

**Course:** Master’s in Data Analytics

**Machine Learning CA1**

**Natural Language Processing in Python**

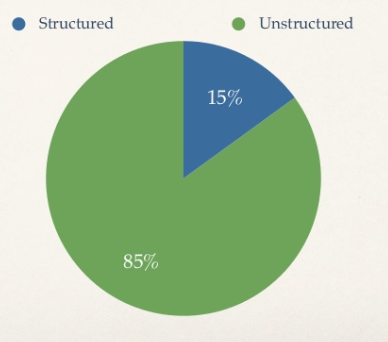
|  |  |
| --- | --- |
| **Student Name(s)** | **Student Number** |
| 1. Sonali Rajesh Shah | 10402114 |
| 1. Neppalli Srityesha | 10509082 |
| 1. Feljose Arockia Shristeca | 10504931 |

# Introduction

**Artificial intelligence (AI) technologies such as machine learning (ML) and deep learning (DL) are dazzling in and of themselves, but believe it or not, leveraged in isolation, they are limited in their potential. These technologies do not interpret data by themselves: they are tied either to deterministic, hard coded software programs created by humans or they are linked to a form of artificial intelligence that can interpret human language into a form ML and DL algorithms can understand. The umbrella term for this gateway AI technology is natural language processing (NLP). It** is a subfield of computer science and artificial intelligence concerned with interactions between computers and human (natural) languages.

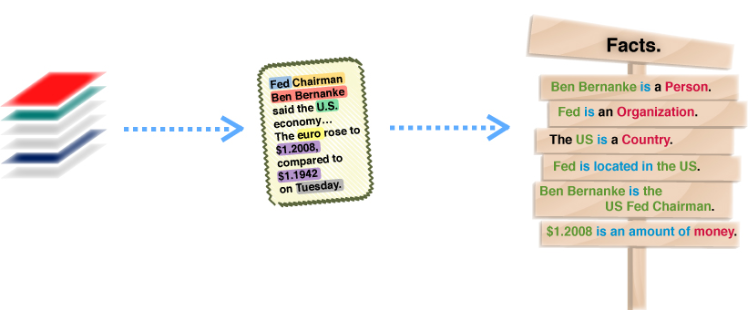
Understanding Text:

Today’s machines can analyze more language-based data than humans, without fatigue and in a consistent, unbiased way. Considering the staggering amount of unstructured data that’s generated every day, from medical records to social media, automation will be critical to fully analyse text and speech data efficiently. This unstructured data needs to be structured for efficient analysis.



### **Text Mining**

Text mining is the automated process of obtaining information from text. Its purpose is to create structured data out of free text content. It is the process of exploring and analysing large amounts of [unstructured text data](https://searchbusinessanalytics.techtarget.com/definition/unstructured-data) aided by software that can identify concepts, patterns, topics, keywords and other attributes in the data. The process can be thought of as slicing and dicing heaps of unstructured, heterogeneous documents into easy-to-manage and interpret data pieces.



## **Natural Language Processing**

Natural language processing is a branch of AI that enables computers to understand, process, and generate language just as people do. It gives machine the ability to read, understand and derive meaning from human languages. It focuses on the semantics of the language rather than just the “text”.

Natural language processing  (or NLP) is a component of text mining that performs a special kind of linguistic analysis that essentially helps a machine “read” text. Natural language processing softwareneeds a consistent knowledge base such as a detailed thesaurus, a lexicon of words, a data set for linguistic and grammatical rules.

Today NLP is booming thanks to the huge improvements in the access to data and the increase in computational power, which are allowing practitioners to achieve meaningful results.

### **Components of NLP:**

There are two components of NLP :-

* Natural Language Understanding
* Natural Language Generation

Natural Language Understanding (NLU):

NLU is a branch of NLP, which helps computers understand and interpret human language by breaking down the elemental pieces of speech. Natural language understanding interprets the meaning that the user communicates and classifies it into proper intents. NLU can digest a text, translate it into computer language and produce an output in a language that humans can understand.

