# CHAPTER 4

# Syntactic Analysis

Syntactic analysis in NLP involves breaking down sentences into their **grammatical components**, such as nouns, verbs, adjectives, and their **relationships**, enabling machines to comprehend the structure and meaning of text

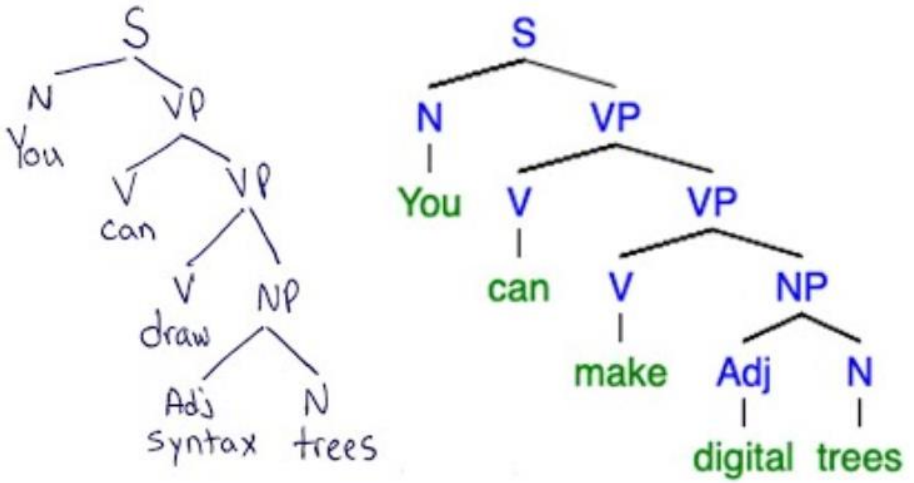
# Syntactic Analysis

- Syntax, Grammar, Parser
- Approaches
- Syntax Tree
- Constituency parsing
- Dependency parsing

# Syntax?

Syntax deals with the **arrangement** and **relationships** of **words** to form **grammatically** correct **sentences**

- Discover **rules** governing the arrangement of words in sentences to form coherent language
- Determine hierarchy and order of words, phrases and clauses



# Syntactic analysis

Syntactic analysis involves parsing a sentence to understand its grammatical structure.

**Main reasons:**

- **Parsing & Understanding:** parse sentences and understand grammatical structure
- **Ambiguity resolution:** solve ambiguities by providing interpretable structures
- **Grammar & Language generation:** generate coherent and grammatically correct sentences
- **Information extraction:** identify syntactic patterns and relationships in texts

# Syntactic analysis approaches

- **Context-Free Grammars (CFG)**

Consist of a set of rules that describe how different components of a sentence can be combined. Help generate parse trees:

  - Constituency grammars: hierarchical structure
  - Dependency grammars: relationships between words

- **Probabilistic CFG (PCFG)**

Assign probabilities to grammar rules

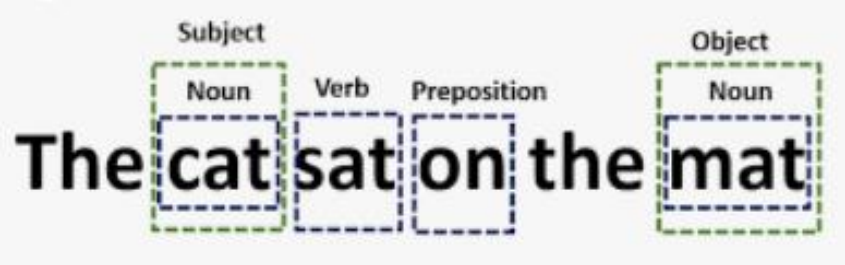
- **Neural-Based (RNN, CNN, Transformer)**

Trained on syntactically annotated Corpora

# Phraser?

Parsing is a fundamental process in syntactic analysis that involves breaking down a sentence into its grammatical components and representing them in a structured form, often as a parse tree or dependency graph

- **Parsing algorithms:**
  - Top-Down
  - Bottom-Up



# Phrase

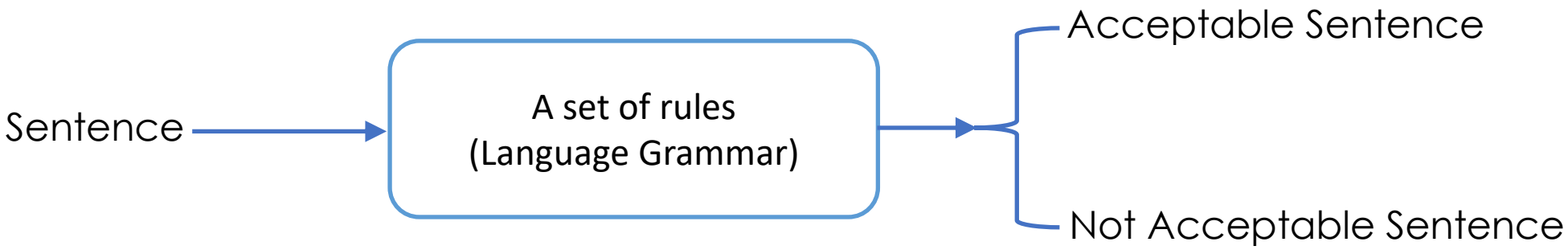
A phrase is a group of words that function together as a single unit within a sentence.

