MICRO-PROJECT REPORT

ON

In Partial fulfilment of Diploma in Computer Engineering

In the subject of

**Python Programming (CM5461)**

**By**

**Ms. Bhagyashree Tekade (19CM003)**

**Mr. Ayush Bulbule (19CM007)**

**Mr. Pratham Gaur (19CM020)**

**Mr. Malhar Joshi (19CM033)**

**Ms. Wruddhi Raut (19CM053)**

**Ms. Akanksha Shewatkar (19CM057)**

**Submitted To**



**Government Polytechnic, Amravati**

(An Autonomous Institute of Govt. of Maharashtra)

Under the guidance of

**Prof. M. R. Torney**

**Lecturer**

Department of Computer Engineering

Government Polytechnic, Amravati

**(2021-2022)**

|  |  |
| --- | --- |
|  | **Government Polytechnic Amravati**  **(An Autonomous Institute of Govt. of Maharashtra)**  **Department of Computer Engineering** |
| *Certificate*  This is to certify that  **Ms. Bhagyashree Tekade, Mr. Ayush Bulbule, Pratham Gaur, Mr. Malhar Joshi Ms. Wruddhi Raut Ms. Akanksha Shewatkar** IDCode **19CM003, 19CM007, 19CM020, 19CM033, 19CM053, 19CM057** Sixth Semester Diploma in Computer Engineering has satisfactorily completed the micro project entitled **“ Natural Language Pre-processing on .csv Data”**  in **(CM5461) Programming with Python** the academic year 2020-21 as prescribed in curriculum. | |

|  |  |
| --- | --- |
| Date: 07 /06 /2022  Place: Amravati | **Prof. M. R. Torney**  Lecturer |

**Institute Vision**

To be a vibrant technical institute of global repute contributing towards the needs of industries & society.

**Institute Mission**

* To develop competent diploma engineers suitable for contemporary industrial environment.
* To include socially accepted ethics & values among budding engineers.
* To Nurture innovations and entrepreneurship .
* To produce engineers with psychomotor & cognitive skills committed to lifelong learning.



**Vision**

Provide skilled professionals in Computer Engineering to contribute towards the advancement of technology useful for society and the industrial environment

**Mission**

**M1.** Impart need-based and value-based education by providing exposure to the latest tools and technologies in the area of computer engineering to satisfy the stakeholders.

**M2**: Upgrade and maintain facilities for quality technical education with continuous effort for excellence in Computer Engineering.

**M3**: Train students with Computer Engineering knowledge to apply it in the general disciplines of design, deployment of software and integration of existing technologies for E-governance and benefit of society.

**M4**: Provide a learning ambiance to enhance innovations, problem-solving skills, leadership qualities, team spirit and ethical responsibilities.

**M5**: Provide an academic environment and consultancy services to the industry and society in the area of Computer Engineering.

**Annexure-I.**

PART A - Plan

Title of Micro-Project – **Natural Language Pre-processing on .csv Data**

**1.0 Brief Introduction**

This project is to clean the data required to feed as input for ml algorithm. In this project we have applied NLP pre-processing concepts such as Removing URL’s Removing Stop words, Removing Special Characters from data.

## **2.0 Aims of the Micro-Project**

1. Cleaning the data set required for Natural Language Processing
2. Applying the concepts of NLP pre-processing
3. Making the dataset ready to be feed as input for other NLP Algo.

**3.0 Course Outcomes Integrated**

By doing this project, we completed the course outcome of

1. Write ‘Python’ programs using arithmetic expressions and control structure.
2. Develop ‘Python’ programs using List, Tuples and Dictionary.
3. Develop/Use functions in Python programs for modular programming approach.
4. Develop ‘Python’ programs using File Input/output operations.

