# Unit I Introduction to Natural Language Processing



## Introduction :

Natural Language Processing

- It is a field of computer science & linguistics concerned with the interactions between computers & human (natural language).
- It is the scientific study of languages from computational perspective.
- It is the technology that is used by machines to understand, analyse, manipulate & interpret human's languages.
- Natural language generation systems convert infⁿ from computer databases into readable human language.
- Advantages
  - helps users to ask questions about subject & get a direct response within seconds.
  - offers exact answers to the question means it does not offer unnecessary & unwanted infⁿ.
  - helps computers to communicate with humans in their languages.
  - improve the efficiency of documentation processes, accuracy of documentation.
- Disadvantages
  - may not show context.
  - unpredictable
  - may require more keystrokes.
  - unable to adapt to the new domain & it has a limited funⁿ that's why NLP is built for a single & specific task only.

Components of NLP:

① Natural Language Understanding (NLU)

- NLU helps the machine to understand and analyse human language by extracting the metadata from content such as concepts, entities, keywords, emotion, relations & semantic roles.
- NLU involves the following tasks.
  - It is used to map the given i/p into useful representation.
  - It is used to analyze different aspects of the language.

② Natural Language Generation (NLG)

NLG acts as a translator that converts the computerized data into natural language representation.

Why NLP is hard?

- Understanding natural language is hard because of inherent ambiguity.
- Because of Huge amount of data resources needed (eg. grammar, dictionary, documents to extract statistics from)
- Computational complexity (intractable) of analyzing a sentence.

Ambiguities :-

- Lexical Ambiguity: words that can be used as adjectives, nouns & verbs
- Semantic Ambiguity: this refers to sentences that have different meaning in different context
- Syntactic Ambiguity: the confusion is created because of the 2 meanings of the sentence given.

Programming languages Vs Natural languages:

- Natural language are those that we use for communicating with each other.
  eg. English, French, Hindi, etc
- Natural language are expensive & easy for us to use.

- Computer programming languages are those that we use for controlling the operations of computer.
  eg. Prolog, C, C++, C#, Java, Python, etc.
- Programming languages are easy for a computer to understand, but they are not expressive.

- Natural language has a grammar & a set of conventions or rules for expressing thoughts.

- A programming language has a STRICT grammar & set of rules.

| Natural Language | Programming Language |
|---|---|
| - Ambiguous | - Non-ambiguous |
| - Context sensitive | - Context free |
| - Informal | - formal |
| - Descriptive | - Perspective |
| - Unstructured | - Structured |
| - Uncontrolled | - Controlled |

Are natural languages regular?
Natural languages aren't regular languages because they are not languages. There is no rigorous definition of what is or is not a member of the language.

eg. English is not a regular language
The white male hired another white male.
A white male who a white male hired
-hired another white male.

Therefore, the language $L_{trg}$ is a subset of English:
$L_{trg} = \{$ A white male (whom a white male)$^n$ (hired)$^n$ hired another white male $| n > 0 \}$

- $L_{trg}$ is not a regular language.
- $L_{trg}$ is the intersection of the natural language English with the regular set
$L_{reg} = \{$ A white male (whom a white male)* (hired)* hired another white male$\}$

Lreg is regular, as it is defined by a regular expression.
- Since, the regular languages are closed under intersection and since Lreg is a regular language, then if English were regular, its intersection with Lreg namely Ltrg would be regular. Since Ltrg is trans-regular. English is not a regular language.

Finite automata for NLP

Mathematically, an automaton can be represented by a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where

- Q is a finite set of ~~state~~ states.
- $\Sigma$ is a finite set of symbols called the alphabet of the automaton.
- $\delta$ is the transition function
- $q_0$ is the initial state from where any i/p is processed ($q_0 \in Q$).
- F is a set of final state/states of Q. ($F \subseteq Q$)

Finite state automation is of 2 types:

① Deterministic finite Automation (DFA)

② Non-deterministic finite Automation (NDFA)

① DFA:

Mathematically, a DFA can be represented by a 5-tuple $(Q, \Sigma, \delta, q_0, F)$

Q : finite set of states

$\Sigma$ : finite set of symbols

$\delta$ : transition fu$^n$ where $\delta: Q \times \Sigma \longrightarrow Q$.

