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

Лабораторна робота №8

Аналіз текстів на мові Python

Тема: Навчання моделей spaCy

Варіант: 1

Виконав
студент групи ІП-11:
Панченко С. В.

Перевірила:
Тимофєєва Ю. С

Київ 2023

ЗМІСТ

1 МЕТА ЛАБОРАТОРНОЇ РОБОТИ

Ознайомитись з додаванням власних прикладів до моделей spaCy та компонентом для класифікації текстів.

2 ЗАВДАННЯ

1. 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.

   Варіант 1.Тематика: музика. Файл music.json.

3 ВИКОНАННЯ

3.1 Завдання перше

Завантажимо пакет.

In [1]:
```
import spacy
nlp = spacy.load("en_core_web_sm")
```

```
2023-05-28 09:45:43.106548: I tensorflow/core/platform/cpu_feature_guard.cc:193] This Tens
orFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the fol
lowing CPU instructions in performance-critical operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flag
s.
2023-05-28 09:45:44.775633: W tensorflow/compiler/xla/stream_executor/platform/default/dso
_loader.cc:64] Could not load dynamic library 'libnvinfer.so.7'; dlerror: libnvinfer.so.7:
cannot open shared object file: No such file or directory
2023-05-28 09:45:44.776230: W tensorflow/compiler/xla/stream_executor/platform/default/dso
_loader.cc:64] Could not load dynamic library 'libnvinfer_plugin.so.7'; dlerror: libnvinfe
r_plugin.so.7: cannot open shared object file: No such file or directory
2023-05-28 09:45:44.776248: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT
Warning: Cannot dlopen some TensorRT libraries. If you would like to use Nvidia GPU with T
ensorRT, please make sure the missing libraries mentioned above are installed properly.
2023-05-28 09:45:46.584440: W tensorflow/compiler/xla/stream_executor/platform/default/dso
_loader.cc:64] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: canno
t open shared object file: No such file or directory
2023-05-28 09:45:46.584472: W tensorflow/compiler/xla/stream_executor/cuda/cuda_driver.cc:
265] failed call to cuInit: UNKNOWN ERROR (303)
2023-05-28 09:45:46.584505: I tensorflow/compiler/xla/stream_executor/cuda/cuda_diagnostic
s.cc:156] kernel driver does not appear to be running on this host (localhost): /proc/driv
er/nvidia/version does not exist
```

Рисунок 3.1.1 - Сутності

Для початку візьмемо сутності з речень та виведемо їх.

In [2]:
```
text = [
    'The American Society of Music Arrangers and Composers (ASMAC) honored legendary film
    'The International Bluegrass Music Association (IBMA) recognized exceptional artists a
    'The Detroit Symphony Orchestra, under the leadership of conductor Leonard Slatkin, pe
    'The International Guitar Festival, held annually in Cordoba, Spain, attracted guitari
]
for sent in text:
    doc = nlp(sent)
    for ent in doc.ents:
        print(ent.text, ent.label_)
```

```
The American Society of Music Arrangers ORG
John Williams PERSON
a Lifetime Achievement Award ORG
The International Bluegrass Music Association ORG
The Detroit Symphony Orchestra ORG
Leonard Slatkin PERSON
The International Guitar Festival ORG
annually DATE
Cordoba GPE
Spain GPE
```

Рисунок 3.1.2 - Сутності

Навчимо модель.

```
In [3]: import random
        from spacy.training import Example

        nlp = spacy.blank("en")
        ruler = nlp.add_pipe("entity_ruler")
        patterns = [
            {"label": "DATE", "pattern": "annualy"},
            {"label": "GPE", "pattern": "Spain"},
            {"label": "GPE", "pattern": "Cordoba"},
            {"label": "PERSON", "pattern": "John Williams"},
        ]

        ruler.add_patterns(patterns)
```

Рисунок 3.1.3 - Навчання моделі

Продемонструємо роботу.

```
In [4]: for sentence in text:
            doc = nlp(sentence)
            for ent in doc.ents:
                print(ent.text, ent.label_)

John Williams PERSON
Cordoba GPE
Spain GPE
```

Рисунок 3.1.4 - Робота моделі

3.2Завдання друге

Дістанемо з json файлу потрібну інформацію та представимо її у csv-форматі.

