Лабораторна робота №8, Обробка та аналіз текстових даних на Python, Варіант 14

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

Тема роботи: Синтаксичні залежності у spaCy
Мета роботи: Ознайомитись з використанням класу Matcher. Ознайомитись із синтаксичними залежностями та їх застосуванням для виявлення намірів

25.05.2024

Завдання:

1. Використати клас Matcher для виділення сутностей. Виділити пункти відправлення за допомогою класу Matcher. Виділити висловлювання користувача, де він просить щось змінити (наприклад, Find me something else.), за допомогою шаблонів. Використати файл з 7-ї лабораторної роботи (не обов'язково усі висловлювання). Продемонструвати роботу.
2. Застосувати синтаксичні залежності для визначення намірів. Використати файл з 7-ї лабораторної роботи (не обов'язково усі висловлювання).

Task:

1. Use the Matcher class to select entities. Select departure points using the Matcher class. Highlight user statements where they ask for something to be changed (for example, Find me something else.) using templates. Use the file from the 7th laboratory work (not necessarily all statements). Demonstrate work.
2. Apply syntactic dependencies to determine intentions. Use the file from the 7th laboratory work (not necessarily all statement).

Task 1

In [ ]:
```
import spacy
from spacy.matcher import Matcher
import json

nlp = spacy.load("en_core_web_sm")
```

In [ ]:
```
with open("flights.json") as file:
    data = json.load(file)
    querries = []
    for dialogue in data:
        for turn in dialogue["turns"]:
            if turn["speaker"] == "USER":
                querries.append(turn["utterance"])
```

```
querries[:10]
```

Out[ ]:
```
['I want to find a one way flight.',
 'I am flying to Phoenix, AZ. I want 1 ticket from Seattle, WA.',
 "I want to depart 11th of March. I don't care which airline.",
 'A different airline please, in Premium Economy.',
 'Find me something else. I want to fly with Southwest Airlines. Look for them
from Atlanta.',
 'Alright.',
 'No thanks for your help.',
 'I would like to find a one way flight. I am interested in flights from Vega
s.',
 'I will be heading to Toronto, Canada. I would like to start my travel on the
13th of March.',
 'What is the airport the flight will be landing at?']
```

In [ ]:
```
doc = nlp(' '.join(querries))
```

In [ ]:
```
matcher = Matcher(nlp.vocab)
pattern = [{"LOWER": {"IN": ["from", "departing"]}}, {"IS_ALPHA": True, "IS_STOP
matcher.add("DEPARTURE_CITY", [pattern])

matches = matcher(doc)

results = []
for match_id, start, end in matches:
    span = doc[start:end]
    # print(span.text)
    results.append(span.text)

results[:10]
```

Out[ ]:
```
['from Seattle',
 'from Atlanta',
 'from Vegas',
 'from Los',
 'from Phoenix',
 'from SD',
 'from NYC',
 'from Las',
 'from Chi',
 'from Philadelphia']
```

In [ ]:
```
# Define a pattern for change requests
matcher = Matcher(nlp.vocab)
change_patterns = [
    [{"LEMMA": "find"}, {"POS": "PRON", "OP": "?"}, {"LEMMA": "something", "OP":
    [{"LEMMA": "change"}, {"POS": "NOUN", "OP": "?"}],
    [{"LEMMA": "alter"}, {"POS": "NOUN", "OP": "?"}],
    [ {"LEMMA": "different"}],

]
matcher.add("CHANGE_REQUEST", change_patterns)

with open("flights.json", "r") as file:
    data = json.load(file)

    for dialogue in data:
        for turn in dialogue['turns']:
```

```
            if turn['speaker'] == 'USER':
                doc = nlp(turn['utterance'])
                matches = matcher(doc)
                for match_id, start, end in matches:
                    span = doc[start:end]
                    # print(f"Matched Change Request: {span.text}")
                    print(f"Matched string: {turn['utterance']}")
```

```
Matched string: A different airline please, in Premium Economy.
Matched string: Find me something else. I want to fly with Southwest Airlines. Lo
ok for them from Atlanta.
Matched string: Find me something else with United Airlines.
Matched string: Find me something else.
Matched string: Find me something else.
Matched string: Try to find me something different. I would like 1 ticket with So
uthwest Airlines.
Matched string: Try to find me something different. I would like 1 ticket with So
uthwest Airlines.
Matched string: Can you find me something else?
Matched string: This Saturday please. Oh, actually, change that to 3 tickets.
Matched string: Can you find something else. I'm just wondering if there is somet
hing that will work out better for us.
Matched string: Find me something else, Delta Airlines instead, economy tickets.
Matched string: I would like to continue trying to make the reservation, let's tr
y a flight to SF. My plans may change, so I only want to find refundable tickets.
Matched string: I need to book two tickets. My plans might change, so I need refu
ndable tickets only. My departure location is Vancouver, BC. We'll be departing t
his Sunday, on our way to Phoenix, USA.
Matched string: Actually, can you change the flight date to the 3rd?
```

