05 Обробка текстових даних засобами пакету NLTK

Natural Language Toolkit (NLTK) є однією з найвідоміших бібліотек мови програмування Python, яка призначена для обробки природної мови (Natural Language Processing, NLP). Вона широко використовується як у наукових дослідженнях, так і в практичних застосуваннях, забезпечуючи різноманітні інструменти для аналізу, обробки, анотації та моделювання текстових даних.

Однією з базових можливостей NLTK є токенізація тексту. Токенізація полягає у розділенні тексту на окремі елементи, такі як слова або речення. Функції `word_tokenize()` і `sent_tokenize()` дозволяють розбити вхідний текст відповідно на слова та речення, що є першим кроком у більшості процесів обробки мови.

Наступним важливим етапом є нормалізація тексту, яка включає приведення слів до нижнього регістру, видалення розділових знаків та стоп-слів. Стоп-слова, наприклад, "і", "або", "але", не несуть суттєвого смислового навантаження і часто видаляються зі статистичного аналізу. NLTK надає вбудовані списки стоп-слів для різних мов, що полегшує цей процес.

е однією важливою складовою обробки тексту є лематизація та стемінг. Стемінг (через алгоритм Портера або Ланкастера) дозволяє звести слова до їх кореневої форми, що корисно для зменшення варіативності текстових даних. Лематизація є більш точною альтернативою стемінгу і виконується з урахуванням частин мови та словникових форм. В NLTK для цього використовується клас `WordNetLemmatizer`.

Бібліотека також підтримує теги частин мови (Part-of-Speech Tagging). Метод `pos_tag()` дозволяє здійснювати синтаксичну анотацію слів, що є основою для глибшого морфологічного аналізу текстів. Частини мови допомагають у виявленні структури речень, семантичного аналізу та побудові залежностей між словами.

Окрім того, NLTK містить засоби для побудови N-грам — послідовностей із n слів, що дозволяє моделювати контекстні залежності між словами. Цей інструмент є корисним у статистичному моделюванні мови та побудові мовних моделей.

Для виконання більш складного аналізу NLTK пропонує інструменти для розпізнавання іменованих сутностей (Named Entity Recognition, NER), аналізу залежностей між словами, а також побудови контекстно-вільних граматик (CFG) та парсингу речень.

Крім того, NLTK містить велику кількість корпусів та лексичних ресурсів, зокрема WordNet — англомовну онтологію, яка використовується для пошуку синонімів, антонімів, гіперонімів та інших семантичних зв’язків між словами.

Таким чином, NLTK забезпечує всебічну підтримку для різноманітних задач обробки природної мови. Завдяки своїй модульності, доступності та освітній спрямованості, цей

пакет є незамінним інструментом як для початківців, так і для досвідчених дослідників у сфері комп'ютерної лінгвістики та машинного навчання.

Детальне вивчення методів обробки текстових даних заслуговує окремої дисципліни, тому нижче наведено тільки прості приклади, які дозволяють отримати уявлення про предмет.

1. Встановлення та імпорт NLTK

Перед використанням потрібно встановити бібліотеку:

```
pip install nltk
```

Потім імпортуються необхідні модулі:

```
import nltk
nltk.download('punkt')  # Завантажуємо токенізатор
```

2. Токенізація (Tokenization)

Токенізація – це процес розбиття тексту на менші частини (слова або речення).

📌 Токенізація речень

```
import nltk
nltk.download('punkt_tab')

from nltk.tokenize import sent_tokenize
text = "NLTK є потужною бібліотекою. Вона використовується для NLP!"
sentences = sent_tokenize(text, language='russian')
print(sentences)
```

✅ `sent_tokenize()` ділить текст на речення. Українська мова підтримується через 'russian', тому що nltk не має окремого токенізатора для української, тому для кириличних мов використовується russian.

📌 Токенізація слів

```
from nltk.tokenize import word_tokenize
words = word_tokenize(text)
print(words)
```

✅ `word_tokenize()` розбиває текст на слова.

3. Видалення стоп-слів (Stop Words Removal)

Стоп-слова – це загальновживані слова (наприклад, "the", "is", "and"), які не несуть особливого сенсу.

```
from nltk.corpus import stopwords
nltk.download('stopwords')

stop_words = set(stopwords.words('english'))

filtered_words = [word for word in words if word.lower() not in
stop_words]
print(filtered_words)
```

✅ **Результат:**

```
['Natural', 'Language', 'Processing', 'exciting', 'field', '.',
'It', 'allows', 'machines', 'understand', 'human', 'language', '.']
```

4. Стемінг (Stemming)

Стемінг – це процес приведення слова до його основної форми (без урахування граматичних змін).

```
from nltk.stem import PorterStemmer
stemmer = PorterStemmer()
stemmed_words = [stemmer.stem(word) for word in filtered_words]
print(stemmed_words)
```

✅ **Результат:**

```
['natur', 'languag', 'process', 'excit', 'field', '.', 'It',
'allow', 'machin', 'understand', 'human', 'languag', '.']
```

📌 Стемінг може змінювати слова неприродним чином (наприклад, `processing` → `process`, `machines` → `machin`).

