A

Seminar report

on

Natural Language Processing with python

Submitted in partial fulfilment of the requirement for the award of degree of information technology

2020 -2021

Department of information technology

RAJKIYA ENGINEERING COLLEGE, ATARRA, BANDA -210201

(Affiliated to Dr.A.P.J Abdul Kalam Technical University, Lucknow UP)

SUBMITTED TO:

Dr. Siddharth Arjariya

(Assistant Professor)

(Dept. of Information Technology)

SUBMITTED BY:

Akanksha Gautam

Roll no.-1773413003

Semester- 8th

PREFACE

I have made this report file on the topic Natural Language Processing with Python; I have tried my best to elucidate all the relevant details to the topic to be included in the report. While in the beginning I have tried to give a general view about this topic.

My efforts and wholehearted co-corporation of each and every one has ended on a successful note. I express my sincere gratitude to Dr. Siddharth Arjariya who assisted me throughout the preparation of this seminar topic. I thank him for providing me the reinforcement, confidence and most importantly the track for the topic whenever I needed it.

Abstract

Natural language processing has recently gained much attention for representing and analysing human language computationally. It has spread its applications in various fields such as machine translation, email spam detection, information extraction, fake news detection, auto correction, speech recognition and question answering etc. NLP depends on all human languages, there are four phases by discussing the different levels of NLP and components of NLU and NLG followed by history and evolution of NLP.

Now NLP is a branch of artificial intelligence that helps the computer understand, interpret and manipulate the human language. NLP draws from many disciplines, including computer science and computational linguistics, in its pursuit to fill the gap between human communication and computer understanding. Natural language process with python coding language which is natural language toolkit (NLTK) processing libraries. This report could be beneficial to those who want to study and learn about NLP and their tools.

Keywords: NLP, machine learning, artificial language, machine translation, human language, communication, NLTK, pre-processing library.

1. Introduction

There are various kinds of research have explained Natural language processing (NLP) and applications that explore how computers can use to understand and manipulate text or speech of natural language.

What is human language?

Human language is that language that the alphabet combined to make words and combination of words make sentences. There are so many languages like English, Hindi, American, Tamil, German and many more. How to program computers to process and analyse large amounts of natural language data. The result is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. [1][2]

Human language is filled with ambiguities that make it incredibly difficult to write software that accurately determines the intended meaning of text or voice data. Homonyms, homophones, sarcasm, idioms, metaphors, grammar and usage exceptions, variations in sentence structure—these just a few of the irregularities of human language that take humans years to learn, but that programmers must teach natural language-driven applications to recognize and understand accurately from the start, if those applications are going to be useful.

What is Natural Language Processing (NLP)?

Language is a method of communication with the help of which we can speak, read and write. For example, we think, we make decisions, plans and more in natural language; precisely, in words. However, the big question that confronts us in this AI era is whether we can communicate in a similar manner with computers. In other words, can human beings communicate with computers in their natural language? It is a challenge for us to develop NLP applications because computers need structured data, but human speech is unstructured and often ambiguous in nature.[3]

NLP combines computational linguistics—rule-based modeling of human language—with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to ‘understand’ its full meaning, complete with the speaker or writer’s intent and sentiment.

It is all about developing applications and services that are able to understand human languages. Some Practical examples of NLP are speech recognition for eg: google voice search, understanding what the content is about or sentiment analysis etc.[4]

NLP use in Real Life

- Information Retrieval(Google finds relevant and similar results).
- Information Extraction(Gmail structures events from emails).
- Machine Translation(Google Translate translates language from one language to another).
- Text Simplification(Rewordify simplifies the meaning of sentences). Shashi Tharoor tweets could be used(pun intended).
- Sentiment Analysis(Hater News gives us the sentiment of the user).
- Text Summarization(Smmry or Reddit’s autotldr gives a summary of sentences).
- Spam Filter(Gmail filters spam emails separately).
- Auto-Predict(Google Search predicts user search results).
- Auto-Correct(Google Keyboard and Grammarly correct words otherwise spelled wrong).

- Speech Recognition(Google WebSpeech or Vocalware).
- Question Answering(IBM Watson's answers to a query).
- Natural Language Generation(Generation of text from image or video data.)

Evolution of natural language processing

the history of NLP into four phases [4]

First Phase (Machine Translation Phase) - Late 1940s to late 1960s

The work done in this phase focused mainly on machine translation (MT). This phase was a period of enthusiasm and optimism.

Let us now see all that the first phase had in it –

- The research on NLP started in early 1950s after Booth & Richens' investigation and Weaver's memorandum on machine translation in 1949.
- 1954 was the year when a limited experiment on automatic translation from Russian to English demonstrated in the Georgetown-IBM experiment.
- In the same year, the publication of the journal MT (Machine Translation) started.
- The first international conference on Machine Translation (MT) was held in 1952 and second was held in 1956.
- In 1961, the work presented in Teddington International Conference on Machine Translation of Languages and Applied Language analysis was the high point of this phase.

Second Phase (AI Influenced Phase) – Late 1960s to late 1970s

In this phase, the work done was majorly related to world knowledge and on its role in the construction and manipulation of meaning representations. That is why, this phase is also called AI-flavored phase.

The phase had in it, the following –

- In early 1961, the work began on the problems of addressing and constructing data or knowledge base. This work was influenced by AI.

- In the same year, a BASEBALL question-answering system was also developed. The input to this system was restricted and the language processing involved was a simple one.
- A much more advanced system was described in Minsky (1968). This system, when compared to the BASEBALL question-answering system, was recognized and provided for the need of inference on the knowledge base in interpreting and responding to language input.

Third Phase (Grammatico-logical Phase) – Late 1970s to late 1980s

This phase can be described as the grammatico-logical phase. Due to the failure of practical system building in the last phase, the researchers moved towards the use of logic for knowledge representation and reasoning in AI.

The third phase had the following in it –

- The grammatico-logical approach, towards the end of decade, helped us with powerful general-purpose sentence processors like SRI's Core Language Engine and Discourse Representation Theory, which offered a means of tackling more extended discourse.
- In this phase we got some practical resources & tools like parsers, e.g. Alvey Natural Language Tools along with more operational and commercial systems, e.g. for database query.
- The work on lexicon in the 1980s also pointed in the direction of grammatico-logical approach.

Fourth Phase (Lexical & Corpus Phase) – The 1990s

We can describe this as a lexical & corpus phase. The phrase had a lexicalized approach to grammar that appeared in late 1980s and became an increasing influence. There was a revolution in natural language processing in this decade with the introduction of machine learning algorithms for language processing

Ambiguity and Uncertainty in Language

Ambiguity, generally used in natural language processing, can be referred to as the ability of being understood in more than one way. In simple terms, we can say that ambiguity is the capability of being understood in more than one way. Natural language is very ambiguous. NLP has the following types of ambiguities –

Lexical Ambiguity

The ambiguity of a single word is called lexical ambiguity. For example, treating the word silver as a noun, an adjective, or a verb.

Syntactic Ambiguity

This kind of ambiguity occurs when a sentence is parsed in different ways. For example, the sentence “The man saw the girl with the telescope”. It is ambiguous whether the man saw the girl carrying a telescope or he saw her through his telescope.

Semantic Ambiguity

This kind of ambiguity occurs when the meaning of the words themselves can be misinterpreted. In other words, semantic ambiguity happens when a sentence contains an ambiguous word or phrase. For example, the sentence “The car hit the pole while it was moving” is having semantic ambiguity because the interpretations can be “The car, while moving, hit the pole” and “The car hit the pole while the pole was moving”.

