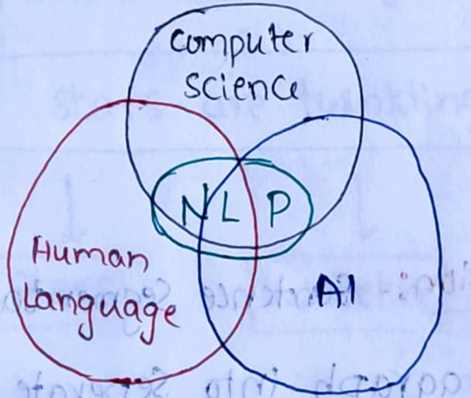
Natural Language Processing:-

Natural Language Processing or NLP refers to the branch of Artificial Intelligence that gives the machines the ability to read, understand and derive meaning from human languages.

→ NLP combines the field of linguistics and computer science to decipher language structure and guidelines and to make models which can comprehend, breaddown and separate significant details from text & speech.



(*) Components of NLP

1) NLU
2) NLG

Natural Language Understanding	Natural Language Generation.
1. 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	1. NLG acts as a translator that converts the computerized data into natural language representation. It involves Text planning, Sentence planning and Text Realization
2. NLU is the process of reading and interpreting language.	2. NLG is the process of writing or generating language.

Applications of NLP:

1. Question Answering Ex:- Alexa

2. Spam Detection Ex:- Spam mail detection

3. Sentiment Analysis Ex:- Delicious food (+ve)
   Unhappy with order (-ve)

4. Machine Translation Ex:- Google Translation (text or Speech)

5. Spelling correction Ex:- Grammerly

6. Chat bot Ex:- Customer support.

7. Information Extraction Ex:- Resume ATS

NLP Pipeline:-

Steps Involved -

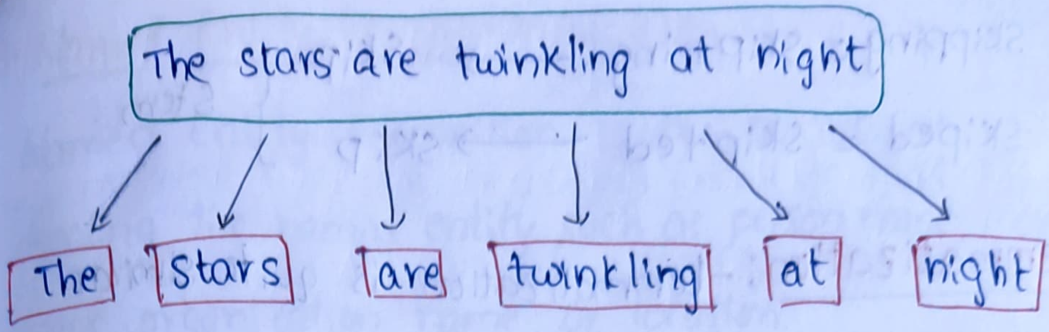
1. Sentence Segmentation:- Sentence segmentation is used to breaks the paragraph into seperate sentences.

Ex:- A boy is playing cricket. Match started at 10 AM.
He is soo tired.

After SS ⇒ 1. A boy is playing cricket
2. Match started at 10 AM.
3. He is soo tired.

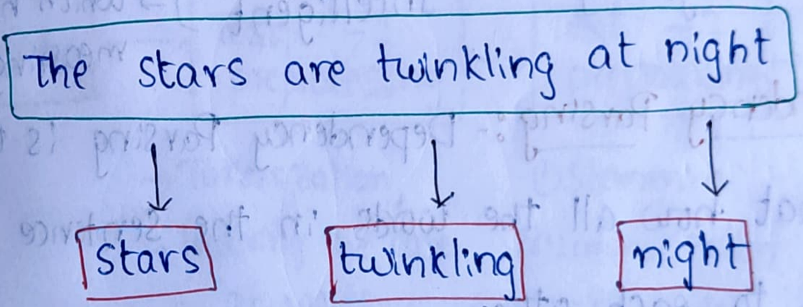
2. Word Tokenization:- Word Tokenization is used to break the sentence into seperate words or tokens.

