## Indian Institute of Information Technology Surat

****

# Lab Report on

# Natural Language Processing (CS 601) Practical

**Submitted by**

### [RAHUL KUMAR SINGH] (UI21CS44)

**Course Faculty**

### Mrs. Nidhi Desai

## Department of Computer Science and Engineering

## Indian Institute of Information Technology Surat

## Gujarat-394190, India

**Jan-2024**

## Lab No: 2

**Aim:**

Study of essential text pre-processing techniques. Write python script for the essential text preprocessing techniques. Store the preprocessed data into a separate column of .CSV file. Compare the outcomes with and without using libraries for the same.

**Description:**

Perform the following task with using inbuilt Python Libraries:

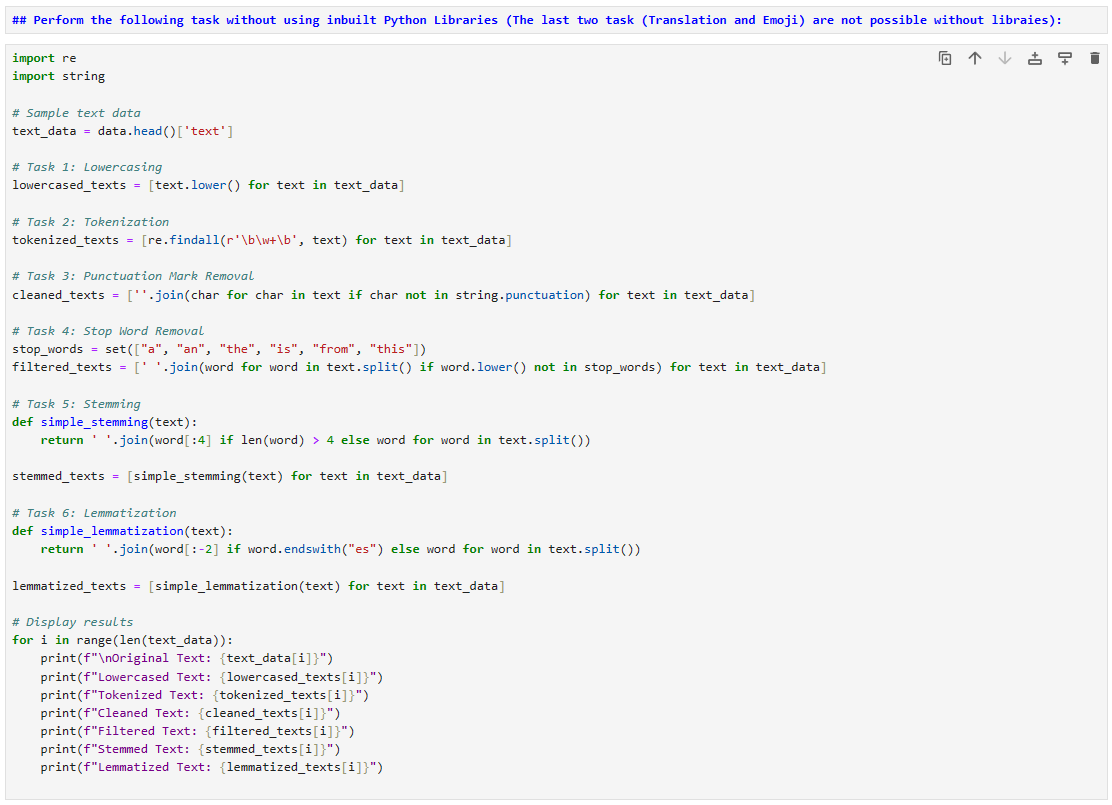
* Lower Casing: Converts text into lower case text. It Helps ensure uniformity in text analysis and processing, as it treats uppercase and lowercase forms of words as the same.
* Tokenization: Break the text into individual words or tokens. It Facilitates analysis at the word level, making it easier to extract meaningful information and perform various natural language processing tasks.
* Punctuation Mark Removal: Eliminate punctuation marks from the text. Enhances the accuracy of text analysis by removing non-alphanumeric characters that don't contribute to the core meaning of the text.
* Stop Word Removal: Exclude common words (stop words) like "and," "the," and "is" that don't carry significant meaning. Improves the efficiency of text processing and analysis by focusing on content-bearing words.
* Stemming: Reduce words to their root or base form by removing suffixes. Aims to group variations of a word together, simplifying analysis and information retrieval. For example, "running" becomes "run."
* Lemmatization: Similar to Stemming but considers the word's context to reduce it to its base or dictionary form (lemma). Results in more accurate representation of the base form of a word, addressing potential ambiguities introduced by stemming.
* Translation: Convert text from one language to another. Facilitates cross-language communication and analysis, enabling understanding of content in different linguistic contexts.
* Emoji to Text: Translate emojis (emotion icons) into their corresponding textual representation. Helps in extracting meaning from textual data that includes emojis, making it easier for analysis and understanding sentiment.
* Word Analysis: Detecting the normality of the words. To find whether words can be categorized under the following 2 categories - Simple & Complex

