CODING AND SOLUTIONING

Assignment Date	18 November 2022
Student Name	K.Devika
Student Roll Number	815819104009
Maximum Marks	

Code layout,readability,reusability

```
import xml.etree.ElementTree
as ET
```

```
import zipfile

from os import listdir

from os.path import isfile, join

from typing import List, Tuple
```

```
import gdown
```

```
def main():
```

```
    url =
"https://drive.google.com/uc?id=1jI1cmxqnwsmC-vbl8dNY6b4aNBtBbKy
3"

    output_path = "Twitter.zip"

    path_train = "Data/train/en"

    path_test = "Data/test/en"
```

```
    data_getter = DataGetter(url, output_path, path_train,
path_test)

    tweet_train, tweet_test = data_getter.get_train_test_docs()



class DataGetter:
    def __init__(self, url: str, output_path: str, path_train:
str, path_test: str):
        self.url = url
        self.output_path = output_path
        self.path_train = path_train
        self.path_test = path_test
        self.download_zip_data_from_google_drive()
        self.unzip_data()

    def download_zip_data_from_google_drive(self):

        gdown.download(self.url, self.output_path, quiet=False)

    def unzip_data(self):

        with zipfile.ZipFile(self.output_path, "r") as zip_ref:
            zip_ref.extractall(".")

    def get_train_test_docs(self) -> Tuple[list, list]:
```

```
        tweets_train_files = self.get_files(self.path_train)
        tweets_test_files = self.get_files(self.path_test)


        t_train = self.extract_texts_from_multiple_files(
            self.path_train, tweets_train_files
        )
        t_test = self.extract_texts_from_multiple_files(
            self.path_test, tweets_test_files
        )
        return t_train, t_test


    @staticmethod
    def get_files(path: str) -> List[str]:


        return [
            file
            for file in listdir(path)
            if isfile(join(path, file)) and file != "truth.txt"
        ]


    def extract_texts_from_multiple_files(
        self, path_to_file: str, files: list
    ) -> List[str]:


        all_docs = []
        for file in files:
            text_in_one_file =
self.extract_texts_from_each_file(path_to_file, file)
```

```
            all_docs.append(text_in_one_file)


        return all_docs


    @staticmethod
    def extract_texts_from_each_file(path_to_file: str,
file_name: list) -> str:


        list_of_text_in_one_file = [
            r.text for r in ET.parse(join(path_to_file,
file_name)).getroot()[0]
        ]
        text_in_one_file_as_string = " ".join(t for t in
list_of_text_in_one_file)


        return text_in_one_file_as_string




if __name__ == "__main__":
    main()
```