5. Лематизація (Lemmatization)

Лематизація – це більш точний процес, ніж стемінг, оскільки вона приводить слово до його базової форми з урахуванням граматичних правил.

```
c
nltk.download('wordnet')
lemmatizer = WordNetLemmatizer()
lemmatized_words = [lemmatizer.lemmatize(word) for word in
filtered_words]
print(lemmatized_words)
```

✅ **Результат:**

```
['Natural', 'Language', 'Processing', 'exciting', 'field', '.',
'It', 'allows', 'machine', 'understand', 'human', 'language', '.']
```

📌 Лематизація враховує граматику (наприклад, `machines` → `machine`).

6. Частиномовний аналіз (POS-tagging)

Аналіз частин мови дозволяє визначити, якими частинами мови є слова у реченні.

```
import nltk

# Завантаження необхідних ресурсів
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

# Токенизация
text = nltk.word_tokenize("NLTK is a powerful library. It is used
for NLP!")

# Частиномовна разметка
pos_tags = nltk.pos_tag(text)

print(pos_tags)
```

✅ **Результат:**

```
[('NLTK', 'NNP'), ('is', 'VBZ'), ('a', 'DT'), ('powerful', 'JJ'),
('library', 'NN'), ('.', '.'), ('It', 'PRP'), ('is', 'VBZ'),
('used', 'VBN'), ('for', 'IN'), ('NLP', 'NNP'), ('!', '.')]
```

📌 **Розшифровка міток POS-тегів:**

- NN – іменник
- VB – дієслово
- JJ – прикметник
- PRP – займенник

7. Аналіз частоти слів (Word Frequency Analysis)

Аналіз частоти слів допомагає визначити, які слова зустрічаються найчастіше.

```
from collections import Counter

word_freq = Counter(filtered_words)
print(word_freq.most_common(5))
```

✅ **Результат:**

```
[('language', 2), ('Processing', 1), ('exciting', 1), ('field', 1),
('allows', 1)]
```

📌 Частота слів допомагає знаходити найважливіші терміни у тексті.

8. Визначення синонімів та антонімів (WordNet)

NLTK має доступ до **WordNet** – бази даних лексичних зв'язків.

```
from nltk.corpus import wordnet
synonyms = wordnet.synsets("good")
print(synonyms[0].definition())  # Визначення першого синоніма
print(synonyms[0].examples())  # Приклади використання
```

✅ **Результат:**

```
'Having desirable or positive qualities especially those suitable
for a thing specified'
['a good report card', 'when she was good she was very very good']
```

📌 Можна також отримати антоніми:

```
antonyms = []
```

```
for syn in wordnet.synsets("good"):
    for lemma in syn.lemmas():
        if lemma.antonyms():
            antonyms.append(lemma.antonyms()[0].name())

print(set(antonyms))
```

✅ **Результат:** `{'evil', 'bad'}`

Література для поглибленого вивчення обробки текстових даних засобами пакету NLTK

Фундаментальні основи обробки природної мови (NLP)

1. JURAFSKY, D., & MARTIN, J. H. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (2nd ed.). Prentice Hall. URL: https://web.stanford.edu/~jurafsky/slp3/ (Доступна 3-тя редакція онлайн)
2. GOLDBERG, Y. (2017). *Neural Network Methods for Natural Language Processing*. Morgan & Claypool Publishers. URL: https://www.morganclaypool.com/doi/abs/10.2200/S00762ED1V01Y201703NLP011

Детальне вивчення пакету NLTK та його функціоналу

3. BIRD, S., KLEIN, E., & LOPER, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media. URL: https://www.nltk.org/book/
4. NLTK documentation. (б.д.). Отримано 4 липня 2025 з https://www.nltk.org/api/nltk.html

Лексичні ресурси та словники (WordNet)

5. MILLER, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11), 39–41. URL: https://dl.acm.org/doi/10.1145/218380.218386

Практичні посібники з NLP у Python (додаткові)

6. **BASU, S. (2020). *Natural Language Processing with Python: A Practical Introduction*. Packt Publishing.** URL:

https://www.packtpub.com/product/natural-language-processing-with-python-a-practical-introduction/9781838827668

7. **STEVENS, T. (2017).** ***Natural Language Processing with Python and spaCy: A Practical Guide*****. O'Reilly Media.** URL: https://www.oreilly.com/library/view/natural-language-processing/9781492042739/