NLU is tasked with communicating with untrained individuals and understanding their intent, meaning that NLU goes beyond understanding words and interprets meaning. NLU is even programmed with the ability to understand meaning despite common human errors like mispronunciations or transposed letters or words.

### **Difficulties in NLU:**

a. Lexical ambiguity: It’s predefined at a very primitive level such as word-level.

b. Syntax Level ambiguity: In this, we can define a sentence in a parsed way in different ways.

c. Referential ambiguity: Referential ambiguity says that we have to refer something using pronouns only.

Natural Language Generation (NLG): Natural language generation (NLG) is the use of artificial intelligence (AI) programming to produce written or spoken narrative from the source taken. It involves generation of text for reports, documents, websites etc. It is done to automate the tedious and monotonous production of text. Data can be assessed, analyzed and communicated with precision, scale and accuracy. NLG software turns structured data into written narrative, writing like a human being but at the speed of thousands of pages per second. It turns structured data into human language but cannot read it.

## **Benefits:**

* Organize massive chunks of textual data
* Perform numerous automated task
* Solve wide range of problems.
* Fast processing of the data is achieved.
* User-friendly

## **Applications:**

* Sentimental Analysis
* Speech Recognition
* Machine Translation
* Spell check
* Keyword search
* Chatbot
* Information extraction
* Advertisement Matching

## **NLP in Python:**

Top NLP libraries

* NLTK
* Gensim
* Polyglot
* TextBlob
* CoreNLP
* PyAudio

# Pre-processing steps in NLP

Before the data is passed through any algorithm it is necessary for the data to be pre-processed or cleaned. Reason the data should be pre-processed is because if it is not than the NLP applications can provide the unexpected results leading to the doubts about the application efficiency. Pre-processing in general means to make the text in the format that is predictable and analysable by the application. Most common pre-processing steps are as follows:

1) Remove Punctuation

2) Tokenization

3) Removal of Stop Words

4) Stemming

5) Lemmatization

6) POS Tagging

**Remove Punctuation:**

Punctuation can provide grammatical context to a sentence which supports our understanding but it does not add any value to machine as they fail to understand them, so as a first step of preprocessing we remove the punctuations.

Example:

Sentence: "How are you? “

Output: How are you

**Tokenization:**

The process by which text is separated into units such as sentences or words is known as tokenization. The sentences after being split are known as tokens. It gives structure to previously unstructured text.

Example:

Sentence: "Have a good day. “

Output: ['Have’,’a’,’good’,’day’,’.’]

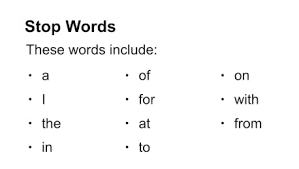
**Stop Words Removal:**

Stop words are commonly used words that are excluded from searches because they don’t tell us much about the data.

Example:

What is motherboard?

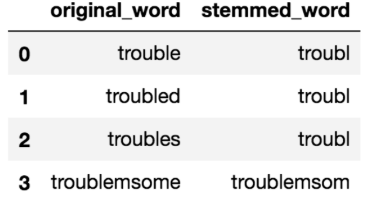
The search engine would only look for the term "motherboard “as it is only relevant rest all words are not that relevant and are stop words. Below is the list of common stop words which are common.



**Stemming:**

Stemming helps reduce a word to its stem form. It often makes sense to treat related words in the same way. It removes suffices, like “ing”, “ly”, “s”, etc. by a simple rule-based approach. It reduces the corpus of words but often the actual words get neglected. There are different algorithms in stemming but the most famous one is Porters Algorithm.

