UNit - I

① NLP :- Natural Language processing.

Natural Language processing is the subfield of Artifical intelligence (AI) focused on enabling Machines to understand, interpret, and generate human Language. chatbots, search engines, translation apps. (like Google translation).

Core challenges of NLP :-

* Ambiguity :-
  * Lexical Ambiguity :- words with multiple meanings (bank as a financial institution vs. riverbank).
  * Syntactic Ambiguity :- Sentences with multiple valid grammatical interpretations. (I saw the man with the telescope).
  * Semantic Ambiguity :- sentences whose meaning changes based on context (The Chicken is ready to eat.)
* contextual understandings :- Difficulty in understanding context-dependent meanings, especially in long conversation (or) documents

Pragmatics :- Recognizing implied meanings. Sarcasm, and humor remains challenging for NLP models.

* Low-Resource Language :-

NLP systems often lack sufficient training data for Low-resource.

* Ethical and Bias concerns :-

* Models inherit biases from training datasets, potentially Leading to unfair (or) prejudiced outputs.

(3) Future of NLP :-

1) Zero-shot and Few-shot Learning.

* Models capable of performing tasks with minimal (or) no labeled data. ex 2G, 6G

2) Cross-Lingual NLP:

* seamless functioning of models across multiple languages without retraining.

I love हिंदी
Eng + Hind = Hinglish

3) Bias mitigation :-

* strategies for identifying and addressing dataset biases.

4) Integration with other Modalities:

* Combining NLP with computer vision and audio processing for enhanced applications.

Tab →

(4) Language Modelling (LM) Vin--

* Language Modelling is the process of predicting the next word in a sequence of given a set of preceding words.

→ * It is the fundamental task in NLP.

→ * serves are back bone for applications like speech recognition, machine translation and text prediction.

Applications of Language Modeling (LM).

*(1) speech Recognition : converting spoken words into a text.

*(2) Machine Translation : Generating accurate translations.

*(3) chat bots : creating human like response

*(4) spell checking : Identifying and correcting errors. (summarly)

Types of Language Models.

1. Grammer Based models.
2. Statistical Models.

* Grammer Based models.

* A Grammer Based model follows prede fined grammer rules. This rules tell the model how to structure sentences correctly.

Sentence structure.

subject + verb + object.

Ex:- The cat eats fish ✓

she loves chocolate ✓

Advantages :-

(1) * High Accuracy in Grammer.

* since it follow strict rules, it ensures that sentences are grammatically correct.

Ex:- She Eats an apple. (✓)

She Eating apple (✗)

* (2) Good for structured Languages

* Works well for Language with clear grammer rules.

Ex: programming Languages. [C, c++, Java, python]

* (3) useful for Language Learning & grammer checking.

* Helps Learners by correcting grammer mistakes.

Ex: Grammarly uses grammar rules to suggest correct sentence structure.

* (4) predictable and Reliable.

* Since rules are predefined, it always follows them, reducing random errors.

Ex: In legal documents, where exact wording is important, it ensures correctness.

Limitations of Grammer - Based models.

* (1) Too Rigid (Not Flexible)

  * Language is always evolving and fixed rules dont always work for informal speech.

  Ex:- Gonna grab a coffee.

  (casual but correct in real life, yet a grammer model may flag it as worng.

* (2) Needs manual Rule creation.

  * Humans must write and update grammer. rules manually, which is time consuming.

  Ex :- Adding new slang (or) cool phrases requries extra work.

* (3) Cannot Handle complex sentences well

  * some sentences follow the rules but still sound awkward.

Ex: The dog which barks loudly is outside.

(Technically correct but unnatural - better as) The loudly barking dog is outside.

(grammatically correct)

* (4) struggles with meaning & context.

* It only checks if the structure is correct but doesn't understand the actual meaning.

* The cat ate the banana.

(Grammatically correct but makes little sense.)

* Parsing Techniques in Grammer-Based LM.

* Parsing in Grammer-Based Models (LMs) refers to analyzing a sentence's structure based on grammatical rules.

* Top-down parsing

  * start with main rule (i.e. a sentence) and breakes it into smaller parts (subject, verb, object etc).

Ex:- If the rule is Sentence -> Noun phrase + verb phase,

it first assumes this structure and tries to match the words.

* Bottom-up parsing.

  * Begins with individual words and builds up to larger structure

Ex:- It first identify 'dog' as a noun 'runs' as a verb then recognizes the phrase "The dog runs".

CYK Algorithm (Cocke-Younger-Kasami).

* A dynamic programming algorithm used for parsing context-Free Grammars efficiently.
* Advantages is Guarantees optimal parsing for CFG's.

