CMPE 409 Machine Translation

Worksheet(Week-06)

1 Download NLTK

Download NLTK package with following instructions

```
>>> import nltk
>>> nltk.download()
```

2 Create a Corpus from Webs

```
print ("collect from web files")

from urllib import request
url="https://www.bbc.com/news/uk-60708450"

response=request.urlopen(url)
raw=response.read().decode("utf8")
print(len(raw))

## See the outputs...

from bs4 import BeautifulSoup
text= BeautifulSoup(raw,"html.parser").get_text()
print(len(text))
print(text)

### Try again

text= BeautifulSoup(raw,"html.parser")
print("Length of the text: ",len(text))

print("printing soup", text)
```

```
counter=0
k=text.find_all("p")
print(type(k))
clean_text=""
for i in k:
    counter=counter+1
    print(counter,i.get_text())
    clean_text = clean_text+ i.get_text()

print("Total",counter)


print("tokenization....")

print(line_tokenize(clean_text))
print("words")

from nltk.tokenize import word_tokenize
token=word_tokenize(text)

print(token)
for i in token:
    print(i)

## Analize outputs
```

3 Default Tagging

```
text="I am a student."
from nltk.tag import DefaultTagger
tagger = DefaultTagger('NN')
s= tagger.tag(['Hello', 'World'])
print (s)
print(tagger.tag(text))
print("Analyze outputs")

print(tagger.tag(text.split()))
print ("Compare with previos outpus")


from nltk.tokenize import word_tokenize
```

```
token_text=word_tokenize(text)
print(type(token_text))
print("with tokenization")
print(tagger.tag(token_text))

## Now change the default Tag

tagger = DefaultTagger('Adj')
print(tagger.tag(token_text))
```

3.1 Evaluating accuracy

```
from nltk.corpus import treebank
test_sents = treebank.tagged_sents()[3000:]
print ("The result is: ")
result=tagger.evaluate(test_sents)

print( result)

## Change default tag and evaluate it again
```

3.2 Untagging

```
print ("untage")
from nltk.tag import untag
print(untag(result)) ##NOTE: we get "result" from the previos section

########### sentences#####

para="We have lecture today. This is a test. We love coding."
from nltk.tokenize import PunktSentenceTokenizer
sentenes=PunktSentenceTokenizer()

sentences=sentenes.tokenize(para)
print(type(sentences))
print(sentences)
mysentence=[ ]
for i in sentences:
    mysentence.append(word_tokenize(i)) ## bu sure you have "word_tokenize"

print (mysentence)
```

```
### Now tag sentences

tagged_sentence= tagger.tag_sents(mysentence)
print("Tagged sentences")
print(tagged_sentence)

print ("Now we untag the tagged sentence")

for i in tagged_sentence:
    print( untag(i))
```

4 Train a Unigram POS tagger

```
print ("Train Unigram")

from nltk.tag import UnigramTagger
from nltk.corpus import treebank
train_sents = treebank.tagged_sents()[:3000]
tagger = UnigramTagger(train_sents)
print("See the out put")
print(treebank.sents()[0])

print("Now we tag the words")

unigrams= tagger.tag(treebank.sents()[0])

print(unigrams)
print("evaluation")
print(tagger.evaluate(train_sents))

### overridng the context model

tagger = UnigramTagger(model={'Pierre': 'NN', 'the': 'DT'})
print(tagger.tag(treebank.sents()[0]))

print(tagger.evaluate(train_sents))
```

4.1 Minimum Frequency cutoff

```
print ("Minimum")
tagger = UnigramTagger(train_sents,cutoff=3)

print("Now we tag the words")

print("evaluation")
print(tagger.evaluate(train_sents))
##Note compre the result with normal tagger.
```

5 Uploading

Show your work to your instructor and upload to learn

6 Resource

This worksheet is prepared from the following books:

- Jacob Perkins, **Python 3 Text Processing with NLTK 3 Cookbook, Packt Publishing**,ISBN: 9781782167853
- Steven Bird, Ewan Klein & Edward Loper, **Natural Language Processing with Python**, O'Reily, June, 2009