Tokenizer genarates the following result.



Each Word is called a token.

3. Removing Stop Words:- In English, there are a lot of words that appear very frequently like "is", "and", "the", "a". NLP pipelines will flag these words as stop words. Stop words might be filtered out before doing any statistical analysis.



4. Stemming:- Stemming is used to normalize words into its base form or root form. For example, Celebrates, celebrated and celebrating, all these words are originated with a single root word "celebrate." The big problem with stemming is that sometimes it produces the root word which may not have any meaning.

Ex:

Intelligence → Intelligen
Intelligent → Intelligen
Intelligently → Intelligen

} Stem

$$\left.\begin{array}{l}\text{skipping} = \text{skip+ing} \longrightarrow \text{skip}\\ \text{skiped} = \text{skip+ed} \longrightarrow \text{skip}\end{array}\right\} \text{Stem.}$$

5. **Lemmatization**:- Lemmatization is quite similar to the stamming. It is used to group different inflected forms of the word called Lemma. The main difference b/w stemming & lemmatization is that it produces the root-word, which has a meaning.

Ex:-

$$\left.\begin{array}{l}\text{Intelligence} \longrightarrow \text{Intelligent}\\ \text{Intelligent} \longrightarrow \text{Intelligent}\\ \text{Intelligently} \longrightarrow \text{Intelligent}\end{array}\right\} \begin{array}{l}\text{Lemma}\\ \Downarrow\\ \Rightarrow \text{which has}\\ \text{meaning}\end{array}$$

6. **Dependency Parsing**:- Dependency Parsing is used to find that how all the words in the sentence are related to each other.

7. **Part of Speech Tagging**:- Now, we must explain the concept of nouns, verbs, articles and other parts of speech to the machine by adding these tags to our words.

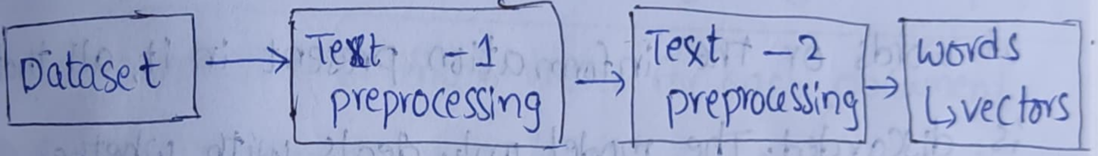
Determiner	Noun	Verb	Adjective	Preposition	Noun
↑	↑	↑	↑	↑	↑
the	stars	are	twinkling	at	night

8. Named Entity Recognition (NER):-

Named Entity Recognition is the process of detecting the named entity such as person name, movie name, organization name, or location.

Ex:- Steve Jobs introduced iPhone.

9. Chunking:- Chunking is used to collect the individual piece of information and grouping them into bigger pieces of sentences.



→ Tokenization,
→ lowering the case of words

① Stemming
② Lemmatization
③ Stopwords

① BOW
② TF IDF
③ word2vec

→ Basic terminologies used in NLP

① CORPUS ⇒ Paragraph

② Documents ⇒ Sentence

③ Vocabulary ⇒ Unique words

④ Word ⇒ Word

Bag of Words (BoW):-

The Bag of Words (BoW) model is a representation that turns arbitrary text into fixed-length vectors by counting how many times each word appears. This process is often referred to as vectorization.

→ Bag of Words works on two things:

1. A known word vocabulary

2. A measure of how many known words are present.

→ The model does not consider the order or structure of words or the information present in it, all that is discarded. The model only deals with whether known words occur in the document, even without considering where in the document.

Steps:-

1. Data collection: consider 3 lines of text as a seperate document which needs to be vectorized.

① the dog sat

② the dog sat in the hat

③ the dog with the hat.

