Machine Learning

NLP

What is NLP?

- Natural Language Processing (NLP) refers to AI method of communicating with an intelligent systems using a natural language such as English
- It is the sub-field of AI that is focused on enabling computers to understand and process human language
- The ultimate objective of NLP is to read, decipher, understand, and make sense of the human languages in a manner that is valuable
- Most NLP techniques rely on machine learning to derive meaning from human languages

Uses cased of NLP

- NLP enables the recognition and **prediction of diseases** based on electronic health records and patient's own speech
- Organizations can determine what customers are saying about a service or product by identifying and extracting information in sources like social media (**sentiment analysis**)
- Companies like Yahoo and Google filter and classify your emails with NLP by analyzing text in emails that flow through their servers and **stopping spam** before they even enter your inbox
- To help **identifying fake news**, a system can be developed to determine if a source is accurate or politically biased, detecting if a news source can be trusted or not
- Amazon's Alexa and Apple's Siri are examples of intelligent **voice driven interfaces** that use NLP to respond to vocal prompts
- NLP is also being used in both the search and selection phases of **talent recruitment**
- NLP is particularly booming in the **healthcare industry**

Terminology

- **Phonology**
  - It is study of organizing sound systematically.
- **Morphology**
  - It is a study of construction of words from primitive meaningful units.
- **Morpheme**
  - It is primitive unit of meaning in a language.
- **Syntax**
  - It refers to arranging words to make a sentence. It also involves determining the structural role of words in the sentence and in phrases.
- **Semantics**
  - It is concerned with the meaning of words and how to combine words into meaningful phrases and sentences.

② ffice go to I .

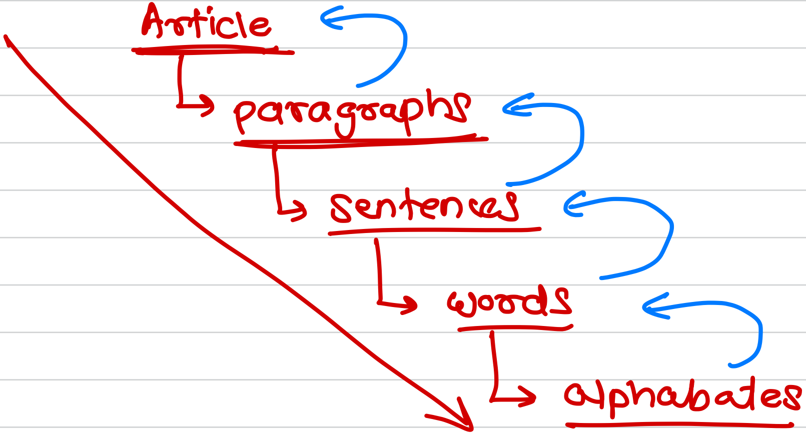
① I go to ffice.

Terminology

is good , is not good

- **Pragmatics**
  - It deals with using and understanding sentences in different situations and how the interpretation of the sentence is affected.
- **Discourse**
  - It deals with how the immediately preceding sentence can affect the interpretation of the next sentence.
- **World Knowledge**
  - It includes the general knowledge about the world.

My name is Amit. I stay in Pune.
I love technologies



Article

paragraphs

sentences

words

alphabates

① I like Apple in my breakfast. : fruit

② Apple has again stood on first place ... : company

India : place name

Steve : person name

Steps in NLP

I go to office

→ Syntax

1 **Lexical Analysis**

It involves identifying and analyzing the structure of words. Lexicon of a language means the collection of words and phrases in a language. Lexical analysis is dividing the whole chunk of txt into paragraphs, sentences, and words → collection of words

2 **Syntactic Analysis**

It involves analysis of words in the sentence for grammar and arranging words in a manner that shows the relationship among the words. The sentence such as “The school goes to boy” is rejected by English syntactic analyzer → grammatically correct sentence

3 **Semantic Analysis**

It draws the exact meaning or the dictionary meaning from the text. The text is checked for meaningfulness. It is done by mapping syntactic structures and objects in the task domain.

↳ words of their meanings (verb / noun / pronoun)

4 **Disclosure Integration**

The meaning of any sentence depends upon the meaning of the sentence just before it. In addition, it also brings about the meaning of immediately succeeding sentence.

