Natural Language Processing (NLP) is a core area of Artificial Intelligence that allows machines to **understand, interpret, analyze, and generate human language**. It lies at the intersection of:

- **Linguistics** (rules of language),
- **Computer Science** (algorithms, data structures), and
- **Artificial Intelligence/Machine Learning** (pattern recognition, learning from data).

**Why important?**

- Human language is inherently ambiguous and context-dependent.
- NLP enables applications such as:
    - Chatbots and voice assistants (Siri, Alexa, Google Assistant)
    - Machine Translation (Google Translate, DeepL)
    - Sentiment Analysis (detecting opinions in tweets/reviews)
    - Information Retrieval (search engines like Google)
    - Speech-to-Text (YouTube captions) and Text-to-Speech (audiobooks)
    - Summarization, Plagiarism detection, Grammar checking (Grammarly)

---

**1. Levels of NLP (Detailed with examples)**

Language can be processed at multiple levels, from sounds to context.

1. **Phonology** — study of sound systems in language.
    - Example: In English, /p/ and /b/ are different phonemes; in Hindi, aspirated vs unaspirated sounds (/p$^h$/ vs /p/).
    - **Applications:** speech recognition (distinguishing words based on pronunciation), text-to-speech.
2. **Morphology** — study of word structure (morphemes = smallest meaning units).
    - Example: *unhappiness* = *un-* (negation) + *happy* (root) + *-ness* (noun suffix).

    - **Applications:** spell checkers, lemmatization, handling morphologically rich languages like Turkish or Hindi.

3. **Syntax** — rules of sentence formation.
    - Example: English: SVO (Subject-Verb-Object): "The cat (S) chased (V) the mouse (O)."
    - **Applications:** parsing, grammar correction, machine translation.
4. **Semantics** — study of meaning of words and sentences.
    - Example: The word *bank* may mean financial bank or river bank.
    - **Applications:** word sense disambiguation, semantic search, QA systems.
5. **Discourse** — coherence and relations between sentences.
    - Example: "Rita bought a book. She read it immediately." → *She* refers to Rita, *it* to book.
    - **Applications:** summarization, dialogue understanding.
6. **Pragmatics** — study of meaning in context and speaker intent.
    - Example: "Can you pass the salt?" is not about ability, but a polite request.
    - **Applications:** dialogue systems, polite and context-aware chatbots.

---

### 2. Preprocessing (Text Cleaning)

Before applying algorithms, text must be standardized.

**Steps with examples:**

- **Lowercasing:** "Natural" → "natural"
- **Punctuation removal:** "Hello!!!" → "Hello"
- **Tokenization:** "I love NLP" → ["I", "love", "NLP"]
- **Stopword removal:** Remove words like *the, is, of.*
- **Stemming:** "running" → "run", "studies" → "studi" (rough root).
- **Lemmatization:** "studies" → "study", "better" → "good" (accurate dictionary form).

- **Normalization:** Handle contractions (don’t → do not), numbers (10k → 10,000).

**Importance:** Reduces noise, decreases vocabulary size, and improves model accuracy.

---

### 3. Stemming vs Lemmatization

- **Stemming:** Fast, rule-based, may produce invalid roots (happi).
- **Lemmatization:** Slower, requires dictionary + POS tagging, but returns valid words (happy).

**Example:** "The children are playing happily."

- Stemming → [child, are, play, happi]
- Lemmatization → [child, be, play, happily]

---

### 4. Morphology

**Types of morphemes:**

- Free morphemes: words that can stand alone (book, run).
- Bound morphemes: must attach to roots (un-, -s, -ed).

**Derivational morphology:** Creates new words or changes POS.

- teach (V) → teacher (N).

**Inflectional morphology:** Adds grammatical features (tense, number).

- play → plays, played, playing.

**Applications:** text search (finding run, running, ran as related), machine translation.

---

### 5. Phonology & Applications

Phonology = study of sound structures.

- **Speech-to-text:** phonemes converted to words.
- **Text-to-speech:** words synthesized into speech.
- **Accent handling:** recognize variations of pronunciation.

- **Spell checking:** detect errors based on phonetic similarity (their/there).

---

### 6. Lexical Analysis & POS Tagging

**Lexical analysis:** identifies words and their properties.

**POS tagging:** assigns part-of-speech categories.

- "Book the ticket" → Book = Verb (to reserve).
- "Read the book" → Book = Noun (object).

**Applications:** parsing, disambiguation, information retrieval.

**Common POS tags:**

- Noun (NN), Verb (VB), Adjective (JJ), Adverb (RB), Determiner (DT).

---

### 7. Syntax, CFG & Parsing

**What is Syntax?** Syntax is the study of how words combine to form valid sentences according to the grammar rules of a language. In NLP, syntax is important because understanding sentence structure is necessary for parsing, machine translation, and grammar checking.

**What is a Context-Free Grammar (CFG)?** A Context-Free Grammar is a formal system that defines the syntactic structure of a language. It consists of:

- **Terminals:** actual symbols/words of the language (e.g., cat, dog, chased).
- **Non-terminals:** abstract syntactic categories (e.g., S for sentence, NP for noun phrase).
- **Production rules:** rules for replacing a non-terminal with a sequence of terminals/non-terminals.
- **Start symbol:** usually S, representing a complete sentence.

