Syllabus: Natural Language Processing Lab

Module 1:

Morphology analysis –survey of English Morphology, Inflectional morphology & Derivational morphology, Tokenization, Stemming and Lemmatization, Stop word removal Self learning topics: Python libraries for NLP (NLTK, spaCy)

Module 2:

Multiple tags & words, Unknown words. Named Entity Recognition (NER) Self learning topics: Text preprocessing

Module 3:

Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF) Self learning topics: Text representations

Module 4:

Word embeddings (Word2Vec, GloVe),Robust Word Sense Disambiguation (WSD), Dictionary based approach Self learning topics:Word2Vec Model

Module 5:

Sentiment Analysis introduction, Sentiment Analysis - Affective lexicons, Sentiment analysis techniques, Aspect based sentiment analysis Self learning topics: Chatbots

Module 6:

Text classification, Text summarization Self learning topics: Algorithms for summarization and classification

Module 1: Morphology Analysis and Preprocessing

1. **What is morphology in NLP?**

   Morphology studies the internal structure of words and how they are formed.

2. **Differentiate between inflectional and derivational morphology.**
   - *Inflectional*: Modifies a word's tense/number (e.g., walk → walked).
   - *Derivational*: Changes the word's class/meaning (e.g., teach → teacher).

3. **What is tokenization?**

   Breaking text into smaller units like words or sentences.

4. **What is stemming?**

   Removing suffixes to get the root word (e.g., "running" → "run").

5. **Define lemmatization.**

   It returns the base or dictionary form of a word, considering its context.

6. **What are stop words?**

   Common words (like "and", "the") often removed in text processing.

7. **Why is text preprocessing important?**

   It prepares raw text for effective analysis and modeling.

8. **Which is better: stemming or lemmatization?**

   Lemmatization is more accurate but slower; stemming is faster but less accurate.

9. **What are some functions of NLTK?**

   Tokenization, POS tagging, stemming, lemmatization, parsing, etc.

10. **What is spaCy used for?**

    An NLP library for fast, production-ready NLP tasks like NER, parsing, and lemmatization.

Module 2: Tagging and Named Entity Recognition

1. **What is POS tagging?**
   Assigning grammatical tags (like noun, verb) to words.
2. **What are multiple tags?**
   A word may have more than one possible tag depending on context.
3. **What is an unknown word (OOV)?**
   A word not present in the training vocabulary.
4. **How can you handle unknown words?**
   Using subword embeddings or fallback methods like character-level models.
5. **What is Named Entity Recognition (NER)?**
   Identifying names of people, places, dates, etc., in text.
6. **Give examples of named entities.**
   “Barack Obama”, “Google”, “Monday”, “India”.
7. **Which NLP libraries support NER?**
   spaCy, NLTK, Stanford NLP.
8. **How is NER useful?**
   Helps in extracting structured information from unstructured text.
9. **What are the steps in text preprocessing?**
   Lowercasing, removing punctuation/stop words, tokenization, stemming/lemmatization.
10. **What is text normalization?**
    Bringing text to a standard format (e.g., converting “U.S.A.” → “USA”).

Module 3: BoW and TF-IDF

1. **What is Bag of Words (BoW)?**

   A representation of text that counts the frequency of words.
2. **What are the limitations of BoW?**

   Ignores grammar, word order, and context.
3. **What is TF-IDF?**

   A weighting technique that reflects how important a word is to a document.
4. **How is TF calculated?**

   Frequency of a term in a document divided by total terms.
5. **How is IDF calculated?**

   Log of total documents divided by number of documents containing the word.
6. **Why is TF-IDF preferred over BoW?**

   It reduces the influence of common terms and highlights unique ones.
7. **What is vectorization in NLP?**

   Converting text into numeric vectors for machine learning models.
8. **What is the problem with sparse matrices in BoW?**

   They consume more memory and are inefficient for computation.
9. **How do you implement TF-IDF in Python?**

   Using TfidfVectorizer from scikit-learn.
10. **What are n-grams?**

    Sequences of *n* words (e.g., bigrams = 2 words).

Module 4: Word Embeddings and WSD

1. **What are word embeddings?**

   Dense vector representations capturing semantic relationships between words.

2. **How does Word2Vec work?**

   Predicts context words (CBOW) or target word (Skip-gram) using neural networks.

3. **What is GloVe?**

   A count-based embedding model using co-occurrence statistics.

4. **Difference between Word2Vec and GloVe?**
   - Word2Vec: Predictive
   - GloVe: Count-based (uses matrix factorization)

5. **What is Word Sense Disambiguation (WSD)?**

   Determining the correct meaning of a word based on context.

6. **What is the dictionary-based approach in WSD?**

   Uses lexical resources like WordNet to find correct meanings.

7. **What is semantic similarity?**

   A measure of how similar two words/phrases are in meaning.

8. **What is a vector space model?**

   Text is represented as vectors in a multi-dimensional space.

9. **What are contextual embeddings?**

   Word vectors that change based on context (e.g., BERT).

10. **Why are embeddings better than BoW?**

    They preserve semantic meaning and reduce dimensionality.

Module 5: Sentiment Analysis

1. **What is sentiment analysis?**

   The process of determining the emotional tone behind a piece of text.
2. **What are affective lexicons?**

   Dictionaries of words with associated emotions/sentiment scores.
3. **What are some lexicons used in sentiment analysis?**

   SentiWordNet, NRC, AFINN.
4. **What is polarity?**

   Indicates whether sentiment is positive, negative, or neutral.
5. **What is subjectivity?**

   Indicates whether the text expresses personal opinions or factual information.
6. **What are the main approaches to sentiment analysis?**
   - Rule-based (lexicons)
   - Machine learning
   - Deep learning
7. **What is aspect-based sentiment analysis?**

   Evaluates sentiment towards specific aspects (e.g., "Camera is great, but battery is bad").
8. **Which libraries support sentiment analysis?**

   TextBlob, VADER, spaCy, NLTK.
9. **What is a chatbot?**

   A conversational AI system that interacts with users via text or voice.
10. **What is the role of NLP in chatbots?**

    It enables understanding user queries and generating relevant responses.

Module 6: Text Classification and Summarization

1. **What is text classification?**

   Categorizing text into predefined groups (e.g., spam, sports, politics).
2. **Which algorithms are used for text classification?**

   Naive Bayes, Logistic Regression, SVM, Decision Trees.
3. **What is text summarization?**

   Producing a concise and coherent summary of a longer document.
4. **What is extractive summarization?**

   Selects important sentences from the original text.
5. **What is abstractive summarization?**

   Generates new sentences to represent the main ideas.
6. **What is a classification pipeline?**

   Text preprocessing → Feature extraction → Classification algorithm
7. **What is TextRank?**

   A graph-based extractive summarization algorithm.
8. **What are common evaluation metrics in classification?**

   Accuracy, Precision, Recall, F1 Score.
9. **Which libraries help with summarization?**

   Gensim, spaCy, Sumy, HuggingFace Transformers.
10. **What are challenges in text classification?**

    Ambiguity, sarcasm, data imbalance, domain-specific vocabulary.