**4.0 Actual Resources Used**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **Name of the Material/Resources** | **QTY** | **Remarks** |
|  | Microsoft word | 1 |  |
|  | VS Code | 1 |  |
|  | Google Colab | 1 |  |

### 6.0 Assessment by Faculty as per Rubrics

|  |  |  |  |
| --- | --- | --- | --- |
| **Process Assessment**  **(03)** | **Product Assessment (02)** | **Total Marks**  **(05)** | **Signature of Faculty** |
|  |  |  |  |

**Annexure-2**

PART B- Plan

Title of Micro-Project – **Natural Language Pre-processing on .csv Data**

**1.0 Brief Introduction**

This project is to clean the data required to feed as input for ml algorithm. In this project we have applied NLP pre-processing concepts such as Removing URL’s Removing Stop words, Removing Special Characters from data.

**2.0 Action Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| S.N | Details of activity | Planned start date | Planned finish date |
| 1. | Discussed and finalized the topic. | 05-04-2022 | 15-04-2022 |
| 2. | Collecting information. | 18-04-2022 | 25-04-2022 |
| 3. | Write the code and execute it | 26-04-2022 | 18-05-2022 |
| 4. | Making Word File of Project. | 11-05-222 | 16-05-2022 |

***3.0 Resources Required***

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **Name of the**  **Material/Resources** | **QTY** | **Remarks** |
| 1. | Python 3.10 | 1 |  |
| 2. | Google-Colab | 1 |  |
| 3. | Microsoft word | 1 |  |
|  | Microsoft Excel | 1 |  |

**Natural Language Pre-processing on .csv Data**

**What is NLP?**

**Natural language processing (NLP)** refers to the branch of computer science—and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.

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.

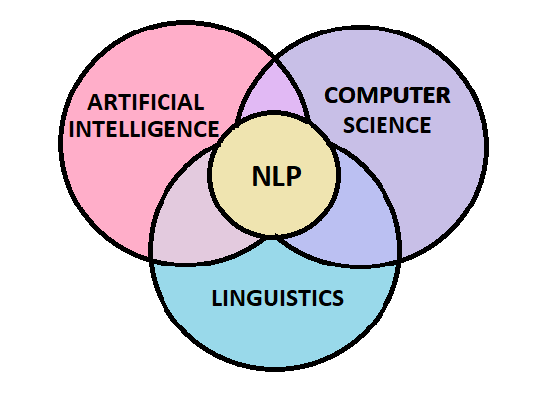


Figure 1.1

NLP drives computer programs that translate text from one language to another, respond to spoken commands, and summarize large volumes of text rapidly—even in real time. There’s a good chance you’ve interacted with NLP in the form of voice-operated GPS systems, digital assistants, speech-to-text dictation software, customer service chatbots, and other consumer conveniences. But NLP also plays a growing role in enterprise solutions that help streamline business operations, increase employee productivity, and simplify mission-critical business processes.

Ever since then, there has been an immense amount of study and development in the field of Natural Language Processing. Today NLP is one of the most in-demand and promising fields of Artificial Intelligence!

There are two main parts to Natural Language Processing:

1. Data Preprocessing
2. Algorithm Development

The idea of this article is to help you get started with NLP and provide you with the basic concepts and tools that are needed to work with text data. We shall be focusing mainly on the first and the most crucial part of Natural Language Processing – Text Preprocessing.

**What is Natural Language Preprocessing?**

The kind of data you get from customer feedback is usually unstructured. It contains unusual text and symbols that need to be cleaned so that a machine learning model can grasp it. Data cleaning and pre-processing are as important as building any sophisticated machine learning model. The reliability of your model is highly dependent upon the quality of your data.

Text processing is a method used under the NLP to clean the text and prepare it for the model building. It is versatile and contains noise in various forms like emotions, punctuations, and text written in numerical or special character forms. We have to deal with these main problems because machines will not understand they ask only for numbers. To start with text processing, certain libraries written in Python simplify this process, and their simple, straightforward syntax gives a lot of flexibility. The first one is NLTK stands for natural language toolkit useful for all tasks like stemming, POS, tokenization, lemmatizing and many more.

**What is NLTK?**

The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP).

It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning. It also includes graphical demonstrations and sample data sets as well as accompanied by a cook book and a book which explains the principles behind the underlying language processing tasks that NLTK supports.



