Практична робота № 7

Тема: Бібліотека NLTK мови Python**.** Токенізація текстів.

Мета: Відпрацювати практичні навички написання програм на мові Python для обробки текстової інформації використовуючи методи бібліотеки NLTK.

Література: https://coderlessons.com/tutorials/mashinnoe-obuchenie/uchebnik-nltk/uchebnik-nltk

Зміст роботи:

Завдання 1.

Дано текстовий рядок (не менше 10 речень). Виконати токенізацію за реченнями та словами, використовуючи інструменти бібліотеки NLTK. Зі списку слів вилучити стоп слова. Визначте частини мови слів автоматично та проаналізуйте на правильність. Результати кожної дії вивести на екран.

Завдання 2.

Виконати операції стемінгу та лематизації використовуючи результат попереднього завдання як вхідні дані. Результати виконаних операцій порівняти, висновки записати у звіт.

Методичні рекомендації.

text = "Hello there! Welcome to this tutorial on tokenizing. After going through this tutorial you will be able to tokenize your text. Tokenizing is an important concept under NLP. Happy learning!"

Для токенізації відповідно до речень використовуйте:

print(sent_tokenize(text))

Результат, який ми отримуємо, такий:

['Hello there!', 'Welcome to this tutorial on tokenizing.', 'After going through this tutorial you will be able to tokenize your text.', 'Tokenizing is important concept under NLP.', 'Happy learning!']

!!!!!Він повертає список з кожним елементом у вигляді пропозиції з тексту.

Для токенізації відповідно до слів, які ми використовуємо:

print(word_tokenize(text))

Результат, який ми отримуємо, такий:

['Hello', 'there', '!', 'Welcome', 'to', 'this', 'tutorial', 'on', 'tokenizing', '.', 'After', 'going', ' through', 'this', 'tutorial', 'you', 'will', 'be', 'able', 'to', 'tokenize', 'your', 'text', '.', 'Tokenizing' , 'is', 'an', 'important', 'conceot', 'under', 'NLP', '.', 'Happy', 'learning', '!']

!!!!!Він повертає список з кожним елементом у вигляді слова з тексту. Тепер вони можуть використовуватися як токени в мовній моделі для навчання.

Повний код:

```
import nltk
from nltk.tokenize import sent_tokenize, word_tokenize

text = "Hello there! Welcome to this tutorial on tokenizing. After going through this tutorial you will be able to tokenize your text. Tokenizing is an important concept under NLP. Happy learning!"

print(sent_tokenize(text))

print(word_tokenize(text))
```

Спочатку ми імпортуємо PorterStemmernltk.stem.porter Далі ми ініціалізуємо stemmer для змінної stemmer.stem().

```
from nltk.stem.porter import PorterStemmer

stemmer = PorterStemmer()

print(stemmer.stem("going"))
```

Лематизація нормалізує слово на основі контексту та словникового запасу тексту. У NLTK ви можете лематизувати пропозиції за допомогою класу WordNetLemmatizer

По-перше, потрібно завантажити ресурс wordnet

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

Як тільки він завантажений, потрібно імпортувати клас WordNetLemmatizer

```
from nltk.stem.wordnet import WordNetLemmatizer

lem = WordNetLemmatizer()
```

Щоб використовувати лематизатор, використовуйте метод .lemmatize() Потрібно два аргументи – слово та контекст. Наприклад «v» для контексту.

Контрольні запитання.

1. Поясніть призначення бібліотеки NLTK.
2. Назвіть методи токенізації текстів за словами, за реченнями?
3. В чому полягає призначення токенізації?
4. В якому вигляді подається результат токенізації?
5. В чому полягає суть стоп-слів?
6. Для чого прибирати стоп-слова з тексту?
7. Як автоматично визначити частини мови слів?
8. Як отримати доступ до методів NLTK ?
9. Що таке стемінг?
10. Що таке лемітизація?
11. Що є результатом виконання стемінгу та лемітизації.
12.