Experiment 2

Name: Pranay Singhvi

UID: 2021300126

NLP Batch A

Aim:

1. Generate word forms from root and suffix information using Add-Delete table.
2. Comparative study of Porter/Snowball/Lancaster Stemmer and Stemmer vs Lemmatizer

1. Generate word forms from root and suffix information using Add-Delete table.

```
!pip install -U word_forms
```

```
Collecting word_forms
  Downloading word_forms-2.1.0-py3-none-any.whl (166 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 166.3/166.3 kB 3.5 MB/s eta 0:00:00
Collecting inflect==4.1.0 (from word_forms)
  Downloading inflect-4.1.0-py3-none-any.whl (31 kB)
Requirement already satisfied: nltk>=3.3 in /usr/local/lib/python3.10/dist-packages (from word_forms) (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk>=3.3->word_forms) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk>=3.3->word_forms) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk>=3.3->word_forms) (
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk>=3.3->word_forms) (4.66.1)
Installing collected packages: inflect, word_forms
  Attempting uninstall: inflect
    Found existing installation: inflect 7.0.0
    Uninstalling inflect-7.0.0:
      Successfully uninstalled inflect-7.0.0
Successfully installed inflect-4.1.0 word_forms-2.1.0
```

```
import nltk
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from nltk.stem import WordNetLemmatizer, PorterStemmer, SnowballStemmer, LancasterStemmer
from nltk.corpus import wordnet as wn
import word_forms as wd
```

```
from word_forms.word_forms import get_word_forms
from prettytable import PrettyTable
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
```

```
words = ["read", "talk", "laugh", "play", "swim", "cook", "draw", "work", "drive", "listen"]
```

```
n = []
for i in range(len(words)):
  n.append(list(get_word_forms(words[i])['v']))
print(n)
```

```
[['read', 'reads', 'reading'], ['talking', 'talked', 'talks', 'talk'], ['laughed', 'laugh', 'laughs', 'laughing'], ['pla
```

```
def count_equal_characters(word1, word2):
  count = 0
  for char1, char2 in zip(word1, word2):
    if char1==char2:
      count+=1
    else:
      break
  return [len(word1) - count, len(word2) - count, count]
```

```
def generate_add_delete_table(word, forms):
  table = PrettyTable()
  table.field_names = ["Root","Delete", "Add", "Chars" ,"Form"]
  for form in forms:
    l = count_equal_characters(word,form)
    table.add_row([word,str(l[0]),str(l[1]),form[l[2]:], form])
  return table


for i in range(len(words)):
  print(generate_add_delete_table(words[i], n[i]))
```

```
+------+--------+-----+-------+---------+
| Root | Delete | Add | Chars |   Form  |
+------+--------+-----+-------+---------+
| read |   0    |  0  |       |   read  |
| read |   0    |  1  |   s   |  reads  |
| read |   0    |  3  |  ing  | reading |
+------+--------+-----+-------+---------+
+------+--------+-----+-------+---------+
| Root | Delete | Add | Chars |   Form  |
+------+--------+-----+-------+---------+
| talk |   0    |  3  |  ing  | talking |
| talk |   0    |  2  |   ed  |  talked |
| talk |   0    |  1  |   s   |  talks  |
| talk |   0    |  0  |       |   talk  |
+------+--------+-----+-------+---------+
+-------+--------+-----+-------+----------+
|  Root | Delete | Add | Chars |   Form   |
+-------+--------+-----+-------+----------+
| laugh |   0    |  2  |   ed  | laughed  |
| laugh |   0    |  0  |       |  laugh   |
| laugh |   0    |  1  |   s   |  laughs  |
| laugh |   0    |  3  |  ing  | laughing |
+-------+--------+-----+-------+----------+
+------+--------+-----+-------+---------+
| Root | Delete | Add | Chars |   Form  |
+------+--------+-----+-------+---------+
| play |   0    |  2  |   ed  |  played |
| play |   0    |  0  |       |   play  |
| play |   0    |  1  |   s   |  plays  |
| play |   0    |  3  |  ing  | playing |
+------+--------+-----+-------+---------+
+------+--------+-----+-------+----------+
| Root | Delete | Add | Chars |   Form   |
+------+--------+-----+-------+----------+
| swim |   0    |  1  |   s   |  swims   |
| swim |   2    |  2  |   um  |   swum   |
| swim |   0    |  4  |  ming | swimming |
| swim |   2    |  2  |   am  |   swam   |
| swim |   0    |  0  |       |   swim   |
+------+--------+-----+-------+----------+
+------+--------+-----+-------+---------+
| Root | Delete | Add | Chars |   Form  |
+------+--------+-----+-------+---------+
| cook |   0    |  1  |   s   |  cooks  |
| cook |   0    |  3  |  ing  | cooking |
| cook |   0    |  2  |   ed  |  cooked |
| cook |   0    |  0  |       |   cook  |
+------+--------+-----+-------+---------+
+------+--------+-----+-------+---------+
| Root | Delete | Add | Chars |   Form  |
+------+--------+-----+-------+---------+
| draw |   2    |  2  |   ew  |   drew  |
| draw |   0    |  0  |       |   draw  |
| draw |   0    |  3  |  ing  | drawing |
| draw |   0    |  1  |   s   |  draws  |
| draw |   0    |  1  |   n   |  drawn  |
+------+--------+-----+-------+---------+
+------+--------+-----+-------+---------+
```

