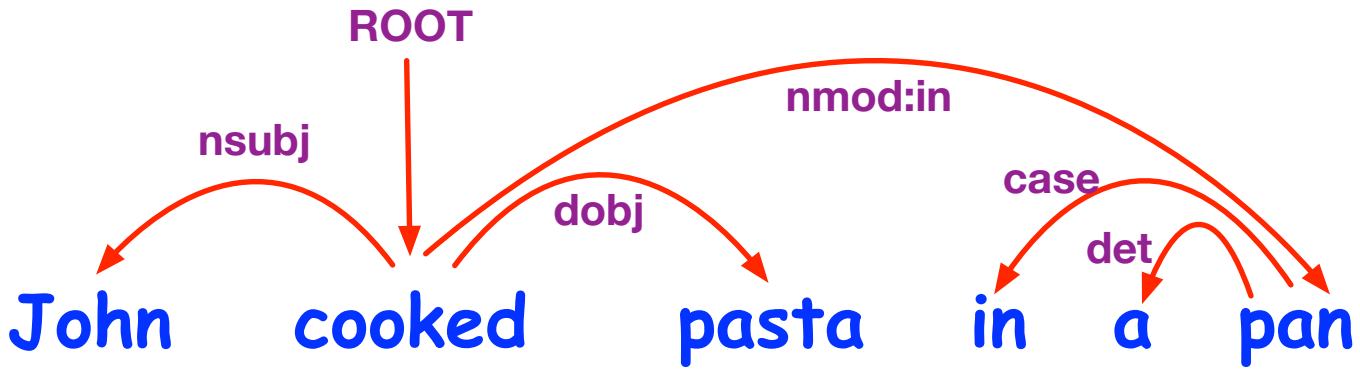


$$if\ (w_i, r, w_j) \in A\ then\ (w_i, r', w_j) \notin A\ \forall\ r' \neq r$$



Dependency Tree

Spanning Node Set: Node set having all the words of the sentence (V_S)



$$G = (V, A)$$

$$V = V_S = \{ROOT,\ John,\ cooked,\ pasta,\ in,\ a,\ pan\}$$

$$A = \{(ROOT,\ PRED,\ cooked), (cooked,\ nsubj,\ John)$$
$$(cooked,\ dobj,\ pasta), (cooked,\ nmod:in,\ pan)$$
$$(pan,\ det,\ a), (pan,\ case,\ in)\}$$

