ОБРОБКА ПРИРОДНОЇ МОВИ (NLP) У PYTHON. РОЗПІЗНАВАННЯ ІМЕНОВАНИХ СУТНОСТЕЙ

ЛЕКЦІЯ 5

РОЗПІЗНАВАННЯ ІМЕНОВАНИХ СУТНОСТЕЙ

- це підзадача видобування інформації, яка спрямована на пошук і класифікацію іменованих сутностей, згаданих в неструктурованому тексті, по заздалегідь певних категорій, таким як імена людей, організації, місця розташування, медичні коди, вираження часу, кількості, грошові значення, відсотки та інше.

- Для цього ви можете використовувати готову попередньо навчену модель NER за допомогою бібліотеки з відкритим вихідним кодом, таку як Spacy або Stanford CoreNLP.

- Тепер, якщо ви гадаєте, що попередньо навчені моделі NER не дають результату відповідно до ваших очікувань, які ви шукаєте (приклад: тварина, об'єкт), недоступні в попередньо навченої моделі NER, тоді ви можете навчити свою власну кастомну модель.

- Для навчання кастомної моделі NER у вас має бути величезна кількість анотованих даних. Для цього ви повинні використовувати який-небудь інструмент анотації, наприклад:

- Brat rapid annotation tool
- GATE
- WebAnno

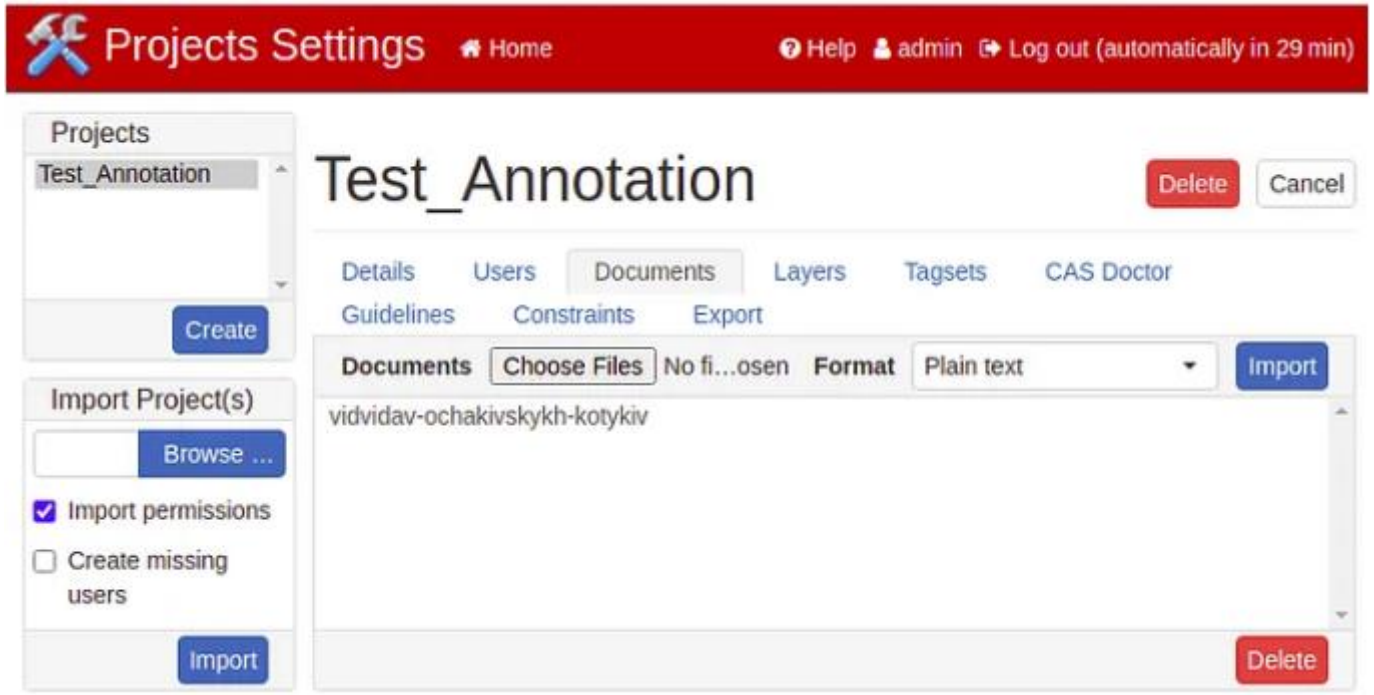
- Наприклад, WebAnno є досить простим, тому що постачається як файл jar, що означає, що вам не потрібно його встановлювати. А також він може використовуватися для складного проекту — кілька користувачів можуть одночасно працювати в одному проекті.