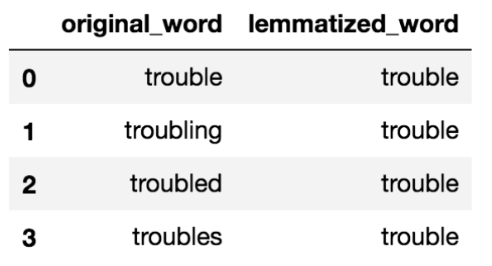
Example of Stemming with Porter Stemmer:



**Lemmatization:**

Lemmatizing derives the canonical form (‘lemma’) of a word. i.e. the root form. It is better than stemming as it uses a dictionary-based approach i.e. a morphological analysis to the root word.

Example:



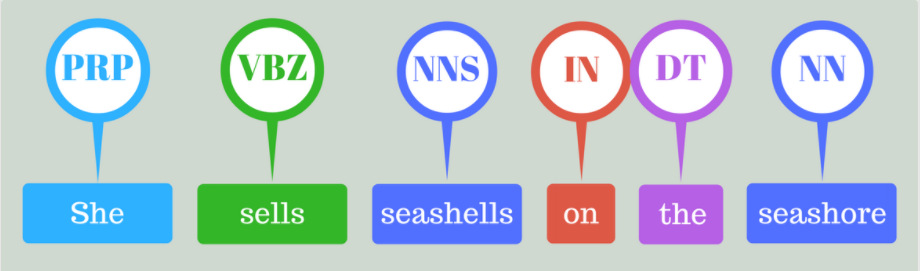
**Difference between Stemming and Lemmatizing:**

Stemming is typically faster as it simply chops off the end of the word, without understanding the context of the word. Lemmatizing is slower and more accurate as it takes an informed analysis with the context of the word in mind.

**POS Tagging:**

POS tagging means associating each word with the part of speech that is noun, verb, adjective, adverb, pronoun, preposition, conjunction etc. It is essential because it helps use create the parse trees which gives us useful information about the named entities and relationship between the words. can also be used to create lemmatizers.

Example:



Where:

* PRP – Personal Noun
* VBZ – verb
* NNS – Noun, plural
* IN – Preposition
* DT – Determiner
* NN – Noun

Vectorization of Data in NLP

Machine Learning Algorithms usually work with the features of the data set to find out the important information from the data. Features are usually numeric attributes on which machine learning algorithms can perform any mathematical operations, but many a times datasets don’t have numeric features in such cases the text values are converted into numeric format which is known as featurization.

Common text vectorization techniques are below:

1)Bag of Words

2)N-Grams

3)TF-IDF

**Bag of Words:**

Bag of words is a way of extracting features from the text for the use of machine learning algorithms. We first tokenize the text, we get the list of the words in the sentences e.g.

“*It was the best of times”*  
*“It was the worst of times”*  
*“It was the age of wisdom”*  
*“It was the age of foolishness”*

We create the list of words present in these sentences which is nothing but the word bag.

Bag of words for the above example:

*‘It’, ‘was’, ‘the’, ‘best’, ‘of’, ‘times’, ‘worst’, ‘age’, ‘wisdom’, ‘foolishness’*

After the Bag of words are identified we then convert the sentences into numeric data of 0’s and 1’s as follows

Let’s take the first sentence as example.

“*It was the best of times”*

We then compare this sentence with the bag of words list and check the frequency of the words.

“it” = 1  
“was” = 1  
“the” = 1  
“best” = 1  
“of” = 1  
“times” = 1  
“worst” = 0  
“age” = 0  
“wisdom” = 0  
“foolishness” = 0

So,  
*“It was the best of times” = [1, 1, 1, 1, 1, 1, 0, 0, 0, 0]*

Rest of the documents will be:

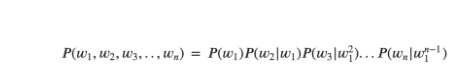
*“It was the worst of times” = [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]*  
*“It was the age of wisdom” = [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]*  
*“It was the age of foolishness” = [1, 1, 1, 0, 1, 0, 0, 1, 0, 1]*

This vector data is then used by the machine learning algorithms to get insights of the data.