$q_0$ initial state from where any i/p is processed ($q_0 \in Q$)

F is a set of final state/states of Q ($F \subseteq Q$)

DFA can be represented by diagraphs called state diagrams where

- The states are represented by vertices.
- The transitions are shown by labeled arcs.
- The initial state is represented by an empty incoming arcs.
- The final state is represented by a double circle.

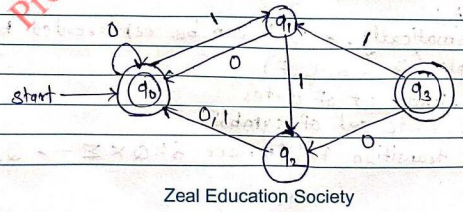
eg. $M = (Q, \Sigma, \delta, q_0, F)$

$Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{0, 1\}$

$F = \{q_0, q_3\}$

| States | 0 | 1 |
|---|---|---|
| $q_0$ | $q_0$ | $q_1$ |
| $q_1$ | $q_0$ | $q_2$ |
| $q_2$ | $q_0$ | $q_0$ |
| $q_3$ | $q_2$ | $q_1$ |



## ② NDFA :

NDFA can be represented by a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where

Q is a finite set of ~~stles~~ states.

$\Sigma$ is a finite set of symbols called the alphabet of the automation.

$\delta$ is the transition fun$^n$ where $\delta : Q \times \Sigma \rightarrow 2^Q$.

$q_0$ is the initial state from where any input is processed ($q_0 \in Q$).

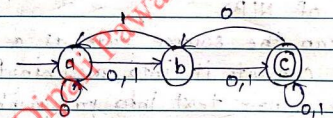
F is a set of final state / states of Q ($F \subseteq Q$)

eg. $M = (Q, \Sigma, q_0, \delta, F)$

where $Q = \{a, b, c\}$

$\Sigma = \{0, 1\}$

$F = \{c\}$




| State | 0 | 1 |
|---|---|---|
| a | a, b | b |
| b | c | a, c |
| c | b, c | c |

Stages of NLP:
There are general 5 stages in NLP:

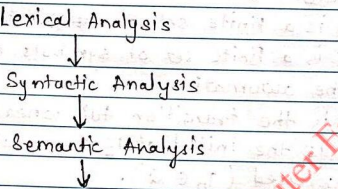



fig. General five stages in NLP

1) Lexical Analysis
- The 1st phase of NLP.
- This phase scan the source code as a stream of characters & converts it into meaningful lexemes. It divides the whole text into paragraphs, sentences & words.

2) Syntactic Analysis (Parsing)
It is used to check grammar, word ~~segm~~ arrangements & shows the relationship among the words.

3) Semantic Analysis
It is concerned with the meaning representation. It mainly focuses on the literal meaning of words, phrases & sentences.

4) Discourse Integration
It depends upon the sentences that proceeds it and also invokes the meaning of the sentences that follow it.

5) Pragmatic Analysis
- It is the 5th & last phase of NLP.
- It helps you to discover the intended effect by applying a set of rules that characterize cooperative dialogues.

Challenges and Issues (Open Problems) in NLP
1) Language differences
Different languages have not only different sets of vocabulary, but also different types of phrasing, different modes of inflection & different cultural norms.

2) Training data
The training data given to an NLP system determines its capabilities. If you feed the system bad or questionable data, it's going to learn the wrong things or learn in an inefficient way.

3) Development time
The total time taken to develop an NLP system is higher. AI evaluates the data points to process them & use them accordingly.

4) Phrasing Ambiguities
5) Misspellings
6) Words with multiple meanings.
7) Phrases with multiple intentions.
8) False positive & uncertainty
9) Keeping a conversation moving.