Dependency Tree

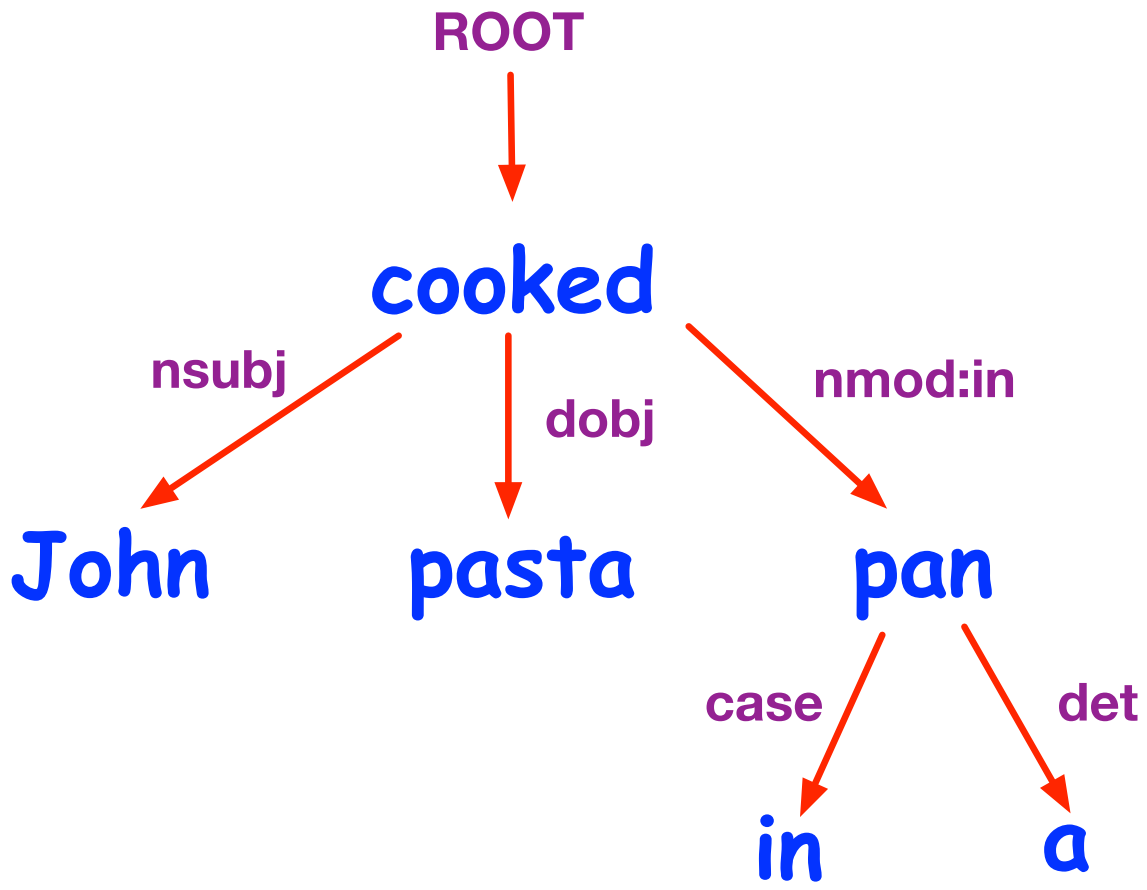
Well-Formed Dependency Graph *G(V,A)* for an input sentence *S* and dependency relation set *R* is any dependency graph that is a directed tree originating out of node w_0 *(ROOT)* and having spanning node set $V = V_S$

Well formed dependency graphs are called as ***Dependency Trees***

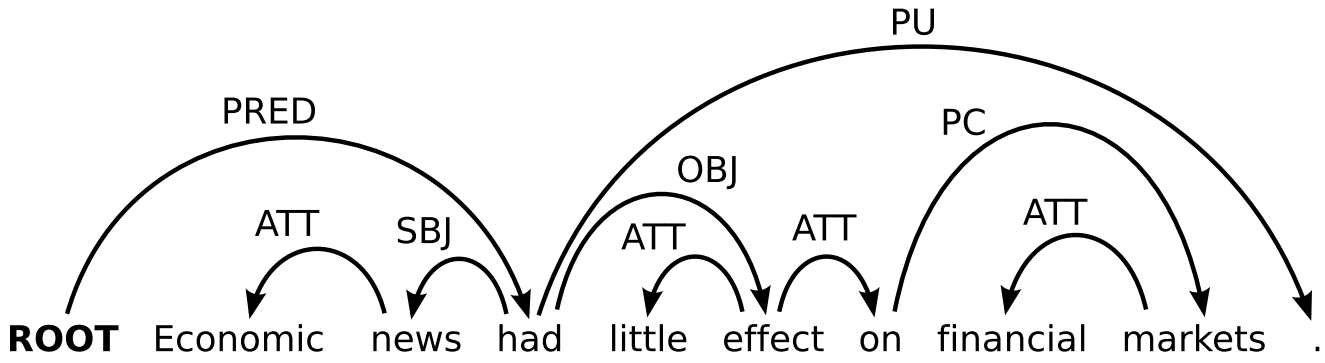
In this lecture, we consider only those parsing systems where dependency graphs are dependency trees