**N-Grams:**

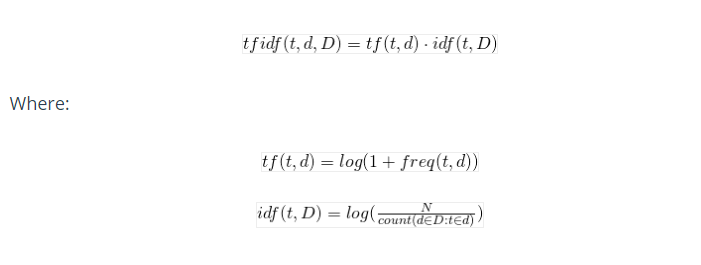
* N-grams are simply all combinations of adjacent words or letters of length n that we can find in our source text. N-grams with n=1 are called unigrams.
* Similarly, bigrams (n=2), trigrams (n=3) and so on can also be used. The basic principle behind n-grams is that they capture the letter or word is likely to follow the given word.
* N-Gram is applied on the body\_text, so the count of each group words in a sentence word is stored in the document matrix.
* It is a language model in which we assign the probability to the sequence of words.

N-Grams can be calculated for a document using following formula:



**TF-IDF:**

* TF-IDF is the technique which helps in text analysis mainly to find out the important words in the text document.
* TF-IDF stands for Term Frequency and Inverse Document Frequency
* TF is calculated by simply calculating the occurrence of word in the document. there are ways to adjust the frequency, by length of a document, or by the raw frequency of the most frequent word in a document.
* IDF means, how common or rare a word is in the entire document set. The closer it is to 0, the more common a word is. This metric can be calculated by taking the total number of documents, dividing it by the number of documents that contain a word, and calculating the logarithm.
* Multiplying these two numbers results in the TF-IDF score of a word in a document. The higher the score, the more relevant that word is in that particular document.
* Mathematical Formula to calculate TF-IDF score is below:



# Applications of NLP with Python

Two applications of NLP along with an explanation in Python has been discussed.

# Speech Recognition

It is the ability of a machine to identify the words in human language and convert them into machine readable format. It generally works using algorithms through acoustic modeling and language modeling.

**Acoustic Modeling**

It is the relationship between the audio signals and the structure of the language ie) grammar, syntax etc.

### **Language Modeling**

It matches the sounds with the words and helps distinguish between words that have similar sounds.

### **Feature Selection**

Its work is to extract those features from the input signal and helps the system identify the speaker.

Any speech recognition program can be evaluated using two factors.

1. Accuracy
2. Speed

## **Programming with Python**

Algorithm for speech recognition in python can be written with the help of 2 libraries. They are,

1. Speech\_recognition
2. PyAudio

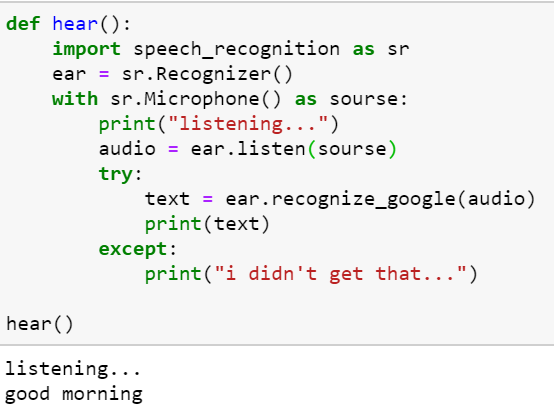
### **Speech Recognition Library**

This library helps in converting spoken words to text. It supports many API’s such as Google Speech, Microsoft Bing, IBM Speech and many more.

### **PyAudio Library**

It is an open source which allows to play and record audio on various platforms such as Windows, Mac, Linux etc. The working of speech\_recognition library totally depends on PyAudio.

With the below code we are converting words spoken into text.



The recognizer gets the input spoken through the microphone which acts as a source. The API recognize\_google helps in converting the words spoken through the microphone into text.

