NATURAL LANGUAGE PROCESSING

Unit I — Linguistic Foundations and Formal Language Models

Topics Covered in this Unit

Introduction to NLP: History, Generic Architecture, Levels of Language & Knowledge Processing
Linguistic Ambiguity in Natural Language: Lexical, Syntactic, Semantic, Pragmatic & Referential
Formal Language Concepts: Alphabets, Regular Expressions, Finite Automata (DFA/NFA) & Limitations
Context-Free Grammars (CFG): 4-Tuple Grammar, Derivations, Parse Trees & Structural Ambiguity
Linguistic Levels: Morphology (Inflectional/Derivational), Morphological Parsing & Finite-State Transducers (FST)
Edit Distance & Spelling Correction: Minimum Edit Distance (Levenshtein DP) & Noisy Channel Model
Approaches to NLP: Rule-based, Data-based (Statistical/ML) & Knowledge-based Approaches
Python-based NLP Toolkits: NLTK, spaCy, TextBlob & Industrial Pipeline Architecture
Case Study: Rule-Based SVO Sentence Validation & Parse Trees ('The cat eats food') via Regex & CFG
High-Yield University Exam Question Bank with Detailed Model Answers & Unit Summary

Compiled in a structured, easy-to-understand, textbook style format
Specially curated for B.Tech / B.E. / BCA / MCA / B.Sc Computer Science & Data Science Students

COURSE SYLLABUS & MAPPING

[UNIT I SYLLABUS — PRESCRIBED UNIVERSITY CURRICULUM]

This section contains the official syllabus for Unit I. Verify with your university curriculum mapping.

Unit I Syllabus Breakdown:

Unit I - Linguistic Foundations and Formal Language Models: Introduction to Natural Language Processing: History of NLP, Generic NLP System, Levels of NLP, Knowledge in Language Processing, Ambiguity in Natural Language, Stages, Challenges, and Applications of NLP. Formal Language Concepts: Alphabets, strings, languages, regular expressions, finite automata (DFA/NFA), and limitations of regular languages for natural language. Context-Free Grammars : CFG components, Derivations, parse trees, Ambiguity. Linguistic Levels: Morphology (roots, affixes, morphological parsing), Finite-State Transducers (FSTs) for morphology, Syntax, semantics, pragmatics. Edit Distance and Spelling Correction: Minimum Edit Distance (MED), Levenshtein distance, Noisy channel model. Approaches to NLP: Rule-based, Data-based, and Knowledge-based Approaches. Python-based NLP libraries: Natural Language Toolkit (NLTK), spaCy, TextBlob, and use cases. Case Study: Design a simple rule-based system to validate and analyze basic English sentences such as “The cat eats food” using regular expressions and Context-Free Grammar (CFG). The system should preprocess the text, define grammar rules for sentence structure (Subject–Verb–Object), generate a parse tree for valid sentences, and explain why regular expressions alone cannot fully capture nested or hierarchical sentence structures.

• Course Code: _____ • Semester / Year: _____

TABLE OF CONTENTS

1.1 Introduction to Natural Language Processing

Definition: Natural Language Processing (NLP)

Natural Language Processing (NLP) is a multidisciplinary field at the intersection of Computer Science, Artificial Intelligence, and Computational Linguistics concerned with enabling computers to understand, interpret, manipulate, and generate human languages in a semantically and pragmatically meaningful manner.

History and Evolution of NLP

- **1. The Symbolic & Rule-Based Era (1950s–1980s):** Early machine translation systems (Georgetown-IBM experiment, 1954), Chomsky's Formal Language Theory, ELIZA rule-based chatbot (Weizenbaum, 1966), and SHRDLU micro-world reasoning. Heavily relied on hand-crafted symbolic grammar rules.
- **2. The Statistical NLP Revolution (1990s–2000s):** Transition from rigid hand-written rules to statistical corpus-driven models. Emergence of N-gram language models, Hidden Markov Models (HMMs), probabilistic context-free grammars (PCFGs), and Support Vector Machines trained on large annotated corpora (Penn Treebank).
- **3. The Neural & Deep Learning Era (2010s–Present):** Distributed dense embeddings (Word2Vec, GloVe), Recurrent Neural Networks (LSTMs, GRUs), the Attention Mechanism (Vaswani et al., 2017), Pretrained Transformers (BERT, GPT), and modern Large Language Models (LLMs).

