Experiment 1

Pranay Singhvi

UID:2021300126

Install NLTK and perform basic Corpus analysis using NLTK such as

(i) frequency distribution (create tabular output, bar graph)

(ii) learn about morphological features of a word by analysing it.

Input – word (user prompt)

Output in tabular format – Root, Category(noun etc), gender, number, tense(present, past)

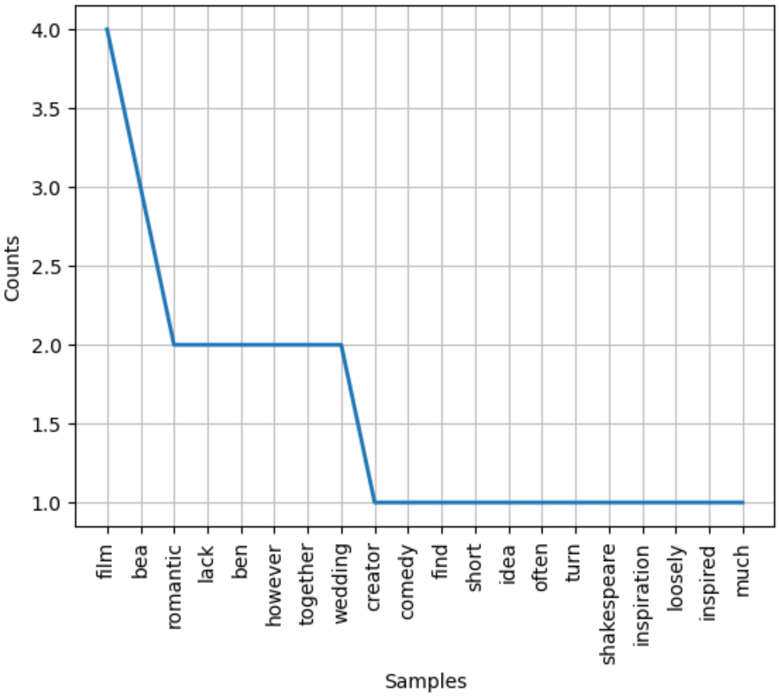
```
from nltk import FreqDist
import nltk
import matplotlib.pyplot as plt
import re
from nltk.corpus import stopwords, wordnet
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
from nltk.tag import pos_tag
from prettytable import PrettyTable
```

```
nltk.download('punkt')
nltk.download('stopwords')
stopwords.words('english')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
True
```

```
with open('/content/drive/MyDrive/NLP/moviereview.txt', 'r', encoding='utf-8') as file:
    text = file.read()
    data = re.sub('[^a-zA-Z]', ' ',text)
    data = data.lower()
    data = data.split()
    data = [word for word in data if not word in set(stopwords.words('english'))]

lemmmatizer=WordNetLemmatizer()
for i in range(len(data)):
    words=nltk.word_tokenize(data[i])
    words = [lemmmatizer.lemmatize(word) for word in words]
    data[i]=' '.join(words)

data = ' '.join(data)

tokens = nltk.word_tokenize(data)

freq_dist = FreqDist(tokens)


print("Word\tFrequency")
print("------------------")

freq_dist.plot(20, cumulative=False)
plt.show()
```

```
Word    Frequency
-----------------
```