Figure 1.2 NLP-NLTK

**What is Spacy?**

**spaCy** is a free and open-source library for Natural Language Processing (NLP) in Python with a lot of in-built capabilities. It’s becoming increasingly popular for processing and analyzing data in NLP. Unstructured textual data is produced at a large scale, and it’s important to process and derive insights from unstructured data. To do that, you need to represent the data in a format that can be understood by computers. NLP can help you do that.

spaCy is a free, open-source library for NLP in Python. It’s written in Cython and is designed to build information extraction or natural language understanding systems. It’s built for production use and provides a concise and user-friendly API.

**What is Pandas?**

Pandas is an opensource Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays. As one of the most popular data wrangling packages, Pandas works well with many other data science modules inside the Python ecosystem, and is typically included in every Python distribution, from those that come with your operating system to commercial vendor distributions like ActiveState’s ActivePython.

**Cleaning and Removing Noise**

It helps to get rid of unhelpful parts of the data, or noise, by converting all characters to lowercase, removing punctuations marks, and removing stop words and typos.

Removing noise comes in handy when you want to do text analysis on pieces of data like comments or tweets. The code in the following sections will be helpful to get rid of the text that interferes with text analysis.

To prepare the text data for the model building we perform text preprocessing. It is the very first step of NLP projects. Some of the preprocessing steps are:

* Removing punctuations like . , ! $( ) \* % @
* Removing URLs
* Removing Stop words
* Lower casing
* Tokenization
* Stemming
* Lemmatization

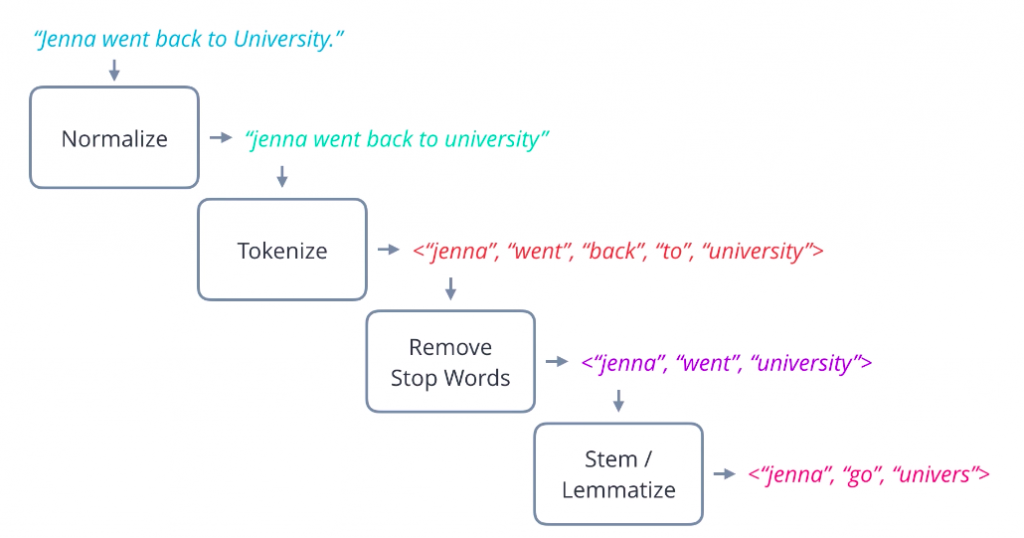
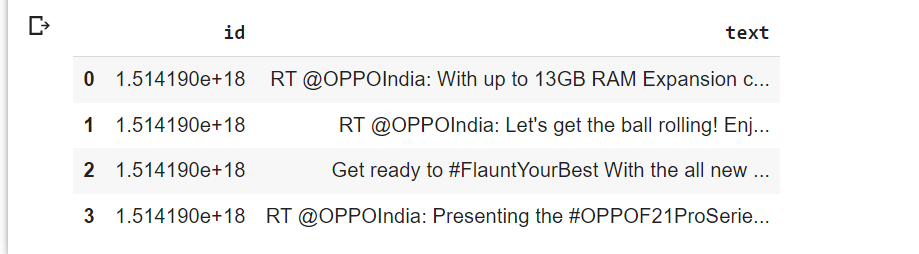