A phrase can consist of a single word or a combination of words.

| | | |
|---|---|---|
| **Sentence** | **S** | **الجملة** |
| **Noun phrase** | **NP** | **جملة إسمية** |
| **Verb phrase** | **VP** | **جملة فعلية** |
| **Prepositional phrase** | **PP** | **جملة الوصل (جار ومجرور)** |
| **Adjective phrase** | **ADJP** | **جملة الصفة (صفة وموصوف)** |
| **Adverb phrase** | **ADVP** | **جملة الحال** |
| **Noun** | **N** | **الإسم** |
| **Verb** | **V** | **الفعل** |
| **Preposition** | **P** | **حرف الجر** |
| **Conjunction** | **CONJ** | **أداة الوصل (حرف العطف)** |
| **Determiner** | **Det** | **أداة التعريف** |

# Grammar

A set of rules that help check whether a sentence belongs to a language or not




A language grammar G is defined by {S, N, T, R}:

- S: Start (Sentence)
- N: Non terminals (Phrases)
- T: Terminals (Words)
- R: Grammar rules

# Grammar

S = Ahmed wrote a book

**S ⇒ NP VP**

**NP ⇒ Det N | N**

**VP ⇒ V NP**

**N ⇒ Ahmed | book**

**V ⇒ wrote**

**Det ⇒ a**

[S [NP [N Ahmed]] [VP [V wrote] [NP[Det a][N book]]]]

# Grammar

S = Karim goes to school

**S ⇒ NP VP**

**NP ⇒ N**

**VP ⇒ V PP**

**PP ⇒ P N**

**N ⇒ Karim | school**

**V ⇒ goes**

**P ⇒ to**

[S[NP [N Karim]][VP[V goes][PP[P to][N school]]]]

# Grammar

S = كتب أحمد الدرس

**S ⇒ VP NP**

**VP ⇒ V N**

**NP ⇒ Det N**

**N ⇒ أحمد | درس**

**V ⇒ كتب**

**Det ⇒ ال**

[S[VP[V كتب][N أحمد]][NP[Det ال][N درس]]]

# Grammar

S = يذهب كريم إلى المدرسة

**S ⇒ VP PP**

**VP ⇒ V N**

**PP ⇒ P NP**

**NP ⇒ Det N**

**N ⇒ كريم | مدرسة**

**V ⇒ يذهب**

**P ⇒ إلى**

**Det ⇒ ال**

[S[VP[V يذهب][N كريم]][PP[P إلى][NP[Det ال][N مدرسة]]]]

# Syntax tree

A syntax tree (or parse tree, or constituency tree) is a graphical representation of the syntactic structure of a sentence or phrase in natural language

**Key elements:**

- **Nodes:** words and phrases
- **Edges:** relationships between words and phrases
- **Root node:** labelled as S (Sentence)
- **Leaves:** individual words
- **Branching:** hierarchical organization of words and phrases

# Syntax tree

S = Ahmed wrote a book

**S ⇒ NP VP**

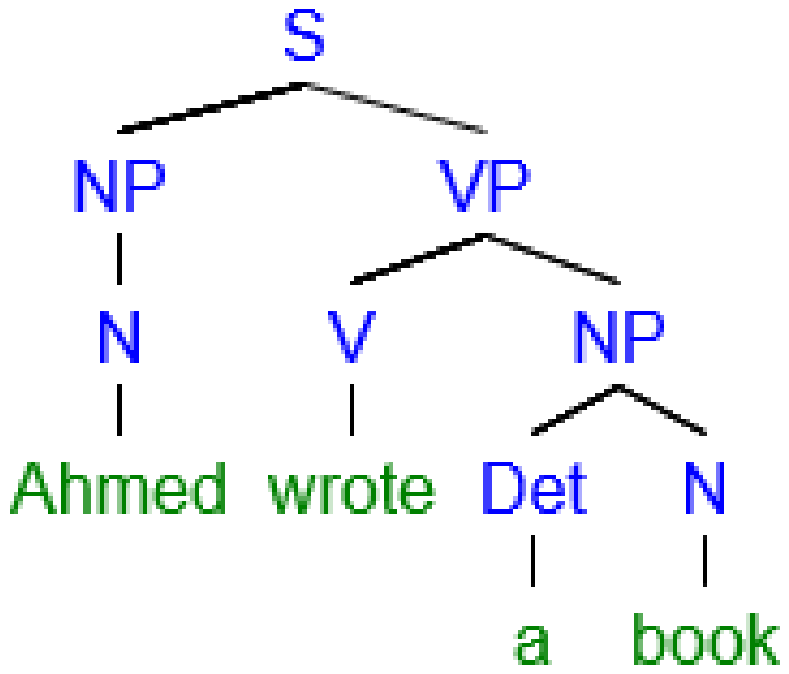
**NP ⇒ Det N | N**

**VP ⇒ V NP**

**N ⇒ Ahmed | book**

**V ⇒ wrote**

**Det ⇒ a**




[S [NP [N Ahmed]] [VP [V wrote] [NP[Det a][N book]]]]