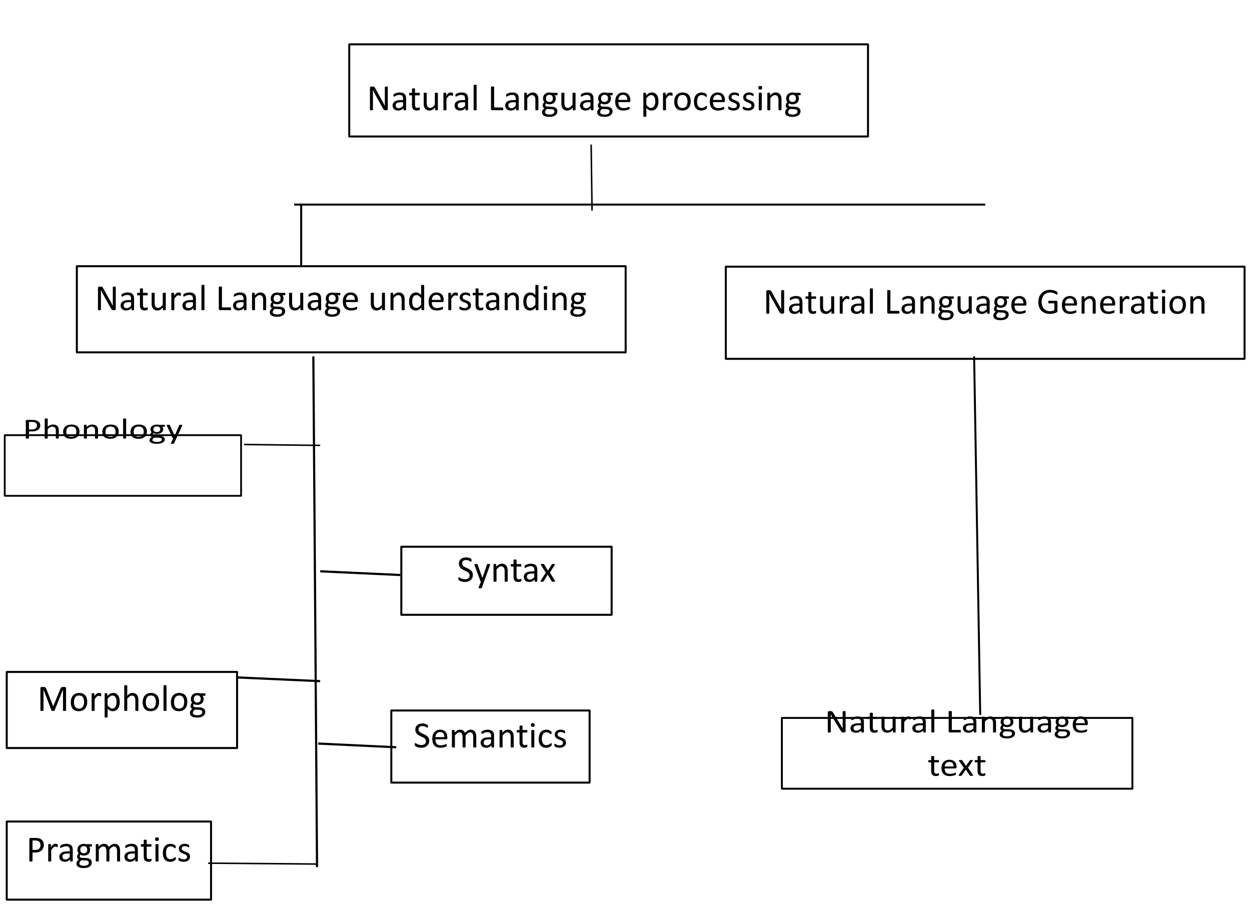
Anaphoric Ambiguity

This kind of ambiguity arises due to the use of anaphora entities in discourse. For example, the horse ran up the hill. It was very steep. It soon got tired. Here, the anaphoric reference of “it” in two situations cause ambiguity.

Pragmatic ambiguity

Such kind of ambiguity refers to the situation where the context of a phrase gives it multiple interpretations. In simple words, we can say that pragmatic ambiguity arises when the statement is not specific. For example, the sentence “I like you too” can have multiple interpretations like I like you (just like you like me), I like you (just like someone else dose).[6]

Components of NLP


Natural Language processing
Natural Language understanding
Natural Language Generation
Phonology
Syntax
Morpholog
Semantics
Pragmatics
Natural Language text

Natural Language understanding - Natural language understanding is a branch of artificial intelligence that uses computer software to understand input in the form of sentences using text or speech.

NLU enables human-computer interaction. It is the comprehension of human language such as English, Spanish and French, for example, that allows computers to understand commands without the formalized syntax of computer languages. NLU also enables computers to communicate back to humans in their own languages.[5][6]

Linguistics is the science which involves the meaning of language, language context and various forms of the language. The various important terminologies of Natural Language Processing are: -

1.Phonology

Phonology is the part of Linguistics which refers to the systematic arrangement of sound. The term phonology comes from Ancient Greek and the term phono- which means voice or sound, and the suffix –logy refers to word or speech. Phonology proper is concerned with the function, behaviour and organization of sounds as linguistic items. Phonology include semantic use of sound to encode meaning of any Human language.[7]

2.Morphology

The different parts of the word represent the smallest units of meaning known as Morphemes. Morphology which comprise of Nature of words, are initiated by morphemes. An example of Morpheme could be, the word precancellation can be morphologically scrutinized into three separate morphemes: the prefix pre, the root canceller, and the suffix.

3.Semantic

Semantic processing determines the possible meanings of a sentence by pivoting on the interactions among word-level meanings in the sentence. This level of processing can incorporate the semantic disambiguation of words with multiple senses; in a cognate way to how syntactic disambiguation of words that can errand as multiple parts-of-speech is adroit at the syntactic level.[7][8]

4.Pragmatic:

Pragmatic is concerned with the firm use of language in situations and utilizes nub over and above the nub of the text for understanding the goal and to explain how extra meaning is read into texts without literally being encoded in them.

5.syntax

It is the set of rules , principles and processes that govern the structure of a sentence in a given language using the terms syntax.

Syntax tree is the hierarchical structure used to represent the syntactic structure of sentences or strings.

Natural Language Generation

Natural Language Generation (NLG) is the process of producing phrases, sentences and paragraphs that are meaningful from an internal representation. It is a part of Natural Language Processing and happens in four phases: identifying the goals, planning on how goals maybe achieved by evaluating the situation and available communicative sources and realizing the plans as a text.[8]

Component of NLG

1.Speaker and Generator

Genrate a text we need to have a speaker or an application and a generator or a program works the application’s into fluent phrase relevant to the situation.

2.Components and Levels of Representation

The process of language generation involves the following interweaved tasks.

- Content selection: Information should be selected and included in the set. Depending on how this information is parsed into representational units, parts of the units may have to be removed while some others may be added by default.
- Textual Organization: The information must be textually organized according to the grammar; it must be ordered both sequentially and in terms of linguistic relations like modifications.
- Linguistic Resources: To support the information’s realization, linguistic resources must be chosen. In the end these resources will come down to choices of particular words, idioms, syntactic constructs etc.
- Realization: The selected and organized resources must be realized as an actual text or voice output.

3.Application or Speaker

This is only for maintaining the model of the situation. Here the speaker just initiates the process and doesn't take part in the language generation.

TECHNOLOGY DESCRIPTION

NLP Implementations

These are some of the successful implementations of Natural Language Processing (NLP):

- **Search engines** like Google, Yahoo, etc. Google search engine understands that you are a tech guy so it shows you results related to you.

- **Social websites feed** like the Facebook news feed. The news feed algorithm understands your interests using natural language processing and shows you related Ads and posts more likely than other posts.
- **Speech engines** like Apple Siri.
- **Spam filters** like Google spam filters. It’s not just about the usual spam filtering, now spam filters understand what’s inside the email content and see if it’s a spam or not. [

NLP tools and approaches:

Python and the Natural Language Toolkit (NLTK)

The Python programming language provides a wide range of tools and libraries for attacking specific NLP tasks. Many of these are found in the Natural Language Toolkit, or NLTK, an open source collection of libraries, programs, and education resources for building NLP programs.