In [5]:
```
import json
with open('music.json', 'r') as file:
    data = json.load(file)
data
```

Out[5]:

```
[{'dialogue_id': '7_00000',
  'services': ['Music_2'],
  'turns': [{'frames': [{'actions': [{'act': 'INFORM_INTENT',
       'canonical_values': ['LookupMusic'],
       'slot': 'intent',
       'values': ['LookupMusic']}],
     'service': 'Music_2',
     'slots': [],
     'state': {'active_intent': 'LookupMusic',
      'requested_slots': [],
      'slot_values': {}}}],
    'speaker': 'USER',
    'utterance': 'I want to listen to some songs.'},
   {'frames': [{'actions': [{'act': 'OFFER',
       'canonical_values': ['Dark Necessities'],
       'slot': 'song_name',
       'values': ['Dark Necessities']},
      {'act': 'OFFER',
       'canonical_values': ['Red Hot Chili Peppers'],
       'slot': 'artist',
       'values': ['Red Hot Chili Peppers']},
      {'act': 'OFFER',
       'canonical_values': ['The Getaway'],
       'slot': 'album',
       'values': ['The Getaway']}],
     'service': 'Music_2',
     'service_call': {'method': 'LookupMusic', 'parameters': {}},
     'service_results': [{'album': 'The Getaway',
       'artist': 'Red Hot Chili Peppers',
       'genre': 'Pop',
       'song_name': 'Dark Necessities'},
      {'album': 'Pitch Perfect',
       'artist': 'Anna Kendrick',
       'genre': 'Soundtracks',
       'song_name': 'Cups'},
      {'album': 'Dedicated',
       'artist': 'Carly Rae Jepsen',
       'genre': 'Pop',
       'song_name': 'Now That I Found You'},
      {'album': 'Prequelle',
       'artist': 'Ghost',
       'genre': 'Metal',
       'song_name': 'Dance Macabre'},
      {'album': 'Illuminate',
       'artist': 'Shawn Mendes',
       'genre': 'Pop',
       'song_name': 'Treat You Better'},
      {'album': 'Brave Enough',
       'artist': 'Lindsey Stirling',
       'genre': 'Pop',
       'song_name': 'The Arena'},
      {'album': 'Gotta Be Me',
       'artist': 'Cody Johnson',
       'genre': 'Country',
       'song_name': 'With You I Am'},
      {'album': 'Caroline',
       'artist': 'Citizen Soldier',
       'genre': 'Rock',
       'song_name': 'Let It Burn'},
      {'album': 'The Black',
       'artist': 'Asking Alexandria',
       'genre': 'Metalcore',
       'song_name': 'The Black'},
      {'album': 'Dangerous Woman',
       'artist': 'Ariana Grande',
       'genre': 'Pop',
       'song_name': 'Let Me Love You'}],
     'slots': [{'exclusive_end': 41, 'slot': 'song_name', 'start': 25},
      {'exclusive_end': 70, 'slot': 'artist', 'start': 49},
      {'exclusive_end': 97, 'slot': 'album', 'start': 86}]}],
    'speaker': 'SYSTEM',
    'utterance': 'Okay, how about the song Dark Necessities by the Red Hot Chili Peppers o
n their album The Getaway?'},
   {'frames': [{'actions': [{'act': 'INFORM',
       'canonical_values': ['Born This Way'],
       'slot': 'album',
       'values': ['Born This Way']},
      {'act': 'INFORM',
```

Рисунок 3.2.1 - JSON-файл

Зчитаємо дані до датафрейму.

In [6]:
```
import pandas as pd
filtered_data = []
for dialog in data:
    for turn in dialog['turns']:
        last_frame = turn['frames'][-1]
        if turn['speaker'] == 'USER':
            state = last_frame['state']
            active_intent = state['active_intent']
            filtered_data.append((turn['utterance'], active_intent))
df = pd.DataFrame(filtered_data, columns=['text', 'theme'])
df
```

Out[6]:

	text	theme
0	I want to listen to some songs.	LookupMusic
1	Let's look for something else, any genre will ...	LookupMusic
2	Yes, that's great.	LookupMusic
3	Yes, play it.	PlayMedia
4	Actually, play it on the kitchen speaker.	PlayMedia
...	...	...
908	Yeah, that sounds good.	LookupSong
909	Yes, please play it.	PlaySong
910	No, actually cast it on my kitchen device.	PlaySong
911	Yeah, that's exactly right.	PlaySong
912	Thanks, that's all for now.	PlaySong

913 rows × 2 columns

Рисунок 3.2.2 - Зчитування даних до датафрейму

Розділимо дані на навчальні та тестові.

