Computational Linguistics (CL) and Natural Language Processing (NLP)

CL&NLP

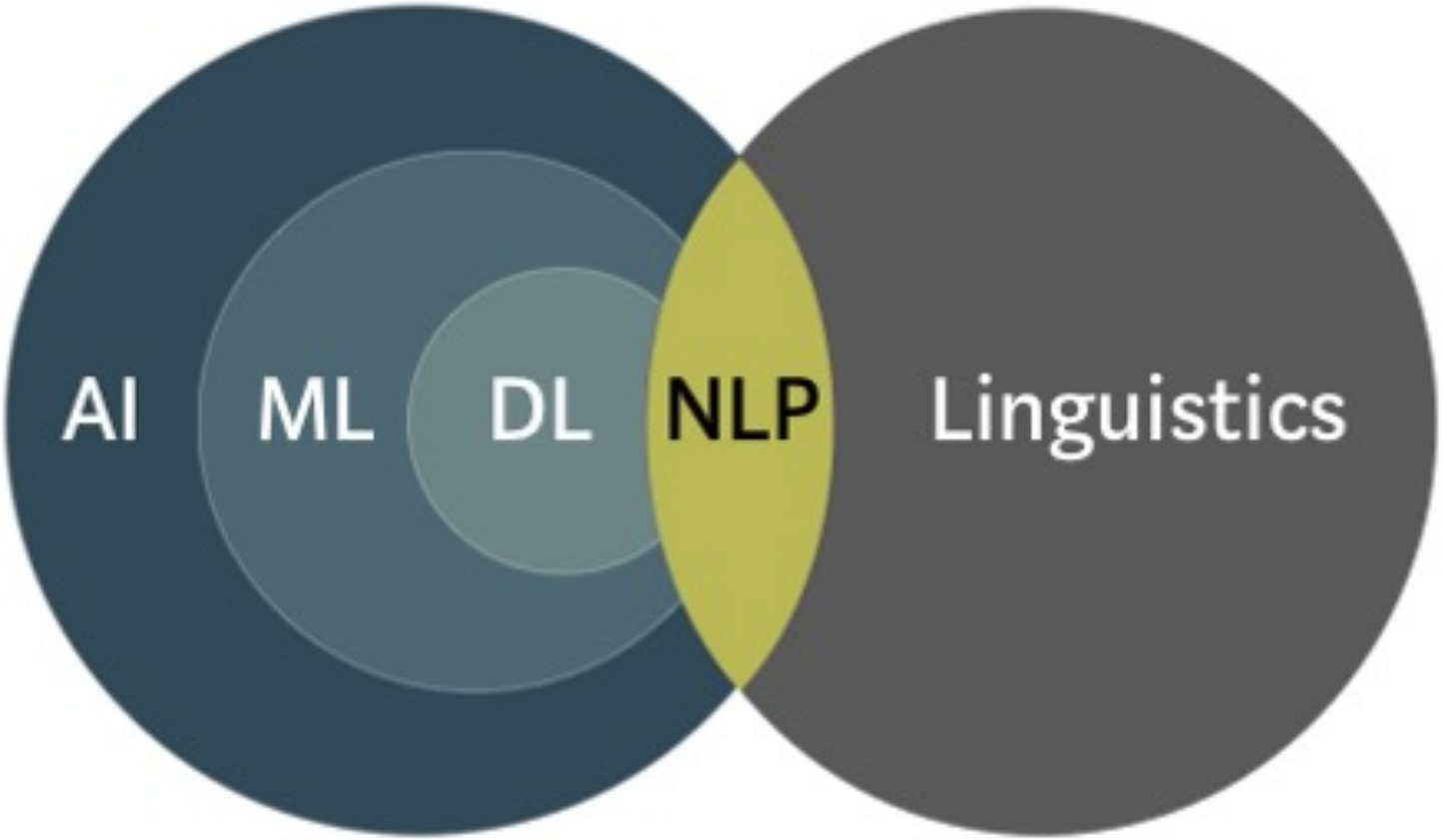
- Introduction to CL & NLP-
- What is Computational Linguistics?-
- What is NLP?
- Applications of CL and NLP in real-world systems
- Differences and similarities between CL and NLP
- Basic NLP Pipeline: Text pre-processing, tokenization, lemmatization, stemming-
- Linguistic essentials: Phonology, Morphology, Syntax, Semantics

