**What is Natural Language Processing (NLP)?**

Natural-language processing (NLP) is an area of computer science and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to fruitfully process large amounts of natural language data.

NLP algorithms are machine learning algorithms based. NLP learns by analysing a set of examples (i.e. a large corpus, like a book, down to a collection of sentences), and making a statistical inference, instead of coding large sets of rules. We can organize the massive chunks of text data and solve a wide range of problems such as — automatic summarization, machine translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation etc.

As we all know, the text is the most unstructured form of all the available data. It is important to cleaning and standardize this text and make it noise free. The idea is to take the raw text and turn into something which can be utilized by an ML algorithm to carry out prediction. We will talk about few important techniques using NLTK.

**import nltk**

**text = open("file\_name.txt").read()**

**text**

**Sentence Segmentation**

We break the articles into sentences. Often, we must do analysis at sentences level. For example, we want to check the number of sentences in an article and number of words in a sentence.

**text = open("file\_name.txt ").read()**

**sents = nltk.sent\_tokenize(text)**

**Tokenization**

Tokenization breaks unstructured data, text, into chunks of information which can be counted as discrete elements. This immediately turns an unstructured string (text document) into a more usable data, which can be further structured, and made more suitable for machine learning. Here we take the first sentence and we get each word as token. Below are two different ways i.e RegexpTokenizer & Word Tokenize.

**from nltk.tokenize import RegexpTokenizer**

**tokenizer = RegexpTokenizer(r'\w+')**

**tokenizer.tokenize(sents[0])**

**or**

**from nltk import word\_tokenize**

**tokens = word\_tokenize(sents[0])**

**StopWords**

Consider words like a, an, the, be etc. These words don’t add any extra information in a sentence. Such words can often create noise while modelling. Such words are known as Stop Words. We filter each sentence by removing the stop words as shown below:

**from nltk.corpus import stopwords**

**stop\_words=set(stopwords.words('english'))**

**Filtered text:**

**filtered\_sentence = []**

**for word in tokens:**

**if word not in stop\_words:**

**filtered\_sentence.append(word)**

**print(tokens) print("====================================================================") print(filtered\_sentence)**

**Stemming And Lemmatization**

Some words represent the same meaning. For example, Copy, copied, copying. The model might treat them differently, so we tend to strip such words to their core. We can do that by stemming or lemmatisation. Stemming and Lemmatization are the basic text processing methods for English text.

**Stemming**

It helps to create groups of words which have similar meanings and works based on a set of rules, such as remove “ing” if words are ending with “ing”. Different types of stemmers in NLTK are PorterStemmer, LancasterStemmer, SnowballStemmer.

**Lemmatization**

It uses a knowledgebase called WordNet. Because of knowledge, lemmatization can even convert words which are different and cant be solved by stemmers, for example converting “came” to “come”.

***#-----PorterStemmer------***

**from nltk.stem.porter import PorterStemmer**

**porter\_stemmer = PorterStemmer()**

**porter\_stemmer.stem("crying")**

**output: ‘cri’**

***#-----LancasterStemmer------***

**from nltk.stem.lancaster import LancasterStemmer**

**lancaster\_stemmer = LancasterStemmer()**

**lancaster\_stemmer.stem("crying")**

**output: ‘cry’**

***#-----SnowballStemmer------***

**from nltk.stem import SnowballStemmer**

**snowball\_stemmer = SnowballStemmer("english")**

**snowball\_stemmer.stem("crying")**

**output: ’cri’**

**from nltk.stem import WordNetLemmatizer**

**wordnet\_lemmatizer = WordNetLemmatizer()**

**wordnet\_lemmatizer.lemmatize("came", pos="v")**

**output: ‘come’**