In [7]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(
    df.theme, df.text, test_size=0.3, random_state=0)
train = pd.concat([y_train, x_train], axis=1)
test = pd.concat([y_test, x_test], axis=1)
```

Рисунок 3.2.3 - Навчальні та тестові дані

Визначимо функцію перетворення даних на документи.

In [8]:
```
from tqdm.auto import tqdm

def make_data(data: pd.DataFrame, labels: list[str]):
    data_tuples = list(data.itertuples(index=False, name=None))
    nlp = spacy.load('en_core_web_trf')
    docs = []
    tt = tqdm(nlp.pipe(data_tuples, as_tuples=True), total=len(data_tuples))
    for doc, label in tt:
        for l in labels:
            doc.cats[l] = int(l == label)
        docs.append(doc)
    return docs
```

Рисунок 3.2.4 - Функція перетворення даних на документи

Перетворимо дані на документи.

```
In [9]: from spacy.tokens import DocBin
        labels = list(set(df.theme.to_list()))

        train_docs = make_data(train, labels)
        test_docs = make_data(test, labels)

          0%|          | 0/639 [00:00<?, ?it/s]
          0%|          | 0/274 [00:00<?, ?it/s]
```

Рисунок 3.2.5 - Перетворення даних на документи

Збережемо документи до файлів.

```
In [10]: train_bin = DocBin(docs=train_docs)
         train_bin.to_disk('textcat_data/textcat_train.spacy')

         test_bin = DocBin(docs=test_docs)
         test_bin.to_disk('textcat_data/textcat_test.spacy')
```

Рисунок 3.2.6 - Збереження до файлів

```
In [11]: %reset
```

Заповнимо config-файл.

```
In [12]: import os
         os.system('python3 -m spacy init fill-config ./textcat_base_config.cfg ./textcat_config.cf

2023-05-28 08:33:01.784483: I tensorflow/core/platform/cpu_feature_guard.cc:193] This Tens
orFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the fol
lowing CPU instructions in performance-critical operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flag
s.
2023-05-28 08:33:02.709334: W tensorflow/compiler/xla/stream_executor/platform/default/dso
_loader.cc:64] Could not load dynamic library 'libnvinfer.so.7'; dlerror: libnvinfer.so.7:
cannot open shared object file: No such file or directory
2023-05-28 08:33:02.709396: W tensorflow/compiler/xla/stream_executor/platform/default/dso
_loader.cc:64] Could not load dynamic library 'libnvinfer_plugin.so.7'; dlerror: libnvinfe
r_plugin.so.7: cannot open shared object file: No such file or directory
2023-05-28 08:33:02.709401: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT
Warning: Cannot dlopen some TensorRT libraries. If you would like to use Nvidia GPU with T
ensorRT, please make sure the missing libraries mentioned above are installed properly.
2023-05-28 08:33:03.534366: W tensorflow/compiler/xla/stream_executor/platform/default/dso
_loader.cc:64] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: canno
t open shared object file: No such file or directory
2023-05-28 08:33:03.534390: W tensorflow/compiler/xla/stream_executor/cuda/cuda_driver.cc:
265] failed call to cuInit: UNKNOWN ERROR (303)
2023-05-28 08:33:03.534407: I tensorflow/compiler/xla/stream_executor/cuda/cuda_diagnostic
s.cc:156] kernel driver does not appear to be running on this host (localhost): /proc/driv
er/nvidia/version does not exist
✔ Auto-filled config with all values
✔ Saved config
textcat_config.cfg
You can now add your data and train your pipeline:
python -m spacy train textcat_config.cfg --paths.train ./train.spacy --paths.dev ./dev.spa
cy
Out[12]: 0
```

Рисунок 3.2.7 - Заповнення config-файлу

Натренуємо модель.