# Syntax tree

S = Karim goes to school

**S ⇒ NP VP**

**NP ⇒ N**

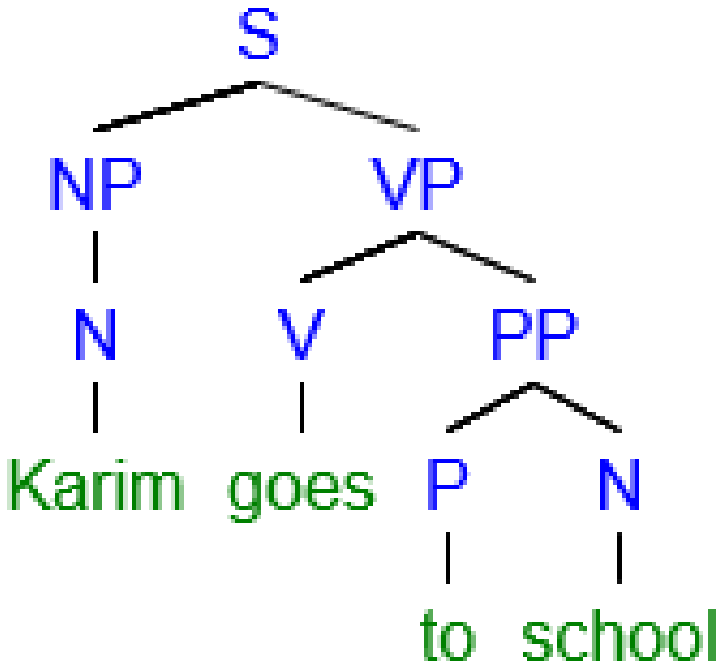
**VP ⇒ V PP**

**PP ⇒ P N**

**N ⇒ Karim | school**

**V ⇒ goes**

**P ⇒ to**




[S[NP [N Karim]][VP[V goes][PP[P to][N school]]]]

# Syntax tree

S = كتب أحمد الدرس

**S ⇒ VP NP**

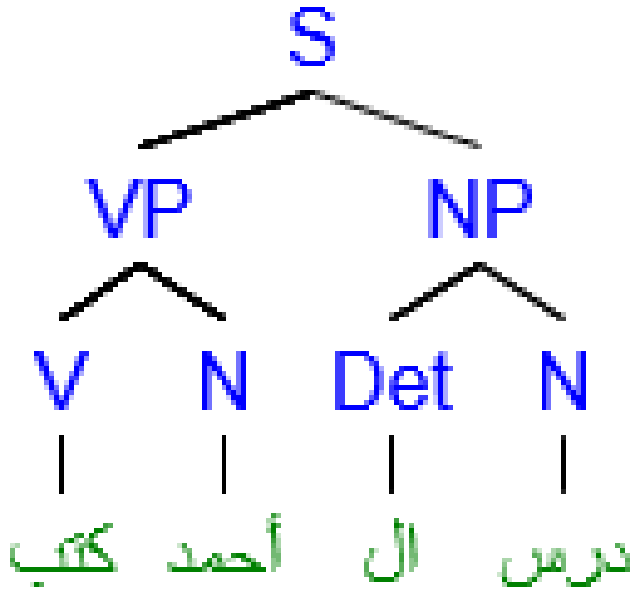
**VP ⇒ V N**

**NP ⇒ Det N**

**N ⇒ أحمد | درس**

**V ⇒ كتب**

**Det ⇒ ال**




[S[VP[V كتب][N أحمد]][NP[Det ال][N درس]]]