**Application of speech recognition**

1. Its need is evolving rapidly in speech recognition.
2. In telecommunication sectors to serve clients with their queries.
3. In automated identification while collecting confidential details from customers.

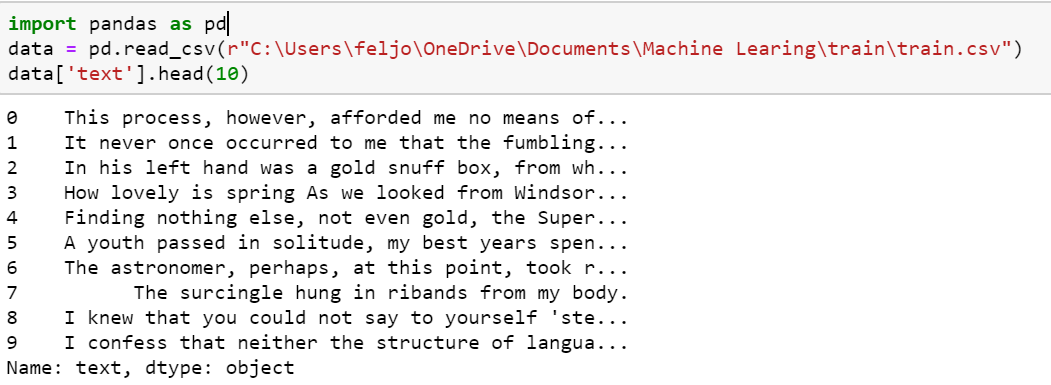
# Text Analysis

The very basic application of NLP is text analysis. For this NLTK library is used. This library is called the mother of all NLP libraries because it gives us a clear idea of the step by step process and used for educational purposes as it helps to understand the background process.

NLTK provides diverse natural language algorithms. It is a well-documented, easy to use and an open source library. It contains more than 50 corpora and algorithms for tokenization, stemming etc.

### **Simple keyword frequency analysis**

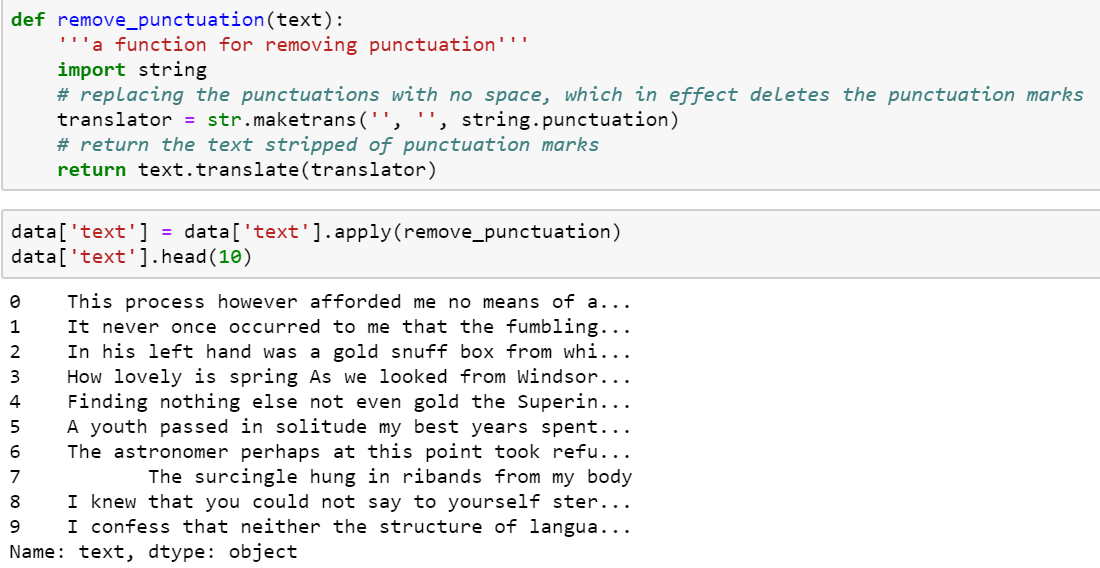
For this text analysis a excel file containing many phrases from 3 different authors has been used.