The NLTK includes libraries for many of the NLP tasks listed above, plus libraries for subtasks, such as sentence parsing, word segmentation, stemming and lemmatization (methods of trimming words down to their roots), and tokenization (for breaking phrases, sentences, paragraphs and passages into tokens that help the computer better understand the text). It also includes libraries for implementing capabilities such as semantic reasoning, the ability to reach logical conclusions based on facts extracted from text.[9]

Install nltk

If you are using Windows or Linux or Mac, you can install NLTK **using pip**:

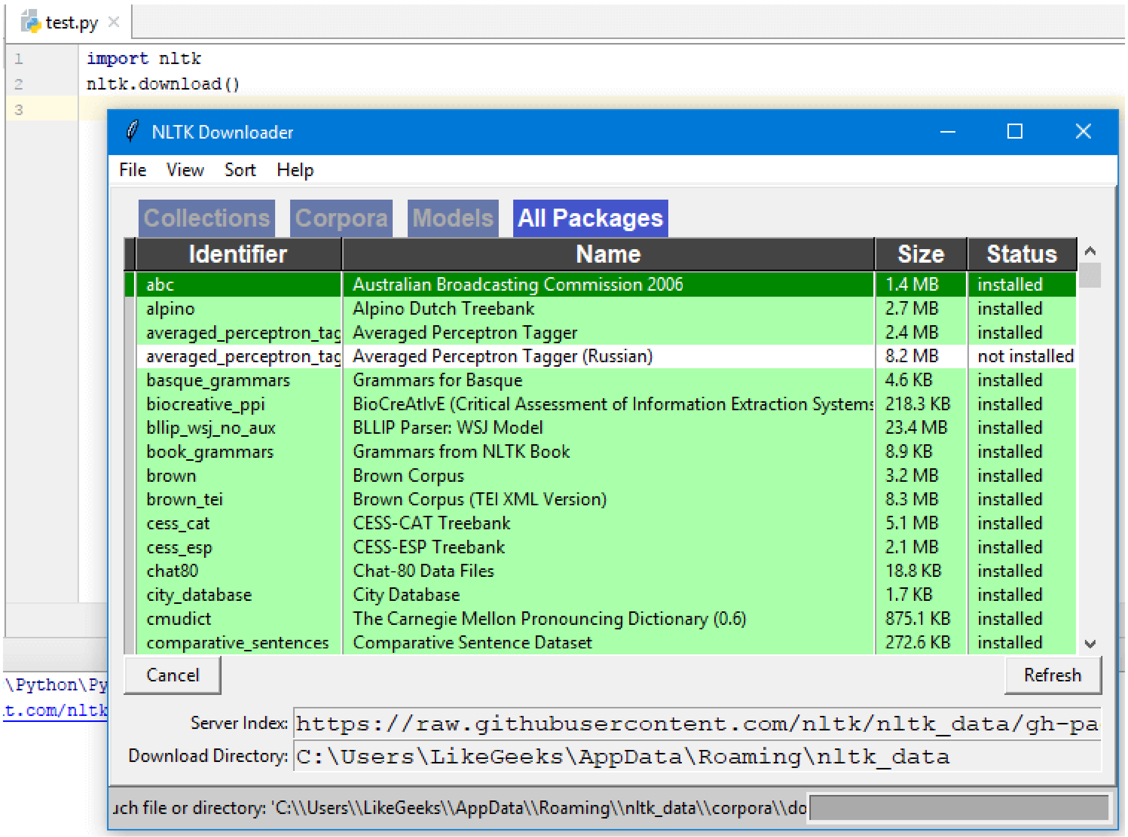
```
$ pip install nltk
```

You can use NLTK on Python 2.7, 3.4, and 3.5 at the time of writing this post.

Import nltk

install the NLTK packages by running the following code:

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



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

```
\Python\Py
t.com/nltk
```

Identifier	Name	Size	Status
abc	Australian Broadcasting Commission 2006	1.4 MB	installed
alpino	Alpino Dutch Treebank	2.7 MB	installed
averaged_perceptron_tag	Averaged Perceptron Tagger	2.4 MB	installed
averaged_perceptron_tag	Averaged Perceptron Tagger (Russian)	8.2 MB	not installed
basque_grammars	Grammars for Basque	4.6 KB	installed
biocreative_ppi	BioCreAtIvE (Critical Assessment of Information Extraction Systems	218.3 KB	installed
bllip_wsj_no_aux	BLLIP Parser: WSJ Model	23.4 MB	installed
book_grammars	Grammars from NLTK Book	8.9 KB	installed
brown	Brown Corpus	3.2 MB	installed
brown_tei	Brown Corpus (TEI XML Version)	8.3 MB	installed
cess_cat	CESS-CAT Treebank	5.1 MB	installed
cess_esp	CESS-ESP Treebank	2.1 MB	installed
chat80	Chat-80 Data Files	18.8 KB	installed
city_database	City Database	1.7 KB	installed
cmudict	The Carnegie Mellon Pronouncing Dictionary (0.6)	875.1 KB	installed
comparative_sentences	Comparative Sentence Dataset	272.6 KB	installed

Server Index: https://raw.githubusercontent.com/nltk/nltk_data/gh-pa

Download Directory: C:\Users\LikeGeeks\AppData\Roaming\nltk_data

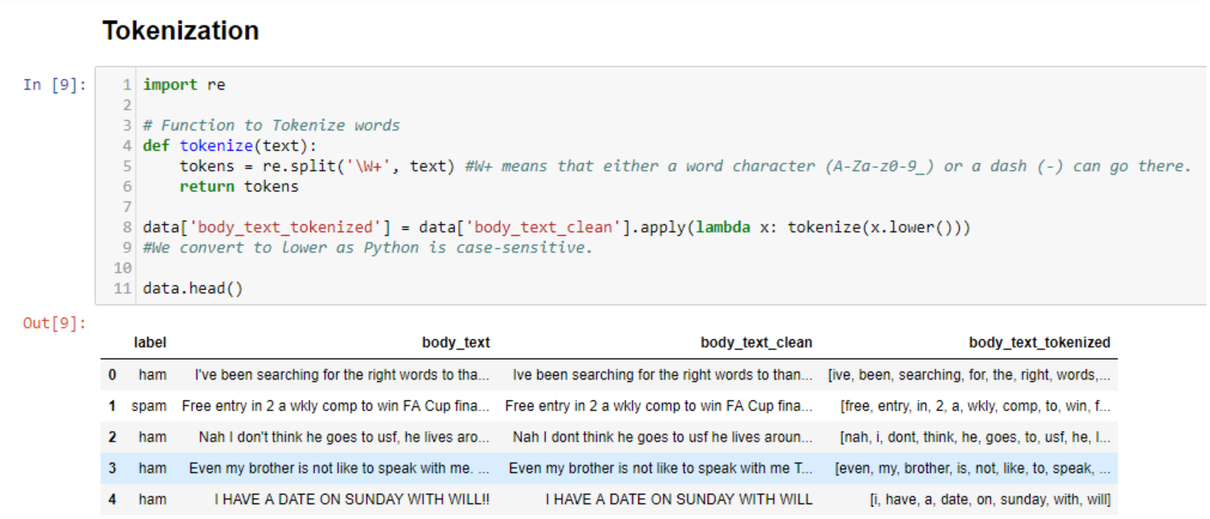
```
ıch file or directory: 'C:\\Users\\LikeGeeks\\AppData\\Roaming\\nltk_data\\corpora\\do
```

We can install all packages since they have small sizes, so no problem.

Type of pre-processing libraries:

1.Tokenizing

Tokenizing separates text into units such as sentences or words. It gives structure to previously unstructured text.



Tokenization

In [9]:
```python
import re

# Function to Tokenize words
def tokenize(text):
    tokens = re.split('\W+', text) #W+ means that either a word character (A-Za-z0-9_) or a dash (-) can go there.
    return tokens

data['body_text_tokenized'] = data['body_text_clean'].apply(lambda x: tokenize(x.lower()))
#We convert to lower as Python is case-sensitive.

data.head()
```

Out[9]:

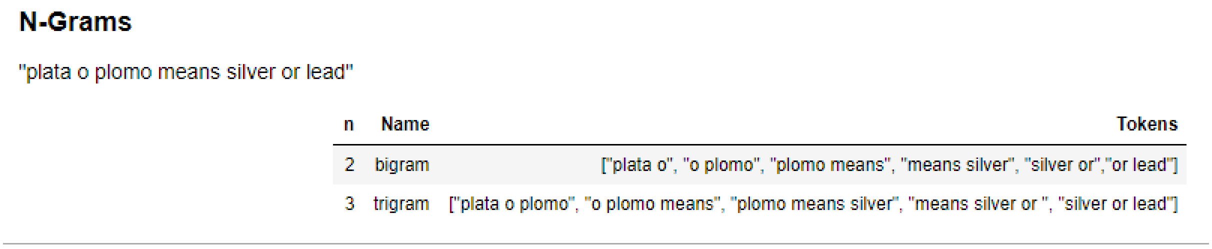
	label	body_text	body_text_clean	body_text_tokenized
0	ham	I've been searching for the right words to tha...	Ive been searching for the right words to than...	[ive, been, searching, for, the, right, words,...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...	[free, entry, in, 2, a, wkly, comp, to, win, f...
2	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...	[nah, i, dont, think, he, goes, to, usf, he, l...
3	ham	Even my brother is not like to speak with me. ...	Even my brother is not like to speak with me T...	[even, my, brother, is, not, like, to, speak, ...
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL	[i, have, a, date, on, sunday, with, will]

In body_text_tokenized, we can see that all words are generated as tokens.

