**TEXT ANALYSIS & SENTIMENT ANALYSIS OF CHARLES DICKEN’S OLIVER TWIST ON PYTHON**

**PROJECT INFORMATION**

Text Analysis is the process through which computer systems understand human language texts and then extract certain information from the texts. For the computer system to understand these human texts and extract information, the text needs to be processed in a readable form that the computer will understand.

The goal of this project is to find out the word that was frequently mentioned and the number of times it was mentioned in the famous classical novel, Oliver Twist written by Charles Dickens.

For the project, certain text analysis operations were carried out and are highlighted below:

**Installation of a Natural Language Processing Tool: Natural Language ToolKit**

Natural Language Toolkit, also known as NLTK, was employed in this project to build python programs that understood human language to produce accurate information. At the commencement of the project, NLTK was installed alongside its packages. Below is the input codes that downloaded NLTK,

*import nltk*

*nltk.download()*

After the installation of NLTK and was confirmed to be working, the dataset was loaded into the program. The dataset was sourced from the internet. It was extracted from the online digital library – Project Gutenberg.

The dataset: <http://www.gutenberg.org/cache/epub/730/pg730.txt>

To load the online text into the program, this function was called:

*from urllib import request*

**Cleaning of the Text**

For the computer to achieve the project goal, the dataset was cleaned. There are some elements in human language data that are not useful in text analysis. It is best to discard these unnecessary elements to have a clean dataset.

The cleaning of the text involved the removal of numbers, punctuation marks and stop words. Stop words are words that the computer system regards as noises in a text. They are removed because they are usually abundant in the text and they are ignored during natural language processing and it is best to filter them out from the dataset. Words such as a, an, the, he, almost, about, any, various, many and so on, are stop words. In the dataset used for this project, 179 stop words were filtered out from the text. The stop word function was imported from the NLTK corpus

*from nltk.corpus import stopwords*

**Whitespace**

After the text was cleaned, the items in the text looked muddled up so they were split by whitespace to space them. This process turned the text to a list variable.

**Normalization**

The text was converted to lowercase. This process is a type of normalization that reduces noises in the text and also help make the items in the text look uniform.

**Tokenization**

This is the process that breaks the big parts of the text – paragraphs- into tokens (words and sentences). To tokenize the text, the text was converted from the list variable to a string.

*from nltk.tokenize import sent\_tokenize, word\_tokenize*

**Stemming and Lemmatization**

To make the text analyzable, stemming and lemmatization were performed on the text. These operations reduced the length of the text by converting words to their root and base words. To carry out these operations, the following NLTK packages were imported:

*from nltk.stem.porter import PorterStemmer*

*from nltk.stem import WordNetLemmatizer*

**Frequency Distribution**

This showed the word that was frequently mentioned in the text and the number of times it was mentioned.

The word ‘mr’ was mentioned 1344 times in the text. Also, the first ten most frequent words were realized and visually represented together with their frequency counts using Matplot Lib and WordCloud.

Matplot Lib was imported and WordCloud was downloaded into the program.

**WHY TEXT ANALYSIS?**

Text analysis is important because it gives accurate results. If I took up the task of finding the word that was frequently mentioned in the text – Oliver Twist by Charles Dickens on my own, I would not produce an accurate result like this. It would have been a cumbersome task for me that would have taken me so much time and energy. With text analysis, difficult tasks like this project are effectively carried out.

**SENTIMENT ANALYSIS**

This analysis is carried out to show the emotion of the text. TextBlob is the tool that was used to analyze the text. The sentiment polarity of the text is 0.08228097558751675 while the Sentiment subjectivity is 0.45884504140555105.

*from textblob import TextBlob*

NLTK was used to tell the exact sentiment of the text and the result showed that the text possessed a Positive Sentiment.

*from nltk.sentiment.vader import SentimentIntensityAnalyzer*