Basics of text processing:
Many redundant words such as stopwords, misspellings & slang can be found in most text & document datasets. Noise & superfluous characteristics can degrade system performance in many algorithms. There are several text cleaning & pre-processing approaches & methodologies.

Tokenization

- It is a pre-processing technique that divides a stream of text into tokens which are words, phrases, symbols or other meaningful pieces.
- Both text classification & text mining necessitate the use of a parser to handle the tokenization of documents.

eg. He opted to sleep for another four hours after four hours of sleep.
The tokens in this scenario are as follows:
'After', 'four', 'hours of', 'sleeping', 'he', 'decided', 'to', 'sleep', 'for', 'another', 'four'.

## Stemming

A single word might exist in multiple forms, all of which have same semantic meaning. Stemming is one way for combining different variants of words into the same feature space.
The stem of the word 'studying' is 'study'.

## Lemmatization

It replaces a words suffix with a different one or eliminates the suffix to reveal the fundamental word form (lemma).

## Part of Speech Tagging

- This step involved taking each word from the previous step & classify it as to what part of speech it represents.
- This is an essential step for identifying the meaning behind a text.
- Identifying the nouns allows us to figure out who or what the given text is about.
- Then the verbs and adjectives let us understand what entities do, or how they are described or any other meaning we can get from a text.

# Unit II Language Syntax and Semantics

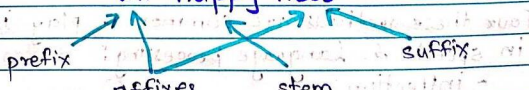


## Morphological Analysis

### What is Morphology ?

* It is study of word information. Some words can be divided into two parts but still have meaning.
- Eg. handsome (hand + some)
* Many words have meaning in themselves but some words have meaning only when used with other words.
- Eg. sleeping, colourful.
* Some words can stand alone.
- Eg. free, cat, run, tree, table, local etc.
* Words parts must be combined in the correct way and systematically.
* Morphology deals with the syntax of complex words and parts of word also called morphemes, as well as with the semantics of their lexical meanings.
* Morpheme is defined as "minimal meaning-bearing unit in a language.
* Morphology handles the formation of words by using morphemes base form (stem, lemma) eg. believe affixes (suffixes, prefixes, infixes) eg. un-, able, -ly.

eg. word - unhappyness morphemes



There are 3 morphemes, each carrying a certain amount of meaning.
un means "not"
ness means "being in a state or condition
happy is a free

* Types of Morphemes

Divide morphemes into 2 broad classes of morphemes
- stem :- main morpheme of the word
- affixes :- add additional meaning of various kinds.
  affixes are further divided into prefixes, suffixes, infixes & circumfixes.
  - prefixes precede the stem
  - suffixes follow the stem
  - circumfixes do both
  - infixes are inserted inside the stem

eg. The word cats is composed of a stem cat & the suffix s.
The word undo is composed of a stem do & the prefix un
The good examples of circumfixes are eg. Unreachable, Unbelievable.
Infixes in which a morpheme is inserted in the middle of the word, occur very commonly

Four these methods are common & play important roles in speech & language processing:
- inflection
- derivation
- compounding
- cliticization

Inflection is the combination of a word stem with a grammatical morpheme, usually resulting in a word of the same class as the original stem and usually filling some syntactic fun like argeement. eg. clean(v), cleaning (v)

Derivation is the combination of word stem with a grammatical, morpheme, usually resulting is a word of the different class, often with a meaning hard to predict exactly. eg. delight (v), delightful (adj)

Compounding is the combination of multiple words stems together.
eg. notebook, bookstore, fireman, etc.

Clitization is the combination of a word stem with a clitic
A clitic is a morpheme that acts syntactically like a word, but is reduced in form & attached to another word.
eg. English morpheme've in the word I've is a clitic.

★ Inflectional Morphology
Inflectional affixes are only suffixes which express grammatical features such as singular/plural, past/present forms of verbs. The study of such morphemes is called inflectional morphemes.
eg. bigger (big + er)
loves (love + s)

English has a relatively simple inflectional system only nouns, verbs & sometimes adjectives can be inflected & the no. of possible inflectional affixes is quite small.