There are three type of tokenization process:

- Bigram
- Trigram
- Ngram

N-grams are simply all combinations of adjacent words or letters of length n that we can find in our source text. Ngrams with n=1 are called unigrams. Similarly, bigrams (n=2), trigrams (n=3) and so on can also be used

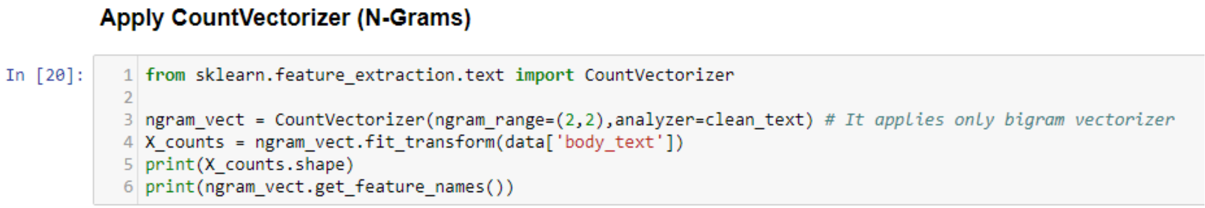


N-Grams

"plata o plomo means silver or lead"

n	Name	Tokens
2	bigram	["plata o", "o plomo", "plomo means", "means silver", "silver or","or lead"]
3	trigram	["plata o plomo", "o plomo means", "plomo means silver", "means silver or ", "silver or lead"]

Unigrams usually don't contain much information as compared to bigrams and trigrams. The basic principle behind n-grams is that they capture the letter or word that is likely to follow the given word. The longer the n-gram (higher *n*), the more context you have to work with.



Apply CountVectorizer (N-Grams)

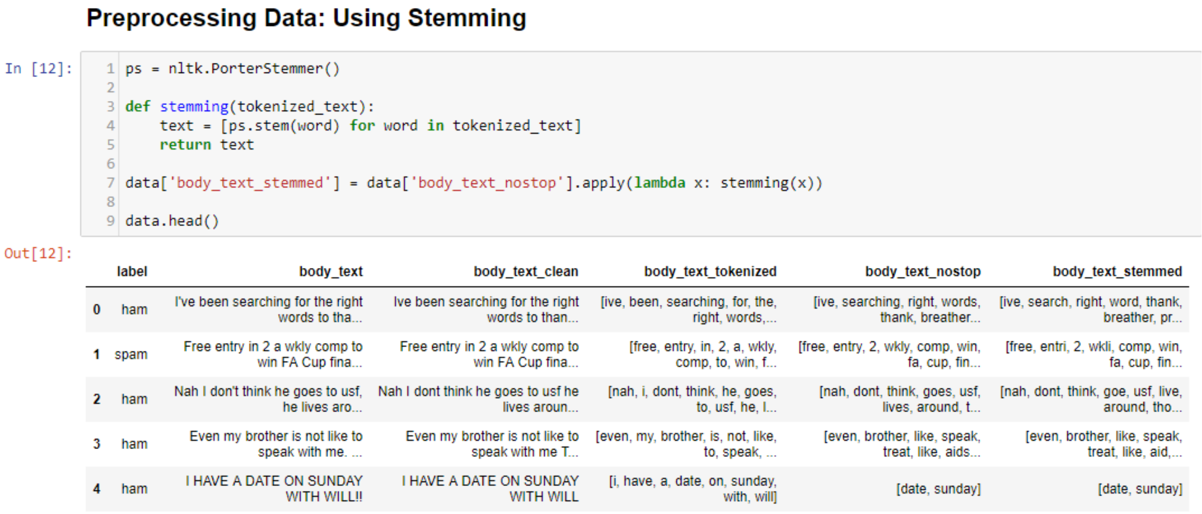
In [20]:
```python
from sklearn.feature_extraction.text import CountVectorizer

ngram_vect = CountVectorizer(ngram_range=(2,2),analyzer=clean_text) # It applies only bigram vectorizer
X_counts = ngram_vect.fit_transform(data['body_text'])
print(X_counts.shape)
print(ngram_vect.get_feature_names())
```

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.

2. 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. eg: Entitling,Entitled->Entitl.



Preprocessing Data: Using Stemming

In [12]:

```
ps = nltk.PorterStemmer()

def stemming(tokenized_text):
    text = [ps.stem(word) for word in tokenized_text]
    return text

data['body_text_stemmed'] = data['body_text_nostop'].apply(lambda x: stemming(x))

data.head()
```

Out[12]:

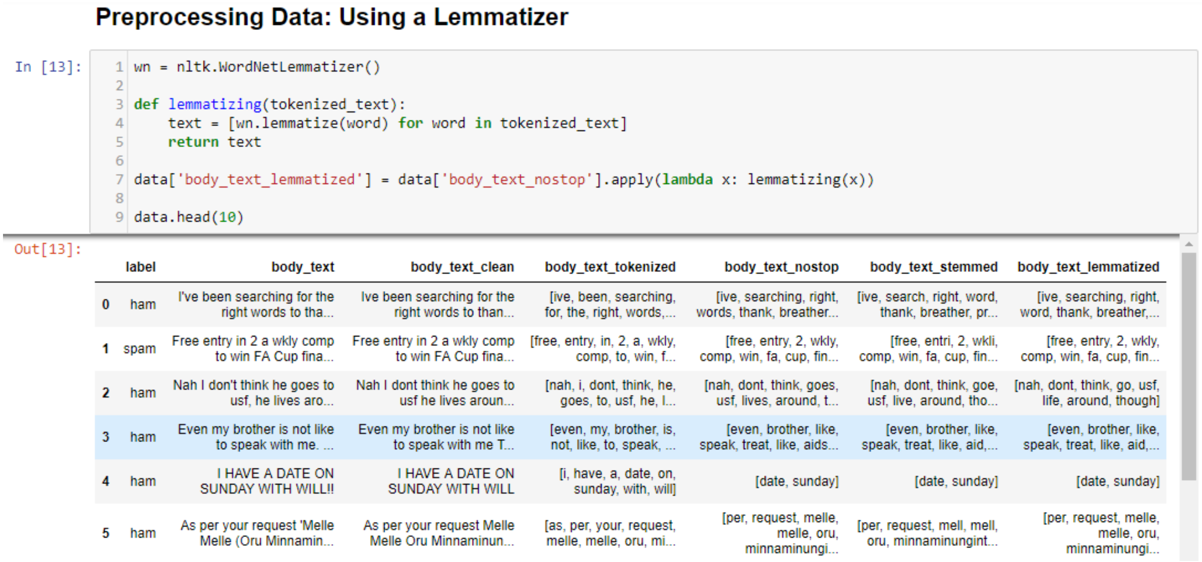
	label	body_text	body_text_clean	body_text_tokenized	body_text_nostop	body_text_stemmed
0	ham	I've been searching for the right words to tha...	Ive been searching for the right words to than...	[ive, been, searching, for, the, right, words,...	[ive, searching, right, words, thank, breather...	[ive, search, right, word, thank, breather, pr...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...	[free, entry, in, 2, a, wkly, comp, to, win, f...	[free, entry, 2, wkly, comp, win, fa, cup, fin...	[free, entri, 2, wkli, comp, win, fa, cup, fin...
2	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...	[nah, i, dont, think, he, goes, to, usf, he, l...	[nah, dont, think, goes, usf, lives, around, t...	[nah, dont, think, goe, usf, live, around, tho...
3	ham	Even my brother is not like to speak with me. ...	Even my brother is not like to speak with me T...	[even, my, brother, is, not, like, to, speak, ...	[even, brother, like, speak, treat, like, aids...	[even, brother, like, speak, treat, like, aid,...
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL	[i, have, a, date, on, sunday, with, will]	[date, sunday]	[date, sunday]

In body_text_stemmed, words like entry,wkly are stemmed to entri,wkli even though they don't mean anything.

3. Lemmatizing

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.eg: Entitling, Entitled->Entitle.

In Short, 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.