# Syntax tree

S = يذهب كريم إلى المدرسة

**S ⇒ VP PP**

**VP ⇒ V N**

**PP ⇒ P NP**

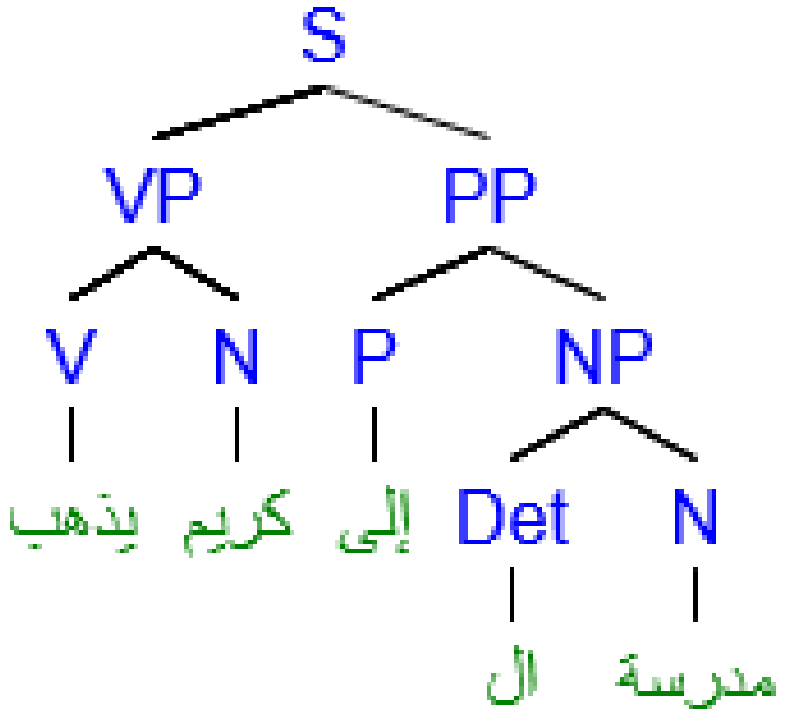
**NP ⇒ Det N**

**N ⇒ كريم | مدرسة**

**V ⇒ يذهب**

**P ⇒ إلى**

**Det ⇒ ال**




[S[VP[V يذهب][N كريم]][PP[P إلى][NP[Det ال][N مدرسة]]]]

# Syntax tree

S = The quick fox jumped over the lazy dog

**S ⇒ NP VP**

**NP ⇒ Det ADJ N**

**VP ⇒ V PP**

**PP ⇒ P NP**

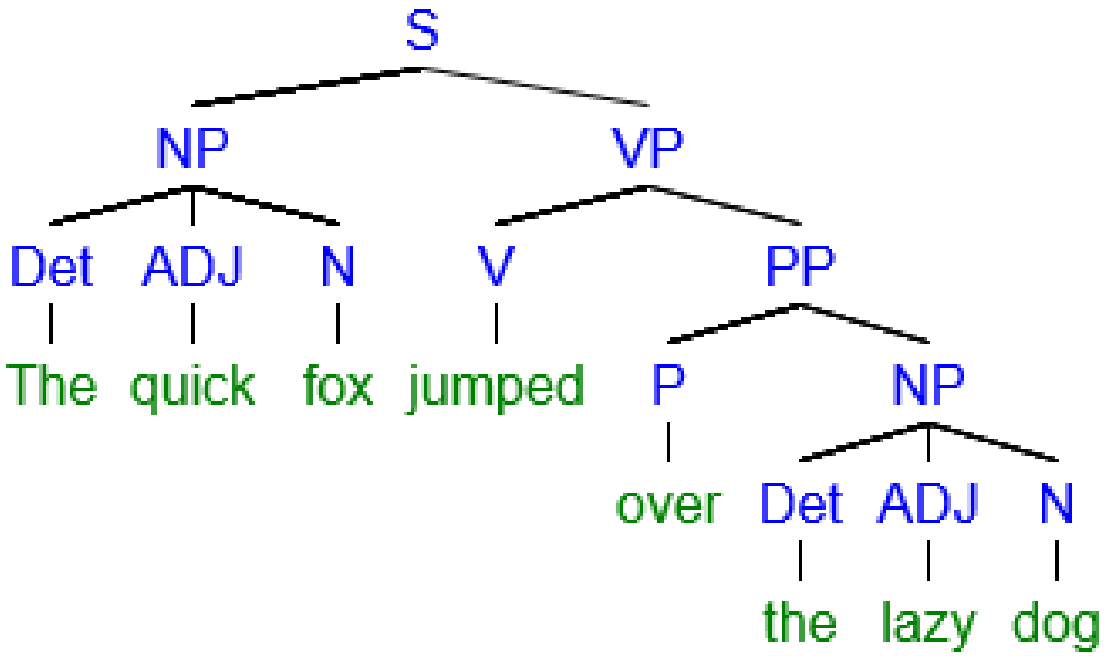
**N ⇒ fox | dog**

**V ⇒ jumped**

**ADJ ⇒ quick | lazy**

**P ⇒ over**

**Det ⇒ the**




[S[NP[Det The][ADJ quick][N fox]][VP[V jumped][PP[P over][NP[Det the][ADJ lazy][N dog]]]]

# Syntax tree

S = The quick fox and the lazy dog

**S ⇒ NP CONJ NP**

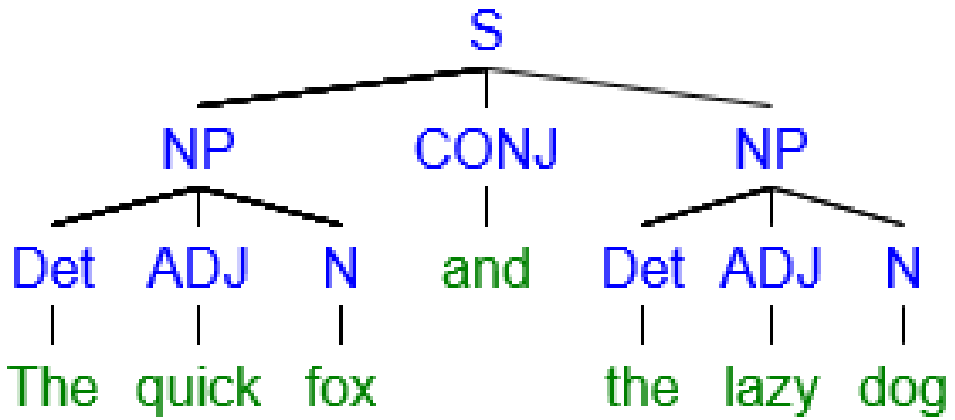
**NP ⇒ Det ADJ N**

**CONJ ⇒ and**

**N ⇒ fox | dog**

**ADJ ⇒ quick | lazy**

**Det ⇒ the**




[S[NP[Det The][ADJ quick][N fox]][CONJ and][NP[Det the][ADJ lazy][N dog]]]