Difference between CL and NLP

Imagine a civil engineer and a construction worker:

- **CL** is like the **civil engineer** who designs bridges, calculates loads, and understands the underlying physics and mathematics.
- **NLP** is like the **construction worker** who builds the bridge using those designs and tools to achieve a specific, practical goal (a functioning bridge).

In essence, CL and NLP are two sides of the same coin – studying and leveraging the power of language in the digital world.


AI
ML
DL
NLP
Linguistics

Basic NLP Pipeline

- Text pre-processing,
- Tokenization,
- Lemmatization,
- Stemming

1. Text Preprocessing

The goal is to **clean and normalize** the text.

🔧 **Tasks involved:**

- **Lowercasing**: Makes all characters lowercase for uniformity.

  > "Apple is Tasty" → "apple is tasty"

- **Removing Punctuation**: Helps reduce noise.

  > "Hello, world!" → "Hello world"

- **Removing Stop Words**: Words that don't carry much meaning (e.g., "is", "the", "a").

  > "This is a cat" → "cat"

- **Removing Numbers/Special Characters**: Optional, depending on the use case.

  > "I scored 95%" → "I scored"

- **Correcting Spellings**: "recieve" → "receive" (can be done with libraries like `TextBlob`, `SymSpell`)

---

2. Tokenization

Tokenization breaks text into **smaller meaningful units** (tokens).

🧱 Types of Tokenization:

- **Word Tokenization**:

  > "ChatGPT helps students." → ["ChatGPT", "helps", "students", "."]

- **Sentence Tokenization**:

  > "NLP is cool. It's useful." → ["NLP is cool.", "It's useful."]

- Tools: `nltk.word_tokenize()`, `spaCy`, `split()` in Python

3. Lemmatization

Lemmatization reduces a word to its **base/dictionary form** called a *lemma*, considering **grammar and context**.

🔍 **Example:**

Word	Lemma
running	run
better	good
studies	study

- More accurate than stemming
- Requires POS tagging (e.g., `spaCy`, `WordNetLemmatizer` in NLTK)

---

4. Stemming

Stemming cuts off **prefixes or suffixes** to reach the root word, but doesn't ensure the root is a real word.

🔧 **Example:**

Word	Stem
playing	play
studies	studi
caring	care

- Faster but less accurate
- Tools: `PorterStemmer`, `SnowballStemmer` (in NLTK)

♦ Key Differences Table

Feature	Stemming	Lemmatization
Output	Root form (maybe invalid)	Dictionary form (valid word)
Technique	Rule-based cutting	Vocabulary + Morphology
Accuracy	Lower	Higher
Speed	Faster	Slower
Example	`caring` → `car`	`caring` → `care`

Linguistic essentials

- Phonology
- Morphology
- Syntax
- Semantics

1. Phonology – *Sound system of a language*

- **Definition**: The study of the sound patterns and systems of languages.
- **Focus**: How speech sounds function in a language (not their physical production).
- **Example**: In English, the plural "-s" in "cats" /s/ vs. "dogs" /z/ follows phonological rules.
- **Application in NLP**: Useful in speech recognition and text-to-speech systems.

2. Morphology – *Structure of words*

- **Definition**: The study of the internal structure of words and how they are formed.
- **Focus**: Morphemes – the smallest meaning-carrying units (e.g., *un-*, *happy*, *-ness*).
- **Types**:
  - **Inflectional** (e.g., talk → talked)
  - **Derivational** (e.g., happy → happiness)
- **Application in NLP**: Stemming, lemmatization, morphological analysis.

