S. B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT & RESEARCH, NAGPUR.

Practical No. 2

Aim: To Perform a Text Preprocessing using following steps

1. Noise Removal 2. Lexicon Normalization

Name of Student: _____

Roll No.: _____

Semester/Year: IV/VII

Academic Session: 2025-2026

Date of Performance: _____

Date of Submission: _____

AIM: To Perform a Text Preprocessing using following steps 1. Noise Removal 2. Lexicon Normalization 3. Object Standardization

OBJECTIVE/EXPECTED LEARNING OUTCOME:

The objectives and expected learning outcome of this practical are:

- Able to understand a Text Preprocessing

HARDWARE AND SOFTWARE REQUIRMENTS:

Hardware Requirement:

Software Requirement:

THEORY:

Any piece of text which is not relevant to the context of the data and the end-output can be specified as the noise. A general approach for noise removal is to prepare a dictionary of noisy entities, and iterate the text object by tokens (or by words), eliminating those tokens which are present in the noise dictionary.

Normalization is a pivotal step for feature engineering with text as it converts the high dimensional features (N different features) to the low dimensional space (1 feature), which is an ideal ask for any ML model.

The most common lexicon normalization practices are :

Stemming: Stemming is a rudimentary rule-based process of stripping the suffixes (“ing”, “ly”, “es”, “s” etc) from a word.

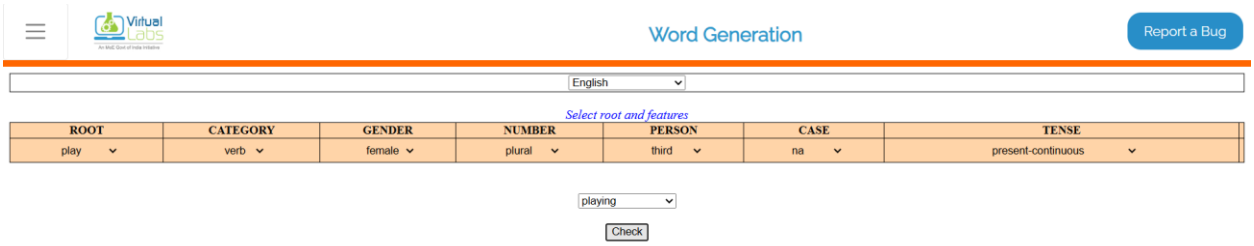
Lemmatization: Lemmatization, on the other hand, is an organized & step by step procedure of obtaining the root form ofthe word, it makes use of vocabulary (dictionary importance of words) and morphological analysis (word structure and grammar relations)

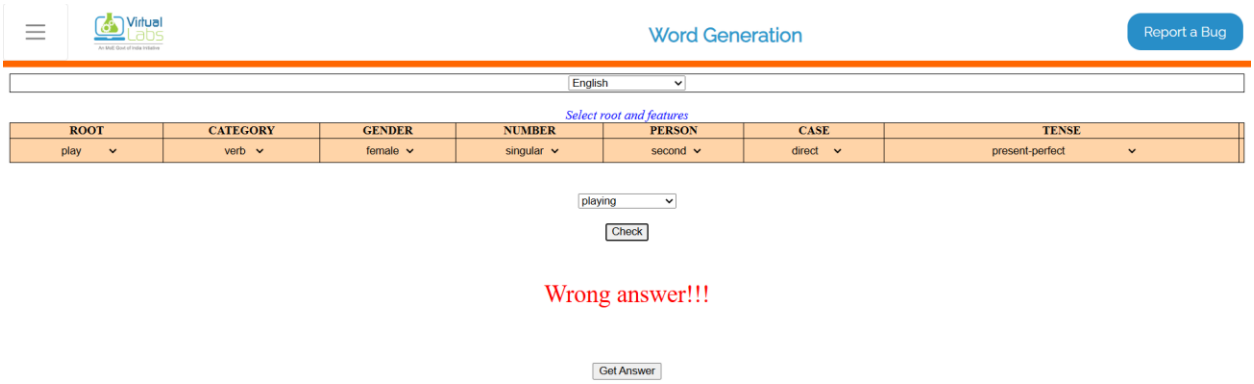
CODE:

```
import nltk
from nltk.stem import PorterStemmer

# Download the 'punkt' tokenizer data (if not already downloaded)
nltk.download('punkt')
nltk.download('punkt_tab') # Added download for punkt_tab

# Create a Porter Stemmer instance
stemmer = PorterStemmer()

# Get input from the user
user_input = input("Enter a word or sentence to stem: ")

# Tokenize the input into words
words = nltk.word_tokenize(user_input)

print("\nOriginal Word → Stemmed Word")
# Stem each word and print the result
for word in words:
    root = stemmer.stem(word)
    print(f"{word:12} → {root}")
```

OUTPUT (SCREENSHOT):

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
Enter a word or sentence to stem: Cared

Original Word → Stemmed Word
Cared        → care
```





CONCLUSION:

DISCUSSION AND VIVA VOCE:

- What is Text Preprocessing
- Define Stemming
- Explain Lemmatization with example

REFERENCE:

- *www.w3schools.com*
- *www.tutorialsmade.com*
- *https://www.javatpoint.com/*