Familiar with nltk package
Lab-1

Topics to be covered…..

- Tokenizing text into sentences
- Tokenizing sentences into words
- Tokenizing sentences using regular expressions
- Filtering stop words in a tokenized sentence
- Stemming words
- Lemmatizing words
- Parts of Speech Tagging
- Named Entity Recognition

NLTK Package

- **NLTK** is the Natural Language Toolkit, a comprehensive Python library for natural language processing and text analytics.

- **Tokenization** is a method of breaking up a piece of text into many pieces, and is an essential first step for recipes.

- **WordNet** is a dictionary designed for programmatic access by natural language processing systems.

- **NLTK** includes a **WordNet** corpus reader.

Installing nltk

- pip install nltk

Sentence Tokenizer

```
from nltk.tokenize import sent_tokenize
para = "Hello World. It's good to see you. Thanks for buying this book.“
Print(sent_tokenize(para) )
```

o/p

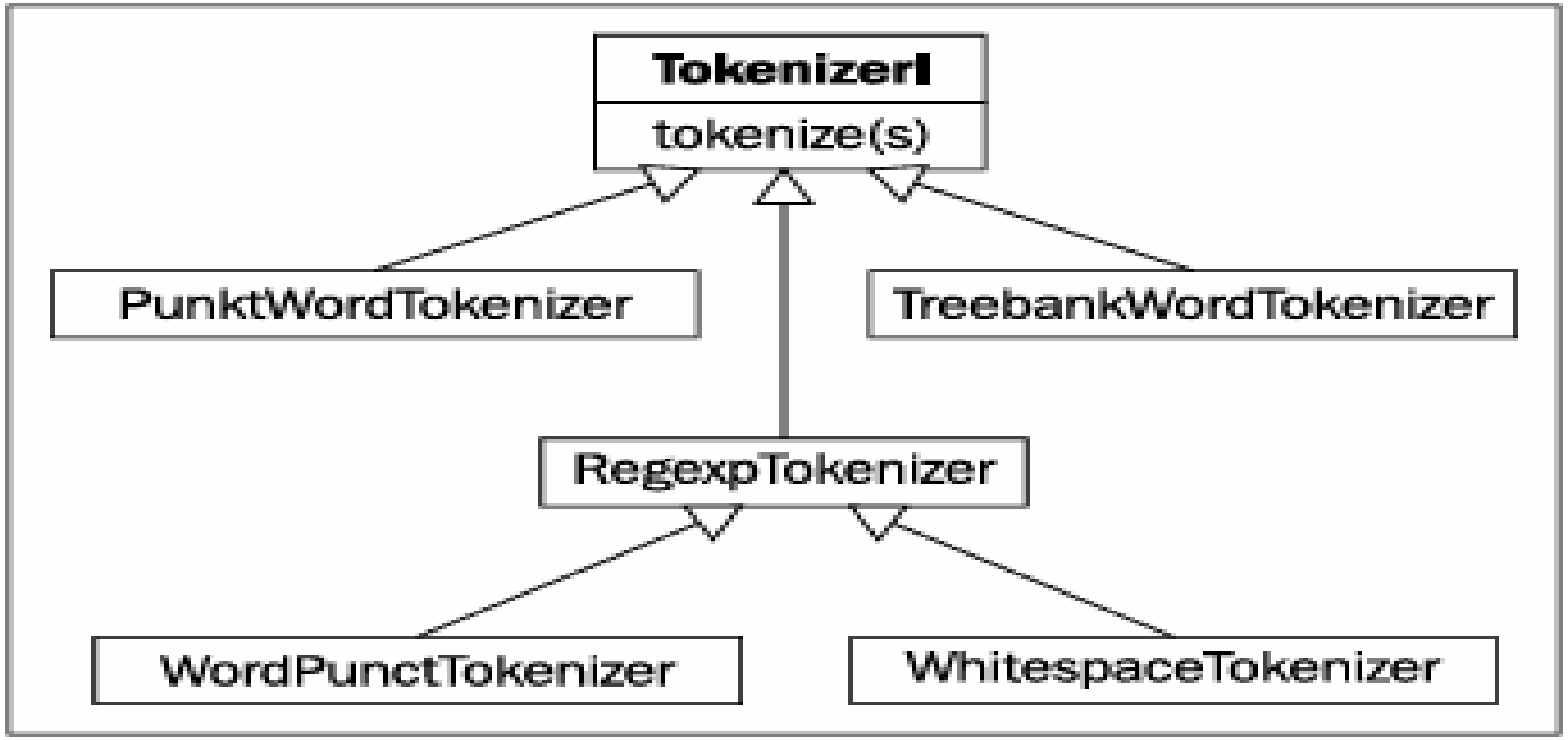
```
['Hello World.', "It's good to see you.", 'Thanks for buying this book.']
```

Word Tokenizer

```
from nltk.tokenize import word_tokenize
para = "Hello World. It's good to see you. Thanks for buying this book.“
Print(word_tokenize(para) )
```

o/p

```
['Hello', 'World', '.', 'It', "'s", 'good', 'to', 'see', 'you', '.', 'Thanks', 'for',
'buying', 'this', 'book', '.']
```


TokenizerI
tokenize(s)
PunktWordTokenizer
TreebankWordTokenizer
RegexpTokenizer
WordPunctTokenizer
WhitespaceTokenizer

Stop words

```
from nltk.corpus import stopwords

print(stopwords.words("english"))
```

Conversion (Upper case to lower case)

```
import re
text = "Natural language processing is an exciting area. Huge budget have been allocated for this."
text = re.sub(r"[^a-zA-Z0-9]", " ", text.lower())
words = text.split()
print(words)
```

Stemming

```
from nltk.stem.porter import PorterStemmer
# Reduce words to their stems
words = "Natural language processing is an exciting area. Huge budget have been allocated for this."

stemmed = [PorterStemmer().stem(w) for w in words]
print(stemmed)
```

Lemmetization

```
from nltk.stem.wordnet import WordNetLemmatizer
# Reduce words to their root form
words = "Natural language processing is an exciting area. Huge budget have been allocated for this."

lemmed = [WordNetLemmatizer().lemmatize(w) for w in words]
print(lemmed)
```

POS Tagging

```
import nltk
from nltk.corpus import stopwords
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
from nltk.tokenize import word_tokenize, sent_tokenize
stop_words = set(stopwords.words('english'))
txt = "Natural language processing is an exciting area. Huge budget have been allocated for this."
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)
```

Thank You