Лабораторна робота №2, Обробка та аналіз текстових даних на Python, Варіант 14

Виконав: студент групи ІП-11, Лошак Віктор Іванович
Перевірив: Юлія Тимофєєва Сергіївна

Тема роботи: Попередня обробка тексту за допомогою NLTK
Мета роботи: Ознайомитись з основними операціями з попередньої обробки тексту та їх реалізацією у бібліотеці NLTK.

15.03.2024

Завдання:

1. Зчитати файл text4.

а) Порахувати кількість речень в тексті;
б) вивести 10 слів, які зустрічаються найчастіше;
в) провести лематизацію слів другого речення, попередньо визначивши частини мови. 2. Використати корпус Brown, сьомий текст категорії adventure.
а) Видалити стоп-слова;
б) Вивести 8 іменників, що зустрічаються найчастіше.

Task:

1. Read the file text4.

a) Count the number of sentences in the text;
b) display 10 words that occur most often;
c) perform lemmatization of the words of the second sentence, having previously determined the parts of speech. 2. Use the Brown corpus, the seventh text of the category adventure.
a) Remove stop words;
b) List 8 nouns that occur most often.

In [ ]:
```
import nltk
```

Task 1

In [ ]:
```
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.probability import FreqDist
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk import pos_tag
from nltk.corpus import wordnet as wn

with open('text4.txt', 'r', encoding='utf-8') as file:
    text = file.read()

text
```

Out[ ]:
```
'“Then we had a talk as to what we should do, and Frank was all for\nopenness,
but I was so ashamed of it all that I felt as if I should\nlike to vanish away
and never see any of them again—just sending\na line to pa, perhaps, to show hi
m that I was alive. It was awful\nto me to think of all those lords and ladies
sitting round that\nbreakfast-table and waiting for me to come back. So Frank t
ook my\nwedding-clothes and things and made a bundle of them, so that I should
\nnot be traced, and dropped them away somewhere where no one could find\nthem.
It is likely that we should have gone on to Paris to-morrow, only\nthat this go
od gentleman, Mr. Holmes, came round to us this evening,\nthough how he found u
s is more than I can think, and he showed us very\nclearly and kindly that I wa
s wrong and that Frank was right, and that\nwe should be putting ourselves in t
he wrong if we were so secret. Then\nhe offered to give us a chance of talking
to Lord St. Simon alone, and\nso we came right away round to his rooms at once.
Now, Robert, you have\nheard it all, and I am very sorry if I have given you pa
in, and I hope\nthat you do not think very meanly of me.”\n\nLord St. Simon had
by no means relaxed his rigid attitude, but had\nlistened with a frowning brow
and a compressed lip to this long\nnarrative.\n'
```

In [ ]:
```python
# a) Count the number of sentences
sentences = sent_tokenize(text)
num_sentences = len(sentences)
print(f"Number of sentences: {num_sentences}")
```

```
Number of sentences: 6
```

In [ ]:
```python
# b) Display 10 words that occur most often
words = word_tokenize(text.lower())
filtered_words = [word for word in words if word.isalnum()]
fdist = FreqDist(filtered_words)
most_common_words = fdist.most_common(10)
print("10 most common words:", most_common_words)
```

```
10 most common words: [('and', 15), ('to', 13), ('i', 10), ('that', 10), ('we',
6), ('a', 6), ('was', 6), ('of', 6), ('should', 5), ('so', 5)]
```

In [ ]:
```python
# c) Lemmatization of the words in the second sentence
lemmatizer = WordNetLemmatizer()

second_sentence = sentences[1]
second_sentence_tokens = word_tokenize(second_sentence)
tagged_tokens = pos_tag(second_sentence_tokens)
tagged_tokens[:5]
```

Out[ ]:
```
[('It', 'PRP'), ('was', 'VBD'), ('awful', 'JJ'), ('to', 'TO'), ('me', 'PRP')]
```

На жаль pos_tag використовує систему тегування що не співпадає з тією яку використовує WordNetLemmatizer. Щоб виправити це використаємо функцію `get_wordnet_pos` для конвертації

In [ ]:
```python
tagged_tokens[0][0], tagged_tokens[0][1], wn.NOUN
```

Out[ ]:
```
('It', 'PRP', 'n')
```

In [ ]:
```python
def get_wordnet_pos(treebank_tag):
    if treebank_tag.startswith('J'):
        return wn.ADJ
    elif treebank_tag.startswith('V'):
        return wn.VERB
    elif treebank_tag.startswith('N'):
```

```
        return wn.NOUN
    elif treebank_tag.startswith('R'):
        return wn.ADV
    else:
        return wn.NOUN
```

In [ ]:
```
lemmatized_words = [lemmatizer.lemmatize(word, get_wordnet_pos(tag)) for word, t
print("Second sentence:", second_sentence)
print("Lemmatized words of the second sentence:", lemmatized_words)
```

```
Second sentence: It was awful
to me to think of all those lords and ladies sitting round that
breakfast-table and waiting for me to come back.
Lemmatized words of the second sentence: ['It', 'be', 'awful', 'to', 'me', 'to',
'think', 'of', 'all', 'those', 'lord', 'and', 'lady', 'sit', 'round', 'that', 'br
eakfast-table', 'and', 'wait', 'for', 'me', 'to', 'come', 'back', '.']
```

Task 2

In [ ]:
```
from nltk.corpus import brown
from nltk.corpus import stopwords
from nltk.probability import FreqDist
from nltk import pos_tag

# a) Extract words from the seventh text of the 'adventure' category
adventure_texts = brown.fileids(categories='adventure')
seventh_text_id = adventure_texts[6]
words = brown.words(fileids=seventh_text_id)
' '.join(words[:10]), len(words)
```

Out[ ]:
```
('The flat , hard cap was small , but he', 2403)
```

In [ ]:
```
filtered_words = [word.lower() for word in words if word.lower() not in stopword
' '.join(filtered_words[:10])
filtered_words[:10]
```

Out[ ]:
```
['flat',
 'hard',
 'cap',
 'small',
 'thrust',
 'back',
 'head',
 'tie',
 'hell',
 'could']
```

In [ ]:
```
# b) Count and list 8 most common nouns
fdist = FreqDist(filtered_words)
most_common_nouns = [(word, freq) for word, freq in fdist.most_common() if pos_t
print("8 most common nouns:", most_common_nouns)
```

```
8 most common nouns: [('barton', 25), ('man', 11), ('dill', 11), ('hague', 9),
('rankin', 8), ('night', 7), ('valley', 7), ('kodyke', 7)]
```

Висновок:

В ході виконання даної лабораторної роботи я ознайомився з основними методами попередньої обробки тексту та їх реалізацією за допомогою бібліотеки NLTK в

Python. Лабораторна робота дала мені змогу практично застосувати процеси, такі як токенізація, видалення стоп-слів, лематизація та частотний аналіз слів, до реальних текстових даних.

Я працював з корпусом Brown, конкретно з сьомим текстом категорії "adventure", що надало мені практичний досвід роботи з реальними текстовими корпусами та допомогло усвідомити важливість видалення стоп-слів для поліпшення якості подальшого аналізу.

Лабораторна робота допомогла мені зрозуміти, як здійснюється підготовка тексту до аналізу та як можна ефективно використовувати бібліотеку NLTK для обробки та аналізу текстових даних в Python. Знання та навички, отримані в ході виконання цієї роботи, будуть корисні для майбутніх проектів, пов'язаних з обробкою та аналізом текстових даних.