Лабораторна робота №7

Знайомство з об’єктами бібліотеки spaCy

Мета роботи: Ознайомитись з вирішенням задач обробки природньої мови за допомогою бібліотеки spaCy.

Короткі теоретичні відомості

spaCy — бібліотека Python з відкритим кодом для сучасної обробки природньої мови. spaCy постачається з попередньо підготовленими мовними моделями та векторами слів для понад 60 мов.

Коли викликається nlp до тексту, spaCy застосовує деякі етапи обробки. Першим кроком є токенізація для створення об’єкта Doc. Потім об’єкт Doc обробляється далі за допомогою тегера, синтаксичного аналізатора та засобу розпізнавання сутностей.

nlp = spacy.load("en_core_web_sm")
doc = nlp(text)
print ([token.text for token in doc])

Найновіші версії spaCy, після версії 3.4, мають моделі для української мови.

nlp1 = spacy.load("uk_core_news_sm")

Також можна виділити окремі речення:

```
for sent in doc.sents:
   print(sent.text)
```

Для кожного токену можна вивести леми:

```
for token in doc1:
   print(token.text, token.lemma_)
```

За допомогою атрибуту text можна вивести сам текст до обробки:

doc1.text

Окрім того, можна вивести речення

sentences = list(doc.sents)

Сутності:

doc.ents

Також можна перетворити об’єкт doc в json-формат

doc.to_json()

Для об’єкту токен також можна вивести власне токен:

token.text

Його ідентифікатор

token.i

Його індекс

token.idx

Відповідний об’єкт doc

token.doc

Відповідне речення:

token.sent

Та перевірити чи починається це речення з даного токену

token.is_sent_start

Можна перевірити чи є токен стоп-словом:

```
for token in doc:
   print(token, token.is_stop)
```

Розбір залежностей пов’язаний з аналізом структур речень через залежності між лексемами. Синтаксичний аналізатор залежностей позначає синтаксичні зв’язки між лексемами речення та з’єднує синтаксично пов’язані пари лексем. Залежність або відношення залежності - це спрямований зв'язок між двома токенами. spaCy призначає кожному токену мітку залежності, так само як і з іншими лінгвістичними властивостями, такими як лема або тег частини мови. spaCy показує відносини залежності з спрямованими дугами.

Вилучення інформації на основі правил є незамінним для будь-якого конвеєра обробки природньої мови. Певні типи об’єктів, як-от час, дата та номери телефонів, мають різні формати, які можна розпізнати за набором правил, без необхідності навчання статистичних моделей.

Клас Matcher може зіставляти попередньо визначені правила з послідовністю токенів в об’єктах Doc і Span; крім того, правила можуть посилатися на лексему або її лінгвістичні атрибути.

```
import spacy
from spacy.matcher import Matcher
doc = nlp("Good morning, I want to reserve a ticket.")
matcher = Matcher(nlp.vocab)
pattern = [{"LOWER": "good"},{"LOWER": "morning"},{"IS_PUNCT": True}]
matcher.add("morningGreeting", [pattern])
matches = matcher(doc)
for match_id, start, end in matches:
   m_span = doc[start:end]
   print(start, end, m_span.text)
```

ORTH і TEXT подібні до LOWER: вони означають точну відповідність тексту токена, включаючи регістр. LENGTH використовується для визначення довжини токена. Наступний блок атрибутів токенів — IS_ALPHA, IS_ASCII і IS_DIGIT. Ці функції зручні для пошуку символів чисел і звичайних слів. IS_PUNCT, IS_SPACE і IS_STOP зазвичай використовуються в шаблонах, які включають деякі допоміжні токени та відповідають токенам пунктуації, пробілу та стоп-слова. IS_SENT_START є ще одним корисним атрибутом; він відповідає лексемам початку речення. LIKE_NUM, LIKE_URL і LIKE_EMAIL – це атрибути, які відповідають токенам, які виглядають як числа, URL-адреси та електронні адреси.