In every step a function is defined and is called when applying the step to the data.

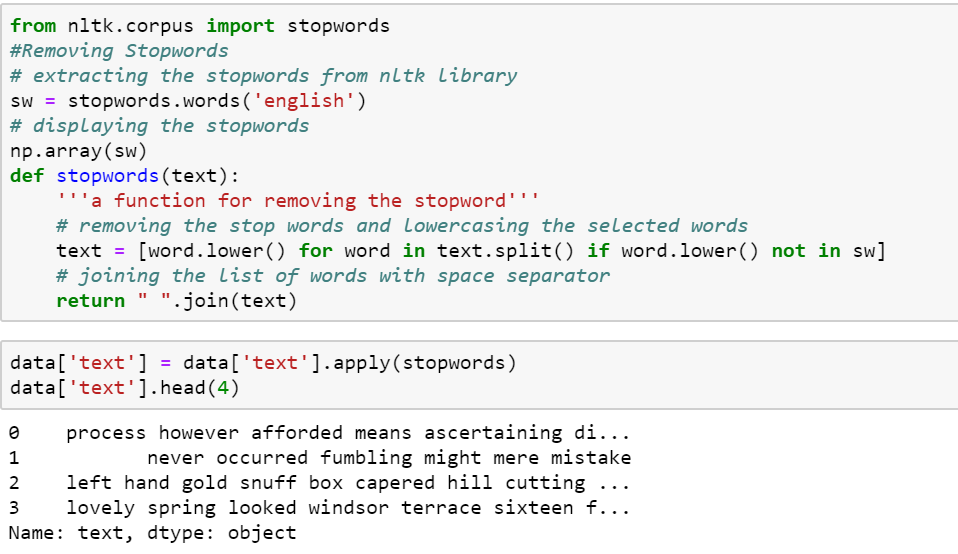
### **Remove Punctuations**

In this step punctuations are removed from the text with the help of maketrans(str1, str2) function. This replaces str1 with str2 and passes it to the translator function. This function replaces the all cases in the string.



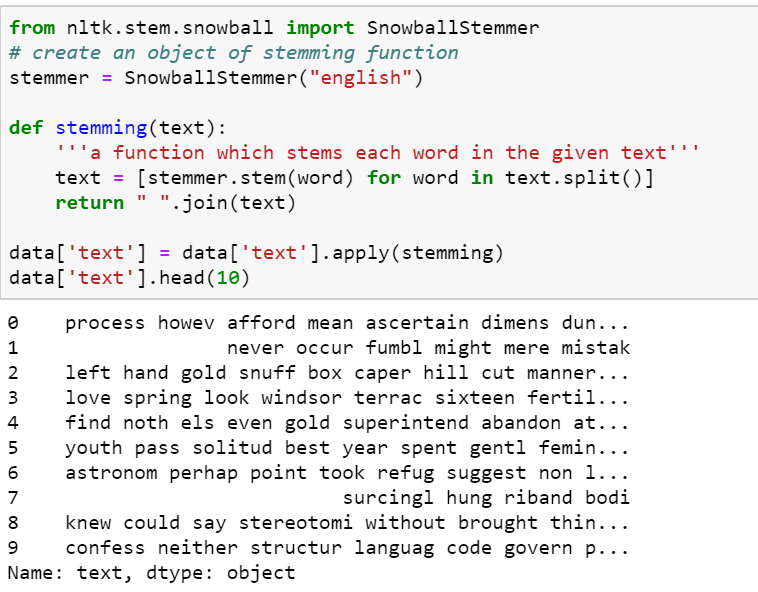
### **Removal of Stopwords**

Stopwords can be imported from the nltk.corpus package. Generally, stopwords are stored in lowercase. Since python is a case sensitive package, all words are converted into lowercase and then the stopwords are removed. Stopwords can be removed for many languages by specifying the language. Here we specify the language ‘engish’.