```
In [13]: os.system('python3 -m spacy train textcat_config.cfg --verbose --output ./textcat_output -
```

```
2023-05-28 08:33:06.648592: I tensorflow/core/platform/cpu_feature_guard.cc:193] This Tens
orFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the fol
lowing CPU instructions in performance-critical operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flag
s.
2023-05-28 08:33:07.441757: W tensorflow/compiler/xla/stream_executor/platform/default/dso
_loader.cc:64] Could not load dynamic library 'libnvinfer.so.7'; dlerror: libnvinfer.so.7:
cannot open shared object file: No such file or directory
2023-05-28 08:33:07.441817: W tensorflow/compiler/xla/stream_executor/platform/default/dso
_loader.cc:64] Could not load dynamic library 'libnvinfer_plugin.so.7'; dlerror: libnvinfe
r_plugin.so.7: cannot open shared object file: No such file or directory
2023-05-28 08:33:07.441823: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT
Warning: Cannot dlopen some TensorRT libraries. If you would like to use Nvidia GPU with T
ensorRT, please make sure the missing libraries mentioned above are installed properly.
2023-05-28 08:33:08.174281: W tensorflow/compiler/xla/stream_executor/platform/default/dso
_loader.cc:64] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: canno
t open shared object file: No such file or directory
2023-05-28 08:33:08.174301: W tensorflow/compiler/xla/stream_executor/cuda/cuda_driver.cc:
265] failed call to cuInit: UNKNOWN ERROR (303)
2023-05-28 08:33:08.174319: I tensorflow/compiler/xla/stream_executor/cuda/cuda_diagnostic
s.cc:156] kernel driver does not appear to be running on this host (localhost): /proc/driv
er/nvidia/version does not exist
[2023-05-28 08:33:08,541] [DEBUG] Config overrides from CLI: ['paths.train', 'paths.dev']
ℹ Saving to output directory: textcat_output
ℹ Using CPU

========================== Initializing pipeline ==========================
[2023-05-28 08:33:08,824] [INFO] Set up nlp object from config
[2023-05-28 08:33:08,835] [DEBUG] Loading corpus from path: textcat_data/textcat_test.spac
y
[2023-05-28 08:33:08,837] [DEBUG] Loading corpus from path: textcat_data/textcat_train.spa
cy
[2023-05-28 08:33:08,837] [INFO] Pipeline: ['tok2vec', 'textcat']
[2023-05-28 08:33:08,840] [INFO] Created vocabulary
[2023-05-28 08:33:10,639] [INFO] Added vectors: en_core_web_lg
[2023-05-28 08:33:10,639] [INFO] Finished initializing nlp object
[2023-05-28 08:33:10,987] [INFO] Initialized pipeline components: ['tok2vec', 'textcat']
[2023-05-28 08:33:11,003] [DEBUG] Loading corpus from path: textcat_data/textcat_test.spac
y
[2023-05-28 08:33:11,005] [DEBUG] Loading corpus from path: textcat_data/textcat_train.spa
cy
[2023-05-28 08:33:11,008] [DEBUG] Removed existing output directory: textcat_output/model-
best
[2023-05-28 08:33:11,014] [DEBUG] Removed existing output directory: textcat_output/model-
last
✔ Initialized pipeline

============================== Training pipeline ==============================
ℹ Pipeline: ['tok2vec', 'textcat']
ℹ Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS TEXTCAT  CATS_SCORE  SCORE
---  ------  ------------  ------------  ----------  ------
  0       0          0.00          0.16       18.10    0.18
  3     200         52.66         19.13       55.77    0.56
  8     400        123.32         13.75       60.43    0.60
 13     600        549.94         11.21       60.63    0.61
 20     800       2481.08          8.79       62.16    0.62
 29    1000       6631.92          8.48       66.73    0.67
 39    1200      13886.48          7.83       63.07    0.63
 52    1400      22349.42          6.75       61.08    0.61
 67    1600      42729.82          5.85       60.03    0.60
 86    1800      69465.50          5.00       58.51    0.59
109    2000     130166.79          4.64       59.43    0.59
137    2200     229572.31          4.12       60.46    0.60
170    2400     244185.19          3.81       61.25    0.61
203    2600     286993.83          3.39       60.17    0.60
✔ Saved pipeline to output directory
textcat_output/model-last
```