# Syntax tree

Exercise: Generate syntax trees for the following sentences:

S1 = The tiger is extremely dangerous

S2 = يذهب محمد إلى مدرسة الحي

S3 = جاء الرجل الصالح

S4 = رأيت الرجل واقفا

# Dependency parsing

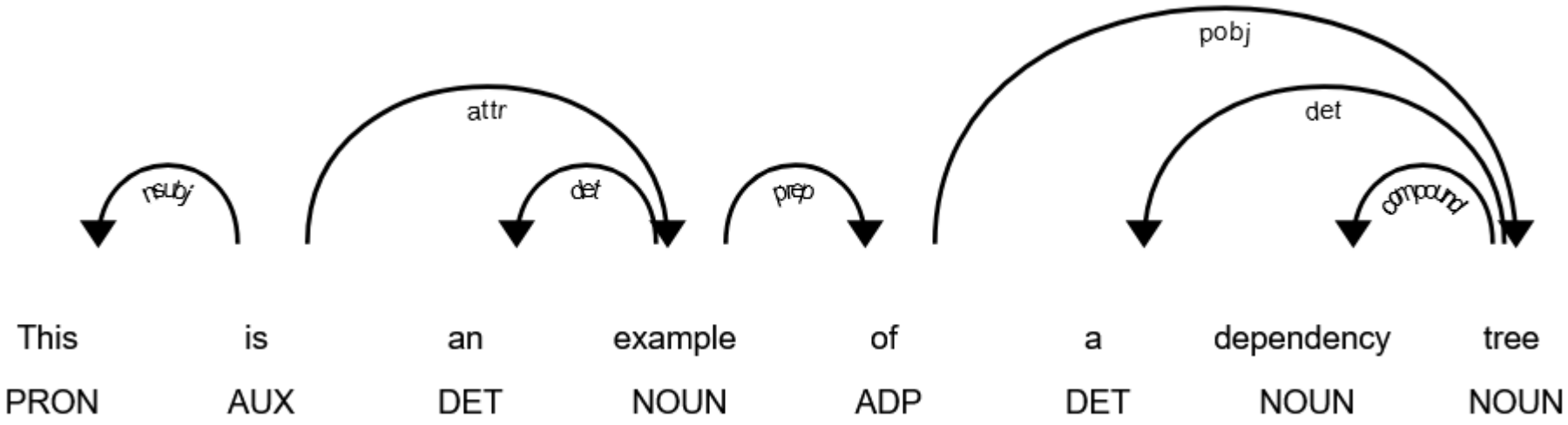
Dependency parsing consists of discovering **dependencies** between **words** in a sentence.

Dependencies are **word-word relations** or links that are typically asymmetric. These relations are established based on the positional relationship between words within a sentence:

- Unravelling the relationships between words in a sentence
- Analyzing how words depend on one another
- Representing syntactic and semantic relationships within the sentence

# Dependency Tree

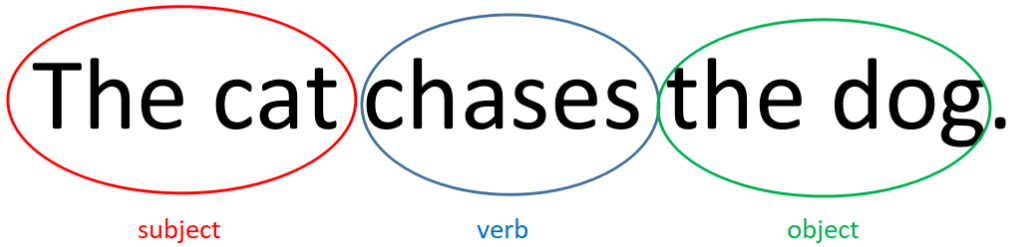
Dependency trees consist of **nodes** and directed **edges**. Each word in a sentence is represented as a node, and the relationships between these words are depicted as directed edges connecting the nodes. These directed edges indicate which word **governs** or **modifies** another word