3. Syntax – *Structure of sentences*

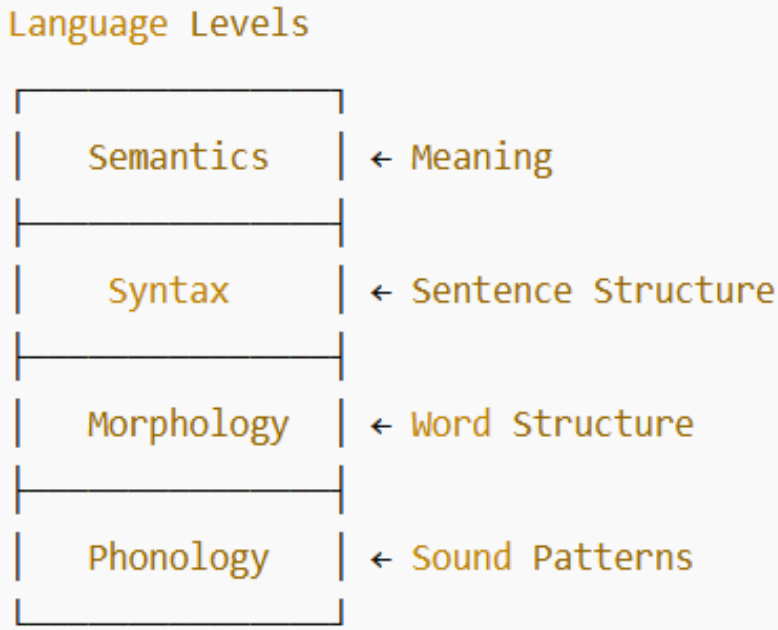
- **Definition**: The study of how words combine to form grammatical sentences.
- **Focus**: Sentence structure and word order rules.
- **Example**: "The cat sat on the mat" is correct; "Cat the mat on sat" is not.
- **Application in NLP**: Parsing, grammar checking, machine translation.

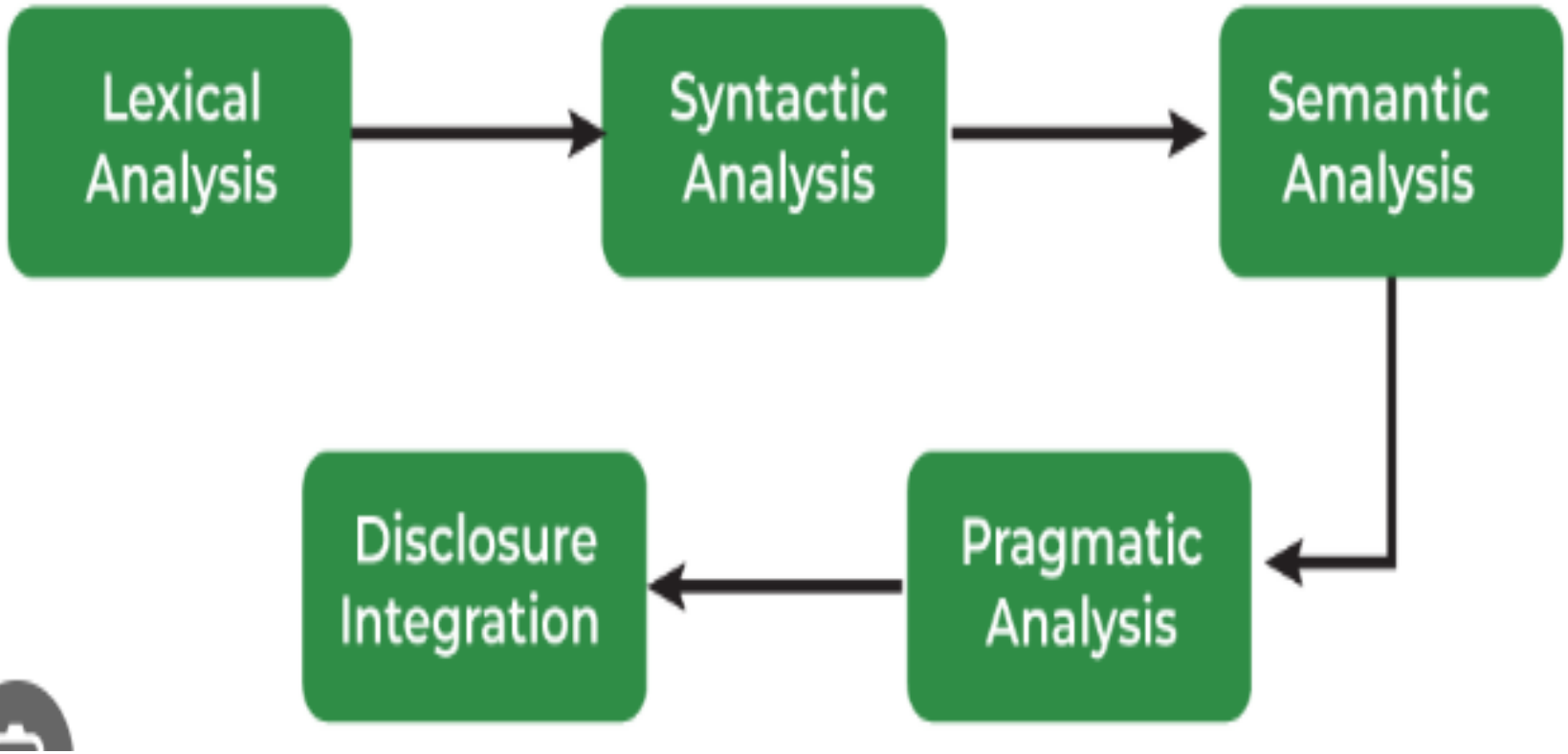
4. Semantics – *Meaning of words and sentences*