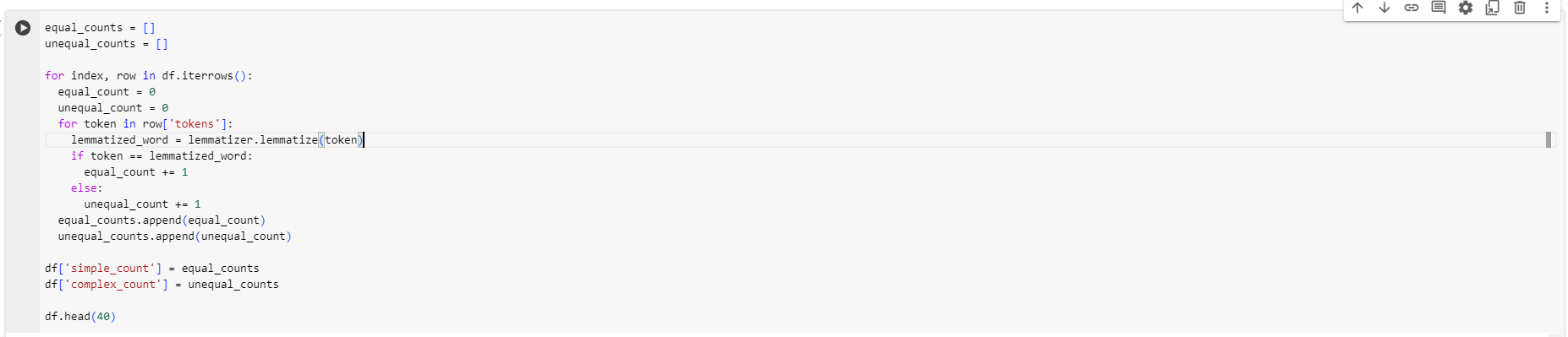
## Source Code:

****

****

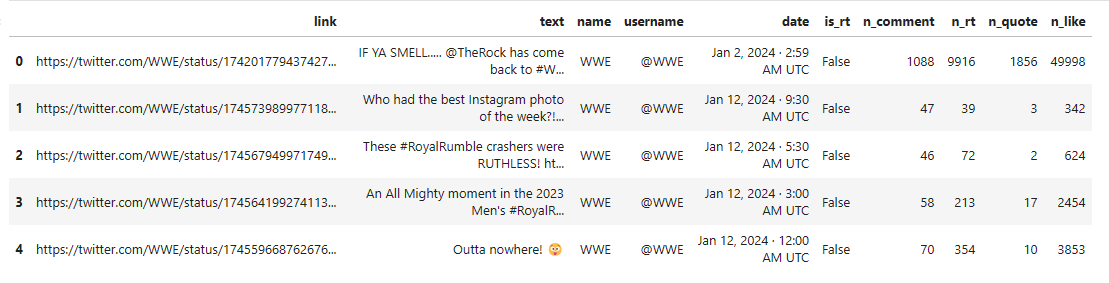
****

****

****

## Output:

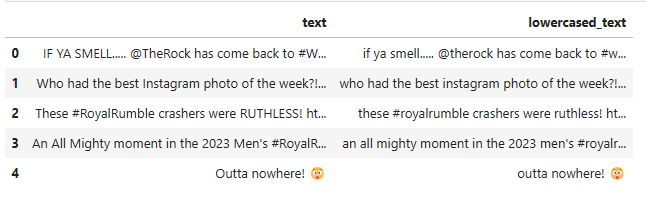
**Twitter Data:**





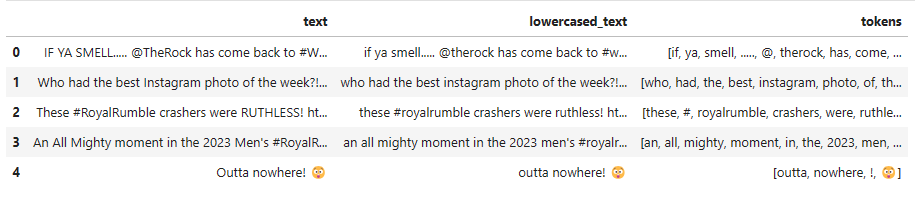
**Perform the following task with using inbuilt Python Libraries:**