2.Comparative study of Porter/Snowball/Lancaster Stemmer and Stemmer vs Lemmatizer

```
from nltk.stem import SnowballStemmer, PorterStemmer, LancasterStemmer, WordNetLemmatizer
import re
from nltk.corpus import stopwords


nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
with open('/content/test.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'))]


stemmers = SnowballStemmer(language='english')
porter_stemmer = PorterStemmer()
stemmer = LancasterStemmer()
lemmatizer = WordNetLemmatizer()

stemmed_words = [stemmer.stem(word) for word in words]


table = PrettyTable()
table.field_names = ["Word","Snowball Stemmer","Porter Stemmer", "Lancaster Stemmer", "Lemmatizer"]
for word in data:
  table.add_row([word,stemmers.stem(word),porter_stemmer.stem(word),stemmer.stem(word),lemmatizer.lemmatize(word)])

print(table)
```

```
+-------------+------------------+----------------+--------------------+-------------+
|     Word    | Snowball Stemmer | Porter Stemmer | Lancaster Stemmer |  Lemmatizer |
+-------------+------------------+----------------+--------------------+-------------+
|     cat     |       cat        |      cat       |        cat         |     cat     |
|    chased   |      chase       |     chase      |        chas        |    chased   |
|    mouse    |       mous       |      mous      |        mous        |    mouse    |
|    garden   |      garden      |     garden     |        gard        |    garden   |
|  completed  |     complet      |    complet     |       complet      |  completed  |
|    puzzle   |      puzzl       |     puzzl      |        puzzl       |    puzzle   |
|    record   |      record      |     record     |       record       |    record   |
|     time    |       time       |      time      |        tim         |     time    |
|     sun     |       sun        |      sun       |        sun         |     sun     |
|     set     |       set        |      set       |        set         |     set     |
|    behind   |      behind      |     behind     |       behind       |    behind   |
|  mountains  |     mountain     |    mountain    |      mountain      |   mountain  |
|   casting   |       cast       |      cast      |        cast        |   casting   |
|     warm    |       warm       |      warm      |        warm        |     warm    |
|     glow    |       glow       |      glow      |        glow        |     glow    |
|    across   |      across      |     across     |       across       |    across   |
|    valley   |      valley      |     valley     |       valley       |    valley   |
|     long    |       long       |      long      |        long        |     long    |
|     day     |       day        |      day       |        day         |     day     |
|     work    |       work       |      work      |        work        |     work    |
|   decided   |      decid       |     decid      |       decid        |   decided   |
|    relax    |      relax       |     relax      |       relax        |    relax    |
|    watch    |      watch       |     watch      |       watch        |    watch    |
|    movie    |       movi       |      movi      |        movy        |    movie    |
|  scientist  |    scientist     |   scientist    |        sci         |  scientist  |
|  conducted  |     conduct      |    conduct     |       conduc       |  conducted  |
|    series   |       seri       |      seri      |        sery        |    series   |
| experiments |      experi      |     experi     |       expery       |  experiment |
|     test    |       test       |      test      |        test        |     test    |
|  hypothesis |    hypothesi     |   hypothesi    |      hypothes      |  hypothesis |
|   children  |     children     |    children    |       childr       |    child    |
|   giggled   |      giggl       |     giggl      |       giggl        |   giggled   |
|    played   |       play       |      play      |        play        |    played   |
|     park    |       park       |      park      |        park        |     park    |
|   despite   |      despit      |     despit     |       despit       |   despite   |
|     rain    |       rain       |      rain      |        rain        |     rain    |
|   athletes  |      athlet      |     athlet     |       athlet       |   athlete   |
|  continued  |     continu      |    continu     |      continu       |  continued  |
|   training  |      train       |     train      |       train        |   training  |
|    author   |      author      |     author     |        auth        |    author   |
|  carefully  |       care       |      care      |        car         |  carefully  |
|   crafted   |      craft       |     craft      |       craft        |   crafted   |
|   sentence  |     sentenc      |    sentenc     |        sent        |   sentence  |
|    create   |      creat       |     creat      |        cre         |    create   |
| captivating |      captiv      |     captiv     |        capt        | captivating |
|    story    |      stori       |     stori      |       story        |    story    |
|    music    |      music       |     music      |        mus         |    music    |
|    echoed   |       echo       |      echo      |        echo        |    echoed   |
|     hall    |       hall       |      hall      |        hal         |     hall    |
|   filling   |       fill       |      fill      |        fil         |   filling   |
|     room    |       room       |      room      |        room        |     room    |
|   melodic   |      melod       |     melod      |       melod        |   melodic   |
|     tune    |       tune       |      tune      |        tun         |     tune    |
|   gathered  |      gather      |     gather     |        gath        |   gathered  |
|    around   |      around      |     around     |       around       |    around   |
```