- **Definition**: The study of meaning in language.
- **Focus**: Literal meanings of words, phrases, and sentences.
- **Example**: "He kicked the bucket" literally vs. idiomatic meaning ("he died").
- **Application in NLP**: Word sense disambiguation, semantic analysis, question answering.

Diagram Summary:

mathematica




Lexical Analysis
Syntactic Analysis
Semantic Analysis
Pragmatic Analysis
Disclosure Integration

♦ 1. Lexical Analysis

- **What it does**: Breaks down the input text into tokens (words, punctuation).
- **Example**: `"I love NLP."` → [`I`, `love`, `NLP`, `.`]
- **Focus**: Identifying the basic units of language.

♦ 2. Syntactic Analysis (Parsing)

- **What it does**: Analyzes the grammatical structure of a sentence.
- **Example**: Recognizes parts of speech and sentence structure (subject-verb-object).
- **Focus**: Ensures the sentence follows grammar ru ↓

♦ 3. **Semantic Analysis**

- **What it does**: Determines the **meaning** of words and phrases in context.
- **Example**: Understanding that "bank" means a financial institution (not a river bank).
- **Focus**: Word meanings, relationships, and contextual understanding.

4. Pragmatic Analysis

- **What it does**: Understands the **intent** behind the sentence, based on context.
- **Example**: Interpreting "Can you pass the salt?" as a request, not a question about ability.
- **Focus**: Speaker's meaning, intention, tone.

♦ 5. **Discourse Integration**

- **What it does**: Links the meaning across sentences to form coherent understanding.
- **Example**:
  - Sentence 1: "Ravi lost his phone."
  - Sentence 2: "He is upset."
  - Linking “He” to “Ravi”.
- **Focus**: Co-reference, sentence connections, overall discourse.

Part of Speech	Role	Examples
Noun	Name	cat, school, honesty
Pronoun	Replaces noun	he, it, ours
Verb	Action/being	eat, run, is
Adjective	Describes noun	red, tall
Adverb	Describes verb/adjective	slowly, very
Preposition	Shows position/time	on, at, before
Conjunction	Connects words/clauses	and, but, because
Interjection	Expresses emotion	wow, hey, ouch

- **POS (Part-of-Speech) Tagging** is the process of assigning a **part of speech (such as noun, verb, adjective, etc.)** to each word in a sentence, based on its meaning and context.