* Parsing Techniques in Grammar-Based Lm

Aspect	Grammar Based Lm	statical Lm
Approach	Rule-based	Data-driven
Interpretability	Highly interpretable.	Less interpretable.
Flexibility	Rigid structures	Adaptive to data.
Handling Ambiguity	Limited	Better Ambiguity handling.
Scalability	Difficult to scale.	Scales well.

Applications of Grammer - Based LM.

1. Voice - controlled system :- Parsing restricted voice commands ("Turn off the fan").
2. Text to - speech system :- Ensuring grammatically accurate speech synthesis.
3. Education Tools :- Grammer - checking software to guide language Learners.
4. compiler Design :- validating syntax in Programming languages.

statistical Language modeling (SLM)

* SLM is a way of predicting the next word in a sentence based on probability.
* It helps computers understand and generate human like text,

* Unigram model (one-word at a time).
  * Assumes that each word is independent of the previous ones.

Ex: I love NLP

I, love NLP.

* Bigram model
  * considers one preceding word to predict the next word.
  (word + Net word)

I love pizza,

I love pasta.

user type I it check most common words comes after "I" most eg love.

(3) Trigram model.

* Looks at groups of three words (word + next two words).

Ex:- If a chatbot sees "I love pizza" many times, it learns that after "I love", "pizza is likely.

* Better than Unigram model.

Neural Language Models (NLMs)

* uses neural networks (complex mathematical functions) to understand entire sentences instead of just word pairs (or) triplets.

Ex:- If a user type "I want to eat", a neural model (like GPT (or) BERT). can predict "a burger" (or) some food based on deep context

Probabilistic Approaches in statistical Language modeling.

PASLM help predict the next word in a ~~sequence~~ sentence using probability. These approaches assign a likelihood to each possible word based on past observation.

* Smoothing Techniques.

To handle zero probabilities (unseen word sequences), smoothing techniques are applied.

1. Add-one (Laplace) smoothing :
   Adds 1 to all word counts.

2. Good-Turing Discounting :
   Adjusts probabilisties for unseen event.

3. Kneser-Ney smoothing :
   Redistributes probabilisties effectively across N-Gram levels

Backoff : If a higher-order N-Gram probability is unavailable, fall back to lower-order probabilities.

Interpolation : combine probabilities from different N-Gram levels for better predictions.

Evaluation Metrics for Language models

When we build a language model (LM). We need to check how good it is at predicting text. Evaluation metrics help measures the model's performance.

* **Perplexity (PPL)**
  * Measures how well the model predicts a sentence.
  * Lower is better. (a perfect model has low perplexity).

Cross - Entropy

* Measures the average bits required to encode a message.

challenges in statistical LM.

* **Data Sparsity :**
  * some words combination appear very rarely, so the model doesn't learn them well.
* **out of vocabulary :**
  * If the model has never seen a word before. it doesn't know how to handle it.
* **Ex: context Length**
  * Traditional models (like n-grams) only look at a few previous words and can't understand long sentences properly.

computational overhead :

More complexity, more Resources.

Ambiguity : Multiple meanings are Hard.

Applications in statistical LM

* Search Engines : Enhancing query relevance.

* Text Generation : Creating context aware text content.

Finite - state Automata (FSA)

A Finite state Automaton (FSA) is like a simple machine that moves through different states based on inputs. It is used to recognize patterns in text, speech, and programming.