```
Out[13]: 0
```

Рисунок 3.2.8 - Тренування моделі

Завантажимо найкращу модель.

In [11]:
```
import spacy
nlp_textcat = spacy.load('textcat_output/model-best')
print(f'text: {test["text"].iloc[100]}')
print(f'orig_cat: {test["theme"].iloc[100]}')
print(f'predicted_cat: {nlp_textcat(test["text"].iloc[100]).cats}')
```

```
text: Sure. Play it on the speaker in the kitchen.
orig_cat: PlaySong
predicted_cat: {'NONE': 2.456112088111695e-05, 'LookupMusic': 0.005323320627212524, 'PlayS
ong': 0.030409136787056923, 'PlayMedia': 0.9623216390609741, 'LookupSong': 0.0019213437335
565686}
```

Рисунок 3.2.9 - Завантаження найкращої моделі

ДОДАТОК А ТЕКСТИ ПРОГРАМНОГО КОДУ

Тексти програмного коду

(Найменування програми (документа))

Жорсткий диск

(Вид носія даних)

(Обсяг програми (документа), арк.)

Студента групи ІП-113 курсу

Панченка С. В

```
import spacy
nlp = spacy.load("en_core_web_sm")
text = [
'The American Society of Music Arrangers and Composers (ASMAC) honored legendary film composer John Williams with a Lifetime Achievement Award for his contributions to the industry.',
'The International Bluegrass Music Association (IBMA) recognized exceptional artists and musicians who preserve and promote the tradition of bluegrass music.',
'The Detroit Symphony Orchestra, under the leadership of conductor Leonard Slatkin, performed a groundbreaking multimedia concert that merged classical music with digital art.',
'The International Guitar Festival, held annually in Cordoba, Spain, attracted guitarists from around the globe to participate in masterclasses and competitions.'
]
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": "DATE", "pattern": "annualy"},
{"label": "GPE", "pattern": "Spain"},
{"label": "GPE", "pattern": "Cordoba"},
{"label": "PERSON", "pattern": "John Williams"},
]
ruler.add_patterns(patterns)
for sentence in text:
doc = nlp(sentence)
for ent in doc.ents:
print(ent.text, ent.label_)
import json
with open('music.json', 'r') as file:
data = json.load(file)
data
import pandas as pd
filtered_data = []
for dialog in data:
for turn in dialog['turns']:
```

```
      last_frame = turn['frames'][-1]
      if turn['speaker'] == 'USER':
      state = last_frame['state']
      active_intent = state['active_intent']
      filtered_data.append((turn['utterance'], active_intent))
      df = pd.DataFrame(filtered_data, columns=['text', 'theme'])
      df
      from sklearn.model_selection import train_test_split
      x_train, x_test, y_train, y_test = train_test_split(
      df.theme, df.text, test_size=0.3, random_state=0)
      train = pd.concat([y_train, x_train], axis=1)
      test = pd.concat([y_test, x_test], axis=1)
      from tqdm.auto import tqdm
      def make_data(data: pd.DataFrame, labels: list[str]):
      data_tuples = list(data.itertuples(index=False, name=None))
      nlp = spacy.load('en_core_web_trf')
      docs = []
      tt = tqdm(nlp.pipe(data_tuples, as_tuples=True), total=len(data_tuples))
      for doc, label in tt:
      for l in labels:
      doc.cats[l] = int(l == label)
      docs.append(doc)
      return docs
      from spacy.tokens import DocBin
      labels = list(set(df.theme.to_list()))
      train_docs = make_data(train, labels)
      test_docs = make_data(test, labels)
      train_bin = DocBin(docs=train_docs)
      train_bin.to_disk('textcat_data/textcat_train.spacy')
      test_bin = DocBin(docs=test_docs)
      test_bin.to_disk('textcat_data/textcat_test.spacy')
      get_ipython().run_line_magic('reset', '')
      import os
      os.system('python3 -m spacy init fill-config ./textcat_base_config.cfg
./textcat_config.cfg')
      os.system('python3 -m spacy train textcat_config.cfg --verbose --
output ./textcat_output --paths.train ./textcat_data/textcat_train.spacy --
paths.dev ./textcat_data/textcat_test.spacy')
      import spacy
      nlp_textcat = spacy.load('textcat_output/model-best')
      print(f'text: {test["text"].iloc[100]}')
      print(f'orig_cat: {test["theme"].iloc[100]}')
      print(f'predicted_cat: {nlp_textcat(test["text"].iloc[100]).cats}')
```