Dependency Tree

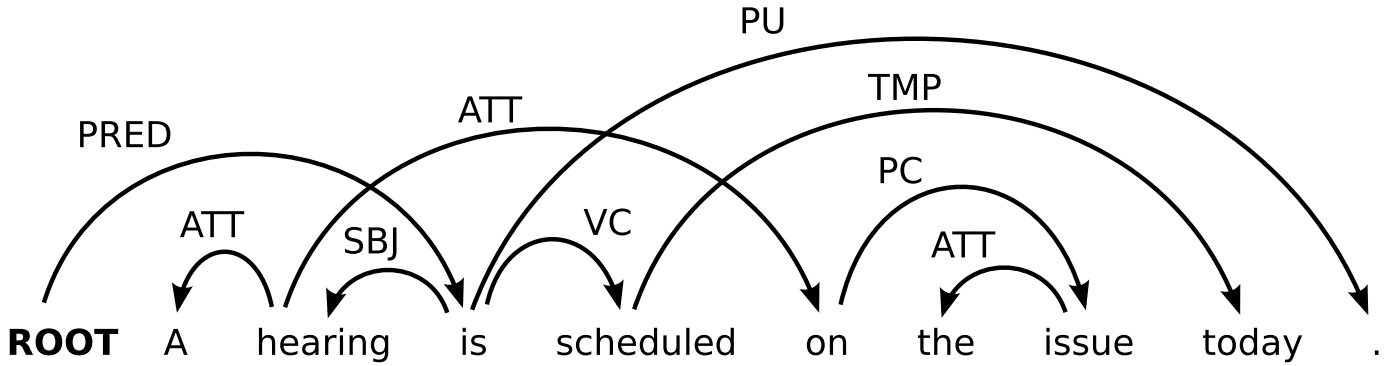
In this lecture, we consider only those parsing systems where dependency graphs are dependency trees