Table 1: Nominal Inflection.

| | Regular Nouns | | Irregular Nouns | |
|---|---|---|---|---|
| Singular | cat | thrush | mouse | ox |
| Plural | cats | thrushes | mice | oxen |

* English verbal inflection is more complicated than nominal inflection

English has 3 kinds of verbs

- main verbs (eat, sleep, impeach)
- modal verbs (can, will, should)
- primary verbs (be, have, do)

Table 2: Morphological Forms for regular verbs.

| Morphological forms | Regularly Inflected Verbs | | | |
|---|---|---|---|---|
| stem | talk | merge | cry | map |
| -s form | talks | merges | cries | maps |
| -ing participle | talking | merging | crying | mapping |
| Past form or -ed participle | talked | merged | cried | mapped |

Table 3 : Morphological forms of irregular verbs

| Morphological form Classes | Irregularly Inflected Verbs | | |
|---|---|---|---|
| stem | eat | catch | put |
| -s form | eats | catches | puts |
| -ing participle | eating | catching | putting |
| past form | ate | caught | put |
| -ed participle | eaten | caught | put |

★ Derivational Morphology

- Derivation is concerned with the way morphemes are connected to existing lexical forms as affixes.
- The common kind of derivation in English is the formation of new nouns often from verbs or adjectives. This process is called nominalization.

Table 4 : formation of new nouns

| Suffix | Base Verb/ Adjective | Derived Noun |
|---|---|---|
| -ation | computerize (V) | computerization |
| -ee | appoint (V) | appointee |
| -er | kill (V) | killer |
| -ness | fuzzy (A) | fuzziness |

| Suffix | Base Noun / Verb | Derived Adjective |
|---|---|---|
| -al | computation (N) | computational |
| -able | embrace (V) | embraceable |
| -less | clue (N) | clueless |

## ★ Morphological parsing with finite state Transducers (FST)

* FST allow the surface structure to be mapped into the list of morphemes.
* FST are useful for both analysis & generation bez the mapping is bidirectional. This approach is known as two-level morphology.
* FST is a type of finite automation which maps betⁿ two sets of symbols.

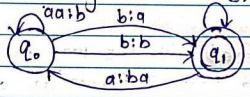



fig. A finite set transducer

* FSA defines a formal lang. by defining a set of strings, an FST defines a relⁿ betⁿ sets of strings. FST is as machine that reads one string & generates another.
* FST can be defined with 7 parameters

Q : a finite set of N states $q_0, q_1 \ldots q_{N-1}$

Σ : a finite set corresponding to the i/p alphabet

Δ : a finite set corresponding to the o/p alphabet

$q_0 \in Q$ : start state

$F \subseteq Q$ : set of final states

$\delta(q,w)$ : the transition funⁿ betⁿ states; Given a state $q \in Q$ & string $w \in \Sigma^*$, $\delta(q,w)$ returns a set of new states $Q' \in Q$. δ is thus a funⁿ from $Q \times \Sigma^*$ to $2^Q$.

* **Syntactic Analysis**

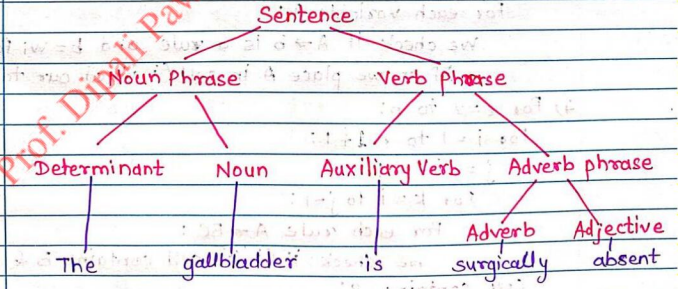
It consists of analysis of words in the sentence for - grammar & ordering words in a way that shows the relationship among the words.