**1. Lower Casing**

****

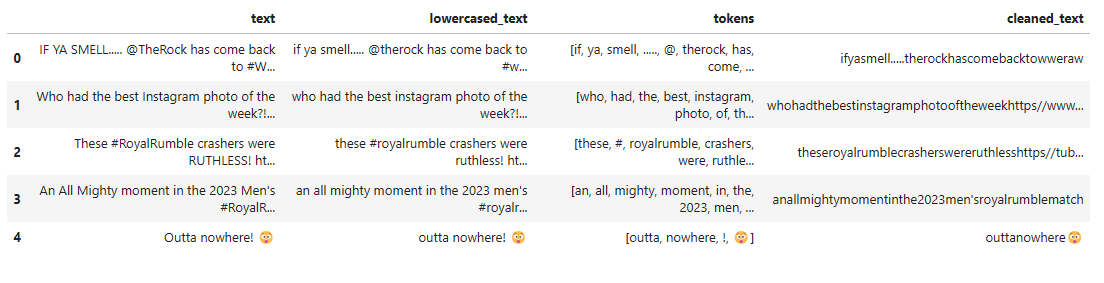


**2. Tokenization**

****

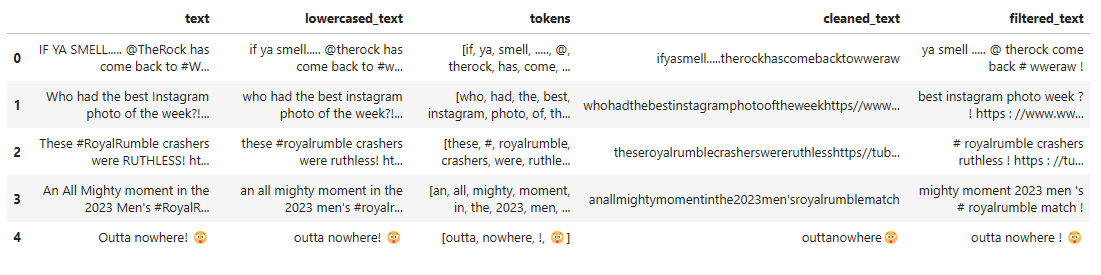


**3. Punctuation Mark Removal**

****

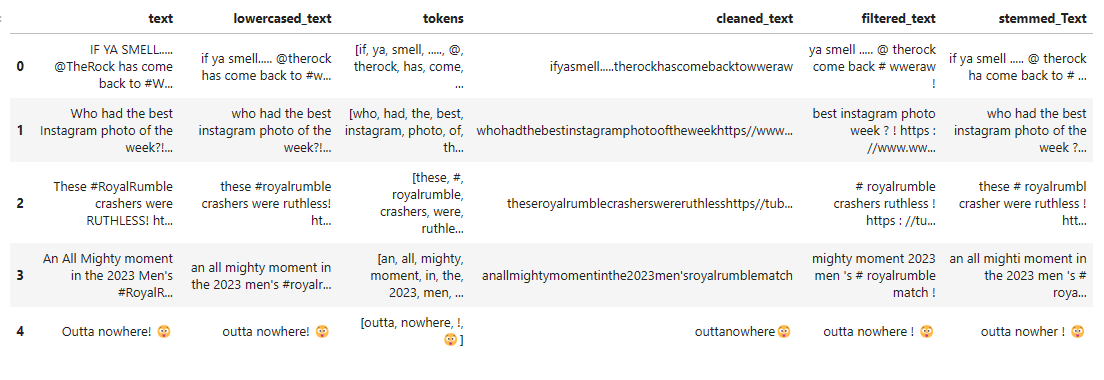


**4. Stop Word Removal**

****

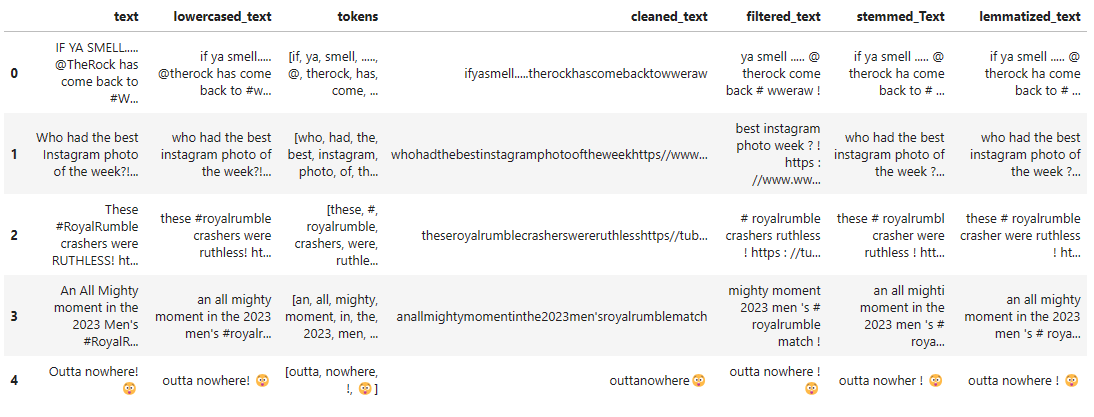


**5. Stemming**

****

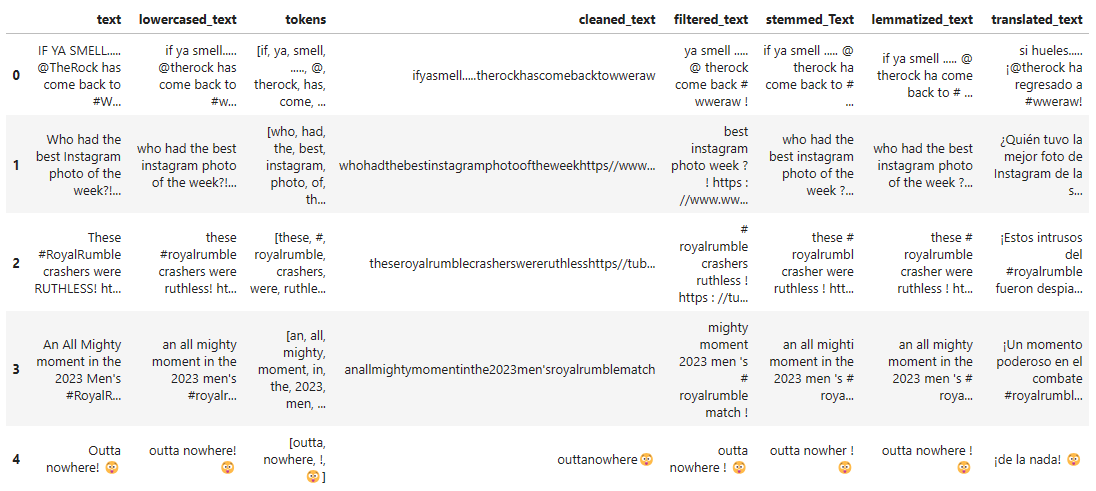


**6. Lemmatization**

****



**7. Translation**

****

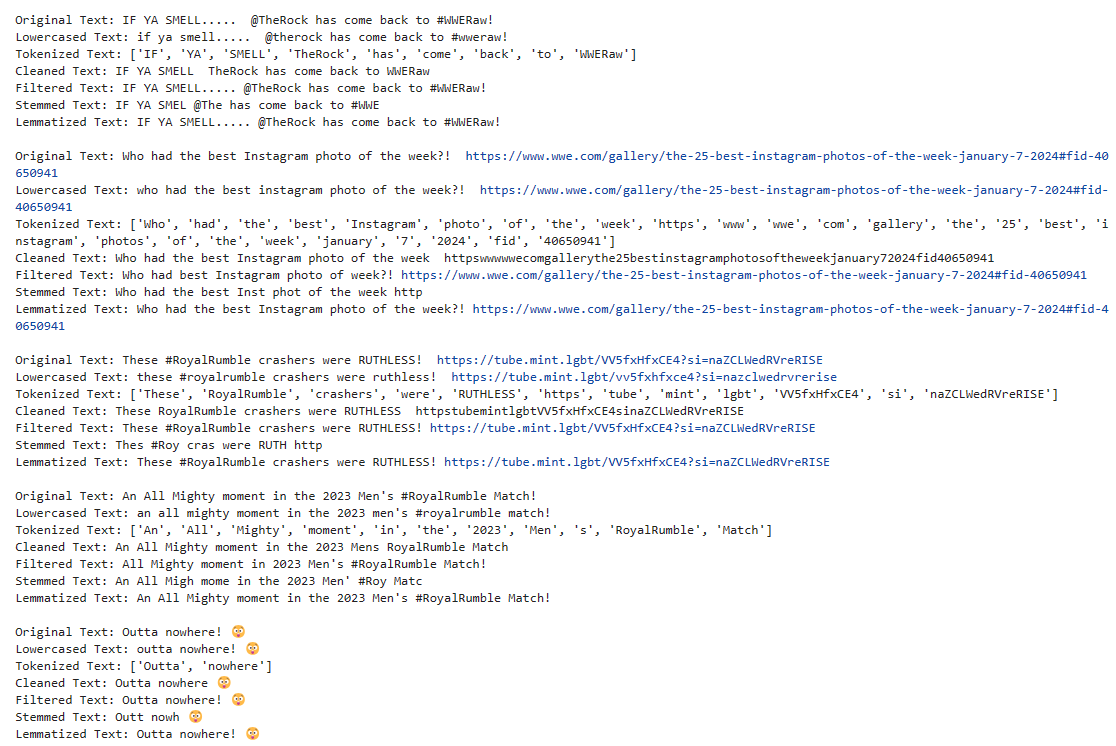


**8. Emoji to text**

****



**Perform the following task without using inbuilt Python Libraries (Wont’t work for last two tasks):**

****



**Word Analysis:  
**

## Conclusion:

* Lowercasing ensures uniformity, treating uppercase and lowercase forms equally, preventing discrepancies in analysis.
* Tokenization breaks down text into meaningful units, enabling granular analysis at the word level and facilitating various natural language processing tasks.
* Punctuation mark removal eliminates non-alphanumeric characters, reducing noise and focusing on the core meaning of the text.
* Stop word removal improves efficiency by excluding common words, allowing a focus on content-bearing words and enhancing the relevance of analysis.
* Stemming and lemmatization contribute to word form normalization, reducing words to their base form for better consistency and information retrieval.
* Translation enables the understanding of text in different languages, fostering cross-language communication and analysis.
* Emoji-to-text conversion aids in extracting emotional context from textual data, contributing to sentiment analysis and understanding user expressions.