AIM: NLP processing of any one Indian regional language.

THEORY:

POS Tagging:

POS Tagging (Parts of Speech Tagging) is a process to mark up the words in text format for a particular part of a speech based on its definition and context. It is responsible for text reading in a language and assigning some specific token (Parts of Speech) to each word. It is also called grammatical tagging.

Example:
Input: Everything to permit us.
Output: [('Everything', NN),('to', TO), ('permit', VB), ('us', PRP)]

Steps Involved in the POS tagging example:

- Tokenize text (word_tokenize)
- apply pos_tag to above step that is nltk.pos_tag(tokenize_text)

NLTK POS Tags Examples are as below:

Abbreviation	Meaning
CC	coordinating conjunction
CD	cardinal digit
DT	determiner
EX	existential there
FW	foreign word
IN	preposition/subordinating conjunction
JJ	This NLTK POS Tag is an adjective (large)
JJR	adjective, comparative (larger)
JJS	adjective, superlative (largest)
LS	list market
MD	modal (could, will)
NN	noun, singular (cat, tree)
NNS	noun plural (desks)
NNP	proper noun, singular (sarah)
NNPS	proper noun, plural (indians or americans)
PDT	predeterminer (all, both, half)
POS	possessive ending (parent\ 's)
PRP	personal pronoun (hers, herself, him, himself)
PRP$	possessive pronoun (her, his, mine, my, our)
RB	adverb (occasionally, swiftly)

Abbreviation	Meaning
RBR	adverb, comparative (greater)
RBS	adverb, superlative (biggest)
RP	particle (about)
TO	infinite marker (to)
UH	interjection (goodbye)
VB	verb (ask)
VBG	verb gerund (judging)
VBD	verb past tense (pleaded)
VBN	verb past participle (reunified)
VBP	verb, present tense not 3rd person singular(wrap)
VBZ	verb, present tense with 3rd person singular (bases)
WDT	wh-determiner (that, what)
WP	wh- pronoun (who)
WRB	wh- adverb (how)

The above NLTK POS tag list contains all the NLTK POS Tags. NLTK POS tagger is used to assign grammatical information of each word of the sentence. Installing, Importing and downloading all the packages of POS NLTK is complete.

In Corpus there are two types of POS taggers:

- Rule-Based
- Stochastic POS Taggers

1.Rule-Based POS Tagger: For the words having ambiguous meaning, rule-based approach on the basis of contextual information is applied. It is done so by checking or analyzing the meaning of the preceding or the following word. Information is analyzed from the surrounding of the word or within itself. Therefore words are tagged by the grammatical rules of a particular language such as capitalization and punctuation. e.g., Brill's tagger.

2.Stochastic POS Tagger: Different approaches such as frequency or probability are applied under this method. If a word is mostly tagged with a particular tag in training set then in the test sentence it is given that particular tag. The word tag is dependent not only on its own tag but also on the previous tag. This method is not always accurate. Another way is to calculate the probability of occurrence of a specific tag in a sentence. Thus the final tag is calculated by checking the highest probability of a word with a particular tag.

A) POS TAGGING FOR ENGLISH LANGUAGE:

SOURCE CODE:

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('averaged_perceptron_tagger')
stop_words = set(stopwords.words('english'))

txt = "Sukanya, Rajib and Naba are my good friends. " \
  "Sukanya is getting married next year. " \
  "Marriage is a big step in one’s life." \
  "It is both exciting and frightening. " \
  "But friendship is a sacred bond between people." \
  "It is a special kind of love between us. " \
  "Many of you must have tried searching for a friend "\
  "but never found the right one."

# sent_tokenize is one of instances of
# PunktSentenceTokenizer from the nltk.tokenize.punkt module

tokenized = sent_tokenize(txt)
for i in tokenized:

  # Word tokenizers is used to find the words
  # and punctuation in a string
  wordsList = nltk.word_tokenize(i)

  # removing stop words from wordList
  wordsList = [w for w in wordsList if not w in stop_words]

  # Using a Tagger. Which is part-of-speech
  # tagger or POS-tagger.
  tagged = nltk.pos_tag(wordsList)

  print(tagged)
```

OUTPUT:

```
[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 averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
[('Sukanya', 'NNP'), (',', ','), ('Rajib', 'NNP'), ('Naba', 'NNP'), ('good', 'JJ'), ('friends', 'NNS'), ('.', '.')]
[('Sukanya', 'NNP'), ('getting', 'VBG'), ('married', 'VBN'), ('next', 'JJ'), ('year', 'NN'), ('.', '.')]
[('Marriage', 'NN'), ('big', 'JJ'), ('step', 'NN'), ('one', 'CD'), ('’', 'NN'), ('life.It', 'NN'), ('exciting', 'VBG'), ('frightening', 'NN'), ('.', '.')]
[('But', 'CC'), ('friendship', 'NN'), ('sacred', 'VBD'), ('bond', 'NN'), ('people.It', 'NN'), ('special', 'JJ'), ('kind', 'NN'), ('love', 'VB'), ('us', 'PRP'), (
[('Many', 'JJ'), ('must', 'MD'), ('tried', 'VB'), ('searching', 'VBG'), ('friend', 'NN'), ('never', 'RB'), ('found', 'VBD'), ('right', 'JJ'), ('one', 'CD'), ('.'
```

[('Sukanya', 'NNP'), (',', ','), ('Rajib', 'NNP'), ('Naba', 'NNP'), ('good', 'JJ'), ('friends', 'NNS'), ('.', '.')]
[('Sukanya', 'NNP'), ('getting', 'VBG'), ('married', 'VBN'), ('next', 'JJ'), ('year', 'NN'), ('.', '.')]

[('Marriage', 'NN'), ('big', 'JJ'), ('step', 'NN'), ('one', 'CD'), ('’', 'NN'), ('life.It', 'NN'), ('exciting', 'VBG'), ('frightening', 'NN'), ('.', '.')]
[('But', 'CC'), ('friendship', 'NN'), ('sacred', 'VBD'), ('bond', 'NN'), ('people.It', 'NN'), ('special', 'JJ'), ('kind', 'NN'), ('love', 'VB'), ('us', 'PRP'), ('.', '.')]
[('Many', 'JJ'), ('must', 'MD'), ('tried', 'VB'), ('searching', 'VBG'), ('friend', 'NN'), ('never', 'RB'), ('found', 'VBD'), ('right', 'JJ'), ('one', 'CD'), ('.', '.')]

B) POS TAGGING FOR HINDI LANGUAGE:

SOURCE CODE:

```
from nltk.tag import tnt
from nltk.corpus import indian
import nltk
import pandas as pd
import numpy as np

nltk.download('punkt')
nltk.download('indian')
text = "इराक के विदेश मंत्री ने अमरीका के उस प्रस्ताव का मजाक उड़ाया है , जिसमें अमरीका ने संयुक्त राष्ट्र के प्रतिबंधों को इराकी नागरिकों के लिए कम हानिकारक बनाने के लिए कहा है ।"
def hindi_model():
   train_data = indian.tagged_sents('hindi.pos')
   tnt_pos_tagger = tnt.TnT()
   tnt_pos_tagger.train(train_data)
   return tnt_pos_tagger
tokensHindi = nltk.word_tokenize(text)
print(tokensHindi)
model = hindi_model()
new_tagged = (model.tag(nltk.word_tokenize(text)))
#print(new_tagged)
array=np.array(new_tagged)
for i in array:
   print(i[0]," ==> ", i[1]," ")
```

OUTPUT:

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

['इराक', 'के', 'विदेश', 'मंत्री', 'ने', 'अमरीका', 'के', 'उस', 'प्रस्ताव', 'का', 'मजाक', 'उड़ाया', 'है', ',', 'जिसमें', 'अमरीका', 'ने', 'संयुक्त', 'राष्ट्र', 'के', 'प्रतिबंधों', 'को', 'इराकी', 'नागरिकों', 'के', 'लिए', 'कम', 'हानिकारक', 'बनाने', 'के', 'लिए', 'कहा', 'है', '।']

```
इराक ==> NNP
के ==> PREP
विदेश ==> NNC
मंत्री ==> NN
ने ==> PREP
अमरीका ==> NNP
के ==> PREP
उस ==> PRP
प्रस्ताव ==> NN
का ==> PREP
मजाक ==> NVB
उड़ाया ==> VFM
है ==> VAUX
, ==> PUNC
जिसमें ==> PRP
अमरीका ==> NNP
ने ==> PREP
संयुक्त ==> NNC
राष्ट्र ==> NN
के ==> PREP
प्रतिबंधों ==> NN
को ==> PREP
इराकी ==> JJ
नागरिकों ==> NN
के ==> PREP
लिए ==> PREP
कम ==> INTF
हानिकारक ==> JJ
बनाने ==> VNN
के ==> PREP
लिए ==> PREP
कहा ==> VFM
है ==> VAUX
। ==> PUNC
```

CONCLUSION:

From this practical, I have learned and implemented POS tagging of English and hindi language.