Natural Language Processing (NLP)

Natural Language Processing (NLP) is a field of artificial intelligence that teaches computers to understand, interpret, and generate human language, both written and spoken. By combining computational linguistics and machine learning, NLP enables computers to process complex human communication like accents, slang, and grammar, allowing them to power everyday applications such as voice assistants (Siri, Alexa), search engines, chatbots, translation apps, and spell-checkers

NLP Process

NLP encompasses a wide array of techniques that aimed at enabling computers to process and understand human language.

- **Text Processing**: Dividing text into words,converting them to the base form and removing punctuations by applying normalization techniques to prepare raw text.
- **Syntax & Parsing**: POS tagging by parsing and analyzing grammar and breaking the sentence into phrases.
- **Semantic Analysis**: Identifying text, determining its meaning in that scenario and finding references from the earlier conversation.
- **Information Extraction**: Entity and relation extraction between key entities.
- **Text Classification**: Understanding sentiments, topic, and spam detection.
- **Language Generation**: Includes translation, summarization and text generation.
- **Speech Processing**: Converts speech to text (recognition) and vice versa(sysnthesis).
- **Question Answering**: Retrieval-based QA chooses the best answer, while generative QA creates them.
- **Dialogue Systems**: Power chatbots to have conversation with users.
- **Sentiment & Emotion Analysis**: Detects emotions and understands opinions.

NLP Techniques

1)TF-IDF(Term Frequency-Inverse Document Frequency)

- TF : Measures how often a word occurs in a document to determine its importance

  TF(T,D) =frequency of T in document d/Total terms in D

- IDF : Reduces the weight of common words across multiple documents while increasing the weight of rare words

  IDF(T,d) = log(Total number of documents(d)/Number of documents with T)

- Calculation of value : TF-IDF= TF*IDF
- Applications:
  - Document similarity and Clustering : Converting documents into numerical vectors helps in comparision,grouping
  - Text Classification : Helps in spam detection,topic classification
  - Keyword Extraction:Ranks words by importance to make summaries,highlights
  - Recommendation Systems:To suggest related products,articles

2)**Bag Of Words(BoW)**

- It turns sentences into a collection of words and counts each words frequency irrespective of its order or grammar
- Vocabulary:It is the list of all unique words from the entire dataset
- Document Representation:A vector where each element shows a frequency of words from the vocabulary used as a feature.
- Working:After selecting the most frequent words,you choose the top n words with the highest frequency and build an BoW model from it by a binary matrix where each row corresponds to a sentence and each column represents one of the top N frequent words. A one in the matrix shows that the word is present in the sentence and a zero shows its absence and then we finally create a word cloud to identify most common words.
- Advantages
  - It is computationally efficient,versatile
  - Easy to implement,interpret
- Limitations
  - It ignores context ,does not capture meanings so might miss important relationships
  - With large datasets, most word vectors will be sparse,making it inefficient

3)One-Hot Encoding

- Converts categorical data into binary vectors to feed as input to neural networks
- Working:First stage is to determine the categorical variables ,then count the frequency of distinct words to identify potential groups,the third stage is to make a binary vector for each of the categories called one hot encoded vector
- Drawbacks
    - Produces high dimentional sparse vectors which can be costly to process
    - Does not use semantic connection between words

4)Word Embeddings

1. Word2Vec

    - It is a neutral word embedding technique using distributed representation models
    - Words with similar meaning have similar vectors by representing words as continuos vector spaces

2. Skip-Gram
    - It learns distributed representations of words in a continuous vector space, with the main objective to predict context words(words surrounding aroud the target) given a target word which is the opposite of CBOW
    - It preserves syntactical information, converts to lower dimentions with similar meaning vectors placed close to each other in space, with other parameters such as window being adjustable
    - It is chosen when semantic relationships are important