★ **Syntactic representations of Natural Language**

- Syntactic phenomena
  eg. subject of a verb, relative clause, small clause
- Mathematical representation type
  eg. phrase structure tree, dependency tree.
- Formal syntactic description:

a) Mapping from phenomena to representations
b) Chooses representation for a specific phenomenon also called analysis.
c) Phenomena extracted in representation are the interpretation
d) Formal description is a syntactic theory if it makes prediction





## ★ Parsing Algorithms

### CYK Algorithm

* CYK means Cocke - Kasami - Younger.
* It is a parsing algorithm for context free grammars. In order to apply CYK algo. to a grammar, it must be in Chomsky Normal Form. It uses a dynamic programming algo. to tell whether a string is in the language of a grammar.

**Algorithm :**

Let w be the n length string to be parsed. And G represents the set of rules in our grammar with start state S.

1) Construct a table DP for size n x n.
2) If w = e (empty string) & S → e is a rule in G then we accept the string else we reject.
3) For i = 1 to n :
   For each variable A :
   We check if A → b is a rule and b = wi for some i :
   If so, we place A in cell (i, i) of our table.
4) For l = 2 to n :
   For i = 1 to n - l + 1 :
   j = i + l - 1
   For k = i to j - 1 :
   For each rule A → BC :
   We check if (i, k) cell contains B & (k+1, j) cell contains C :
   If so, we put A in cell (i, j) of our table

5) We check if S is in $(1,n)$:
If so, we accept the string else, we reject

Eg. Grammar G be:

$S \rightarrow AB \mid BC$
$A \rightarrow BA \mid a$
$B \rightarrow CC \mid b$
$C \rightarrow AB \mid a$

Check if baaba is in L(G).

① first insert single length rules into our table

| | b | a | a | b | a |
|---|---|---|---|---|---|
| b | {B} | | | | |
| a | | {A,C} | | | |
| a | | | {A,C} | | |
| b | | | | {B} | |
| a | | | | | {A,C} |

② Then fill the remaining cells of our table

| | b | a | a | b | a |
|---|---|---|---|---|---|
| b | {B} | {S,A} | $\phi$ | $\phi$ | {S,A,C} |
| a | | {A,C} | {B} | {B} | {S,A,C} |
| a | | | {A,C} | {A,C} | {B} |
| b | | | | {S,C} | {S,A} |
| a | | | | {B} | {A,C} |

③ Observe that S is in the cell (1,5).
Hence, the string baaba belongs to L(G).

- Time Complexity : $O(n^3 \cdot |G|)$
  where $|G|$ is the no. of rules in the given grammar.

- Space Complexity : $O(n^2)$

★ Probabilistic context-free grammers

Probabilistic context-free grammar (PCFG) also known as the Stochastic Context-Free Grammar (SCFG)
It is defined by following components.
N : a set of non-terminal symbols (or variables)
Σ : a set of terminal symbols
R : a set of rules or productions, each of the form
$A \rightarrow \beta [p]$
where, A is a non-terminal
β is a string of symbols from the infinite set of strings $(\Sigma \cup N)^*$
p is a number betⁿ 0 & 1 expressing $P(\beta|A)$
S : a designated ~~part~~ start symbol.

★ Statistical parsing

* It is a task of computing the most probable parse of a sentence given a probabilistic (or weighted) CFG.
* The weights of the probabilistic or weighted CFG are typically learned on a corpus of texts.
* A CFG in Chomsky normal form consists of rules
  $A \rightarrow BC$
  $A \rightarrow a$

where A, B and C are non-terminal symbols & a is a terminal symbol.

## Semantic Analysis

* Semantic analysis is to draw exact meaning or you can say dictionary meaning from the text.
* It allows computers to understand and interpret sentences, paragraphs or whole documents, by analyzing their grammatical structure & identifying relationships between individual words in particular context.
* Semantic features are theoretical units of meaning-holding components which are used for representing word meanin These features play a vital role in determining the kind of lexical reln which exists betn words in a language.
* Semantics of a language provide meaning to its constructs like tokens and syntax structure. Semantics help interpret symbols, their types & their relations with each other.
* It judges whether the syntax structure constructed in the source program derives any meaning or not. Following tasks should be performed in semantic analysis :-
  - Scope resolution
  - Type checking
  - Array bound checking

## Lexical semantic

* This technique calculates the sentiment orientations of the whole document, or set of sentences from semantic orientation of lexicons.