2. Determine the Vocabulary:

Vocabulary is defined as the set of all the words found in the documents. The words in the document above : the, dog, sat, in, ~~the~~, hat, with

3. Counting: The vectorization process involves counting the number of times each word appears.

Document	the	dog	Sat	in	hat	with
The dog Sat	1	1	1	0	0	0
The dog sat in the hat	2	1	1	1	1	0
The dog with the hat	1	1	0	0	1	1

This generates a 6-length vector for each document.

→ As you can see, the bow vector only contains info about what words occurs and how many times without contextual information or where they occur.

4. Managing Vocabulary:

As we can see from prev. example, as vocabulary grows the vector representation in the documents also grows. This means that for very large documents, books the vector length can stretch up to thousands of positions. Since each document can also contain a few known words, that create a lot of empty lots with zeros called a sparse vector.

→ We use data cleaning methods to reduce the size of the vocabulary. This includes ignoring case, punctuation, fixing misspelt words, ignoring stop words.

5. Scoring words :- Scoring the words is simply attaching a numerical value to mark the occurence of the words. In above example, scoring was binary:

→ presence or absence of words.

other scoring methods include:

- Counts: this is to count every time the word appears in the document.
- Frequencies: Calculate the frequency of the words in a document in contrast to the total words in the document.

Disadvantages:-

① Sparsity

② Ordering of words :

③ Semantic meaning not able to capture.

Advantages

① Simple & Intuitive.

N-grams

N-gram is a sequence of the N-words in the modeling of NLP.

→ Unigram or One gram :- There is a one-word sequence

Ex:- This is a sentence ⟶ This, is, a, sentence

→ bi-gram or two-gram:- Two-word sequence.

Ex:- This is a sentence ⟷ This is, is a, a sentence

→ Tri-gram or three-gram:- Three-word sequence.

Ex:- This is a sentence ⟶ This is a, is a sentence

→ Same way we can Calculate N-grams

Applications:- Speech recognition, machine translation etc.

TF-IDF:-

TF-IDF stands for Term Frequency - Inverse Document Frequency, and the tf-idf weight is a weight often used in information retrieval and text mining. this weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the words in the corpus.

→ Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

→ TF-IDF can be successfully used for stop-words filtering in various subject fields including.

text summarization and classification.

TF - Term Frequency: Which measures how frequently a term occurs in a document.

$$TF = \frac{\text{No. of times term } t \text{ appers in a document}}{\text{Total no. of terms in the document.}}$$

IDF - Inverse Document Frequency: Which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of" "that" may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones.

$$IDF = \log_e\left[\frac{\text{Total no. of documents}}{\text{No. of documents with term } t \text{ in it}}\right]$$

Example:- Lets consider 3 sentences (documents)

① Good boy

② Good girl

③ Boy girl Good

("Here boy should be given more importance or weight than good, since less frequent in the corpus)

→ Find the vocabulary in the sentences and find TF, IDF

TF X IDF

	Sent1	Sent2	Sent3
good	$1/2$	$1/2$	$1/3$
boy	$1/2$	0	$1/3$
girl	0	$1/2$	$1/3$

words	IDF
good	$\log_e(3/3) = 0$
boy	$\log_e(3/2)$
girl	$\log_e(3/2)$

→ Now Multiply TF, and IDF to obtain TF-IDF weight

	good	boy	girl
Sent 1	0	$1/2 \times \log_e(3/2)$	0
sent 2	0	0	$1/2 \log_e(3/2)$
sent 3	0	$1/3 \log_e(3/2)$	$1/3 \log_e(3/2)$

Advantages

1. Intuitive

2. Word Importance is getting capture

Disadvantages

1. Sparsity

2. Out of Vocabulary.

Word Embeddings :-

Word Embeddings are a type of word representation that allows words with similar meanings to have a similar representation.

→ Embeddings translate large sparse vectors into a lower-dimensional space that preserves semantic relationships.