*Notes by **Izam Mohammed***

NLP

- *NLP basics*
  - *Working - pipeline*
  - *Application*
    - *Chatbot*
    - *Sentiment analysis*
    - *NER*
    - *Email spam detection*
  - *Challenges*
    - *Handling informal words*
    - *Ambiguity in words*
    - *Dialects (wannabe and all)*
    - *Lack of data in different languages*
- *Tokenization*
  - *What is it*
  - *Types*
    - *Word tokenization*
    - *Subword tokenization (Bite pair encoding -> BPE)*

      BPE ensures that the most common words are represented in the vocabulary as a single token while the rare words are broken down into two or more subword tokens and this is in agreement with what a subword-based tokenization algorithm does.

      Subword → sub, word

      Smarter → smart, er
    - *Sentence*
    - *Character level*
  - *Why tokenization, over 'split the sentence'*
  - *OOV words (Out of vocabulary)*
- *Corpus*
  - *What is it → collection of documents*
  - *Corpora → plural of corpus*
  - *Types*
    - *General*
    - *Special → specific topic*
    - *Multilingual*
    - *Monolingual*
    - *Parallel → same content, multiple languages*
    - *Balanced, imbalanced → category*
    - *Annotated → dedicated to NLP*

- *Stemming*
  - *What is it → Reduce the word into root form*
  - *Types*
    - *Porter → simple*
    - *Lancaster → extreme stemming, faster*
    - *Snowball → multiple languages*
  - *Challenges*
    - *Over stemming → remove too many affix (prefix + suffix)*
    - *Under stemming → not enough*
  - *Purpose / Importance*
    - *Reduce data redundancy → duplicates in data*
    - *Information retrieval*
    - *Normalization*
- *Lemmatization*
  - *What is it → consider the context and meaning of words*
  - *Types*
    - *WordNet Lemmatizer → nltk*
      Utilize the Wordnet lexical database to find the lemma
    - *Spacy Lemmatizer → spacy*
      Use rule-based and statistical-based methods to find the lemma
  - *Challenges*
    - *Slower, computationally intensive*
    - *Need dictionary lookup*
- *NER*
  - *What is it → Identify and categorize the entities in the corpus*
  - *Goal → extract structured info in unstructured data*
  - *Types of entity → Person, Org, Place, Money, Date, Time, Percentage*
  - *Usecase*
    - *Chatbot*
    - *Q&A*
    - *Information Extraction*
    - *Identify hashtags*
  - *Techniques*
    - *Rule-based*
    - *Statistical approach → ml model*
    - *Deep learning approach*
  - *Challenges*
    - *Ambiguity*
    - *Model generalization → biased to a domain*
    - *Language variation*
- *POS tagging*
  - *What → Assign grammatical categories (pos)*

- *Types*
  - *Noun (N) → cat*
  - *Verb (V) → walk*
  - *Adverb (ADV)→ very *describes a verb*
  - *Adjective (ADJ) → happy *describes a noun*
  - *Pronoun (PRON)→ he, it*
  - *Preposition (PREP)→ in, on*
  - *Conjunction (CONJ)→ and, but*
  - *Interjection (INTJ)→ wow, oh*

➢ *Ngrams*

- *What → continuous sequences of n items*
- *Types*
  - *Unigrams*
  - *Bigrams*
  - *Trigrams*
  - *4-gram, 5-grams …*
- *Application*
  - *Text prediction*
  - *Spell checker*
  - *Translation*
- *Python from scratch implementation*

➢ *Vectorization*

- *What → Convert word to numeric format*
- *Types*
  - *Label Encoding*
  - *One-hot-encoding*
  - *CountVectorizer (BOW)*
  - *TF - IDF*
    - *TF(t, d) = No. of times the term appeared in doc / Total No. terms in the doc*
    - *IDF(t, D) = log((no. of documents/ No of the documents contain t) +1)*
  - *Word2Vec*

    Represent word as a vector trained with an FFN network
    - *CBOW*

      *Context → word *fake problem solving*
    - *Skip-grams*

      Word → word (negative sampling)
  - *Glove → Global Vectors for Word Representation*

    Use a Co-occurrence matrix
  - *Fasttext*

    Operate on sub-word level. excels in handling rare words or words not seen during training

➢ *Stop words*
  - *Words that have no meaning*

➢ *Parsing*

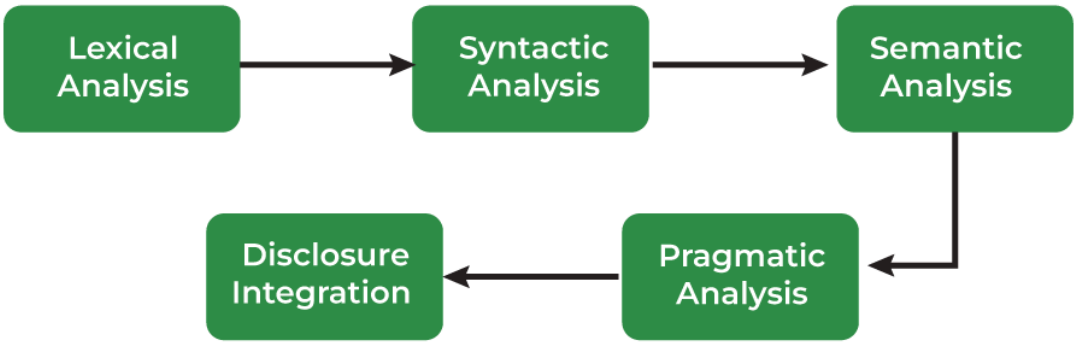
process of analyzing the grammatical structure of a sentence and how they relate to each other.

➢ *NLU vs NLP vs NLG*
  - *NLP → NLU + NLG*
  - *NLU → Natural Language Understanding*

Speech recognition, sentiment analysis
  - *NLG → Natural Language Generation*

Chatbots, Voice assistants

➢ *Phases of NLP*



  - *Lexical Analysis → tokenization*
  - *Syntactic Analysis → Parsing*
  - *Semantic Analysis → focuses on extracting the meaning of words.*
  - *Pragmatic Analysis →*

involves considering the context, speaker's intentions, and shared knowledge to interpret the true message
  - *Discousure Integration →*

Connects individual sentences into a logical discourse

➢ *Cosine Similarity*

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

  -

➢ *Visualize the text*

- *t-SNE → t-Distributed Stochastic Neighbor Embedding*
  the goal of t-SNE is to take a high-dimensional dataset and project it into a lower-dimensional space (usually 2D or 3D)

➢ *What are the options for a chatbot*

- *Rasa nlu*
- *Google dialogue flow*
- *Amazon alex*
- *Microsoft Luis*
- *Use a pre-trained transformer from Huggingface*
- *Use a LSTM model and train*
- *Use any LLM and RAG system*
- *Take an LLM and finetune*