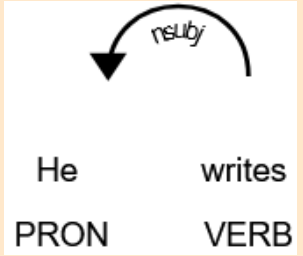
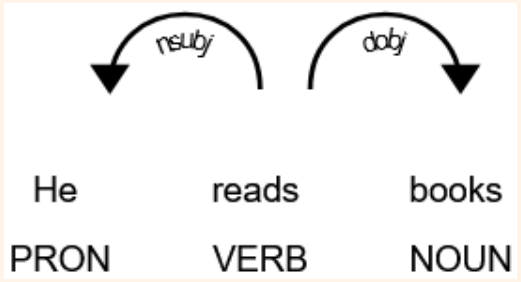
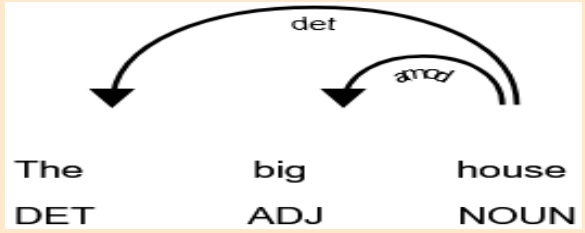
# Dependency Relationships

Discovering dependency relationships helps us answer essential questions about a sentence's structure:

- What is the subject of the sentence**?**
- Which words are objects of verbs**?**
- How do adjectives modify nouns**?**
- What are the adverbs modifying**?**



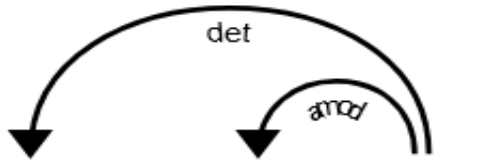
# Main Dependency Relationships

| Dependency | Example |
| --- | --- |
| **Subject-Verb:** represents the relationship between the subject and the governing verb. (e.g: in the sentence "**He wrote**" "He" has a subject-verb dependency with "wrote." |  |
| **Object-Verb:** indicates the direct object of a verb. (e.g: In "**He reads books**," "books" has an object-verb dependency with "reads." |  |
| **Modifier-Head:** captures how words modify other words. (e.g: In "**The big house**," "big" has a modifier-head dependency with "house." |  |
| **Adverbial-Verb:** demonstrates how adverbs modify verbs. (e.g: In "**He writes well**," "well" has an adverbial-verb dependency with "writes." |  |

# Universal Dependency Types

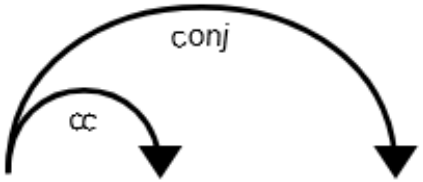
| | | |
|---|---|---|
| **det** | Determiner relationship between a nominal head and the determiner | **The** red book |
| **amod** | Adjectival modifier that modifies the meaning of a noun | The **big** house |
| **advmod** | Adverbal modifier that modifies the meaning of a verb | He writes **well** |
| **nsubj** | Entity that acts as the subject or agent in a clause | **She** eats apples |
| **cc/conj** | Linkages related to words connected by coordinating conjunctions | Blood **and** bone |
| **aux** | Auxiliary or secondary verb in the clause. | He **is** reading a book |
| **acomp** | Adjective complement, complementing or modifying the verb | She is afraid **of spiders** |
| **prep** | Prepositional modifier that modifies nouns, verbs, adjectives, or prepositions | **Over** the table |
| **pobj** | Object of a preposition, typically the head of a noun phrase following a preposition | Over the **table** |
| **dobj** | Direct object, represents the noun or noun phrase that directly receives the action of the verb | She eats **apples** |

# Universal Dependency Types




| The | big | house |
|---|---|---|
| DET | ADJ | NOUN |


| He | writes | well |
|---|---|---|
| PRON | VERB | ADV |




| Blood | and | bone |
|---|---|---|
| NOUN | CCONJ | NOUN |

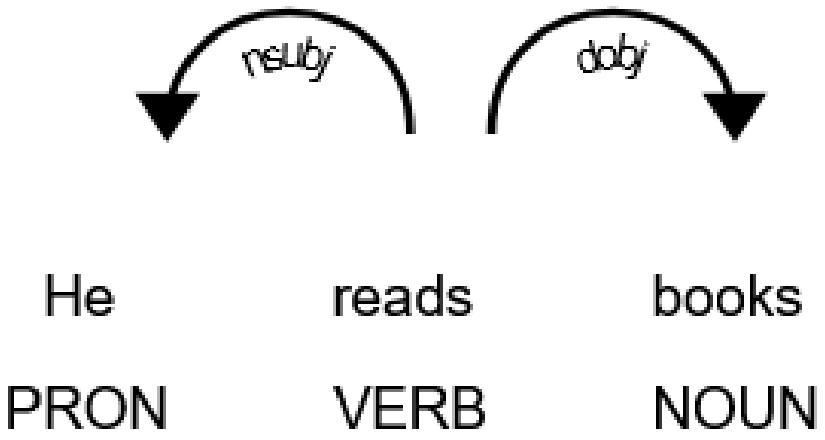

| He | is | reading | a | book |
|---|---|---|---|---|
| PRON | AUX | VERB | DET | NOUN |

