**EXPERIMENT NO.4**

**AIM:**

Lemmatization

**RESOURCES REQUIRED:**

Python 3, NLTK toolkit, Text editor, 4 GB RAM and above, i5 processor and above

**THEORY:**

**Lemmatization:**

Lemmatization is the process of grouping together the different inflected forms of a word so they can be analysed as a single item. Lemmatization is similar to stemming but it brings context to the words. So it links words with similar meaning to one word.

Text pre-processing includes both Stemming as well as Lemmatization. Many times people find these two terms confusing. Some treat these two as same. Actually, lemmatization is preferred over Stemming because lemmatization does morphological analysis of the words.

Applications of lemmatization are:

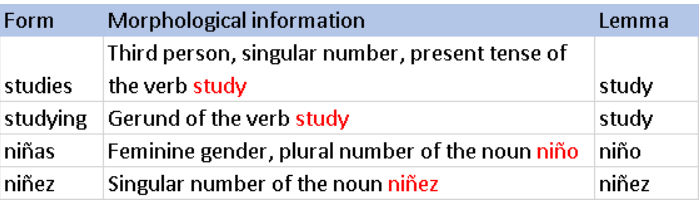
* Used in comprehensive retrieval systems like search engines.
* Used in compact indexing

Examples of lemmatization:

rocks : rock

corpora : corpus

better : good



**CONCLUSION:**

Lemmatization is a basic text pre-processing operation in many natural language processing tasks. It is similar to stemming but unlike stemming, it does not truncate any affixes from the morpheme but rather reduces the inflected form to its actual stem. Therefore, lemmatization provides a better result when compared to stemming.

**CODE:**

from random import choice

from nltk import word\_tokenize

from nltk.corpus import brown

from nltk.stem import WordNetLemmatizer

samples = choice(brown.paras(categories="fiction"))

corpus = " ".join([" ".join(sample) for sample in samples])

print(f"Original corpus :\n{corpus}\n")

tokens = word\_tokenize(corpus)

print(f"Tokenized words : \n{tokens}\n")

lemma = WordNetLemmatizer()

lem\_words = [lemma.lemmatize(token) for token in tokens]

print(f"Lemmatized words :\n{lem\_words}")

**OUTPUT:**

****