Preprocessing Data: Using a Lemmatizer

In [13]:

```
wn = nltk.WordNetLemmatizer()

def lemmatizing(tokenized_text):
    text = [wn.lemmatize(word) for word in tokenized_text]
    return text

data['body_text_lemmatized'] = data['body_text_nostop'].apply(lambda x: lemmatizing(x))

data.head(10)
```

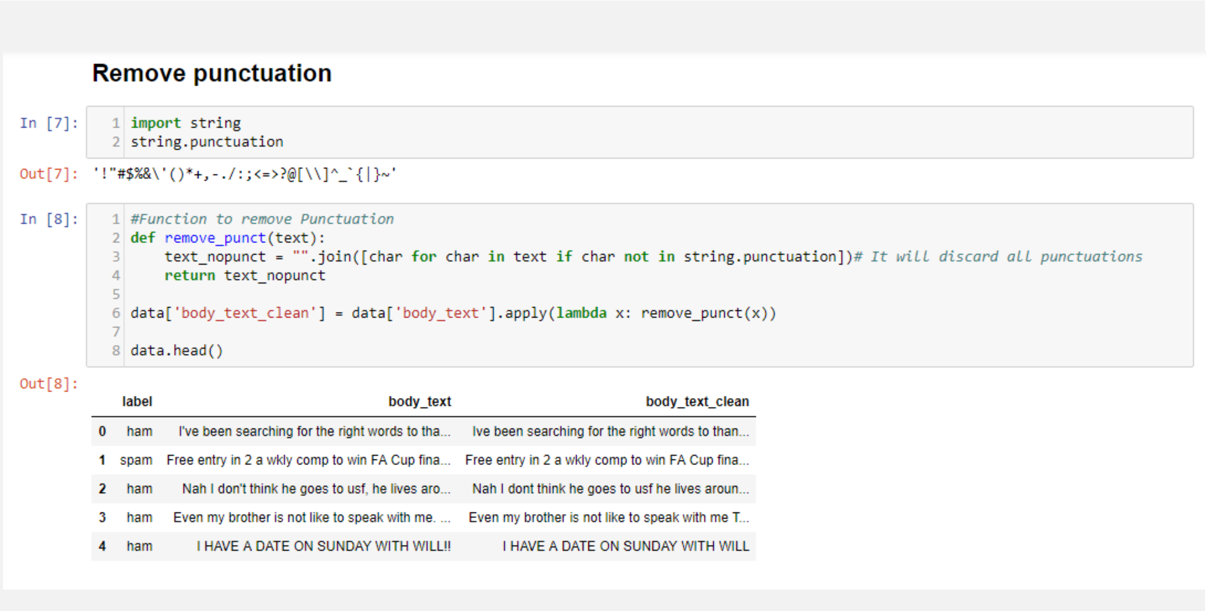
Out[13]:

	label	body_text	body_text_clean	body_text_tokenized	body_text_nostop	body_text_stemmed	body_text_lemmatized
0	ham	I've been searching for the right words to tha...	Ive been searching for the right words to than...	[ive, been, searching, for, the, right, words,...	[ive, searching, right, words, thank, breather...	[ive, search, right, word, thank, breather, pr...	[ive, searching, right, word, thank, breather,...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...	[free, entry, in, 2, a, wkly, comp, to, win, f...	[free, entry, 2, wkly, comp, win, fa, cup, fin...	[free, entri, 2, wkli, comp, win, fa, cup, fin...	[free, entry, 2, wkly, comp, win, fa, cup, fin...
2	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...	[nah, i, dont, think, he, goes, to, usf, he, l...	[nah, dont, think, goes, usf, lives, around, t...	[nah, dont, think, goe, usf, live, around, tho...	[nah, dont, think, go, usf, life, around, though]
3	ham	Even my brother is not like to speak with me. ...	Even my brother is not like to speak with me T...	[even, my, brother, is, not, like, to, speak, ...	[even, brother, like, speak, treat, like, aids...	[even, brother, like, speak, treat, like, aid,...	[even, brother, like, speak, treat, like, aid,...
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL	[i, have, a, date, on, sunday, with, will]	[date, sunday]	[date, sunday]	[date, sunday]
5	ham	As per your request 'Melle Melle (Oru Minnamin...	As per your request Melle Melle Oru Minnaminun...	[as, per, your, request, melle, melle, oru, mi...	[per, request, melle, melle, oru, minnaminungi...	[per, request, mell, mell, oru, minnaminungint...	[per, request, melle, melle, oru, minnaminungi...

In body_text_stemmed, we can word like chances are lemmatized to chance whereas it is stemmed to chance.

4. Remove punctuation

Punctuation can provide grammatical context to a sentence which supports our understanding. But for our vectorizer which counts the number of words and not the context, it does not add value, so we remove all special characters. eg: How are you?->How are you



Remove punctuation

In [7]:

```
import string
string.punctuation
```

Out[7]:

```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

In [8]:

```
#Function to remove Punctuation
def remove_punct(text):
    text_nopunct = "".join([char for char in text if char not in string.punctuation])# It will discard all punctuations
    return text_nopunct

data['body_text_clean'] = data['body_text'].apply(lambda x: remove_punct(x))

data.head()
```

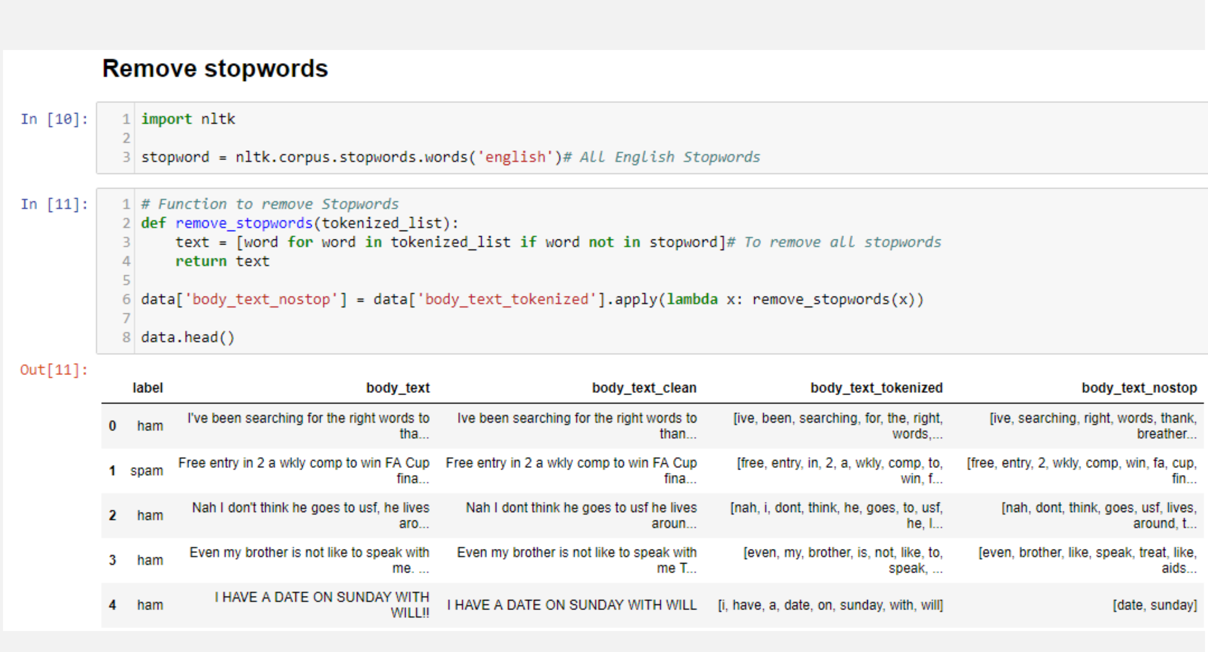
Out[8]:

	label	body_text	body_text_clean
0	ham	I've been searching for the right words to tha...	Ive been searching for the right words to than...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...
2	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...
3	ham	Even my brother is not like to speak with me. ...	Even my brother is not like to speak with me T...
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL

In body_text_clean, we can see that all punctuations like I've-> I've are omitted.

5. **Remove stopwords**

Stopwords are common words that will likely appear in any text. They don’t tell us much about our data so we remove them. eg: silver or lead is fine for me-> silver, lead, fine.



Remove stopwords