Generic Architecture of an NLP System

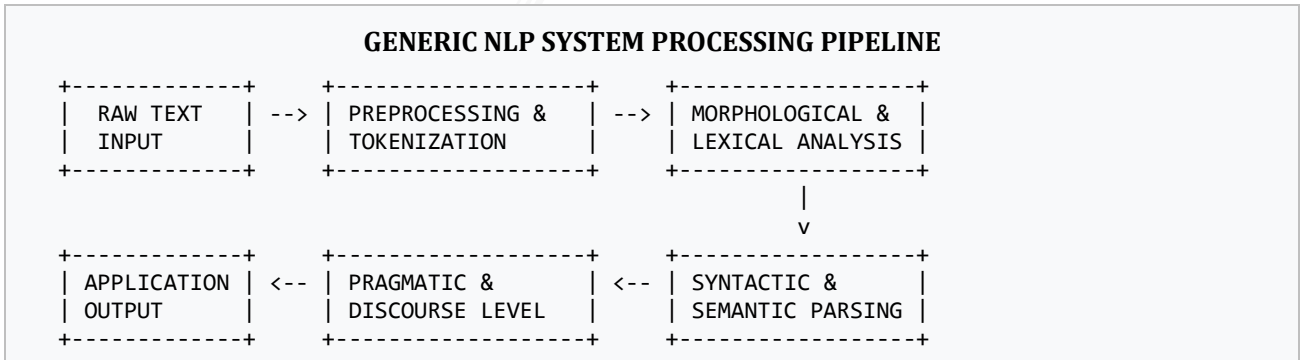

Figure: Hierarchical data flow from raw speech/text through linguistic levels to actionable output

Levels of Language Processing (Knowledge in NLP)

Linguistic Level	Primary Linguistic Focus	Core Computational Tasks & Examples
1. Phonetics & Phonology	Sound waves, phonemes, and acoustic properties of human speech.	Speech-to-Text, Automatic Speech Recognition (ASR), Text-to-Speech (TTS).
2. Morphology	Internal structure of words, morphemes, roots, prefixes, and suffixes.	Stemming, Lemmatization, Morphological Parsing (e.g., 'unbreakable' -> 'un-' + 'break' + '-able').
3. Syntax	Grammatical rules governing word order and sentence structure.	Part-of-Speech (POS) Tagging, Context-Free Grammar Parsing, Dependency Parsing.
4. Semantics	Literal dictionary meaning of words, phrases, and compositional logic.	Word Sense Disambiguation (WSD), Semantic Role Labeling, Named Entity Recognition.
5. Pragmatics	Contextual meaning, speaker intention, and non-literal implied meaning.	Irony/Sarcasm detection, sentiment nuance, speech act classification.
6. Discourse	Multi-sentence context, narrative flow, and coreference resolution.	Anaphora Resolution (resolving pronouns: 'Alice called Mary. She was happy.'), Summarization.

Types of Ambiguity in Natural Language

- **1. Lexical Ambiguity:** A single word possesses multiple meanings or part-of-speech tags. Example: 'I went to the bank' ('bank' = financial institution vs river bank).
- **2. Syntactic / Structural Ambiguity:** A sentence can be parsed into multiple distinct grammatical parse trees. Example: 'I saw the man with a telescope' (Did I use the telescope to see the man, or did the man have a telescope?).
- **3. Semantic Ambiguity:** Multiple logical interpretations of a sentence. Example: 'Every student read a book' (Did all students read the exact same book, or did each student read a different book?).
- **4. Pragmatic Ambiguity:** Meaning depends on social context and real-world background. Example: 'Can you pass the salt?' (Literally a query about physical ability; pragmatically a polite command).
- **5. Anaphoric / Referential Ambiguity:** Pronoun references are unclear across sentences. Example: 'The trophy would not fit into the brown suitcase because it was too big.' ('it' refers to the trophy).