Curiosity Questions

1.What is paradigm class? Give example

In linguistic terms, a "paradigm class" denotes a grouping of words or forms that share a common grammatical feature, such as tense, number, gender, or case. These classes serve to categorize words that undergo similar grammatical modifications.

Let's illustrate a paradigm class in English by focusing on verb conjugation in the present tense, using the regular verbs "talk," "walk," and "eat." These verbs alter their forms in the present tense based on the subject (e.g., I talk, you talk, he/she/it talks). The paradigm class for regular verbs in the present tense encompasses these various forms:

1. I talk
2. You talk
3. He/She/It talks
4. We talk
5. You talk
6. They talk

In this instance, the paradigm class encompasses the distinct forms of the verb "talk" resulting from the application of the grammatical rule for present tense conjugation.

It's important to recognize that irregular verbs may exhibit unique forms deviating from a regular pattern, making them less suitable for classification into a neat paradigm class. Nevertheless, for regular verbs and numerous other grammatical features, paradigm classes assist linguists and language learners in comprehending the systematic changes that words undergo based on their grammatical context.

2. What are the different types of morphemes. Give example of each.

---> Morphemes, the smallest units of meaning in a language, are classified into two main categories: free morphemes and bound morphemes. The latter can be further divided into prefixes and suffixes, each contributing to the nuanced structure of words.

1. **Free Morphemes:** Free morphemes are independent units that can stand alone as complete words. *Example:* In the word "cat," both "cat" and "free" are free morphemes, capable of conveying meaningful content on their own.
2. **Bound Morphemes:** Bound morphemes, in contrast, lack standalone meaning and must attach to a free morpheme to convey significance.

   a. **Prefixes:** Prefixes are bound morphemes affixed to the beginning of a word. *Example:* In "unhappy," the prefix "un-" denotes negation.

   b. **Suffixes:** Suffixes, attached to the end of a word, are bound morphemes. *Example:* In "happily," the suffix "-ly" indicates manner or quality.

   c. **Infixes:** Some languages employ infixes, which are bound morphemes inserted inside a word. *Example:* Tagalog uses infixes for verb conjugation, as seen in "linuto" (cooked) derived from the root "luto" (cook) with the infix "-in-".

   d. **Roots and Stems:** Roots and stems, while often considered as standalone units, become bound morphemes when they require additional morphemes for complete meaning. *Example:* In "unhappiness," "happi-" serves as the root, while both "un-" and "-ness" are bound morphemes.

Comprehending these morphemic distinctions is pivotal for morphological analysis and the exploration of word structure within the realm of linguistics.

Conclusion

In this experiment I learnt about diffenent word forms and diffenrence between different stemmer (like porter, snowball etc.) and lemmatizer. I also made add and delete table for all different type of word form.