Task 2

In [ ]:

```
results = {}

with open('flights.json', 'r') as file:
    data = json.load(file)

    for dialogue in data:
        for turn in dialogue['turns']:
            if turn['speaker'] == 'USER':
                doc = nlp(turn['utterance'])
                # print(F"Sentence:", doc.text)
                # print("Intentions:")
                intentions = []
                for token in doc:
                    if token.dep_ == "dobj":  # Check for direct objects in the
                        dobj = token.text  # Text of the direct object
                        # conj = [t.text for t in token.conjuncts]  # List any c
                        # The verb (action) associated with the direct object
                        verb = token.head
                        # print(verb, dobj)
                        intentions.append(verb.text+' '+ dobj)

                if intentions:
                    results[doc.text] = intentions

for i, key in enumerate(results.keys()):
    if i> 10:
        break
```

```
    print(F"Sentence:", key)
    print("Intentions:", results[key])
```

```
Sentence: I want to find a one way flight.
Intentions: ['find flight']
Sentence: I am flying to Phoenix, AZ. I want 1 ticket from Seattle, WA.
Intentions: ['want ticket']
Sentence: I want to depart 11th of March. I don't care which airline.
Intentions: ['depart 11th', 'care airline']
Sentence: Find me something else. I want to fly with Southwest Airlines. Look for
them from Atlanta.
Intentions: ['Find something']
Sentence: I would like to find a one way flight. I am interested in flights from
Vegas.
Intentions: ['find flight']
Sentence: I will be heading to Toronto, Canada. I would like to start my travel o
n the 13th of March.
Intentions: ['start travel']
Sentence: What time does the flight get there?
Intentions: ['get time']
Sentence: No. I appreciate your help.
Intentions: ['appreciate help']
Sentence: I need to find a one way flight.
Intentions: ['find flight']
Sentence: I will need flights from Los Angeles. Can you find me something with Al
aska Airlines?
Intentions: ['need flights', 'find something']
Sentence: Okay thank you anyways, that is all I need.
Intentions: ['thank you']
```

Висновок:

В ході виконання даної лабораторної роботи я ознайомився з методами синтаксичного аналізу текстів за допомогою бібліотеки spaCy, зокрема з класами Matcher та PhraseMatcher для виділення сутностей та визначення намірів користувачів. Завдання полягало у використанні цих класів для виявлення місць відправлення та висловлювань з проханням змін у діалогах з файлу "flights.json".

Мною було створено шаблони для виділення пунктів відправлення, що дозволяло виявляти локації у реченнях користувачів, які запитували про авіарейси. Також я розробив шаблони для ідентифікації фраз, де користувачі просять про зміну чи вибір іншого варіанту, що важливо для систем автоматизованого обслуговування клієнтів.

Окрім того, я застосував синтаксичні залежності для аналізу залежностей у реченнях, що дало змогу глибше зрозуміти структуру висловлювань та визначати наміри користувачів більш точно. Це дозволило визначати наміри користувачів не тільки за ключовими словами, а й за структурою речень.