* Semantic orientation can be positive, negative or neutral.
* The dictionary of lexicons can be created manually as well as automatically generated.

The different approaches to lexicon-based approach are:

(1) Dictionary based approach
Dictionary is created by taking a few words initially. Then an online dictionary, thesaurus or WordNet can be used to expand that dictionary by incorporating synonyms & antonyms.

(2) Corpus-based approach
The two methods

(a) Statistical approach:
- The words which show erratic behavior in positive behavior are considered to have +ve polarity.
- If they show negative recurrence in negative text they have negative polarity.
- If the freq. is equal in both +ve & -ve text then the word has neutral polarity

(b) Semantic approach
assigns sentiment values to words & the words which are semantically closer to those words, this can be done by finding synonyms & antonyms with respect to that word.

© Lexical semantics

- also known as lexicosemantic, as a subfield of linguistic semantics, is the study of word meanings.
- The following are the steps involved in lexical semantics
* Classification of lexical items like words, sub-words, affixes, etc. is performed in lexical semantics.
* Decomposition of lexical items like words, sub-words affixes etc. is performed in lexical semantics.
* Differences as well as similarities betn various lexical semantics structures is also ~~aa~~ analyzed.

★ Relations among lexemes & their senses

* A lexeme is an ~~tdr~~ individual entry in the lexicon. ☲
* A lexicon is meaning structure holding meaning relations of lexeme. A lexeme may have different meanings. A lexeme's meaning component is known as one of the its senses.

Sense Relation

* As the relations of meaning betn words, as expressed in synonymy, hyponymy & antonymy.
* Two major types –
  - sense reln of inclusion, esp. hyponymy & synonymy
  - sense reln of exclusion, esp. complementarity & antonymy.

## Homonymy

It is the relationship betⁿ words that are homonyms - words that have different meanings but are pronounced the same or spelled the same or both.

eg. Homonymy with meaning & sentences

- Cache - cash
- Scents - sense
- chile - chili
- choir - Quire
- Site - Sight
- Facts - Fax
- Finnish - Finish

## Polysemy

When symbol, word or phrase means many different ~~meaning~~ things that called polysemy.

The verb get is a good eg. of polysemy - it can mean procure, become or understand.

Types of polysemy

4 types :

autohyponymy
automeronymy
autosuperordination
autoholonymy

## Hyponymy

- It is the subordinate relation of a word.
- It is semantic rel$^n$ of being subordinate or belonging to a lower rank or class.
  eg. sweep, wipe & scrub are hyponyms of clean.
- Words on the superordinate level are called hypernym & words on the subordinate level are called hyponyms.

## WordNet

- It is a large lexical database of English, which was created by Princeton.
- It is part of the NLTK corpus.
- Nouns, verbs, adjectives & adverbs all are grouped into set of synsets. i.e. cognitive synonyms.
  Here each set of synsets express a distinct meaning.

## Word Sense Disambiguation (WSD)

It is the problem of determining which sense (meaning) of a word is activated by the use of the word in a particular context, a process which appears to be largely unconscious in people.

## Dictionary based approach

* In this approach dictionary is created by taking a few words initially. Then an online dictionary, thesaurus or WordNet can be used to expand that dictionary by incorporating synonyms & antonyms of those words.
* The dictonary is expanded till no new words can

be added to that dictionary. The dictionary can be redefined by manual inspection

Latent semantic Analysis.

It is a natural language processing method that uses the statistical approach to identify the association among the words in a document.

eg. mobile, phone, cell phone, telephone are all similar but if we pose a query like 'The cell phone has been ringing' then the documents which have 'cell phone' are only retrieved whereas the documents containing the mobile, phone, telephone are not retrieved