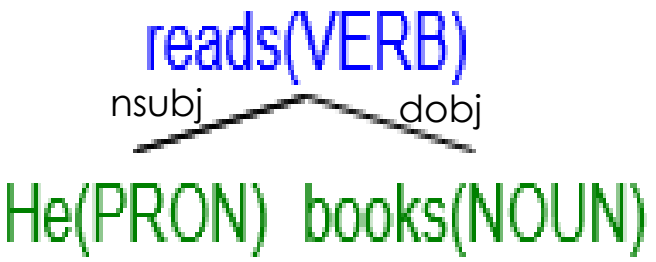

| She | is | afraid | of | spiders |
|---|---|---|---|---|
| PRON | AUX | ADJ | ADP | NOUN |

# Dependency parsing

## Examples

He reads books

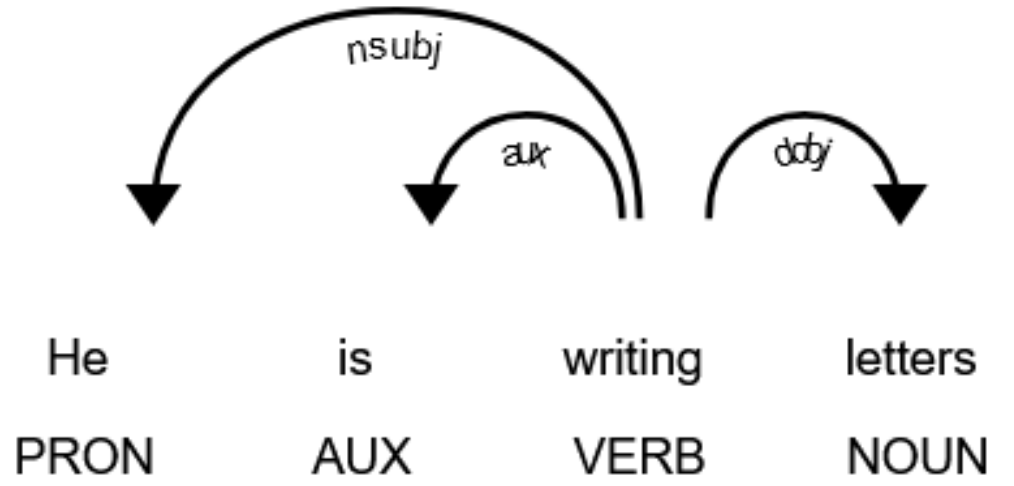


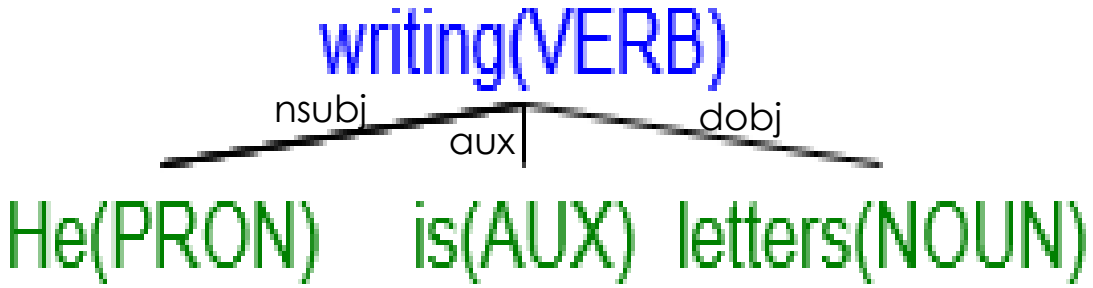



# Dependency parsing

## Examples

He is writing letters

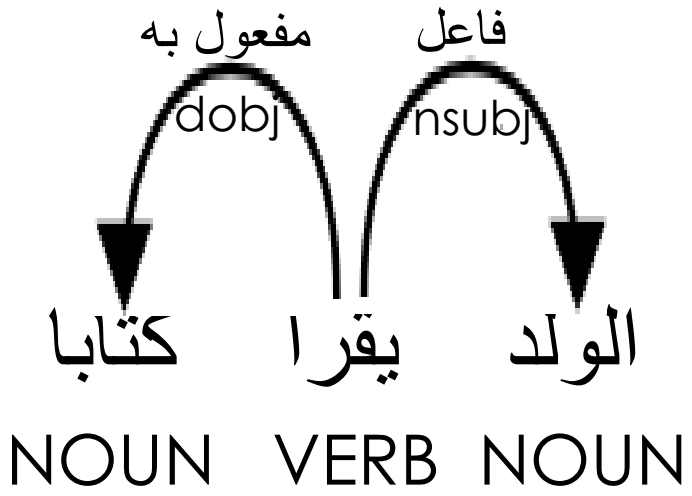


