**Міністерство освіти і науки України**

**Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

**Звіт**

з лабораторної роботи №7 з дисципліни

«Обробка та аналіз текстових даних на Python»

„Навчання моделей spaCy”

**Виконав(ла)**

(шифр, прізвище, ім'я, по батькові)

*ІП-11 Головня Олександр Ростиславович*

**Перевірив**

(прізвище, ім'я, по батькові)

Київ 2024

**Мета роботи:** Ознайомитись з додаванням власних прикладів до моделей

spaCy та компонентом для класифікації текстів.

**Завдання до лабораторної роботи:**

Створити кілька своїх прикладів у форматі json за тематикою варіанту

(англійською або українською мовою) для розпізнавання нового типу

сутностей (обрати самостійно). Створити програму, що додає ці

приклади до існуючої моделі spaCy, навчає модель. Продемонструвати

роботу.

2. Застосувати компонент TextCategorizer для визначення намірів. Дані для

навчання за тематикою варіанту обрати самостійно або скористатись

вказаним файлом (utterance містить висловлювання, intent - намір). Дані

файли містять приклади діалогів користувачів з системою-помічником за

певною тематикою, наприклад, замовлення квитків і т.д. Навчити

компонент та продемонструвати роботу.

**Результат виконання коду:**

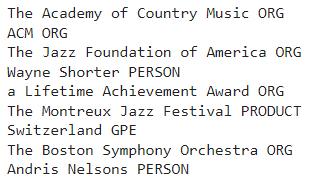
**Для початку я взяв сутності з речень та вивів їх:**

for sent in text:

    doc = nlp(sent)

    for ent in doc.ents:

        print(ent.text, ent.label\_)

****

**Далі навчив модель і продемонстрував роботу:**

import random

from spacy.training import Example

nlp = spacy.blank("en")

ruler = nlp.add\_pipe("entity\_ruler")

patterns = [

    {"label": "ORG", "pattern": "ACM"},

    {"label": "GPE", "pattern": "Switzerland"},

    {"label": "PERSON", "pattern": "Andris Nelsons"},

    {"label": "PERSON", "pattern": "Wayne Shorter"},

]

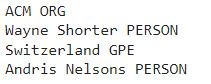
ruler.add\_patterns(patterns)

for sentence in text:

    doc = nlp(sentence)

    for ent in doc.ents:

        print(ent.text, ent.label\_)

****

Завдання 2:

Взяв приклади даних для навчання

Ініціалізував модель spaCy

nlp = spacy.blank("en")

textcat = nlp.add\_pipe("textcat")

textcat.add\_label("ticket\_order")

textcat.add\_label("event\_info")

textcat.add\_label("proceed\_booking")

textcat.add\_label("thanks\_goodbye")

textcat.add\_label("not\_interested")

Навчання моделі:

optimizer = nlp.begin\_training()

for i in range(20):

    losses = {}

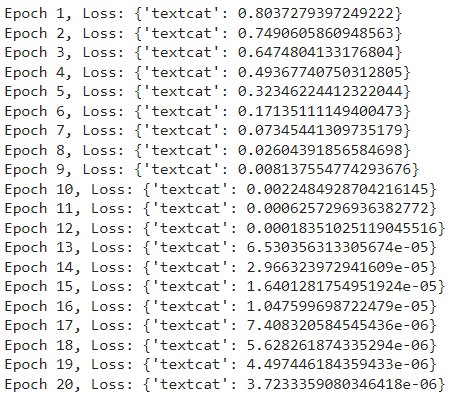
    for text, label in train\_data:

        doc = nlp.make\_doc(text)

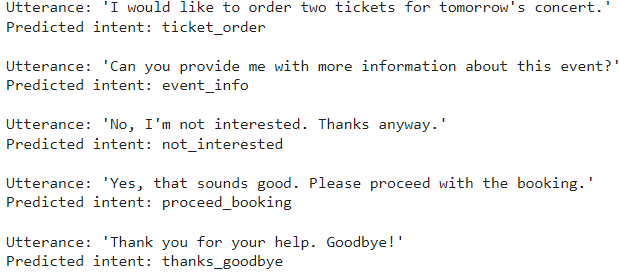
        example = Example.from\_dict(doc, {"cats": {label: 1.0}})

        nlp.update([example], sgd=optimizer, losses=losses)

    print(f"Epoch {i+1}, Loss: {losses}")



Демонстрація роботи на прикладах:



**Висновок:** Під час розробки лабораторної роботи, я ознайомився на практиці з з додаванням власних прикладів до моделей spaCy та компонентом для класифікації текстів.

**Код програми:**

import spacy

nlp = spacy.load("en\_core\_web\_sm")

text = [

    'The Academy of Country Music (ACM) honored outstanding performers and songwriters who have significantly contributed to the evolution of country music.'

    'The Jazz Foundation of America celebrated the career of iconic saxophonist Wayne Shorter with a Lifetime Achievement Award, recognizing his impact on the jazz world.'

    'The Montreux Jazz Festival in Switzerland drew musicians from various genres to engage in workshops, jam sessions, and international competitions.'

    'The Boston Symphony Orchestra, led by maestro Andris Nelsons, presented an innovative concert series that blended classical symphonies with contemporary dance performances.'

]

for sent in text:

    doc = nlp(sent)

    for ent in doc.ents:

        print(ent.text, ent.label\_)

import random

from spacy.training import Example

nlp = spacy.blank("en")

ruler = nlp.add\_pipe("entity\_ruler")

patterns = [

    {"label": "ORG", "pattern": "ACM"},

    {"label": "GPE", "pattern": "Switzerland"},

    {"label": "PERSON", "pattern": "Andris Nelsons"},

    {"label": "PERSON", "pattern": "Wayne Shorter"},

]

ruler.add\_patterns(patterns)

for sentence in text:

    doc = nlp(sentence)

    for ent in doc.ents:

        print(ent.text, ent.label\_)

import spacy

from spacy.training import Example

# Приклади даних для навчання

train\_data = [

    ("I want to order two tickets for tomorrow's concert.", "ticket\_order"),

    ("Can you provide me with more information about this event?", "event\_info"),

    ("No, I'm not interested. Thanks anyway.", "not\_interested"),

    ("Yes, that sounds good. Please proceed with the booking.", "proceed\_booking"),

    ("Thank you for your help. Goodbye!", "thanks\_goodbye"),

]

# Ініціалізація моделі spaCy

nlp = spacy.blank("en")

textcat = nlp.add\_pipe("textcat")

textcat.add\_label("ticket\_order")

textcat.add\_label("event\_info")

textcat.add\_label("proceed\_booking")

textcat.add\_label("thanks\_goodbye")

textcat.add\_label("not\_interested")

# Навчання моделі

optimizer = nlp.begin\_training()

for i in range(20):

    losses = {}

    for text, label in train\_data:

        doc = nlp.make\_doc(text)

        example = Example.from\_dict(doc, {"cats": {label: 1.0}})

        nlp.update([example], sgd=optimizer, losses=losses)

    print(f"Epoch {i+1}, Loss: {losses}")

# Демонстрація роботи моделі на прикладах

test\_data = [

    "I would like to order two tickets for tomorrow's concert.",

    "Can you provide me with more information about this event?",

    "No, I'm not interested. Thanks anyway.",

    "Yes, that sounds good. Please proceed with the booking.",

    "Thank you for your help. Goodbye!",

]

for text in test\_data:

    doc = nlp(text)

    intent = max(doc.cats, key=doc.cats.get)

    print(f"Utterance: '{text}'\nPredicted intent: {intent}\n")