Figure 1.3

Mounting the data set and working on it. We have taken twitter’s twitts data set to work on it. Data present initially before applying the concepts of NLP preprocessing:



Applying NLP Preprocessing on texts:

1. **Removing Punctuations**: In this step, all the punctuations from the text are removed. string library of Python contains some pre-defined list of punctuations such as ‘!”#$%&'()\*+,-./:;?@[\]^\_`{|}~’

Code:

#removing punctuation

def remove\_punctuation(text):

    punctuationfree="".join([i for i in text if i not in string.punctuation])

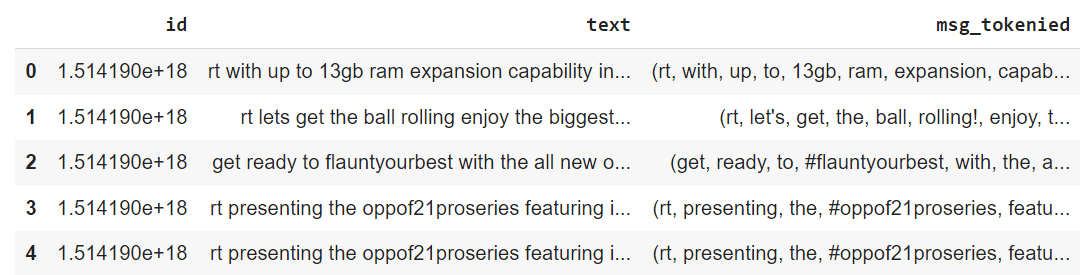
    return punctuationfree

#storing the puntuation free text

data['text']= data['text'].apply(lambda x:remove\_punctuation(x))

data.head()

Output:



1. **Lowering the text:** It is one of the most common preprocessing steps where the text is converted into the same case preferably lower case. But it is not necessary to do this step every time you are working on an NLP problem as for some problems lower casing can lead to loss of information.

For example, if in any project we are dealing with the emotions of a person, then the words written in upper cases can be a sign of frustration or excitement.

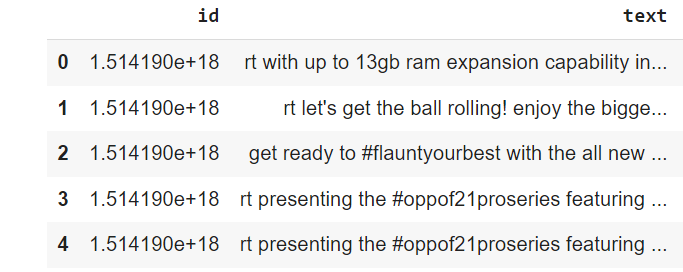
Code:

# Text to lower case

data['text']= data['text'].apply(lambda x: x.lower())

data.head()

Output:



1. **Tokenization:** In this step, the text is split into smaller units. We can use either sentence tokenization or word tokenization based on our problem statement.

Code:

#data tokenization

tokenizer = Tokenizer(nlp.vocab)

def tokenization(text):

    tokens = tokenizer(text)

    return tokens

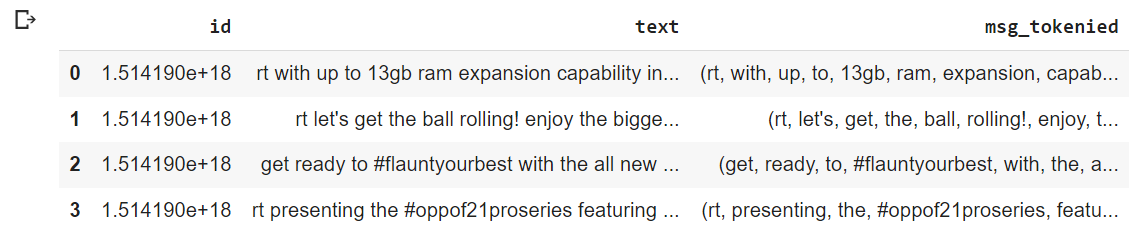
# applying function to the column

data['msg\_tokenied']= data['text'].apply(lambda x: tokenization(x))

# print(data['msg\_tokenied'])

data.head(4)

Output:



1. **Removing Links:** Sometimes while processing data links does not convey any meaning so links removal is also a part of NLP Preprocessing.