◆ What is POS Tagging?

POS tagging labels each word in a sentence with its correct grammatical role.

◆ What is POS Tagging?

POS tagging labels each word in a sentence with its correct grammatical role.

Example:

swift Copy Edit

```
Sentence: She is reading a book.

POS tags:
She/PRONOUN is/VERB reading/VERB a/DETERMINER book/NOUN
```

🔷 Common Parts of Speech Used in Tagging

Tag	Description	Example
NN	Noun (singular)	book, cat
NNS	Noun (plural)	books, cats
VB	Verb (base form)	eat, run
VBD	Verb (past tense)	ate, ran
VBG	Verb (gerund)	eating, running
JJ	Adjective	big, red
RB	Adverb	quickly, very
PRP	Personal pronoun	he, she
IN	Preposition	in, on
DT	Determiner	the, a

🔷 Types of POS Tagging

1. Rule-Based POS Tagging

- Uses a set of hand-written rules to identify POS tags.
- **Example Rule:** If a word ends in *-ly*, it's likely an adverb.
- **Limitation:** Fails in ambiguous contexts.

2. Statistical POS Tagging

- Uses **machine learning models** like **HMM (Hidden Markov Model)** or **Maximum Entropy**.
- Learns from annotated corpora.
- **Example Tool:** Stanford POS Tagger.

3. Neural POS Tagging

- Uses **deep learning (e.g., LSTM, BERT)** for high accuracy.
- Automatically learns features from data.
- **Example Frameworks:** spaCy, Transformers (HuggingFace).

🔷 POS Tagging Example (Detailed)

Sentence: *The quick brown fox jumps over the lazy dog.*

Word	POS Tag	Meaning
The	DT	Determiner
quick	JJ	Adjective
brown	JJ	Adjective
fox	NN	Noun
jumps	VBZ	Verb (3rd person singular)
over	IN	Preposition
the	DT	Determiner
lazy	JJ	Adjective
dog	NN	Noun

✅ Why Parsing is Required in NLP

1. **Understanding Sentence Structure**

- Parsing reveals how words in a sentence relate to each other.
- It breaks down a sentence into phrases and grammatical units like **noun phrase (NP)**, **verb phrase (VP)**, etc.
- Example:
  "The cat sat on the mat."
  → Helps identify "The cat" as the subject (NP), "sat on the mat" as the predicate (VP).

2. **Disambiguation**

- Natural language is **ambiguous** — a single sentence can have multiple meanings.
- Parsing helps disambiguate by analyzing **syntax** (structure).
- Example:
  "I saw the man with a telescope."
  → Did I use a telescope, or does the man have a telescope? Parsing helps figure that out.

3. **Machine Translation**

- To translate properly, machines must understand the **syntax** of the source and target language.
- Parsing ensures correct word order, grammar, and meaning during translation.

4. **Question Answering Systems**

- Parsing helps extract the subject, object, verb, and question type.
- Example:
  "Who wrote the book?" → Helps locate the **subject** (**"who"**) and **action** (**"wrote"**).

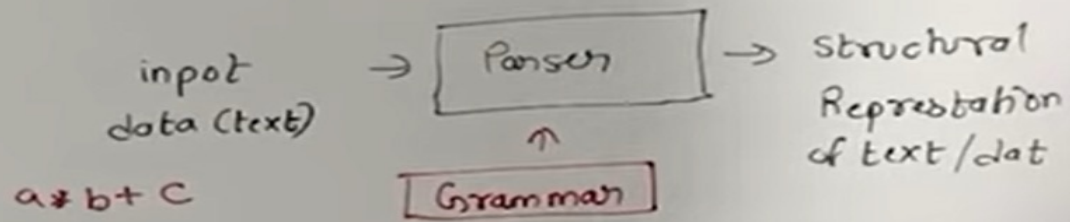
5. **Information Extraction**

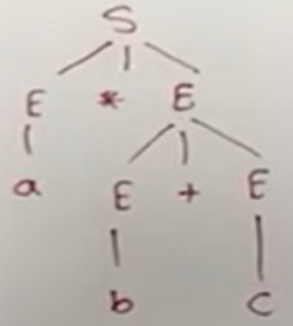
- Parsing identifies key information from unstructured text.
- For example, extracting **names, places, dates**, or **relationships**.