```
In [10]:
import nltk

stopword = nltk.corpus.stopwords.words('english')# All English Stopwords
```

```
In [11]:
# Function to remove Stopwords
def remove_stopwords(tokenized_list):
    text = [word for word in tokenized_list if word not in stopword]# To remove all stopwords
    return text

data['body_text_nostop'] = data['body_text_tokenized'].apply(lambda x: remove_stopwords(x))

data.head()
```

Out[11]:

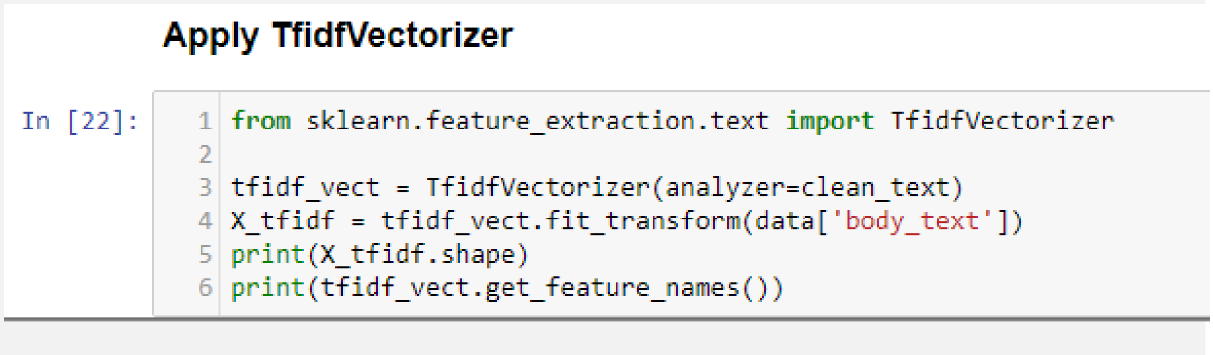
	label	body_text	body_text_clean	body_text_tokenized	body_text_nostop
0	ham	I've been searching for the right words to tha...	Ive been searching for the right words to than...	[ive, been, searching, for, the, right, words,...	[ive, searching, right, words, thank, breather...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...	[free, entry, in, 2, a, wkly, comp, to, win, f...	[free, entry, 2, wkly, comp, win, fa, cup, fin...
2	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...	[nah, i, dont, think, he, goes, to, usf, he, l...	[nah, dont, think, goes, usf, lives, around, t...
3	ham	Even my brother is not like to speak with me. ...	Even my brother is not like to speak with me T...	[even, my, brother, is, not, like, to, speak, ...	[even, brother, like, speak, treat, like, aids...
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL	[i, have, a, date, on, sunday, with, will]	[date, sunday]

In body_text_nostop, all unnecessary words like been, for, the are removed.

6. **TF-IDF**

It computes “relative frequency” that a word appears in a document compared to its frequency across all documents. It is more useful than “term frequency” for identifying “important” words in each document (high frequency in that document, low frequency in other documents).
Note: Used for search engine scoring, text summarization, document clustering.

Check my previous post — In the TF-IDF Section, I have elaborated on the working of TF-IDF.



Apply TfidfVectorizer

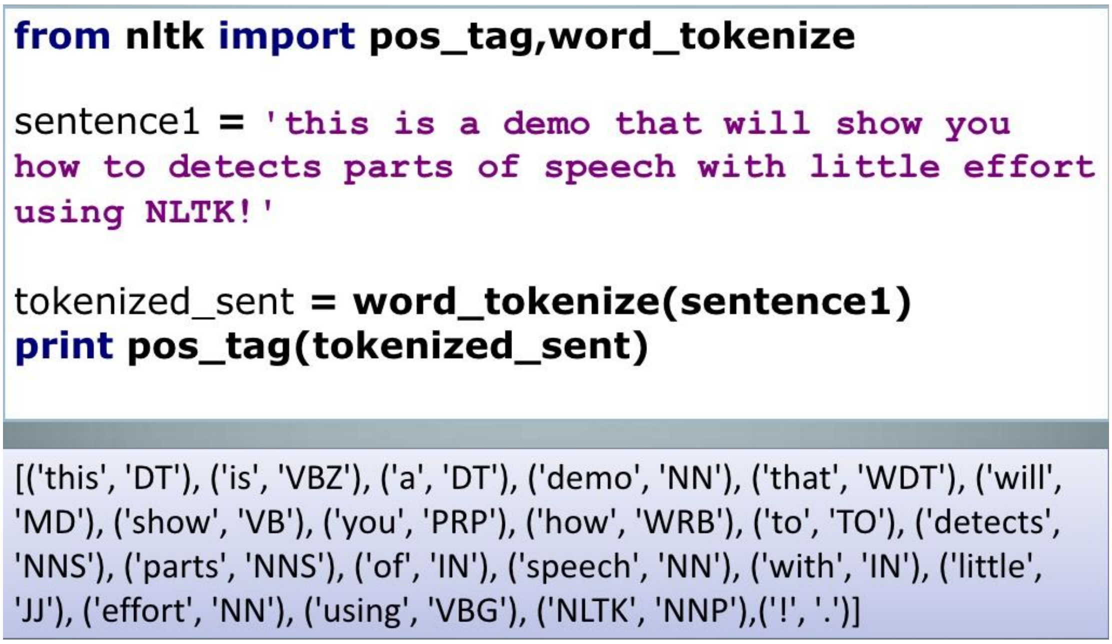
```
In [22]:
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf_vect = TfidfVectorizer(analyzer=clean_text)
X_tfidf = tfidf_vect.fit_transform(data['body_text'])
print(X_tfidf.shape)
print(tfidf_vect.get_feature_names())
```

TF-IDF is applied on the body_text, so the relative count of each word in the sentences is stored in the document matrix. (Check the repo).

Note: Vectorizers outputs sparse matrices. **Sparse Matrix** is a matrix in which most entries are 0. In the interest of efficient storage, a sparse matrix will be stored by only storing the locations of the non-zero elements.

7. **Part of speech**

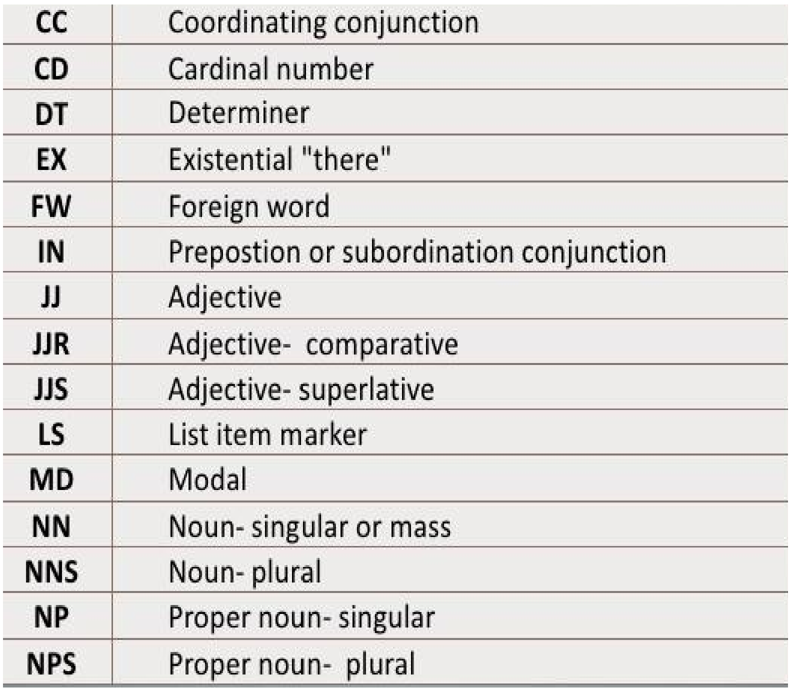


```
from nltk import pos_tag,word_tokenize

sentence1 = 'this is a demo that will show you
how to detects parts of speech with little effort
using NLTK!'