Projective vs Non-Projective Dependency Parse

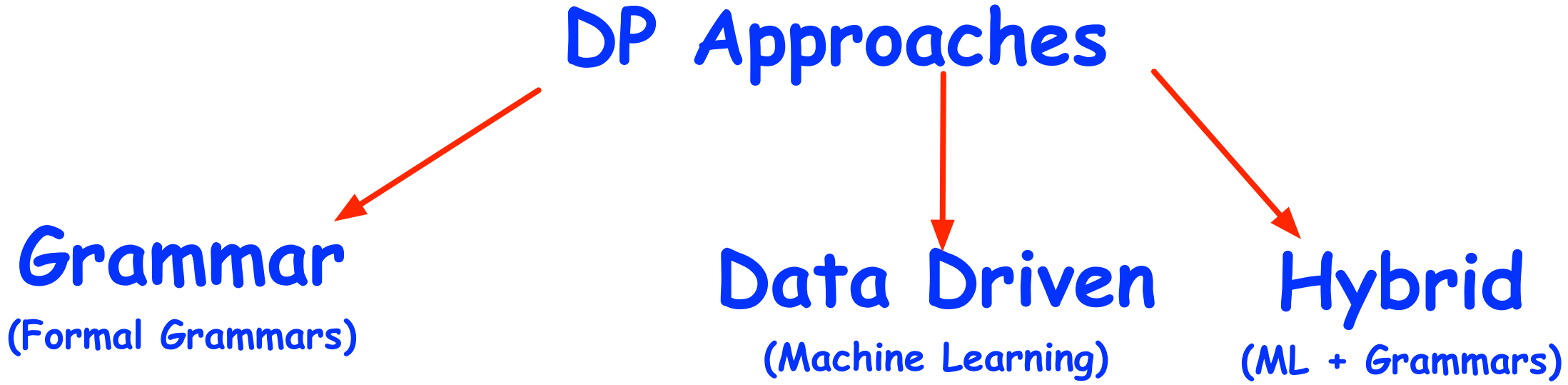


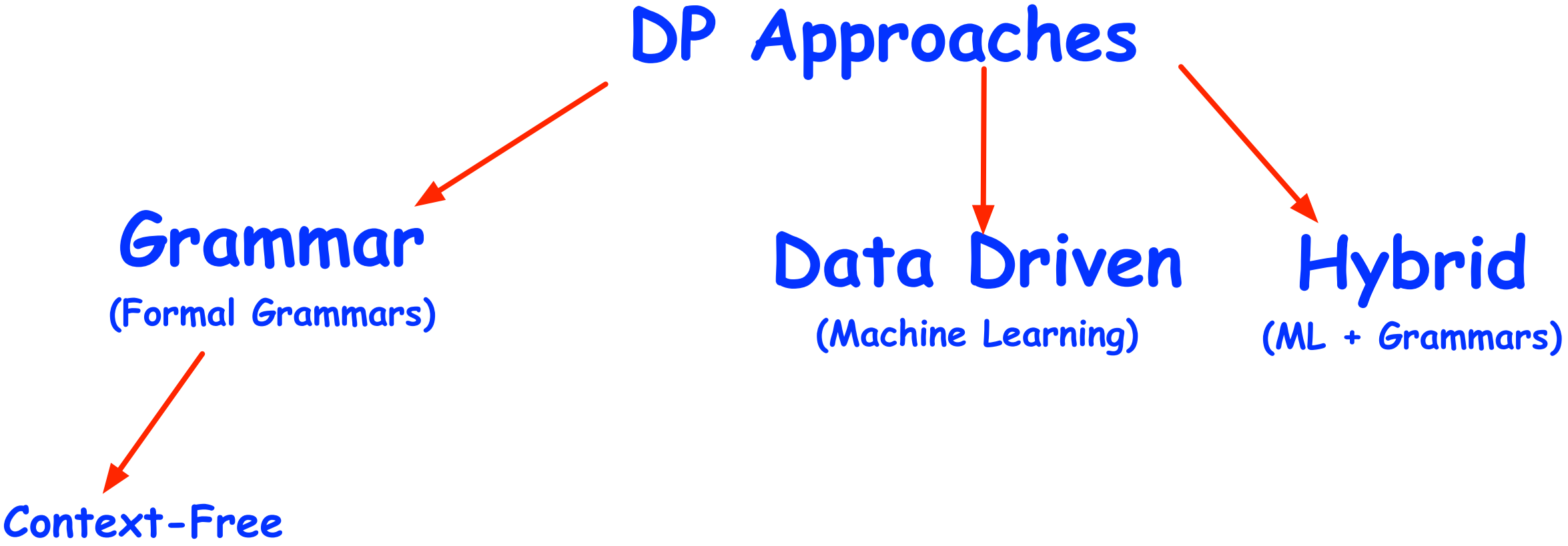
Projective Dependency Tree



Non-Projective Dependency Tree

DP Approaches


DP Approaches
Grammar
(Formal Grammars)
Data Driven
(Machine Learning)
Hybrid
(ML + Grammars)


DP Approaches
Grammar
(Formal Grammars)
Context-Free
Data Driven
(Machine Learning)
Hybrid
(ML + Grammars)

DP Approaches

Grammar
