NLP PROGRAM-7

A program for lemmatizing words using WordNet

Soundarya G_ 2048057

- Try using a minimum of 10 different words, use the results, and based on that interpretations can be given.
- Mention references such as papers as well for interpretation.

```
# import these modules
import nltk
nltk.download('wordnet')
```

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

```
nltk.download('all')
```

```
[nltk_data] Downloading collection 'all'
[nltk_data]    |
[nltk_data]    | Downloading package abc to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/abc.zip.
[nltk_data]    | Downloading package alpino to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/alpino.zip.
[nltk_data]    | Downloading package biocreative_ppi to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/biocreative_ppi.zip.
[nltk_data]    | Downloading package brown to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/brown.zip.
[nltk_data]    | Downloading package brown_tei to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/brown_tei.zip.
[nltk_data]    | Downloading package cess_cat to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/cess_cat.zip.
[nltk_data]    | Downloading package cess_esp to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/cess_esp.zip.
[nltk_data]    | Downloading package chat80 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/chat80.zip.
[nltk_data]    | Downloading package city_database to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/city_database.zip.
[nltk_data]    | Downloading package cmudict to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/cmudict.zip.
[nltk_data]    | Downloading package comparative_sentences to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/comparative_sentences.zip.
[nltk_data]    | Downloading package comtrans to /root/nltk_data...
[nltk_data]    | Downloading package conll2000 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/conll2000.zip.
[nltk_data]    | Downloading package conll2002 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/conll2002.zip.
[nltk_data]    | Downloading package conll2007 to /root/nltk_data...
[nltk_data]    | Downloading package crubadan to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/crubadan.zip.
[nltk_data]    | Downloading package dependency_treebank to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/dependency_treebank.zip.
[nltk_data]    | Downloading package dolch to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/dolch.zip.
[nltk_data]    | Downloading package europarl_raw to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/europarl_raw.zip.
[nltk_data]    | Downloading package floresta to /root/nltk_data...
```

```
[nltk_data]    |   Unzipping corpora/floresta.zip.
[nltk_data]    | Downloading package framenet_v15 to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/framenet_v15.zip.
[nltk_data]    | Downloading package framenet_v17 to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/framenet_v17.zip.
[nltk_data]    | Downloading package gazetteers to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/gazetteers.zip.
[nltk_data]    | Downloading package genesis to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/genesis.zip.
[nltk_data]    | Downloading package gutenberg to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/gutenberg.zip.
[nltk_data]    | Downloading package ieer to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/ieer.zip.
```

```
!pip install simplemma
```

```
Collecting simplemma
  Downloading simplemma-0.3.0-py3-none-any.whl (44.6 MB)
     |████████████████████████████████| 44.6 MB 8.9 kB/s
Installing collected packages: simplemma
Successfully installed simplemma-0.3.0
```

```
import simplemma
```

Lemmatization

- In contrast to stemming, lemmatization is a lot more powerful.
- It looks beyond word reduction and considers a language's full vocabulary to apply a morphological analysis to words, aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma.
- Lemmatization is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item. Lemmatization is similar to stemming but it brings context to the words. So it links words with similar meanings to one word.
- Applications of lemmatization are:

```
- Used in comprehensive retrieval systems like search engines.
- Used in compact indexing
```

Different Approaches on Lemmatization

1. Wordnet Lemmatizer

Wordnet is a publicly available lexical database of over 200 languages that provides semantic relationships between its words. It is one of the earliest and most commonly used lemmatizer technique.

- It is present in the nltk library in python.
- Wordnet links words into semantic relations. (eg. synonyms)
- It groups synonyms in the form of synsets.

```
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
```

```
list_of_words = ['rocks','corpora','kites', 'babies', 'dogs', 'flying', 'smiling','driving',
for words in list_of_words:
  print(words + "\t : " + lemmatizer.lemmatize(words))
```

```
rocks    : rock
corpora  : corpus
bring    : bring
kites    : kite
babies   : baby
dogs     : dog
flying   : flying
smiling  : smiling
driving  : driving
died     : died
tried    : tried
feet     : foot
```

```
list_of_words_uppercase = ['FEET','CALFS','CHILDREN','WOMEN']
for words in list_of_words_uppercase:
  print(words + " : " + lemmatizer.lemmatize(words))
```

```
FEET : FEET
CALFS : CALFS
CHILDREN : CHILDREN
WOMEN : WOMEN
```

```
list_of_words_lowercase = ['feet','calfs','children','women']
for words in list_of_words_lowercase:
  print(words + " : " + lemmatizer.lemmatize(words))
```

```
feet : foot
calfs : calf
children : child
women : woman
```

```
list_of_words_suffix = ['sitting','seated','feeted','striped']
for words in list_of_words_suffix:
  print(words + " : " + lemmatizer.lemmatize(words))
```

```
sitting : sitting
seated : seated
feeted : feeted
striped : striped
```