1.2 Formal Language Concepts & Finite Automata

Formal Language Fundamentals

• *Alphabet (Sigma): A finite, non-empty set of basic atomic symbols (e.g., Sigma = {a, b} or ASCII characters).*

• *String (w): A finite sequence of symbols chosen from Sigma. Length |w| is symbol count. Empty string is epsilon.*

• *Language (L): A set of strings over alphabet Sigma (L subset of Sigma*).*

Regular Expressions & Finite State Automata (DFA/NFA)

- **Regular Expressions (Regex):** Algebraic notation defining Regular Languages using Concatenation (ab), Alternation/Union (a|b), and Kleene Star Closure (a* = zero or more repetitions of a).
- **Deterministic Finite Automaton (DFA):** Formally defined as a 5-tuple: M = (Q, Sigma, delta, q_0, F), where Q is finite state set, Sigma is input alphabet, delta: Q x Sigma -> Q is transition function, q_0 in Q is start state, and F subset of Q is set of accepting states.
- **Non-Deterministic Finite Automaton (NFA):** Transition function maps to power set delta: Q x (Sigma union {epsilon}) -> P(Q). Every NFA can be converted to an equivalent DFA via Subset Construction (Powerset algorithm).

Limitations of Regular Languages for Natural Language

According to Noam Chomsky's hierarchy of formal languages, Regular Languages (Type 3) are computationally INSUFFICIENT to represent human natural language syntax for the following reasons:

- **1. Inability to Count & Match Arbitrary Recursion:** Natural languages contain center-embedded recursive structures of arbitrary depth. For example, nested English clauses: 'The cheese [that the rat [that the cat chased] ate] was rotten' corresponds to the formal language L = {a^n b^n : n >= 1}. By the Pumping Lemma for Regular Languages, finite state automata cannot count unbounded n because they lack memory storage!
- **2. Lack of Memory Stack:** Finite automata have a finite number of internal states and zero auxiliary memory. Context-Free Grammars (Type 2) equipped with a Pushdown Stack (PDA) are required to track nested matching syntactic dependencies.

1.3 Context-Free Grammars (CFG) & Parsing

Formal Definition: Context-Free Grammar (4-Tuple)

A Context-Free Grammar (CFG) is formally defined as a 4-tuple: G = (V, Sigma, R, S)

- *V: A finite set of Non-Terminal syntactic variables (e.g., {S, NP, VP, Det, N, V}).*
- *Sigma: A finite set of Terminal symbols / vocabulary words disjoint from V (e.g., {'the', 'cat', 'eats', 'food'}).*
- *R: A finite set of Production Rules of the form: A -> alpha, where A in V and alpha in (V union Sigma)*.*
- *S in V: The distinguished Start Symbol representing a complete sentence (S).*

Derivations & Parse Trees

- **Derivations:** A sequence of production rule applications replacing non-terminals until only terminal words remain. Leftmost Derivation replaces the leftmost non-terminal at every step; Rightmost Derivation replaces the rightmost non-terminal first.
- **Parse Tree:** A hierarchical graphical tree representation of a derivation where the Root is Start symbol S, Interior nodes are non-terminals V, and Leaf nodes (yield) are terminal words Sigma read from left to right.
- **Structural Ambiguity in CFG:** A grammar G is Ambiguous if there exists at least one string w in L(G) that possesses TWO OR MORE distinct parse trees (or two distinct leftmost derivations).

1.4 Morphology & Finite-State Transducers (FST)

1. Morphological Fundamentals

- **Morpheme:** The smallest indivisible meaningful unit of language (e.g., 'cats' contains 2 morphemes: root 'cat' + plural suffix '-s').
- **Roots, Stems & Affixes:** A Root is the primary lexical unit. Affixes attach to stems: Prefixes (un-happy), Suffixes (eat-ing), Infixes (inserted inside root), Circumfixes (attach before and after).
- **Inflectional vs. Derivational Morphology:**
  - Inflectional Morphology: Modifies grammatical form (tense, number, person) WITHOUT changing core meaning or grammatical category (e.g., book -> books [Noun -> Noun], walk -> walked [Verb -> Verb]).
  - Derivational Morphology: Creates a brand new word with a different meaning and often a new grammatical category (e.g., compute [Verb] -> computer [Noun] -> computational [Adjective]).

2. Finite-State Transducers (FSTs) for Morphological Parsing

A Finite-State Transducer (FST) is a 2-tape finite state automaton that maps an input string from one alphabet to an output string in another alphabet. In Two-Level Morphology (Koskenniemi, 1983), FSTs map between the Abstract Lexical Level (e.g., 'cat +N +Pl') and the Surface Orthographic Level (e.g., 'cats').