(Formal Grammars)

Data Driven
(Machine Learning)

Hybrid
(ML + Grammars)

Context-Free

- Exploits a mapping from dependency structures to context-free phrase structure representations.
- Reuses parsing algorithms originally developed for CFG
- E.g. Chart parser, shift reduce parser, etc.

DP Approaches

Grammar
(Formal Grammars)

Data Driven
(Machine Learning)

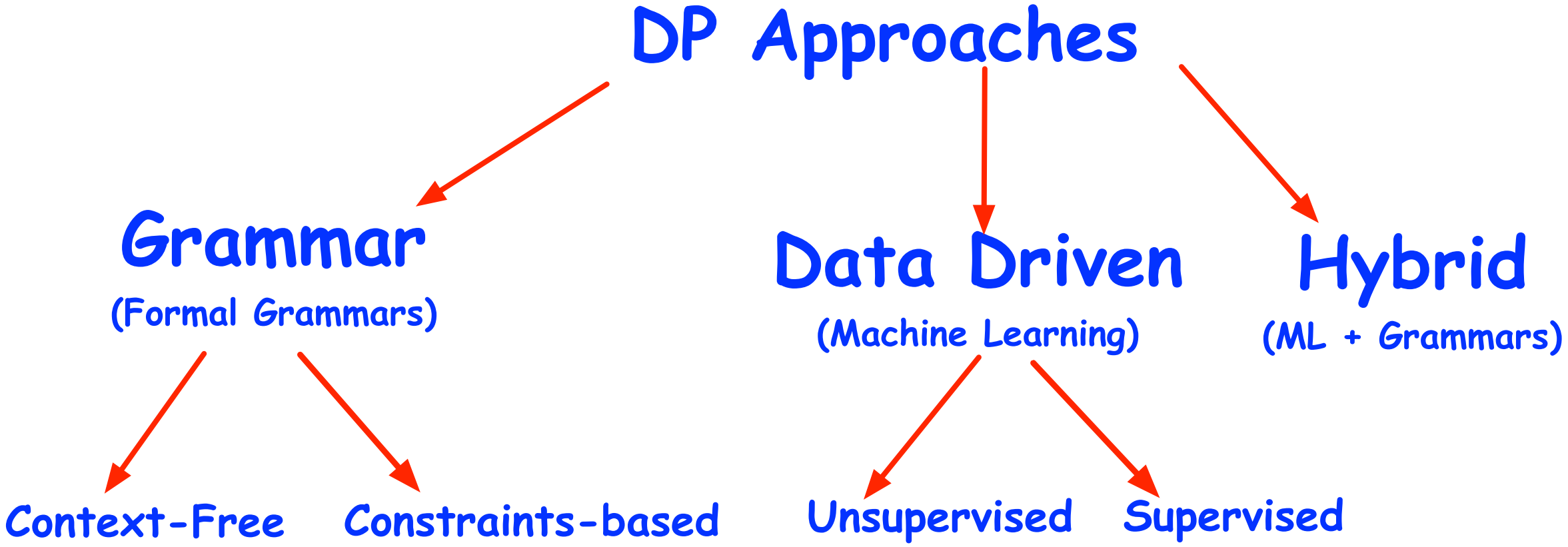
Hybrid
(ML + Grammars)