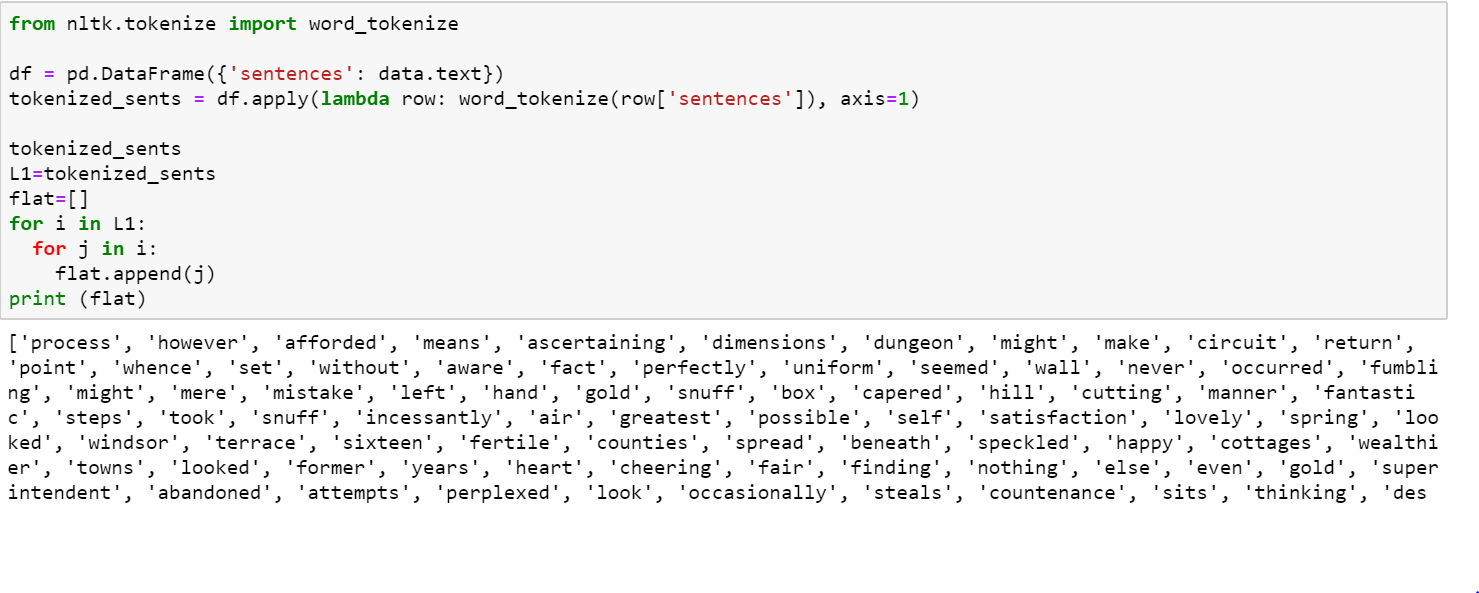
### **Stemming**

SnowballStemmer is used for the stemming process. With the help of this stemmer, stemming can done for nearly 40+ language files.



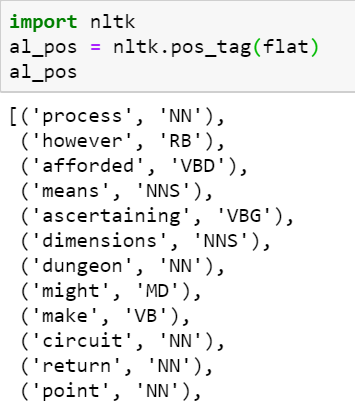
### **Tokenization**

After stemming, tokenization is done to split phrases into words. The word\_tokenize function helps in doing by importing it from nltk.tokenize package.. The lamda function used here is a small anonymous function. It can take any number of argument but can have only one expression.