Morphological features of a word

```
pos_mapping = {
    "CC": "Coordinating conjunction",
    "CD": "Cardinal number",
    "DT": "Determiner",
    "EX": "Existential there",
    "FW": "Foreign word",
    "IN": "Preposition or subordinating conjunction",
    "JJ": "Adjective",
    "JJR": "Adjective, comparative",
    "JJS": "Adjective, superlative",
    "LS": "List item marker",
    "MD": "Modal",
    "NN": "noun, singular or mass",
    "NNS": "noun, plural",
    "NNP": "Proper noun, singular",
    "NNPS": "Proper noun, plural",
    "PDT": "Predeterminer",
    "POS": "Possessive ending",
    "PRP": "Personal pronoun",
    "PRP$": "Possessive pronoun",
    "RB": "Adverb",
    "RBR": "Adverb, comparative",
    "RBS": "Adverb, superlative",
    "RP": "Particle",
    "SYM": "Symbol",
    "TO": "to",
    "UH": "Interjection",
    "VB": "Verb, base form",
    "VBD": "Verb, past tense",
    "VBG": "Verb, gerund or present participle",
    "VBN": "Verb, past participle",
    "VBP": "Verb, non3rd person singular present",
    "VBZ": "Verb, 3rd person singular present",
    "WDT": "Whdeterminer",
    "WP": "Whpronoun",
    "WP$": "Possessive whpronoun",
    "WRB": "Whadverb",
}


def analyze_sentence(sentence):
    words = word_tokenize(sentence)
    tags = pos_tag(words)
    maleWords = [
        "he",
        "him",
        "his",
        "himself",
        "boy",
        "sir",
        "man",
        "gentleman",
        "father",
        "son",
        "brother",
        "uncle",
        "nephew",
        "grandfather",
        "grandson",
        "king",
        "prince",
        "husband",
        "groom",
    ]
    femaleWords = [
        "she",
        "her",
        "hers",
        "hersef",
        "girl",
        "madam",
        "lady",
        "woman",
        "mother",
        "daughter",
        "sister",
        "aunt",
        "niece",
        "grandmother",
        "granddaughter",
        "queen",
        "princess",
        "wife",
        "bride",
        "widow",
    ]
    lemmatizer = WordNetLemmatizer()
    morphological_table = PrettyTable()
    morphological_table.field_names = ["Root", "Category", "Gender", "Number", "Tense"]
    print(f"\nSentence: {sentence}")
    for i in range(len(words)):
        root = lemmatizer.lemmatize(words[i])
        category = pos_mapping[tags[i][1]]
        if words[i].lower() in maleWords:
            gender = "male"
        elif words[i].lower() in femaleWords:
            gender = "female"
        elif "Proper noun" in category and (
            words[i].endswith("i")
            or words[i].endswith("a")
            or words[i].endswith("e")
            or words[i].endswith("y")
        ):
            gender = "female"
        elif "Proper noun" in category:
            gender = "male"
        elif "pronoun" in category:
            gender = "Can't Assume"
        else:
            gender = "neutral"
        number = sentence.count(words[i])

        # check if word is a verb
        if tags[i][1].startswith("V"):
            # determine tense of the verb
            if (
                category == "Verb, past tense" or
                category == "Verb, past participle"
            ):
                tense = "Past"
            elif (
                category == "Verb, gerund or present participle"
                or category == "Verb, non3rd person singular present"
                or category == "Verb, 3rd person singular present"
            ):
                tense = "Present"
            else:
                tense = "Future"
        else:
            tense = "NA"

        # print(f"Root: {root}, Category: {category}, Gender: {gender}, Number: {number}, Tense: {tense}")
        morphological_table.add_row([root, category, gender, number, tense])
    print(morphological_table)


analyze_sentence(input())
```

```
I am Vijay

Sentence: I am Vijay
+-------+--------------------------------------+--------------+--------+---------+
|  Root |               Category               |    Gender    | Number |  Tense  |
+-------+--------------------------------------+--------------+--------+---------+
|   I   |           Personal pronoun           | Can't Assume |   1    |    NA   |
|   am  | Verb, non3rd person singular present |   neutral    |   1    | Present |
| Vijay |              Adjective               |   neutral    |   1    |    NA   |
+-------+--------------------------------------+--------------+--------+---------+
```

Curiosity Questions

1. What is Natural language processing? Discuss various levels of analysis under it with example.

Natural Language Processing (NLP) is a field of artificial intelligence that focuses on the interaction between computers and human language. It involves the development of algorithms and models to enable machines to understand, interpret, and generate human-like text. NLP encompasses various levels of linguistic analysis to make sense of language data.

1. Phonetic and Phonological Analysis: At the basic level, NLP can analyze speech sounds and patterns. For example, speech recognition systems use phonetic analysis to convert spoken words into text.
2. Morphological Analysis: This level involves the study of word structure and formation. NLP systems can break down words into morphemes, the smallest units of meaning. For instance, the analysis of the word "unhappiness" would identify "un-" as a prefix and "-ness" as a suffix.
3. Syntactic Analysis: This level deals with the grammar and structure of sentences. Syntax analysis helps in understanding the relationships between words in a sentence. For instance, in the sentence "The cat chased the mouse," syntactic analysis determines that the cat is the one doing the chasing.
4. Semantic Analysis: NLP aims to understand the meaning of words and how they combine to form meaningful sentences. Semantic analysis helps in extracting the intended meaning from a text.
5. Pragmatic Analysis: This level considers the context and the implied meaning of language. It involves understanding the speaker's intentions and the social context. For example, interpreting sarcasm or identifying speech acts like requests or commands.