- Non-English Languages

```
# TAMIL
list_of_words = ['பாறைகள்','காத்தாடி', 'நாய்கள்', 'புன்னகை', 'இறந்தார்']
for words in list_of_words:
  print(words + "\t : " + lemmatizer.lemmatize(words))
```

```
பாறைகள்கள்    : பாறைகள்கள்
காத்தாடி       : காத்தாடி
நாய்கள்        : நாய்கள்
புன்னகை       : புன்னகை
இறந்தார்       : இறந்தார்
```

```
# GERMAN
list_of_words = ['Hier', 'Sind', 'Vaccines']
for words in list_of_words:
  print(words + " : " + lemmatizer.lemmatize(words))
```

```
Hier : Hier
sind : sind
```

```
Vaccines : Vaccines
```

> **Inference:**

- The lemmas are same as the words since it isn't supporting non-english languages.
- The lemmatization doesn't work properly, if the words are in uppercase.
- If we notice the above words, the plural forms are converted to the singular form.
- The general lemmatization, doesn't trunk the suffix 'ing','ed'.

- Simplemma

```
# Simplemma supports diferent languages
mytokens = ['Hier', 'sein', 'Vaccines']
langdata = simplemma.load_data('de') # German
for token in mytokens:
  print(token + " : " +simplemma.lemmatize(token, langdata))
```

```
Hier : hier
SIND : sein
Vaccines : Vaccines
```

```
# Chaining Languages
langdata = simplemma.load_data('de', 'en')
simplemma.lemmatize('Vaccines', langdata)
```

```
'vaccine'
```

> **Inference:**

- sind means 'are' while sein means 'be'.
- With its multilingual capacity, Simplemma can be configured to tackle several languages of intere

2. Wordnet Lemmatizer with POS tag

```
# a denotes adjective in "pos"
print("better :", lemmatizer.lemmatize("better"))
print("better :", lemmatizer.lemmatize("better", pos ="a"))
```

```
better : better
better : good
```

```
# v denotes verb in "pos"
print("cooking :", lemmatizer.lemmatize("cooking"))
print("cooking :", lemmatizer.lemmatize("cooking", pos ="v"))
```

```
cooking : cooking
cooking : bring
```

```
print("playing :", lemmatizer.lemmatize("playing"))
print("playing :", lemmatizer.lemmatize("playing", pos ="v"))
```

```
playing : playing
playing : play
```

```
print("dogs :", lemmatizer.lemmatize("dogs"))
```

```
print("dogs :", lemmatizer.lemmatize("dogs", pos ="n"))
```

```
dogs : dog
dogs : dog
```

```
print(lemmatizer.lemmatize("the cat is sitting with the bats on the striped mat under many ba
```

```
the cat is sitting with the bats on the striped mat under many badly flying geese
```

```
from nltk.corpus import wordnet

# Define function to lemmatize each word with its POS tag
def pos_tag(nltk_tag):
    if nltk_tag.startswith('J'):
        return wordnet.ADJ
    elif nltk_tag.startswith('V'):
        return wordnet.VERB
    elif nltk_tag.startswith('N'):
        return wordnet.NOUN
    elif nltk_tag.startswith('R'):
        return wordnet.ADV
    else:
        return None
```

```
sentence = 'the cat is sitting with the bats on the striped mat under many badly flying geese

# tokenize the sentence and find the POS tag for each token
pos_tagged = nltk.pos_tag(nltk.word_tokenize(sentence))
print(pos_tagged)
```

```
[('the', 'DT'), ('cat', 'NN'), ('is', 'VBZ'), ('sitting', 'VBG'), ('with', 'IN'), ('the
```

```
# we use our own pos_tagger function to make things simpler to understand.
wordnet_tagged = list(map(lambda x: (x[0], pos_tag(x[1])), pos_tagged))
print(wordnet_tagged)
```

```
[('the', None), ('cat', 'n'), ('is', 'v'), ('sitting', 'v'), ('with', None), ('the', Nor
```

> **Inference:**

- The general sentence lemmatization without POS tagging doesn't trunk the word properly.
- This is the sentence given for lemmatization "the cat is sitting with the bats on the striped mat

References:

> https://www.geeksforgeeks.org/python-lemmatization-approaches-with-examples/

> https://www.machinelearningplus.com/nlp/lemmatization-examples-python/

> https://subscription.packtpub.com/book/application_development/9781782167853/2/ch02lvl1sec20/lemmatizing-words-with-wordnet