tokenized_sent = word_tokenize(sentence1)
print pos_tag(tokenized_sent)
```

```
[('this', 'DT'), ('is', 'VBZ'), ('a', 'DT'), ('demo', 'NN'), ('that', 'WDT'), ('will',
'MD'), ('show', 'VB'), ('you', 'PRP'), ('how', 'WRB'), ('to', 'TO'), ('detects',
'NNS'), ('parts', 'NNS'), ('of', 'IN'), ('speech', 'NN'), ('with', 'IN'), ('little',
'JJ'), ('effort', 'NN'), ('using', 'VBG'), ('NLTK', 'NNP'),('!', '.')]
```



CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential "there"
FW	Foreign word
IN	Prepostion or subordination conjunction
JJ	Adjective
JJR	Adjective- comparative
JJS	Adjective- superlative
LS	List item marker
MD	Modal
NN	Noun- singular or mass
NNS	Noun- plural
NP	Proper noun- singular
NPS	Proper noun- plural

Example of POS:

sent = “ tom is eating a bread ”

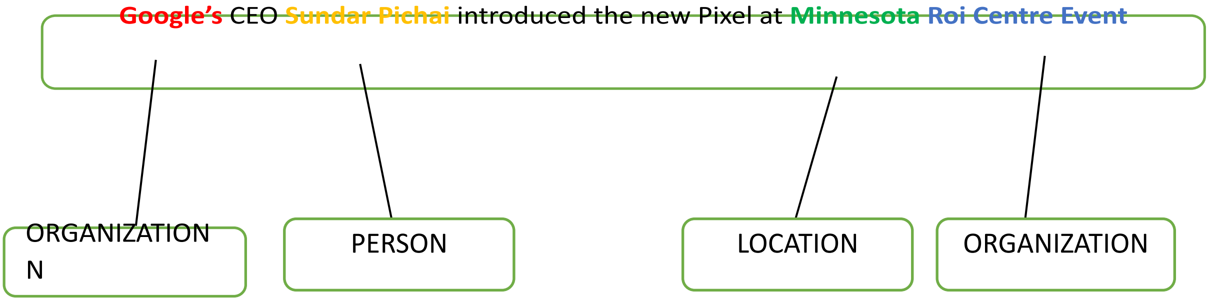
Output = [‘tom’, ‘NNP’] [‘is’, ‘VBZ’] [‘eating’, ‘VBG’] [‘a’, ‘DT’] [‘bread’, ’NN’]

8. **Named Entity Recognition-**

Named entity recognition (NER)is probably the first step towards information extraction that seeks to locate and classify named entities in text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. NER is used in many fields in Natural Language Processing (NLP).

It describes how to build named entity recognizer with NLTK and SpaCy, to identify the names of things, such as persons, organizations, or locations in the raw text.

Example of NER:


Google’s CEO Sundar Pichai introduced the new Pixel at Minnesota Roi Centre Event
ORGANIZATION
N
PERSON
LOCATION
ORGANIZATION

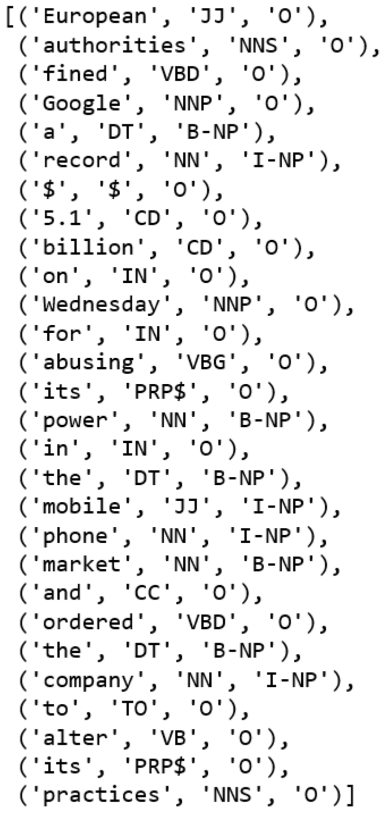
how to build named entity recognizer with NLTK and SpaCy, to identify the names of things, such as persons, organizations, or locations in the raw text.

9. Chunking

We'll implement noun phrase chunking to identify named entities using a regular expression consisting of rules that indicate how sentences should be chunked. Chunk pattern consist of one rule, that a noun phrase, NP, should be formed whenever the chunker finds an optional determiner, DT, followed by any number of adjectives, JJ, and then a noun, NN.

```
ex = 'European authorities fined Google a record $5.1 billion on
Wednesday for abusing its power in the mobile phone market and
ordered the company to alter its practices
```

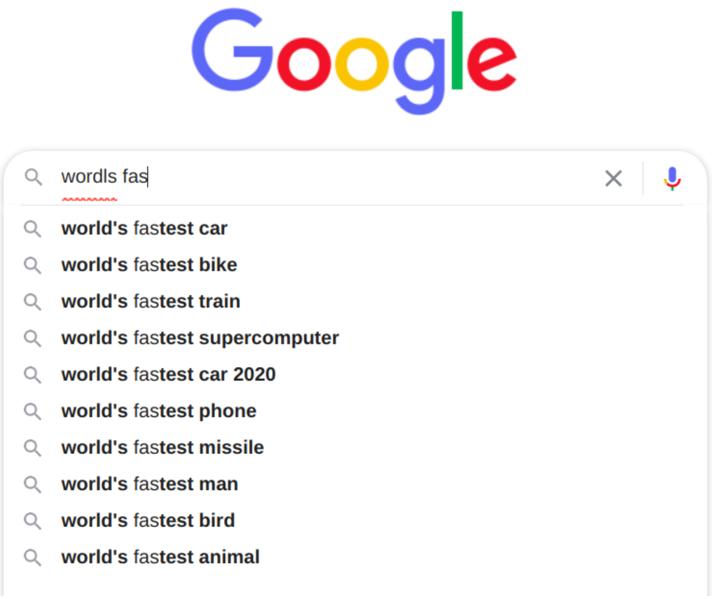
```
from nltk.chunk import conlltags2tree, tree2conlltags
from pprint import pprintiob_tagged = tree2conlltags(cs)
pprint(iob_tagged)
```