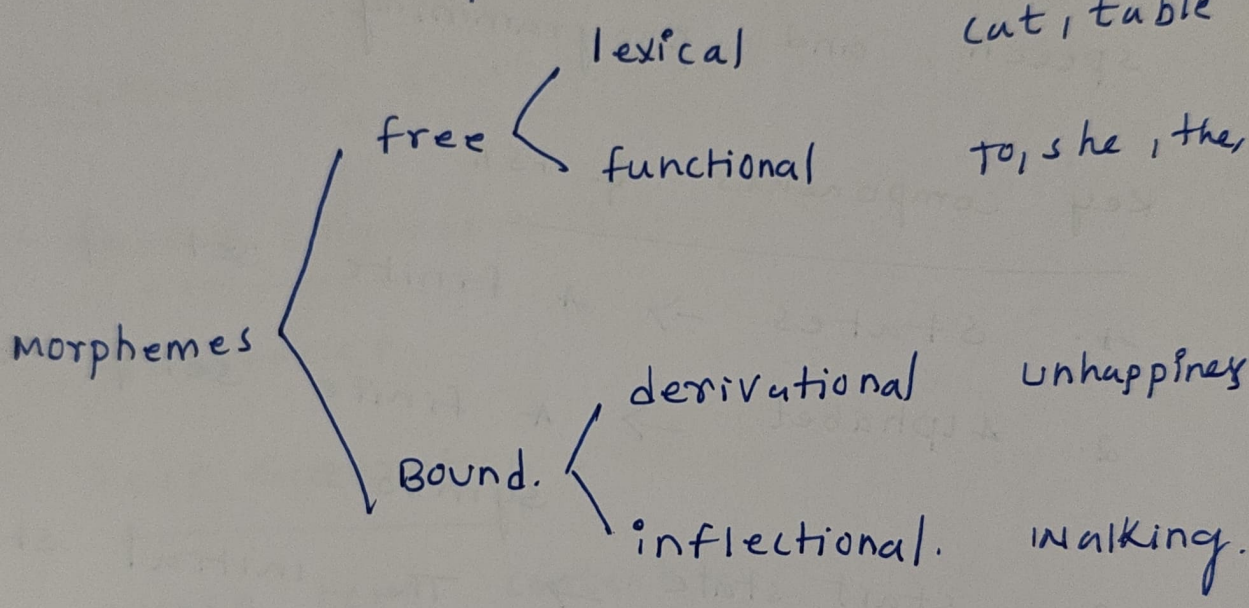
Key components of FSA

1. States → A finite set of states.
2. Alphabet → A finite set of input symbols.
3. start state → The initial state.
4. Accepting states → states that signify valid string recognition.
5. Transition Function : Rules defining tran sitions between state.

English Morphology

* Morphology is the study of how words are formed and structured in a language.

* In English, words are made up of smaller meaningful parts called morphemes.



Free :- Can stand Alone as independent words.

Ex:- cat, run, tree.

Bound :- Cannot stand alone, they need to be attached to another word.

Ex:- plural cats, dogs.
negation unhappy, unclear.

1) Lexical words - Meaningful words.

These words carry the main meaning of a Sentence.

Ex: Nouns → (dog, car, happiness)
verb → (run, eat, jump)
Adjectives → (beautiful, big, fast)
Adverbs → (quickly, softly, yesterday)

The dog runs quickly in the park.

carry the main meaning → Lexical words.

2) Functional words : Grammar Helpers.

* These words don't carry meaning but help connect and structure sentences.

* It shows the relationships between Lexical words.

The dog is running in the park.
is in the → Functional words

* Inflectional Morphology (changes grammer Not meaning).

changes, tense, number, (or) comparison but doesn't create a new word.

Ex: play → played. (Past tense)

big → bigger (comparison)

dog → dogs (plural)

Derivational (creates new words).

* changes the meaning (or) word type (noun → verb, verb → adjective).

Ex: "happy" → "unhappy" (meaning changed).

"teach" → "teacher" (verb - changed)

"act" → "action (verb → noun)

Morphological parsing.

→ Morphological parsing is the process of analyzing words into their morphemes. to extract structure and meaning.

steps in Morphological

1. Tokenization :- Breaking text into individual words.
2. morphological segmentation :- Splitting words into morphemes.
3. Morpheme Tagging :- Identifying each morpheme's role. (e.g. root, prefix, sufic
4. Reconstructing Meaning :- Combining the meanings of morphemes.

Tools for Morphological parsing

* Porter stemmer :- Reduces words to their root form.
* Lancaster stemmer :- Aggressive stemming algorithm.
3. spacy :- NLP library for morphological analysis.

Challenges in Morphological Analysis

* Ambiguity :- Words with multiple meanings
* complex morphology :- Highly inflected Languages (e.g. Finnish, Turkish)
* compound words :- Difficulty breaking down compound words (e.g. notebooks)
* Irregular forms :- Handling irregular verbs and plurals (go -> went.

Low-Resource Language : Limited linguist ic resources for less common languages

Transducers for Lexicon and Rules

1. Transducer : A Transducer is a type of finite-state machine that takes input and produces output.

Ex Input : "play" → output "played".

Input : "jump" → output "jumped".

2. Lexicon Rules ?

Lexicon rules define how words change (like verb tense, plural form,).

* Rule Based Transduction

* If a word ends in "y", replace it with "ied" (e.g, carry → carried)

* If a word ends in "e" add "d" (like → "liked)

Finite state Transducer (FST)

* similar to a Finite state Automaton (FSA) but with two layers.
  * one layer read input (e.g "run").
  * one layer gives output (e.g "running").
* It follows predefined rules to transform one word into another.

Input. "play" → output "played"

Input "jump" → output "jumped".

Applications of FST

1. Morphological Analysis (word structure processing).
   * Helps break words into roots and suffix.

   Input : running → output running (root + suffix).

used in : spell checkers, grammar tools.

Speech Recognition (converting sound to Text)

* converts spoken words into text format

Ex: Input : voice command "call mom"

output : written txt "call mom"

used in : Siri, Alexa, Google Assist ant.

Machine Translation (language conversion).

* Helps convert words between diff erent languages

Ex : dog (English) → output "perro" (spanish)

used in : Google Translate.

Text - to - speech systems (TTS).

* converts written text into spoken words.

Ex : "Hello, how are you ? → Input.

output : spoken words in AI voice.

used in : Audiobooks, screen readers.