**Code:**

# Removing url

import re

count =0

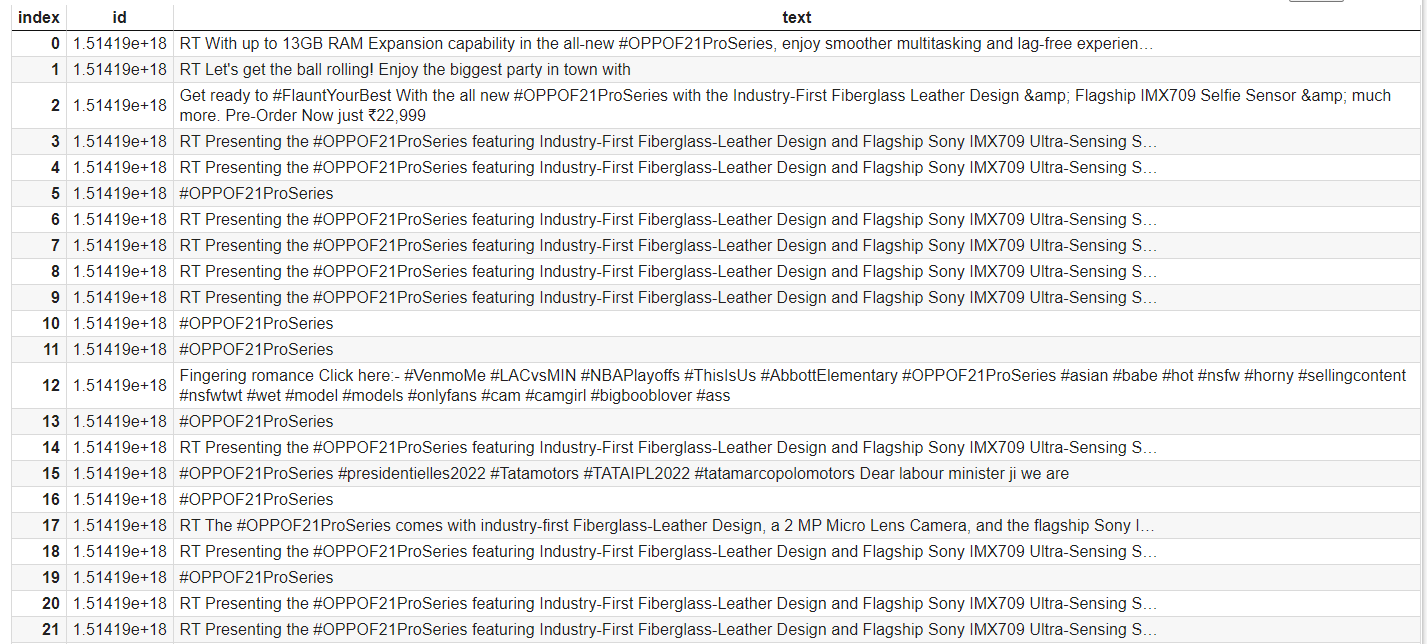
def remove\_urls(text):

  return re.sub(r"http\S+", "", text)

data['text']= data['text'].apply(lambda x:remove\_urls(x))

data.head(25)

# data['text']

Output:

1. **Removing Mentions:** As we are applying NLP on twits data there are mentions of various twitter accounts which is irrelevant to the NLP algorithm. So we have to remove the mentions also.

def remove\_mention(text):

  print(text)

  text\_sp = text.split()

  ans = ' '.join([i for i in text\_sp if '@' not in i])

  print(ans)