- **Composition of Transducers:** FSTs can handle complex spelling change rules (e.g., 'fox +N +Pl' -> 'foxes' [e-insertion rule], 'city +N +Pl' -> 'cities' [y-to-ies rule]) by cascading transducers in sequence.

1.5 Edit Distance and Spelling Correction

Definition: Minimum Edit Distance (Levenshtein Distance)
Minimum Edit Distance (MED) between two strings s_1 and s_2 is the minimum number of character-level editing operations (Insertion, Deletion, Substitution) required to transform string s_1 into string s_2.

The Levenshtein Distance Dynamic Programming Algorithm

Given source string s_1 of length n and target string s_2 of length m, define D[i, j] as the edit distance between prefix s_1[1..i] and s_2[1..j]:

- **Base Conditions:** D[i, 0] = i * cost(delete); D[0, j] = j * cost(insert).
- **Dynamic Programming Recurrence:**
  - D[i, j] = min of:
  - 1. D[i-1, j] + cost(delete(s_1[i]))
  - 2. D[i, j-1] + cost(insert(s_2[j]))
  - 3. D[i-1, j-1] + (0 if s_1[i] == s_2[j] else cost(substitute))
- **Standard Cost Weights:** Levenshtein Metric: Insert=1, Delete=1, Substitute=2 (or 1 depending on metric). Time Complexity: O(n * m).

The Noisy Channel Model for Spelling Correction

The Noisy Channel Model frames spelling correction probabilistically: Suppose a user intended to type correct word w, but transmission noise corrupted it into misspelled surface string x. We seek the most probable intended word w_hat:

- **Bayesian Formulation:** w_hat = argmax_w P(w | x) = argmax_w [(P(x | w) * P(w)) / P(x)] = argmax_w [P(x | w) * P(w)]
- **1. Error Model (Channel Model P(x | w)):** The probability of typing x when w was intended. Computed using a Confusion Matrix tracking character insertions, deletions, substitutions, and transpositions (Damerau-Levenshtein).
- **2. Language Model (Prior P(w)):** The prior probability of word w occurring in natural language corpus (e.g., N-gram frequency).

1.6 Approaches to NLP & Python Toolkits

Approach	Core Principle & Mechanics	Primary Advantages	Major Limitations
Rule-Based Approach	Hand-crafted formal grammar rules, regular expressions, and linguistic dictionaries created by human linguists.	100% interpretable; highly precise for constrained, well-defined domains.	Brittle; fails on informal text; does not scale; high human engineering cost.
Data-Based / Statistical	Statistical machine learning models trained on large annotated text corpora (e.g., HMMs, Naive Bayes, LSTMs).	Robust to noisy inputs; automatically learns patterns; highly scalable.	Requires large labeled training corpora; vulnerable to out-of-domain drift.
Knowledge-Based	Leverages structured ontologies, semantic networks, and knowledge graphs (e.g., WordNet, ConceptNet, DBpedia).	Captures deep relational semantics and factual common-sense knowledge.	Expensive to curate and maintain; struggles with evolving slang.

Python-Based NLP Libraries

- **1. NLTK (Natural Language Toolkit):** The definitive academic and educational NLP library. Features extensive modules for tokenization, stemming (Porter), POS tagging, CFG parsing, and access to over 50 linguistic corpora (Brown, WordNet).
- **2. spaCy:** Industrial-strength, ultra-fast C-optimized NLP library designed for production software pipelines. Features pretrained neural pipelines for tokenization, NER, dependency parsing, and multi-language support.
- **3. TextBlob:** A user-friendly, high-level wrapper built on top of NLTK and Pattern. Excellent for rapid prototyping of sentiment analysis, spelling correction, and noun phrase extraction.

1.7 Case Study: Rule-Based SVO Sentence Analysis

1. Problem Statement & Objective

Design a rule-based NLP system to validate, preprocess, and syntactically analyze basic Subject–Verb–Object (SVO) English sentences such as 'The cat eats food' using Regular Expressions and Context-Free Grammars (CFG).