5 **Pragmatic Analysis**

During this, what was said is re-interpreted on what it actually meant. It involves deriving those aspects of language which require real world knowledge → apple: fruit/ company

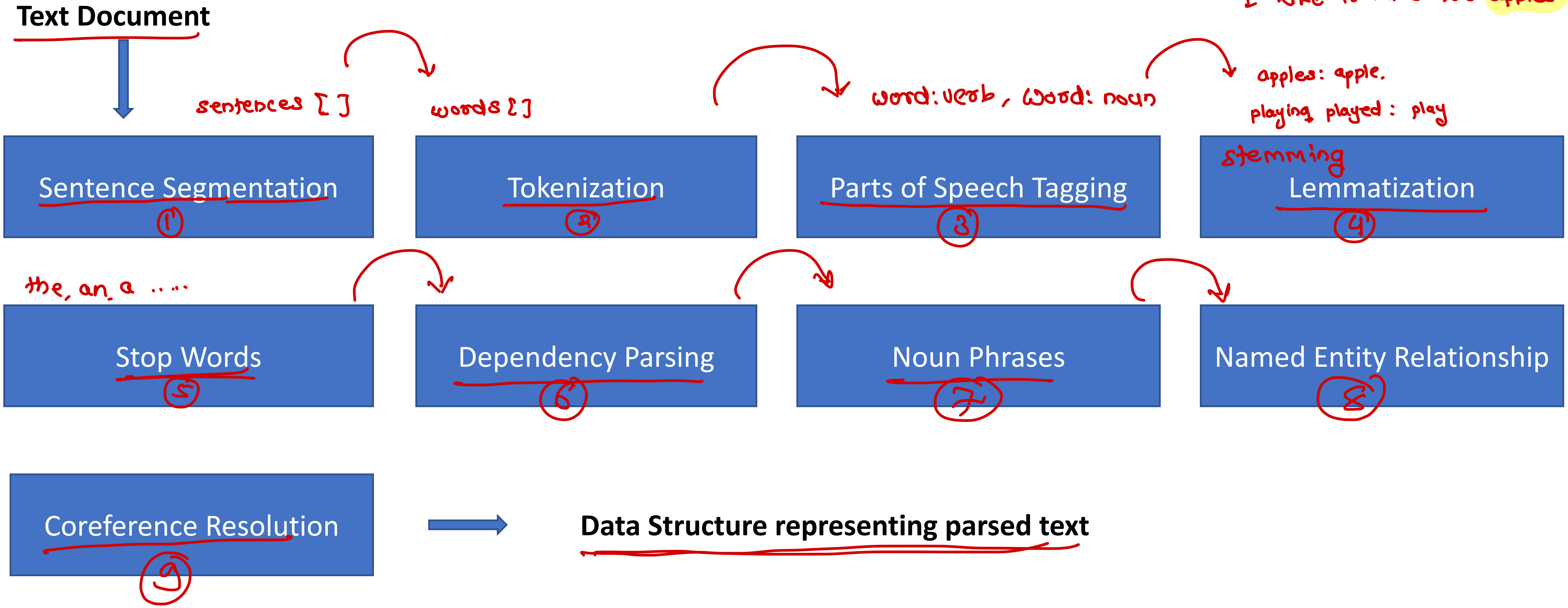
① The restaurant is awesome. : good

② The restaurant is not so bad. : good

③ The restaurant is bad. : bad —

Creating NLP pipeline

I like to have an apple.
I like to have two apples.



Text Document

sentences [ ]

words [ ]

word: verb, word: noun

apples: apple.

playing played : play

Sentence Segmentation ①

Tokenization ②

Parts of Speech Tagging ③

stemming

Lemmatization ④

the, an, a .....

Stop Words ⑤

Dependency Parsing ⑥

Noun Phrases ⑦

Named Entity Relationship ⑧

Coreference Resolution ⑨

Data Structure representing parsed text

Sentence Segmentation

- The first step in the pipeline is to break the text apart into separate sentences
- We can assume that each sentence in English is a separate thought or idea
- It will be a lot easier to write a program to understand a single sentence than to understand a whole paragraph
- Coding a Sentence Segmentation model can be as simple as splitting apart sentences whenever you see a punctuation mark

↳ [. , ? ! . -]

Word Tokenization

- Now that we’ve split our document into sentences, we can process them one at a time
- Tokenization is the process of splitting the sentence into separate words
- Tokenization is easy to do in English. We’ll just split apart words whenever there’s a space between them. And we’ll also treat punctuation marks as separate tokens since punctuation also has meaning.