2. What do you mean by ambiguity in NLP? Explain with suitable example. Discuss various ways to resolve ambiguity in NLP

Ambiguity in Natural Language Processing (NLP) refers to situations where a given piece of language can have multiple interpretations or meanings, making it challenging for machines to accurately understand and process the intended message. Ambiguity can arise at various linguistic levels, including lexical, syntactic, semantic, and pragmatic.

Example of Ambiguity: Consider the sentence, "I saw her with the telescope." Here, the word "saw" is ambiguous. It could mean physically seeing someone using a telescope or merely possessing a telescope. This ambiguity can lead to different interpretations, posing a challenge for NLP systems to discern the intended meaning without additional context.

Ways to Resolve Ambiguity in NLP:

Contextual Analysis: Incorporating contextual information from surrounding words and sentences helps disambiguate meanings. Understanding the broader context aids in selecting the most appropriate interpretation.

Statistical Methods: Leveraging statistical models, such as machine learning algorithms, can help predict the most likely interpretation based on patterns observed in large datasets.

Semantic Role Labeling: Identifying the roles that different words play in a sentence helps in disambiguation. For instance, determining whether a word is acting as a subject or an object can clarify its meaning.

Syntactic and Semantic Constraints: Applying grammatical and semantic rules helps in eliminating unlikely interpretations. For instance, considering the syntactic structure of a sentence can restrict possible meanings.

Pragmatic Considerations: Taking into account pragmatic factors, such as the speaker's intention and the social context, helps resolve ambiguity. Understanding implied meanings aids in selecting the correct interpretation.

3.What is morphology with examples?

Morphology is a subfield of linguistics that studies the internal structure of words and the rules governing their formation. It explores how words are built from smaller units called morphemes, which are the smallest units of meaning in a language. Morphology encompasses the study of prefixes, suffixes, roots, and other morphological elements that contribute to the structure and meaning of words.

There are two main types of morphemes: free morphemes, which can stand alone as words (e.g., "cat," "run"), and bound morphemes, which must attach to a free morpheme to convey meaning (e.g., the "-s" in "cats" or the "-ed" in "walked").

Examples of morphological processes include:

Inflection: This involves adding morphemes to a word to indicate grammatical information such as tense, number, or gender. For instance, adding "-s" to "cat" results in "cats," indicating plurality.

Derivation: This process creates new words by adding affixes (prefixes or suffixes) to existing words. For example, adding "-er" to "teach" creates "teacher."

Compounding: It involves combining two or more words to create a new one. An example is "blackboard," formed by combining "black" and "board."

Understanding morphology is crucial for comprehending the structure and formation of words in a language, providing insights into how words convey meaning through their internal components.

4. Discuss discourse and pragmatic analysis. Discuss reference resolution problem in detail.

Discourse Analysis: Discourse analysis is the examination of language beyond the level of individual sentences, focusing on the structure and organization of larger units such as conversations, paragraphs, or entire texts. It investigates how language elements contribute to the overall meaning in context, considering coherence, cohesion, and rhetorical strategies. Discourse analysis is essential for understanding how linguistic elements connect and contribute to the flow of communication.

Pragmatic Analysis: Pragmatic analysis involves studying language use in context, considering the speaker's intentions, implied meanings, and the social or cultural aspects of communication. Pragmatics helps interpret language beyond its literal meaning, dealing with aspects like politeness, indirect speech acts, and conversational implicatures. Understanding the pragmatic dimension is crucial for effective communication and accurate interpretation of language in various social contexts.

Reference Resolution Problem: Reference resolution is a challenge in natural language processing where the goal is to determine the referent of a pronoun or noun phrase in a given context. Ambiguities arise when multiple entities are possible referents for a pronoun, and resolving this ambiguity is crucial for understanding the meaning of a text. For instance, in the sentence "She gave the book to her mother," determining whether "her" refers to the subject of the sentence or another person is a reference resolution problem. Resolving references accurately is essential for building coherent and contextually aware NLP applications, such as chatbots, machine translation, and text summarization. Various techniques, including syntactic and semantic analysis, are employed to address this challenge in NLP systems.

Conclusion

In this experiment we learnt about the basic corpus analysis using NLTK and also learnt about the morphological features of a word.