2. Context-Free Grammar Definition

- **Non-Terminals:** V = {S, NP, VP, Det, N, V}
- **Terminals:** Sigma = {'the', 'a', 'cat', 'dog', 'eats', 'drinks', 'food', 'water'}
- **Grammar Production Rules:**
  - S -> NP VP
  - NP -> Det N | N
  - VP -> V NP | V
  - Det -> 'the' | 'a'
  - N -> 'cat' | 'dog' | 'food' | 'water'
  - V -> 'eats' | 'drinks'

3. Parse Tree for 'The cat eats food'

PARSE TREE: 'THE CAT EATS FOOD'

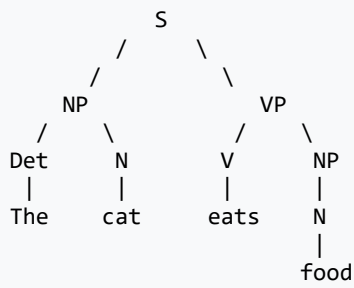

Figure: Hierarchical SVO syntactic parse tree generated by Context-Free Grammar

4. Why Regular Expressions Alone Cannot Capture Sentence Structures

- **1. Flat vs. Hierarchical Syntax:** Regular expressions can only match linear string patterns; they cannot represent nested hierarchical phrase structures (e.g., identifying that 'eats food' is a cohesive Verb Phrase constituent).
- **2. Lack of Recursive Memory:** If a sentence contains nested relative clauses ('The cat [that chased the dog [that barked]] eats food'), regular expressions fail because finite state mechanisms cannot maintain stack-based memory to match open and closed clause boundaries.

1.8 High-Yield University Exam Question Bank

Part A: Short Answer Questions (2 to 5 Marks)

Q1. What is Lexical Ambiguity and Syntactic Ambiguity? Give an example of each. [3 Marks]

- **Answer:** Lexical ambiguity occurs when a word has multiple meanings (e.g., 'bank' = river bank vs financial bank). Syntactic ambiguity occurs when a sentence has multiple grammatical parse trees (e.g., 'I saw the man with a telescope' where the prepositional phrase can attach to the verb 'saw' or the noun 'man').

Q2. Why are regular languages insufficient for modeling natural language? [3 Marks]

- **Answer:** Natural languages contain center-embedded recursive structures (e.g., L = {a^n b^n : n >= 1}). By the Pumping Lemma, regular languages and Finite State Automata lack memory stacks and cannot count unbounded dependencies, requiring Context-Free Grammars.

Q3. State the Noisy Channel Model equation for spelling correction. [3 Marks]

- **Answer:** w_hat = argmax_w P(w|x) = argmax_w [P(x|w) * P(w)], where P(x|w) is the Error Channel Model (confusion matrix probability of typo x given intended word w) and P(w) is the Language Model prior probability.

Part B: Long Answer Questions (10 to 15 Marks Outline Guides)

- **Q4. Explain the six linguistic levels of Natural Language Processing with computational tasks and examples. [12 Marks]**
- **Q5. Formulate Context-Free Grammars (CFG) as a 4-tuple. Explain leftmost derivation, rightmost derivation, parse trees, and structural ambiguity. [12 Marks]**
- **Q6. Compute the Levenshtein Minimum Edit Distance dynamic programming table between source 'EXECUTION' and target 'INTENTION'. [10 Marks]**
- **Q7. Compare Rule-based, Data-based, and Knowledge-based approaches to NLP. Discuss Python toolkits NLTK, spaCy, and TextBlob. [10 Marks]**

1.9 Unit Summary & Key Takeaways

- NLP enables computational systems to process, understand, and generate human natural languages across phonetic, morphological, syntactic, semantic, pragmatic, and discourse levels.
- Ambiguity exists at lexical, syntactic, semantic, pragmatic, and referential linguistic tiers.
- Regular languages cannot model center-embedded recursion; Context-Free Grammars (V, Sigma, R, S) handle hierarchical syntax.
- Morphology studies word structure (inflectional vs derivational), modeled via Finite-State Transducers (FSTs).

- Minimum Edit Distance measures string distance via dynamic programming; Noisy Channel Model combines error models P(x|w) with language priors P(w).
- NLP approaches span Rule-based, Data-based (statistical/ML), and Knowledge-based paradigms implemented via NLTK, spaCy, and TextBlob.