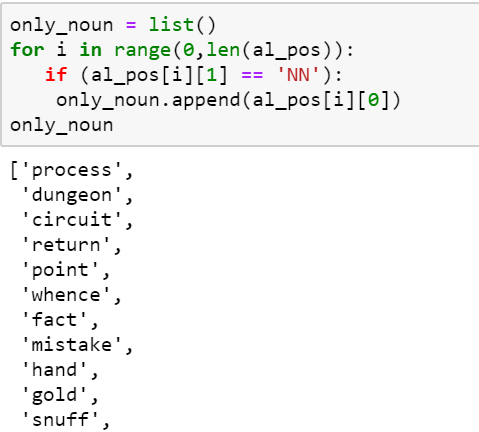


### **Parts of Speech tagging**

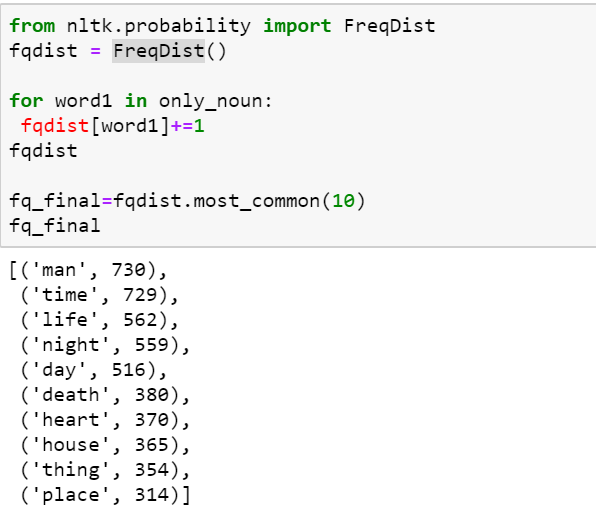
The pos\_tag helps in tagging each tokenized word with its parts of speech.



For this example, we are considering only the NN phrases and it is separated using the below code.



From the words the top 10 frequently used words are been found using the FreqDist() function and then it is plotted using the pyplot in the matplotlib package.



The two columns fetched as output are separated using the below code.

**a= list()**

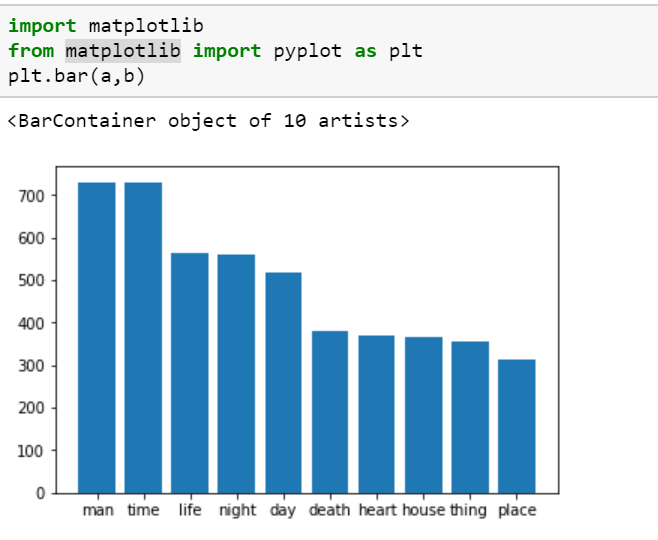
**b= list()**

**for x in fq\_final:**

**a.append(x[0])**

**b.append(x[1])**

Plotting of the top 10 frequent words from the file.



Using text analysis,

1. We can discover the concepts and actions in the document.
2. Get the meaning of the document.
3. Decide a title for the document.

# Conclusion

Thus, in this report we have discussed Natural language processing along with its application and benefits. The pre-processing steps of Nlp are explained in detail and program has been written for two applications of Nlp in Python programming language.