Practical 2

Name : Shivam Tawari
Roll no : A-58
Subject : NLP

Aim: Write a python program for getting the meaning of words using stemming.

Theory:

* Stemming

It is a technique used to extract the base form of the words by removing affixes from them is just like cutting down the branches of tree to its stem. Search engine uses stemming for indexing the word.

* Porter Stemmer

NLTK has a porter stemmer class with the help of which we can easily implement it for the word we want to stem. It is based

suffixes in English are made of combinations of smaller & & simpties suffix.

* Lancaster Stemmer

The Lancaster stemmer are more aggressive & dynamic compared to other two. The stemmer is really faster but algo is really confusing when dealing with small words. The lancaster stemmers same the rules externally & basically uses it algorithm.

* Snowball Stemmer

When compared to porter stemmer it can map non-English words. NLTK has snowball stemmer which can easily implement snowball stemmer algorithm in order to use this stemming we need to create an instance with the name of the language we are using the case the stem method. It will be map non English words too.

* Regexp Stemmer Class

NLTK has regexp stemmer class with the help of which we can easily implement regular expression stemmer algorithm.

It basically takes a single regular expression and removes only prefix or suffix that matches the expression.

* Conclusion:

Hence successfully implemented stemming in python and explored various types of stemmers in NLTK library.

Practical – 2

Shivam Tawari (A-58)

```
# Shivam Tawari A-58

from nltk.stem import PorterStemmer
```

```
[4] word_stemmer = PorterStemmer()
    word_stemmer.stem('writing')
```

```
'write'
```

```
[5] word_stemmer.stem('eating')
```

```
'eat'
```

```
[6] import nltk
    nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
[7] from nltk.stem import PorterStemmer
    from nltk.tokenize import sent_tokenize, word_tokenize
    sentence = "Hello Gopal, You have to build a very good YouTube Channel and I love visiting your Channel."
    words = word_tokenize(sentence)
    ps = PorterStemmer()
    for w in words:
      rootWord=ps.stem(w)
      print(rootWord)
```

```
hello
gopal
,
you
have
to
build
a
veri
good
youtub
channel
and
I
love
visit
your
channel
.
```

```
[8]  import nltk
     from nltk.stem.porter import PorterStemmer
     porter_stemmer = PorterStemmer()
     word_data= "Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefix
     nltk_tokens=nltk.word_tokenize(word_data)
     for w in nltk_tokens:
       print("Actual: %s Stem: %s" %(w,porter_stemmer.stem(w)))
```

```
Actual: Stemming Stem: stem
Actual: is Stem: is
Actual: the Stem: the
Actual: process Stem: process
Actual: of Stem: of
Actual: reducing Stem: reduc
Actual: a Stem: a
Actual: word Stem: word
Actual: to Stem: to
Actual: its Stem: it
Actual: word Stem: word
Actual: stem Stem: stem
Actual: that Stem: that
Actual: affixes Stem: affix
Actual: to Stem: to
Actual: suffixes Stem: suffix
Actual: and Stem: and
Actual: prefixes Stem: prefix
Actual: or Stem: or
Actual: to Stem: to
Actual: the Stem: the

Actual: roots Stem: root
Actual: of Stem: of
Actual: words Stem: word
Actual: known Stem: known
Actual: as Stem: as
Actual: lemma Stem: lemma
Actual: . Stem: .
```