**EXPERIMENT NO.1**

**AIM:** Tokenization of text

**RESOURCES REQUIRED:**

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

**THEORY:**

**TOKENIZATION:**

Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens , perhaps at the same time throwing away certain characters, such as punctuation. Here is an example of tokenization:

Input: Friends, Romans, Countrymen, lend me your ears;

Output: 

These tokens are often loosely referred to as terms or words, but it is sometimes important to make a type/token distinction. A token is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing. A type is the class of all tokens containing the same character sequence. A term is a (perhaps normalized) type that is included in the IR system's dictionary. The set of index terms could be entirely distinct from the tokens, for instance, they could be semantic identifiers in a taxonomy, but in practice in modern IR systems they are strongly related to the tokens in the document. However, rather than being exactly the tokens that appear in the document, they are usually derived from them by various normalization processes.

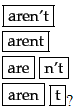
The major question of the tokenization phase is what are the correct tokens to use? In this example, it looks fairly trivial: you chop on whitespace and throw away punctuation characters. This is a starting point, but even for English there are a number of tricky cases. For example, what do you do about the various uses of the apostrophe for possession and contractions?

Mr. O'Neill thinks that the boys' stories about Chile's capital aren't amusing.

For O'Neill, which of the following is the desired tokenization?



And for aren't, is it:



CHALLENGES IN TOKENIZATION

Challenges in tokenization depends on the type of language. Languages such as English and French are referred to as space-delimited as most of the words are separated from each other by white spaces. Languages such as Chinese and Thai are referred to as unsegmented as words do not have clear boundaries. Tokenising unsegmented language sentences requires additional lexical and morphological information. Tokenization is also affected by writing system and the typographical structure of the words. Structures of languges can be grouped into three categories:

Isolating: Words do not divide into smaller units. Example: Mandarin Chinese

Agglutinative: Words divide into smaller units. Example: Japanese, Tamil

Inflectional: Boundaries between morphemes are not clear and ambiguous in terms of grammatical meaning. Example: Latin.

**CONCLUSION:**

The process of segmenting running text into words and sentences is called tokenization.Tokenization is a basic pre-processing step in every NLP task. There are two types of tokenization, sentence and word tokenization. Tokenization has been performed on a simple text corpus.

**CODE:**

import nltk

from nltk.corpus import \*

from random import choice

samples = choice(brown.paras(categories="science\_fiction"))

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

tokenized = nltk.word\_tokenize(sample)

print(f"Original Corpus:\n{sample}\n\nTokenized Corpus:\n{tokenized}")

**OUTPUT:**