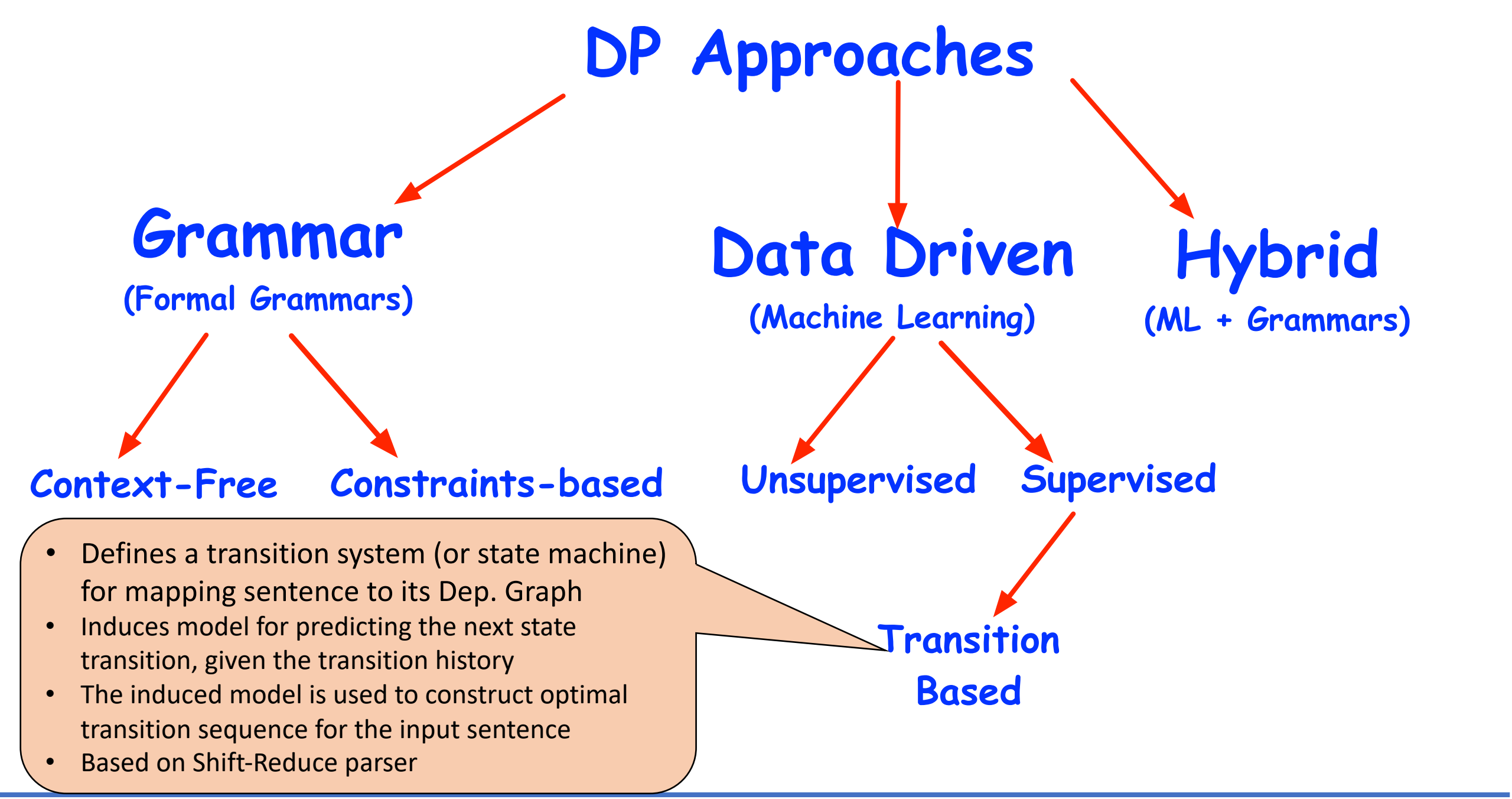
Context-Free

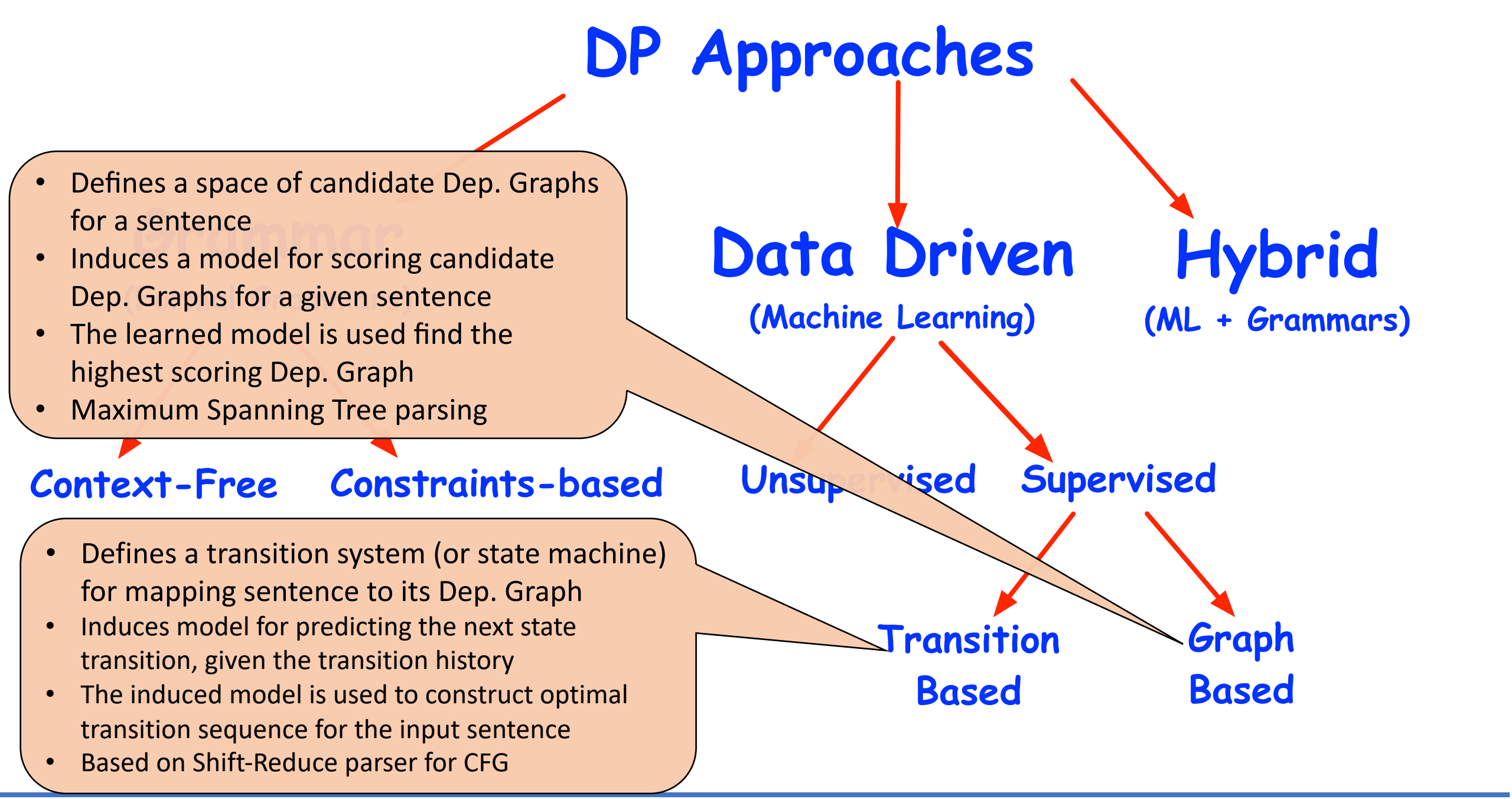
- Exploits a mapping from dependency structures to context-free phrase structure representations.
- Reuses parsing algorithms originally developed for CFG
- E.g. Chart parser, shift reduce parser, etc.

