Experiment No. 3
Apply Various text preprocessing techniques Lemmatization / Stemming.
Date of Performance:
Date of Submission:

Exp. No.: 3

Title: Apply Various text preprocessing techniques Lemmatization and Stemming.

Theory:

Lemmatization and Stemming are key text preprocessing techniques in Natural Language Processing (NLP) used to reduce words to their base or root form.

1. Stemming

- **Concept**: Stemming is the process of reducing a word to its root or base form by chopping off prefixes or suffixes. The resulting stem may not always be a valid word but is typically shorter.
- **Method**: Stemming uses simple rules or heuristics to remove affixes (e.g., "ing", "ed", "s").
- **Output**: Stems may not be linguistically accurate, e.g., "running" becomes "run", "better" becomes "bett".

Example:

Word	Stemmed Form
running	run
studies	studi
better	bett

Common Algorithms:

- **Porter Stemmer**: One of the oldest and most popular stemming algorithms.
- **Snowball Stemmer**: A more advanced version of the Porter stemmer.

2. Lemmatization

- **Concept**: Lemmatization reduces words to their base or dictionary form (known as the "lemma") by considering the word's context and part of speech (POS). Unlike stemming, it produces valid root words.
- **Method**: Lemmatization uses a vocabulary and morphological analysis of words to identify their lemma, factoring in tense, plurality, and word usage.
- **Output**: The result is a linguistically valid word, e.g., "better" becomes "good", "running" becomes "run".

Example:

Word	Lemmatized Form
running	run
studies	study
better	good

- Lemmatization depends on **POS tagging** to achieve accurate reductions, as the lemma of a word can vary depending on its role in a sentence.

Key Differences:

- **Accuracy**: Lemmatization provides more accurate, dictionary-based root words, while stemming may result in non-dictionary words.
- **Processing**: Stemming is faster but less precise, whereas lemmatization requires more processing time and contextual analysis.

When to Use:

- **Stemming**: Useful when speed is more critical than accuracy, such as in quick searches or very large datasets.
- **Lemmatization**: Preferred when semantic understanding and linguistic accuracy are needed, such as in text classification, sentiment analysis, or machine translation.

Code:

```
from nltk.stem import PorterStemmer porter = PorterStemmer() print(porter.stem("play"))
print(porter.stem("playing")) print(porter.stem("plays")) print(porter.stem("played"))
print(porter.stem("hiked"))
play
play
play
play
hike
```

```
from nltk.stem import PorterStemmer porter = PorterStemmer() print(porter.stem("Communication"))
```

```
commun
```

```
import nltk nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
True
```

```
from nltk.stem import WordNetLemmatizer lemmatizer = WordNetLemmatizer() print(lemmatizer.lemmatize("plays", 'v')) print(lemmatizer.lemmatize("played", 'v')) print(lemmatizer.lemmatize("play", 'v')) print(lemmatizer.lemmatize("playing", 'v')) print(lemmatizer.lemmatize("better", 'a')) print(lemmatizer.lemmatize("knew", 'v'))
```

```
play
play
play
play
good
Know
```

```
from nltk.stem import WordNetLemmatizer lemmatizer = WordNetLemmatizer() print(lemmatizer.lemmatize("Communication", 'v'))
```

```
Communication
```

Conclusion:

In conclusion, both stemming and lemmatization are essential text prepossessing techniques in NLP for reducing words to their root forms. Stemming relies on heuristic rules to remove affixes and quickly generate shorter word stems, though these may not always be valid words. Lemmatization, on the other hand, uses linguistic knowledge and part-of-speech tagging to generate accurate base forms of words. While stemming is faster and useful for large datasets with minimal need for precision, lemmatization is preferred when higher accuracy and semantic meaning are crucial, making it ideal for tasks like sentiment analysis and text classification.