# Dependency parsing

## Examples

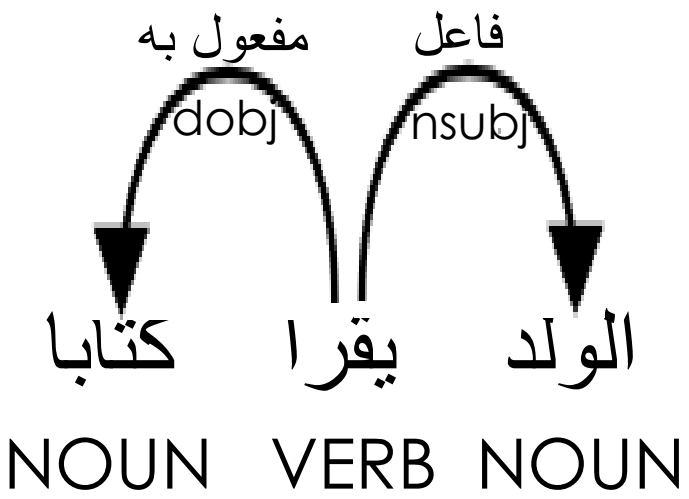
يقرأ الولد كتابا

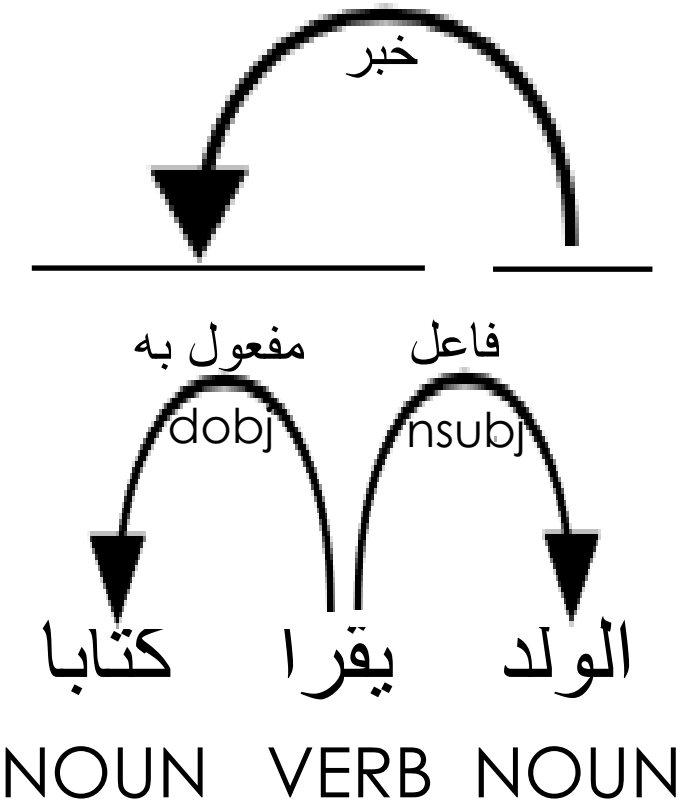



# Dependency parsing

## Examples

الولد يقرأ كتابا

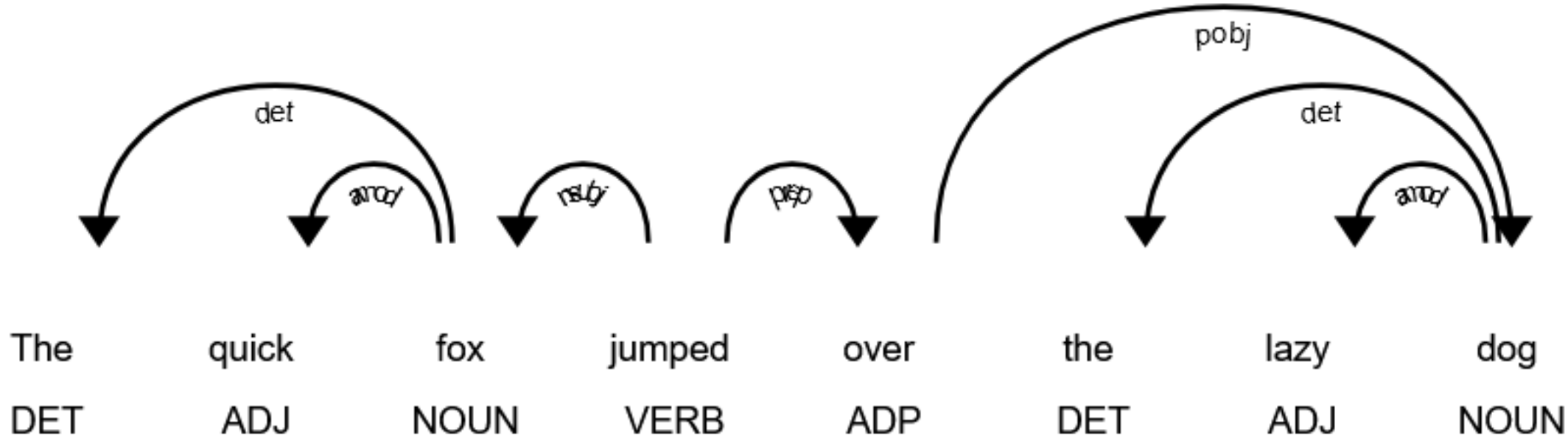


# Dependency parsing

## Examples

The quick fox jumped over the lazy dog



# Dependency parsing

## Examples

The quick fox jumped over the lazy dog