NLP Mid Notes

Stages of Language Processing: (PMLSSDP)

1. **Phonology**: Study of sounds in a language.
    - Example: "She sells seashells by the seashore" has alliteration with the "s"
2. **Morphology**: Study of word structure and formation.
    - Example: "Dogs" is the plural form of "dog."
3. **Lexical**: Study of vocabulary.
    - Example: "Erudite" means learned or scholarly.
4. **Syntactic**: Study of sentence structure.
    - Example: "The boy saw the girl with the telescope" can mean the boy used the telescope.
5. **Semantic**: Study of meaning.
    - Example: "John gave Mary the red ball" means John gave a red ball to Mary.
6. **Discourse**: Study of larger language units like conversations.
    - Example: A paragraph describing a morning routine.
7. **Pragmatic**: Study of language in context.
    - Example: "Do you have plans this weekend?" can mean different things depending on who asks.

Types of Ambiguity in NLP: (PWPSR)

1. **Phonological Ambiguity**: Words sound the same but have different meanings.
    - Example: "Write" vs. "right."
2. **Word Sense Ambiguity**: Words with multiple meanings.
    - Example: "Bank" can mean a financial institution or the side of a river.
3. **Part of Speech Ambiguity**: Words that can function as different parts of speech.
    - Example: "Fish" can be a verb or a noun.
4. **Syntactic Ambiguity**: Sentences that can be parsed in multiple ways.

- Example: "I saw the man with the telescope" can mean two things.

5. **Referential Ambiguity**: Pronouns or nouns that can refer to multiple entities.
    - Example: "He" in "John told Tom that he needs to go to the store" could refer to John or Tom.

What are the factors driving the advancement of Natural Language Processing (NLP) technology?

1. Increases in **computing power**
2. Rise of the **social media platforms,** increasing **text data**
3. Advanced in **ML and DL for better NLP model**
4. Advanced in understanding **sarcasm and cultural references**
5. Growth of **Multilingual NLP programs.**

Compare Between Statistical and Deep Learning NLP: (ADICGL)

Feature	Statistical NLP	Deep Learning NLP
Approach	**Mathematical** and **probability** models	Uses **neural networks** and **deep learning** models
Data Requirements	**Small to moderate** datasets	**Large** scale datasets for training
Interpretability	**More** interpretable	Considered a **black box**
Computational Cost	**Low to moderate**	High (requires **GPU/TPUs**)
Generalization	Struggles with **unseen words** and **context**	Generalizes well with **enough training data**
Linguistic Rules	**Predefined** linguistic rules	**Learns representations** without rules

Regular Expression: functions that allows us to **search a string for a match**

Function	Description
findall	Return a list containing **all matches**
search	Returns **match object if its in the string**
split	List where string has **been split at each match**
sub	Replace **one or many matches with a string**

- **Metacharacters: are characters with a special meaning.**
- **A special sequence: is a \ followed by one of the characters in the list**
- **A set is a set of characters inside a pair of square brackets [ ]**

Stemming VS Lemmatization:

Stemming	Lemmatization
Reduces words to **root** form to find **stem** of the word	Reduces words to **dictionary** form to find **lemma** of the word
Used in **IR** and **Text Mining**	Used in **NLP with high level** of text understanding
Use various rules to **strip off affixes and suffixes** to get base form	Uses **context and syntactic** role of word
Stemmers: Porter stemmer, Snowball, Lancaster	**Lemmatizers: WordNet, Stanford, spaCy**

Stemming:

Strengths:

- Fast and computationally efficient
- **Reduces vocabulary size**, improves text matching and retrieval

Weaknesses:

- Produces stems that are not actual words, have no meaning, or not related to original word
- Cannot handle **irregular forms** or words with **multiple meanings**

Lemmatization:

Strengths:

- Produces Lemmas that are **accurate and meaningful bases**
- Considers **context and syntactic role** of word.
- Can handle **irregular forms** and words that have **multiple meanings.**

Weaknesses:

- Slower and more computationally intensive than stemming.
- Requires **a dictionary or knowledge base** to get correct lemma, (not available in all languages)
- Different results for different forms of the same word depending on context.

Different kinds of Automatic POS Taggers in NLTK:

1. **Default** tagger**:** Assigns default POS tagging in all words in a corpus
2. **Regular expression:** uses RE to **match patterns of words** and assign POS tags to them
3. **N-Gram** tagger: **statistical tagger** that assigns POS tags based on **context of surrounding words**. Uses **frequency distribution** to **predict** POS tag for current word.
4. **Brill Tagger:** uses **rule-based approach to correct mistakes** made by **previous tagger**. It learns a set of **transformational rules from training data to improve accuracy of previous tagger.**

Named Entity Recognition (NER): Identify and categorize specific **named entities** in a text**.** It can be **Person (Bill Gates), Organizations (Google), Location(London) etc.**

4 different text representation techniques:

1. **One-Hot Encoding**: Convert **categorical** data into **numerical** data by creating a **binary vector** of size n. (n is the number of categories)
2. **Bag of Words: Counting number of times each word appeared** in a document, and representing the document as a **vector of word frequencies**.
3. **TF-IDF: Evaluates importance of a word in a document**. Multiply term frequency with inverse document frequency, which is log of **total number of documents** divided by **number of documents containing the word**.
4. **Occurrence, Co-occurrence: Relationships between words.**
    - **Occurrence matrix:** how many times **each word occurs** in the text
    - **Co-occurrence:** how many times each **pair of words occur together.**

What is Negative Sampling?

- Word2Vec models train word embeddings by **reducing the number of words processed in each iteration.**
- Randomly selects a small number of "negative" words (words not in the context).
- **Updates embeddings** for both the "positive" word (word in the context) and the "negative" words.
- Helps the model **differentiate between words that appear** in the context and those that don't.
- Avoids **computing probabilities for every word in the vocabulary**, which is computationally efficient

Illustrate 5 different types of relations that Word2Vec can learn: (SACHP)

1. **Synonyms and Antonyms**
2. **Analogies**
3. **Co-occurrence**
4. **Hierarchical**
5. **POS**

5 hyperparameters that can be adjusted during the construction of Word2Vec (LVCMN)

1. **Learning rate**
2. **Vector dimensionality**
3. **Context window size**
4. **Minimum word count**
5. **Number of negative samples**

3 major Differences between RNN and LSTM

Feature	RNN	LSTM
Long sequences	Struggles because of vanishing gradient	Can handle because of memory cell
Retain Information	Lacks memory mechanism	Has memory cell to store and retrieve information
Gates	One gate, can't control information flow	Three gates (input, forget, output)
Long-term memory tasks	performs poorly on tasks like long sentence	Excels at tasks
Architecture	Simple with fewer parameters	Complex with memory cells and gates
Uses	Short sequences	Machine translation and Time-series prediction