```
[('European', 'JJ', 'O'),
 ('authorities', 'NNS', 'O'),
 ('fined', 'VBD', 'O'),
 ('Google', 'NNP', 'O'),
 ('a', 'DT', 'B-NP'),
 ('record', 'NN', 'I-NP'),
 ('$', '$', 'O'),
 ('5.1', 'CD', 'O'),
 ('billion', 'CD', 'O'),
 ('on', 'IN', 'O'),
 ('Wednesday', 'NNP', 'O'),
 ('for', 'IN', 'O'),
 ('abusing', 'VBG', 'O'),
 ('its', 'PRP$', 'O'),
 ('power', 'NN', 'B-NP'),
 ('in', 'IN', 'O'),
 ('the', 'DT', 'B-NP'),
 ('mobile', 'JJ', 'I-NP'),
 ('phone', 'NN', 'I-NP'),
 ('market', 'NN', 'B-NP'),
 ('and', 'CC', 'O'),
 ('ordered', 'VBD', 'O'),
 ('the', 'DT', 'B-NP'),
 ('company', 'NN', 'I-NP'),
 ('to', 'TO', 'O'),
 ('alter', 'VB', 'O'),
 ('its', 'PRP$', 'O'),
 ('practices', 'NNS', 'O')]
```

Application of NLP

1. **Search Autocorrect and Autocomplete**
   Whenever you search something on Google, after typing 2-3 letters, it shows you the possible search terms. Or, if you search for something with typos, it corrects them and still finds relevant results for us.

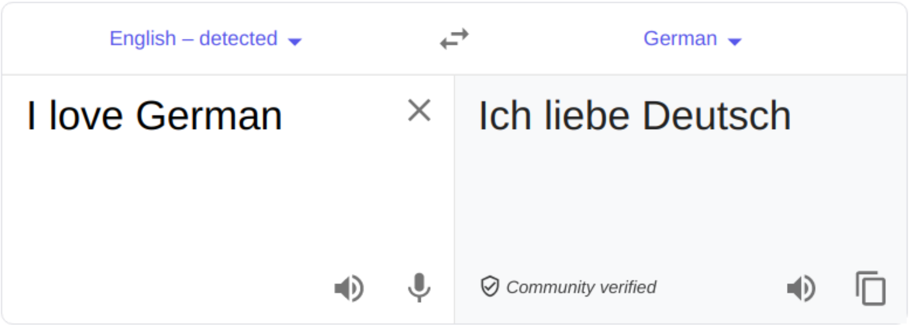


wordls fas

- world's fastest car
- world's fastest bike
- world's fastest train
- world's fastest supercomputer
- world's fastest car 2020
- world's fastest phone
- world's fastest missile
- world's fastest man
- world's fastest bird
- world's fastest animal

2. **Language Translator**

   Machine Translation is the procedure of automatically converting the text in one language to another language while keeping the meaning intact
   Have you ever used Google Translate to find out what a particular word or phrase is in a different language? I'm sure it's a YES!! and the ease with which it translates a piece of text in one language to another is pretty amazing, right? The technique behind it is Machine Translation.



English – detected ⇄ German

I love German

Ich liebe Deutsch

Community verified

Today, tools like Google Translate can easily convert text from one language to another language. These tools are helping numerous people and businesses in breaking the language barrier and becoming successful.

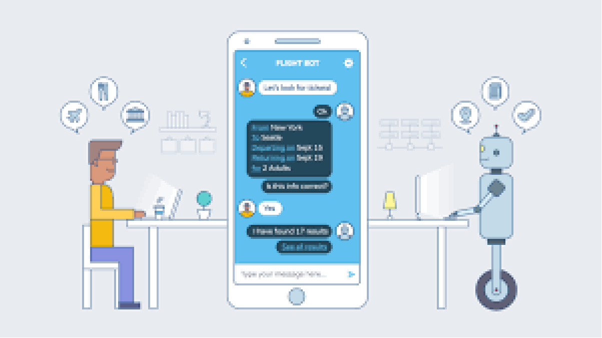
3. Social media Monitoring

More and more people these days have started using social media for posting their thoughts about a particular product, policy, or matter. These could contain some useful information about an individual's likes and dislikes. Hence analyzing this unstructured data can help in generating valuable insights. Natural Language Processing comes to rescue here too.

Various NLP techniques are used by companies to analyze social media posts and know what customers think about their products. Companies are also using social media monitoring to understand the issues and problems that their customers are facing by using their products.

4. Chatbots

Customer service and experience are the most important thing for any company. It can help the companies improve their products, and also keep the customers satisfied. But interacting with every customer manually, and resolving the problems can be a tedious task. This is where Chatbots come into the picture. Chatbots help the companies in achieving the goal of smooth customer experience.



5. Survey Analysis

Surveys are an important way of evaluating a company's performance. Companies conduct many surveys to get customer's feedback on various products. This can be very useful in understanding the flaws and help companies improve their products.

But, the problem arises when a lot of customers take the survey leading to increasing data size. It becomes impossible for a person to read them all and draw a conclusion. That's where companies use natural language processing to analyze the surveys and generate insights from them, like knowing the sentiments of users about an event from the feedbacks and analyzing product reviews to understand the pros and cons.

6. Targeted Advertising

One day I was searching for a mobile phone on Amazon, and a few minutes later, Google started showing me ads related to similar mobile phones on various webpages. I am sure you have experienced it.

Targeted advertising works mainly on Keyword Matching. The Ads are associated with a keyword or phrase, and it is shown to only those users who search for the keyword similar to the keyword with which the advertisement was associated.

7. Hiring and Recruitment

The Human Resource department is an integral part of every company. They have the most important job of selecting the right employees for a company. But, today, in this highly competitive world, recruiters need to review hundreds or sometimes thousands of resumes for a single position. It might take hours for filtering resumes and shortlisting the candidates. It filter automatic.

Yes! With the help of natural language processing, recruiters can find the right candidate with much ease. This simply means that the recruiter would not have to go through every resume and filter the right candidates manually. The technique, like information extraction with named entity recognition, can be used to extract information such as skills, name, location, and education. Then, these features can be used to represent the candidates in the feature space, and then they can be classified into the categories of fit or not-fit for a particular role. Or, they can also be recommended for a different role based on their resume.

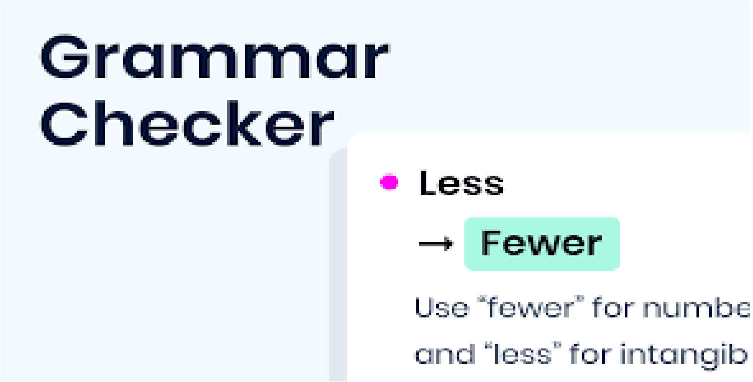
8. Voice Assistants

I am sure you all have already met them, Google Assistant, Apple Siri, Amazon Alexa, ring a bell? Yes, all of these are voice assistants.

A voice assistant is a software that uses speech recognition, natural language understanding, and natural language processing to understand the verbal commands of a user and perform actions accordingly. You might say it is similar to a chatbot, but I have included voice assistants separately because they deserve a better place on this list. They are much more than a chatbot and can do many more things than a chatbot can do.

9. Grammar Checkers

This is one of the most widely used applications of natural language processing. Grammar Checking tools like Grammarly provide tons of features that help a person in writing better content. They can change any ordinary piece of text into beautiful literature. If you want to write an email to your boss or if you're going to write a report or better an article, there is no denying the fact that you need these helpful friends.



Grammar Checker

- Less

→ Fewer

Use "fewer" for numbe and "less" for intangib

These tools can correct grammar, spellings, suggest better synonyms, and help in delivering content with better clarity and engagement. They also help in improving the readability of content and hence allowing you to convey your message in the best possible way.

10. Email Filtering

I'm sure you have, then you might have already noticed that whenever a mail arrives, it gets classified into the sections of primary, social, spam and promotions. And the best thing is that the spam emails are also filtered out to a separate section. Isn't it amazing and beneficial at the same time? Yes, it is, and that's all email filtering is. And I don't have to tell you how much our daily tasks rely on this feature.

The emails are filtered using text classification, which is a natural language processing technique. And as you might have already guessed it. Text

Classification is the process of classification of a piece of text into pre-defined categories.

Conclusion

- NLP is one of the growing technologies. With constant innovation and discovery in that field, it is only expected to grow in the future. This is such an upcoming field that needs many well skilled professionals.
- A lot of research is going into developing a new application and investigation into new techniques that will make statically NLP more feasible in the future.
- If you are interested in working on making computers learn and understand human language, then this is a good time to upskill yourself. NLP offers good prospects and is a high paying field with this technologies like Python for Data Science, Data Science with R, Machine Learning and Deep Learning, Full Stack Web Development, Mobile App Development.
- So we will be able to see improved application of NLP in the forthcoming time

References

[1] https://en.wikipedia.org/wiki/Natural_language_processing

[2] image, https://owlcation.com/humanities/Human-Language-Nature-Vs-Nurture

[3] https://towardsdatascience.com/gentle-start-to-natural-language-processing-using-python-6e46c07addf3

[4] https://www.tutorialspoint.com/natural_language_processing/natural_language_processing_quick_guide.htm

[5] https://www.ibm.com/cloud/learn/natural-language-processing

[6] Shemtov, H. (1997). Ambiguity management in natural language generation. Stanford University.

[7] Knight, K., & Langkilde, I. (2000, July). Preserving ambiguities in generation via automata intersection. In AAAI/IAAI (pp. 697-702).

[8] Nation, K., Snowling, M. J., & Clarke, P. (2007). Dissecting the relationship between language skills and learning to read: Semantic and phonological

contributions to new vocabulary learning in children with poor reading comprehension. Advances in Speech Language Pathology,

[9]
https://towardsdatascience.com/natural-language-processing-nlp-for-machine-learning-d44498845d5b

[10] images
https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.tipsandtricks-hq.com%2Fwp-content%2Fuploads%2F2018%2F10%2Fgmail-filters-web-improvement.jpg ,

https://www.google.com/imgres?imgurl=https%3A%2F%2Fwriter.com%2Fwp-content%2Fuploads%2F2020%2F08%2FGrammar-Check.png

https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.commonsense.org%2Feducation%2Fsites%2Fdefault%-the-privacy-practices-of-the-most-popular-smart-speakers-with-virtual-assistants

https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.marketingdonut

https://www.google.com/imgres?imgurl=https%3A%2F%2Fmiro.medium.com%2Fmax%2F1024%2F1*e8v1xC0NTgoduh_ei9F7Pw.png&imgrefurl=https%3A%2F%2Fchatbotsmagazine.com%2Fwhy-the-world-needs-chatbots

https://cdn.analyticsvidhya.com/wp-content/uploads/2020/07/na10-768x432.jpg

https://cdn.analyticsvidhya.com/wp-content/uploads/2020/04/na3-768x509.jpg

https://cdn.analyticsvidhya.com/wp-content/uploads/2020/07/na7-768x283.png