**Formal definition of CFG:** A CFG is a 4-tuple $(V, \Sigma, R, S)$

- $V$ = finite set of non-terminals
- $\Sigma$ = finite set of terminals
- $R$ = finite set of production rules ($A \to \alpha$)
- $S$ = start symbol ($S \in V$)

**Example CFG rules:**

S → NP VP

NP → Det N | N | Det Adj N

VP → V NP | V NP PP

PP → P NP

Det → the | a

N → cat | mouse | boy | park

Adj → black | tall

V → chased | saw | liked

P → in | on | with

**Explanation:**

- A sentence (S) is made up of a noun phrase (NP) followed by a verb phrase (VP).
- A noun phrase can be just a noun (N), or a determiner (Det) + noun, or Det + adjective + noun.
- Verb phrases may consist of a verb alone, or a verb + NP, optionally followed by a prepositional phrase.

**Parse tree for "The black cat chased the mouse in the park":**

```
S
├─ NP
│ ├─ Det (the)
│ ├─ Adj (black)
│ └─ N (cat)
└─ VP
├─ V (chased)
├─ NP
│ ├─ Det (the)
│ └─ N (mouse)
```

└─ PP

├─ P (in)

└─ NP

├─ Det (the)

└─ N (park)

**Why CFGs matter in NLP:**

- Provides a clear, rule-based structure for sentences.
- Used in constituency parsing to generate parse trees.
- Helps resolve syntactic ambiguity.
- Forms the basis of many natural language parsers.

**Limitations of CFGs:**

- Cannot capture all linguistic phenomena (e.g., context-sensitive dependencies like subject-verb agreement in complex sentences).
- Real-world NLP often supplements CFGs with probabilistic methods (PCFGs) to handle ambiguity.

---

### 8. Semantics

Semantics = study of meaning.

**Challenges:**

- Synonyms (big/large)
- Polysemy (bank = river/finance)
- Context-based meaning

**Approaches:**

- Rule-based logic (formal semantics)
- Statistical semantics (Word2Vec, BERT embeddings)

**Applications:** Question answering, semantic search, translation, summarization.

---

### 9. Discourse & Pragmatics

**Discourse:** linking sentences for coherence.

- "John dropped the glass. It broke." → "It" = glass.

**Pragmatics:** language use in context.

- "It's cold here" → may imply "Please close the window."

**Applications:** dialogue systems, summarization, context-aware AI.

---

### 10. Ambiguity in NLP

**Meaning:** Ambiguity in NLP occurs when a sentence, phrase, or word can have more than one interpretation. Natural languages (like English, Hindi, Marathi, etc.) are full of ambiguities because the same words or structures can mean different things depending on context.

---

**Types of Ambiguity:**

1. **Lexical Ambiguity (Word-level)**
   - A single word has multiple meanings.
   - Examples:
     - *Bank* → (i) River bank, (ii) Financial bank
     - *Bat* → (i) An animal, (ii) A cricket bat
2. **Syntactic Ambiguity (Structural ambiguity)**
   - A sentence can be parsed in more than one way due to grammar structure.
   - Example:
     - *I saw the man with the telescope.*
     - Meaning 1: I used the telescope to see the man.
     - Meaning 2: The man had the telescope.
3. **Semantic Ambiguity (Meaning-level)**
   - The sentence meaning is unclear even if grammar is correct.
   - Example:
     - *Visiting relatives can be boring.*

        - Meaning 1: Relatives who are visiting are boring.
        - Meaning 2: The act of visiting relatives is boring.

4. **Pragmatic Ambiguity (Context-based)**
    - Interpretation depends on real-world context or speaker's intent.
    - Example:
        - *Can you open the door?*
        - Literal: Asking about ability.
        - Pragmatic: A polite request to open the door.
5. **Anaphoric Ambiguity (Reference ambiguity)**
    - Occurs when it is unclear what a pronoun refers to.
    - Example:
        - *Ravi told Ramesh that he was selected.*
        - Who was selected: Ravi or Ramesh?

---

**Why Ambiguity is Important in NLP?**

- Ambiguity makes language processing difficult for machines.
- Many NLP tasks such as parsing, translation, chatbots, question answering, and information retrieval require clear meaning.
- Handling ambiguity is a **core challenge** in NLP.

---

**How NLP Handles Ambiguity:**

- **Statistical Models:** Estimate probabilities of meanings from large corpora.
- **Context Analysis:** Use surrounding words/sentences to infer correct meaning.
- **World Knowledge / Ontologies:** Background knowledge helps disambiguate.
- **Deep Learning Models:** Modern models (BERT, GPT) learn context-based meanings automatically.

### 11. NLP Pipeline

Raw text → Preprocessing → Lexical Analysis → Syntax Parsing → Semantic Analysis → Discourse/Pragmatics → Applications

- Preprocessing: cleaning text.
- Lexical: tokenize, POS tagging.
- Syntax: sentence structure.
- Semantics: meaning representation.
- Discourse/Pragmatics: linking across context.
- Applications: chatbots, MT, QA.

### 12. Parsing Techniques

**Top-down parsing:**

- Starts with the start symbol (S).
- Predicts grammar rules and expands non-terminals until matching the input.
- Example:
  - Input: "the boy saw the cat"
  - Steps:
    1. Start with S → NP VP
    2. NP → Det N → (the boy)
    3. VP → V NP → (saw the cat)
  - Successfully matches input.
- Advantage: concept