Predicting Parts of Speech for Each Token

- Next, we’ll look at each token and try to guess its part of speech — whether it is a noun, a verb, an adjective and so on
- Knowing the role of each word in the sentence will help us start to figure out what the sentence is talking about
- We can do this by feeding each word (and some extra words around it for context) into a pre-trained part-of-speech classification model
- The part-of-speech model was originally trained by feeding it millions of English sentences with each word’s part of speech already tagged and having it learn to replicate that behavior.
- Keep in mind that the model is completely based on statistics — it doesn’t actually understand what the words mean in the same way that humans do
- It just knows how to guess a part of speech based on similar sentences and words it has seen before

Text Lemmatization

- In English words appear in different forms. E.g.
  - I ate an apple
  - I ate two apples

} apple

- Both sentences talk about the noun **apple,** but they are using different inflections
- When working with text in a computer, it is helpful to know the base form of each word so that you know that both sentences are talking about the same concept
- Otherwise the strings like apple and apples will look lie two totally different words to computer
- In NLP, we call finding this process *lemmatization* — figuring out the most basic form or *lemma* of each word in the sentence.
- Lemmatization is typically done by having a look-up table of the lemma forms of words based on their part of speech and possibly having some custom rules to handle words that you've never seen before

playing → play

played → play

Identifying Stop Words

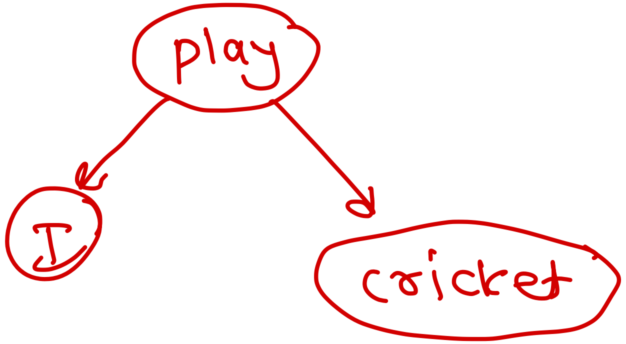
- Next, we want to consider the importance of a each word in the sentence. English has a lot of filler words that appear very frequently like “and”, “the”, and “a”
- When doing statistics on text, these words introduce a lot of noise since they appear way more frequently than other words
- Some NLP pipelines will flag them as **stop words** —that is, words that you might want to filter out before doing any statistical analysis

nltk/ BS4

- Stop words are usually identified by just by checking a hardcoded list of known stop words
- But there’s no standard list of stop words that is appropriate for all applications. The list of words to ignore can vary depending on your application

Dependency Parsing

- The next step is to figure out how all the words in our sentence relate to each other. This is called *dependency parsing*.
- The goal is to build a tree that assigns a single **parent** word to each word in the sentence
- The root of the tree will be the main verb in the sentence



Finding Noun Phrases

- So far, we've treated every word in our sentence as a separate entity
- But sometimes it makes more sense to group together the words that represent a single idea or thing
- We can use the information from the dependency parse tree to automatically group together words that are all talking about the same thing
- Whether or not we do this step depends on our end goal
- But it's often a quick and easy way to simplify the sentence if we don't need extra detail about which words are adjectives and instead care more about extracting complete ideas.

restaurant is bad → bad

restaurant is not bad → good

Named Entity Recognition (NER)

- The goal of *Named Entity Recognition*, or *NER*, is to detect and label these nouns with the real-world concepts that they represent
- But NER systems aren't just doing a simple dictionary lookup. Instead, they are using the context of how a word appears in the sentence and a statistical model to guess which type of noun a word represents.
- Here are just some of the kinds of objects that a typical NER system can tag
  - People's names
  - Company names
  - Geographic locations (Both physical and political)
  - Product names
  - Dates and times
  - Amounts of money
  - Names of events

Coreference Resolution

- Coreference resolution is one of the most difficult steps in our pipeline to implement
- It’s even more difficult than sentence parsing
- It is the process of finding the references from the previous sentences

↳ Disclounare integration