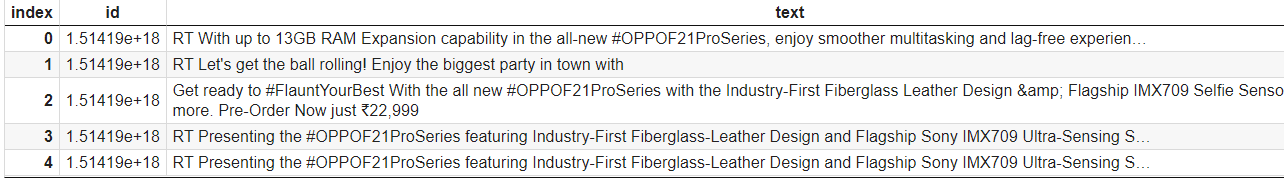
  return ans

#storing the puntuation free text

data['text']= data['text'].apply(lambda x:remove\_mention(x))

data.head()

**Output:**

****

**Project Source Code:**

!pip install spacy

!pip install nltk

!pip install pandas

from google.colab import drive

drive.mount('/content/drive')

#importing the libraries

import pandas as pd

import spacy

import string

import nltk

from spacy.tokenizer import Tokenizer

from spacy.lang.en import English

from nltk.stem.porter import PorterStemmer

load\_model = spacy.load('en', disable = ['parser','ner'])

#defining the object for stemming

porter\_stemmer = PorterStemmer()

nltk.download('stopwords')

#Stop words present in the library

stopwords = nltk.corpus.stopwords.words('english')

string.punctuation

nlp = spacy.load("en\_core\_web\_sm")

sw\_spacy = nlp.Defaults.stop\_words

nlp

# Mounting data

data = pd.read\_csv("/content/drive/MyDrive/NLPDatasets/tweets.csv")

data.head(5)

def remove\_mention(text):

  print(text)

  text\_sp = text.split()

  ans = ' '.join([i for i in text\_sp if '@' not in i])

  print(ans)

  return ans

#storing the puntuation free text

data['text']= data['text'].apply(lambda x:remove\_mention(x))

data.head()

# Removing url

import re

count =0

def remove\_urls(text):

  return re.sub(r"http\S+", "", text)

data['text']= data['text'].apply(lambda x:remove\_urls(x))

data.head(25)

# data['text']

# Removing Tags

count =0

def remove\_mention(text):

  print(text)

  text\_sp = text.split()

  ans = ' '.join([i for i in text\_sp if '#' not in i])

  print(ans)

  return ans

#storing the puntuation free text

data['text']= data['text'].apply(lambda x:remove\_mention(x))

data.head()

#removing punctuation

def remove\_punctuation(text):

    punctuationfree="".join([i for i in text if i not in string.punctuation])

    return punctuationfree

#storing the puntuation free text

data['text']= data['text'].apply(lambda x:remove\_punctuation(x))

data.head()

#data tokenization

tokenizer = Tokenizer(nlp.vocab)

def tokenization(text):

    tokens = tokenizer(text)

    return tokens

# applying function to the column

data['msg\_tokenied']= data['text'].apply(lambda x: tokenization(x))

# print(data['msg\_tokenied'])

data.head(4)

#defining a function for stemming

def stemming(text):

  doc2 = load\_model(text)

  return(" ".join([token.lemma\_ for token in doc2]))

  # stem\_text = [porter\_stemmer.stem(word) for word in text]

  # print(stem\_text)

  # return stem\_text

data['text\_lemmatized']=data['text'].apply(lambda x: stemming(x))

data['text\_lemmatized']

data.head()

from google.colab import files

data.to\_csv('output.csv', encoding = 'utf-8-sig')

files.download('output.csv')

**Google Colab Link:**

<https://colab.research.google.com/drive/1AZh2tvtP9MzwOaJ-dNlGRZ0lJfmF-4Gn?usp=sharing>

**Conclusion:** In this way, we have created a python script project to apply NLP Preprocessing on the data to make it ready for NLP Algorithm.

**References:**

1. <https://www.analyticsvidhya.com/>
2. <https://spacy.io/>
3. <https://www.w3schools.com/python>
4. <https://www.nltk.org/>
5. <https://pandas.pydata.org/>
6. <https://colab.research.google.com/>