У будь-якому текстовому документі є певні терміни, які представляють сутності, які є більш інформативними та мають унікальний контекст порівняно з рештою тексту. Ці сутності відомі як іменовані сутності, і вони більш конкретно

представляють об’єкти реального світу, такі як люди, місця, організації тощо, які зазвичай позначаються власними іменами.

```
import spacy
nlp = spacy.load("en_core_web_sm")
text_nlp = nlp(text)
ner_tagged = [(word.text, word.ent_type_) for word in
text_nlp]
print(ner_tagged)
from spacy import displacy
displacy.render(text_nlp, style='ent', jupyter=True)
```

Завдання до лабораторної роботи

Створити програму, яка:

1. Виконує завдання № 2 лабораторної роботи №1 за допомогою класу Matcher.
2. Виконує завдання відповідно до варіанту засобами бібліотеки spaCy.

Варіант 1.

Файл lab7-1.txt. а) Знайти та вивести стоп-слова, які присутні у тексті. б) Знайти та вивести всі іменники, які присутні у тексті. в) Знайти та вивести числа і дати, які присутні у тексті.

Варіант 2.

Файл lab7-4.txt. а) Знайти та вивести іменники жіночого роду, які присутні у тексті. б) Вивести леми слів першого речення. в) Знайти та вивести організації, які присутні у тексті.

Варіант 3.

Файл lab7-3.txt. а) Знайти та вивести всі слова з тексту, які не є стоп-словами. б) Знайти та вивести всі дієслова, які присутні у тексті. в) Знайти та вивести числа та особи, які присутні у тексті.

Варіант 4.

Файл lab7-5.txt. а) Знайти та вивести дієслова, які присутні у тексті. б) Вивести леми слів останнього речення. в) Знайти та вивести осіб, які присутні у тексті.

Варіант 5.

Файл lab7-2.txt. а) Знайти та вивести всі слова з тексту, які не є стоп-словами. б) Знайти та вивести всі прикметники, які присутні у тексті. в) Знайти та вивести числа та особи, які присутні у тексті.

Варіант 6.

Файл lab7-1.txt. а) Знайти та вивести стоп-слова, які присутні у тексті. б) Знайти та вивести всі дієслова, які присутні у тексті. в) Знайти та вивести країни та особи, які присутні у тексті.

Варіант 7.

Файл lab7-5.txt. а) Знайти та вивести іменники жіночого роду, які присутні у тексті. б) Вивести леми слів передостаннього речення. в) Знайти та вивести місця, які присутні у тексті.

Варіант 8.

Файл lab7-2.txt. а) Знайти та вивести стоп-слова, які присутні у тексті. б) Знайти та вивести всі іменники, які присутні у тексті. в) Знайти та вивести особи і дати, які присутні у тексті.

Варіант 9.

Файл lab7-4.txt. а) Знайти та вивести іменники чоловічого роду, які присутні у тексті. б) Вивести леми слів другого речення. в) Знайти та вивести осіб, які присутні у тексті.

Варіант 10.

Файл lab7-3.txt. а) Вивести леми до слів у тексті. б) Знайти та вивести всі прикметники, які присутні у тексті. в) Знайти та вивести організації та дати, які присутні у тексті.

Варіант 11.

Файл lab7-3.txt. а) Знайти та вивести стоп-слова, які присутні у тексті. б) Знайти та вивести всі іменники, які присутні у тексті. в) Знайти та вивести числа і організації, які присутні у тексті.

Варіант 12.

Файл lab7-2.txt. а) Знайти та вивести стоп-слова, які присутні у тексті. б) Знайти та вивести всі дієслова, які присутні у тексті. в) Знайти та вивести організації та локації, які присутні у тексті.

Варіант 13.

Файл lab7-5.txt. а) Знайти та вивести іменники чоловічого роду, які присутні у тексті. б) Вивести леми слів першого речення. в) Знайти та вивести організації, які присутні у тексті.

Варіант 14.

Файл lab7-1.txt. а) Знайти та вивести всі слова з тексту, які не є стоп-словами. б) Знайти та вивести всі прикметники, які присутні у тексті. в) Знайти та вивести організації та дати, які присутні у тексті.

Варіант 15.

Файл lab7-4.txt. а) Знайти та вивести дієслова, які присутні у тексті. б) Вивести леми слів третього речення. в) Знайти та вивести місця, які присутні у тексті.

Оформити звіт. Звіт повинен містити:

— титульний лист;
— код програми;
— результати виконання коду;

Продемонструвати роботу програми та відповісти на питання стосовно теоретичних відомостей та роботи програми.