Constraints-based

- Views parsing as constraint satisfaction problem
- Grammar is set of constraints on well-formed Dep. Graphs
- Some approaches allow soft weighted constraints and score Dep. Graphs by combination of weights of constraints violated by the graph.


DP Approaches
Grammar
(Formal Grammars)
Data Driven
(Machine Learning)
Hybrid
(ML + Grammars)
Context-Free
Constraints-based
Unsupervised
Supervised


DP Approaches
Grammar
(Formal Grammars)
Data Driven
(Machine Learning)
Hybrid
(ML + Grammars)
Context-Free
Constraints-based
Unsupervised
Supervised
Transition Based
• Defines a transition system (or state machine) for mapping sentence to its Dep. Graph
• Induces model for predicting the next state transition, given the transition history
• The induced model is used to construct optimal transition sequence for the input sentence
• Based on Shift-Reduce parser


DP Approaches
Data Driven
(Machine Learning)
Hybrid
(ML + Grammars)
Unsupervised
Supervised
Context-Free
Constraints-based
Transition Based
Graph Based
• Defines a space of candidate Dep. Graphs for a sentence
• Induces a model for scoring candidate Dep. Graphs for a given sentence
• The learned model is used find the highest scoring Dep. Graph
• Maximum Spanning Tree parsing
• Defines a transition system (or state machine) for mapping sentence to its Dep. Graph
• Induces model for predicting the next state transition, given the transition history
• The induced model is used to construct optimal transition sequence for the input sentence
• Based on Shift-Reduce parser for CFG

Dependency Parsing Model

Dependency Parsing Model

$$\mathcal{M} = (\Gamma,\ \lambda,\ h)$$

Dependency Parsing Model

Dependency Parsing Model

$$\mathcal{M} = (\Gamma,\ \lambda,\ h)$$

Set of constraints Defines set of permissible Dep. Trees

Dependency Parsing Model

Dependency Parsing Model

$$\mathcal{M} = (\Gamma,\ \lambda,\ h)$$

Set of constraints Defines set of permissible Dep. Trees

Set of parameters

Dependency Parsing Model

Dependency Parsing Model

$$\mathcal{M} = (\Gamma,\ \lambda,\ h)$$

Set of constraints Defines set of permissible Dep. Trees

Set of parameters

Parsing Algorithm

Supervised Dependency Parsing

- Given a corpus of annotated sentences: $\mathcal{D} = \{(S_d, G_d)\}_{d=0}^{|\mathcal{D}|}$
- Learn a model to automatically generate the dependency tree for a given sentence

Supervised Dependency Parsing

- Two main computational problems:
  - **Learning:** **Given a training set of annotated sentences, induce a parsing model that can be used to parse a new sentence**

Supervised Dependency Parsing

- Two main computational problems:
  - **Learning:** **Given a training set of annotated sentences, induce a parsing model that can be used to parse a new sentence**
  - **Parsing:** **Given a parsing model and a sentence derive the optimal dependency tree**