Natural Language Toolkit (NLTK)

PREPARED BY:

ENG. FATMA

What is NLTK ?

- NLTK (Natural Language Toolkit 2022) is one of the earliest Python-based NLP development tool,

Kindly check the following link for nltk documentation:

https://www.nltk.org/

What is NLTK ?

- NLTK contains statistical-based text processing libraries of five fundamental NLP enabling technologies and basic semantic reasoning tools including:
  - Word tokenization
  - Stemming
  - POS tagging
  - Text classification
  - Semantic analysis

NLTK installation

- NLTK requires Python versions 3.7, 3.8, 3.9, 3.10 or 3.11.

```
pip install nltk
```

NLTK installation

- Installing NLTK Data:

```
>>> import nltk
>>> nltk.download()
```

What is Tokenization?

- Tokenization is the process of breaking down a text into smaller units called tokens.
- These tokens can be words, phrases, symbols, or other meaningful elements depending on the context of the text and the requirements of the task at hand.

What is Tokenization?

- Tokenization is a fundamental step in natural language processing (NLP) tasks because it allows the computer to understand and process textual data more effectively.

Tokenization with NLTK

1- Word Tokenization:

```
# Create utterance 3 (utt3) and performs tokenization
utt3 = 'Jane lent $100 to Peter early this morning.'
wtokens = nltk.word_tokenize(utt3)
wtokens
```

```
['Jane', 'lent', '$', '100', 'to', 'Peter', 'early', 'this', 'morning', '.']
```

Tokenization with NLTK

- ***Different Between Tokenize() vs Split()***
- Python provides *split()* function to split a sentence of text into words

```
# Use split() to perform word tokenization
words = utt3.split()
words
```

```
['Jane', 'lent', '$100', 'to', 'Peter', 'early', 'this', 'morning.']
```

Tokenization with NLTK

2- sentence Tokenization:

```
import nltk

from nltk.tokenize import sent_tokenize


# Sample text

text = "NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum."


# Perform sentence tokenization

sentences = sent_tokenize(text)


# Print the tokenized sentences

for sentence in sentences:

    print(sentence)
```