6. **Speech Recognition and Synthesis**

- Helps systems understand grammar to process **spoken language** and respond correctly.

Parser





- It is a software component design for taking input data (text) and give structural representation of the input after checking for correct syntax or grammar
- How it is used in NLP
  - Grammar checking
  - Intermediate stage of Semantic Analysis

Basic concept of Grammer and Parse tree (CFG)

- mathematically a grammer G can be written as four touples

(N, T, S, P)

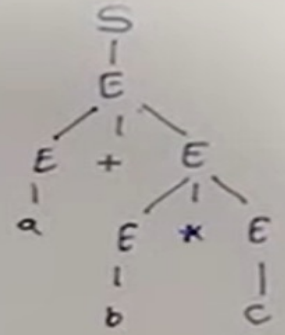
$\vee \Sigma$

N → Non terminal
T → terminal
S → start Symbol
P → Production rules

Ex: $S \to E$

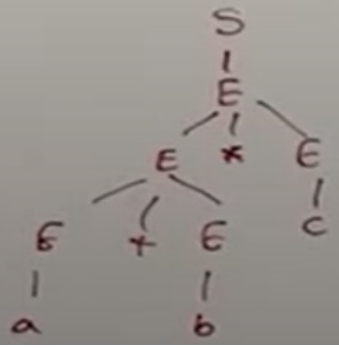
$E \to E+E \mid E*E \mid a \mid b \mid c$

input: a+b*c



Concept of Parse tree

- It is Graphical Representation of Derivation
- Start Symbol is root of Parse tree
- Leaf nodes are terminals
- Interior nodes are non-terminal
- If Parse Properly will create a input text



Example Grammar

Let's define a CFG similar to the image:

mathematica

Copy Edit

```
S → E
E → E + E | E * E | a | b | c
```

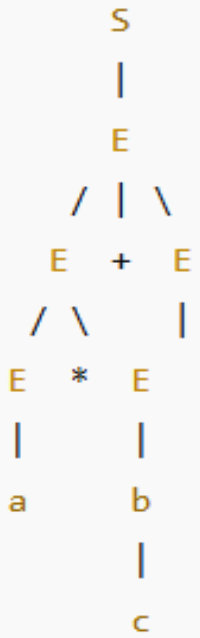
Input String

We will generate a parse tree for the string:

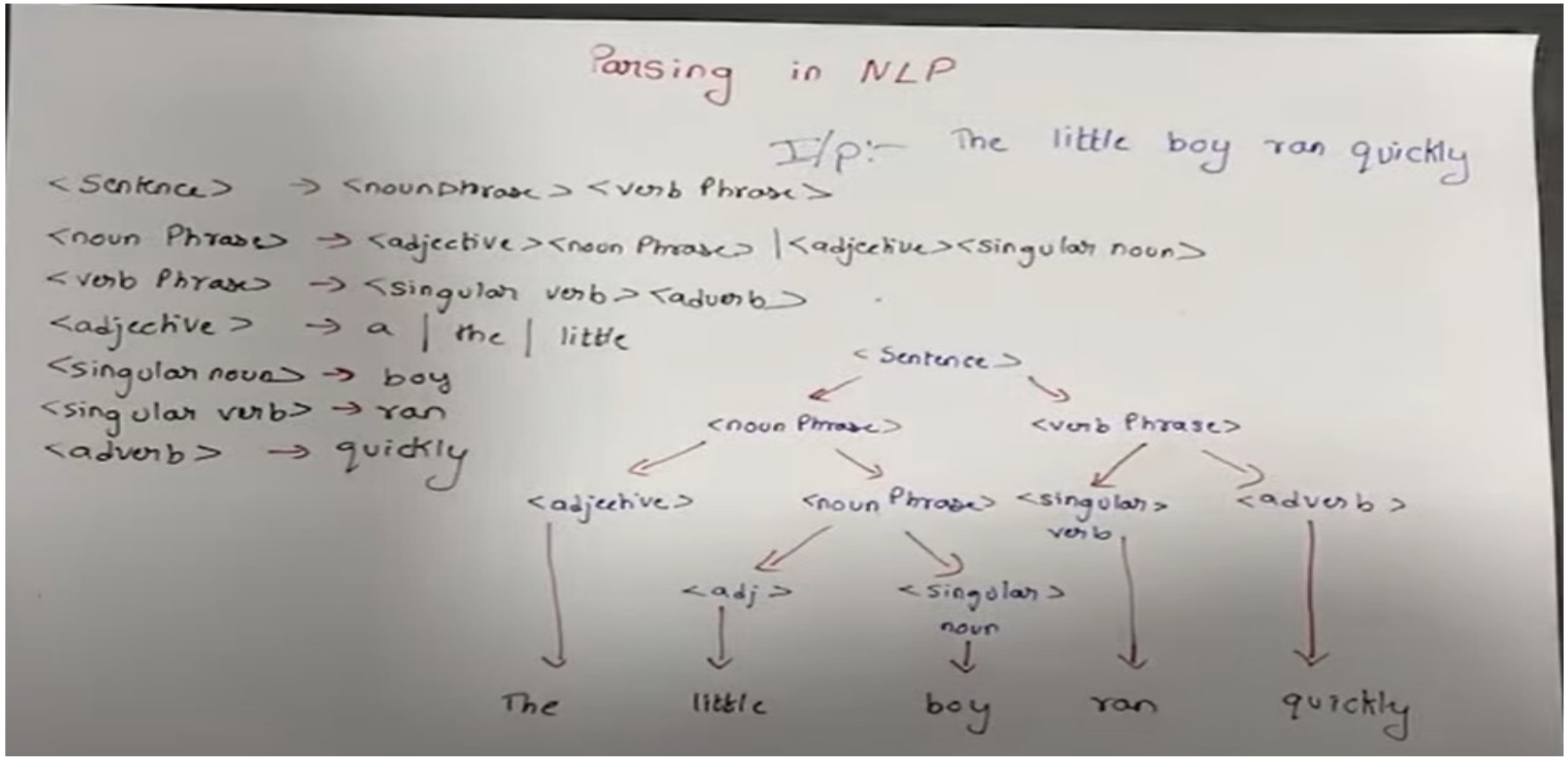
`a * b + c`

Parse Tree for `a * b + c`

mathematica



Parsing in natural Language processing



Grammer

S → VP

VP → Verb NP

NP → Det Nom

Det → that

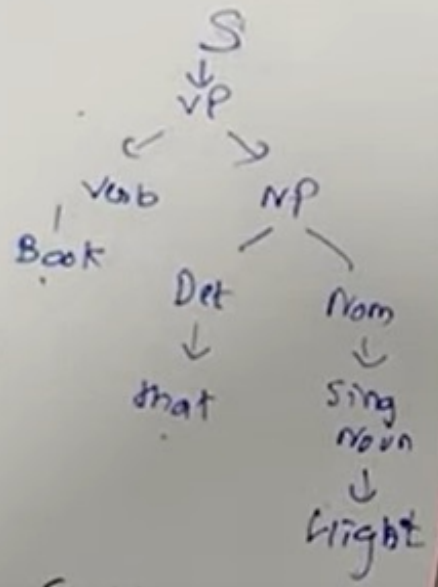
Nom → singular noun

Verb → Book

S Noun → Flight

Input : Book that Flight

Top down



Bottom up