- Щоб запустити це веб-додаток, в консолі використовуйте наступну команду:

```
java -jar webanno-standalone-3.6.7.jar
```

https://webanno.github.io/webanno/releases/3.6.7/docs/user-guide.html

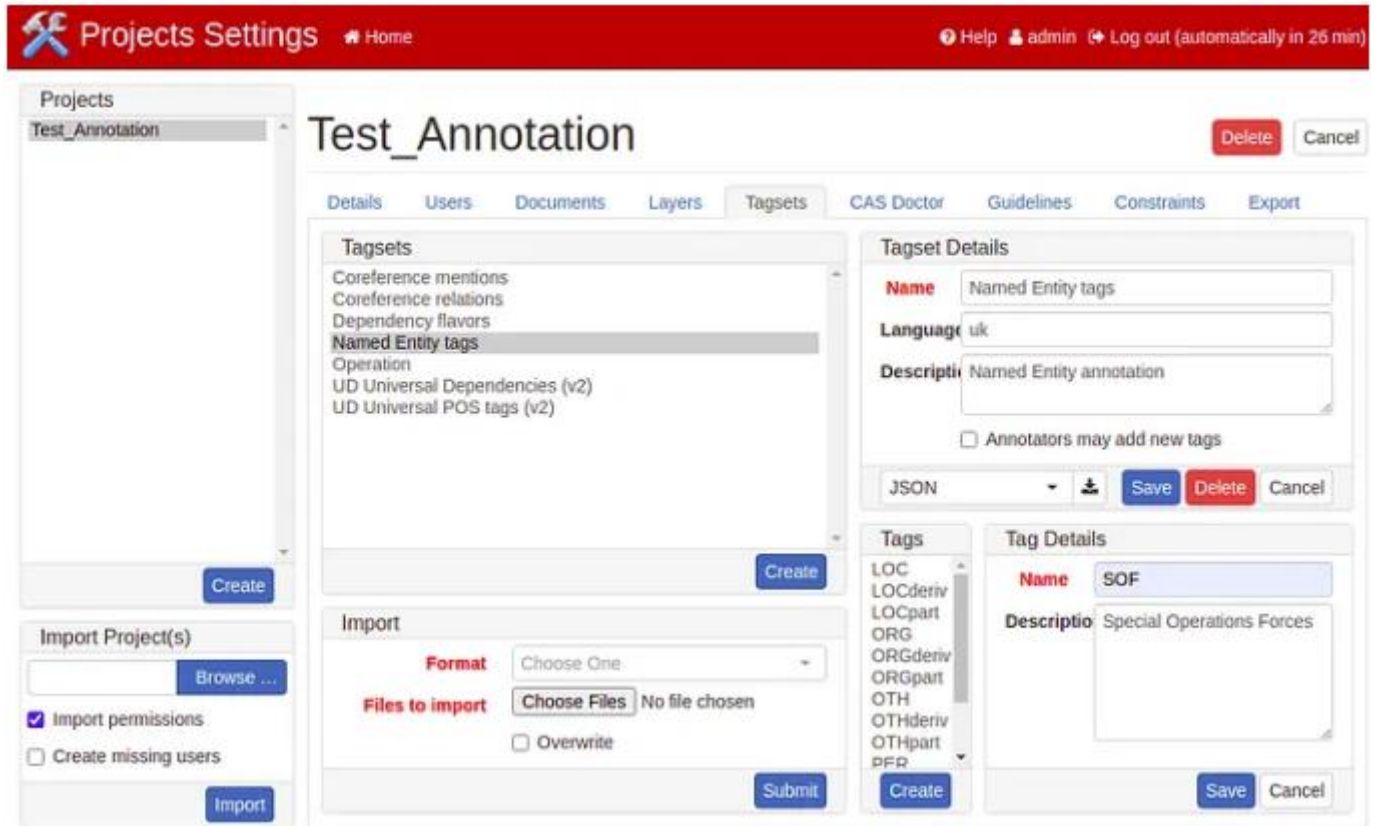
- Створіть новий проект, на сторінці налаштувань проектів, у вкладці “Projects” натисніть “Create”. Напишіть яку-небудь назву проекту. (Приклад: «Test_Annotation») та оберіть “Project type” як annotation.



- Після визначення деталей проекту з’являться кілька вкладок, таких як Користувачі, Документи, Шари, Набори тегів та ін. Звідти виберіть вкладку Документи і зробіть наступне: у списку виберіть формат “Plain text”, завантажте текстовий файл текстового документа, для якого ми будемо готувати навчальні дані та натисніть “Import”. Зразок тексту, який використовувався у вхідному текстовому файлі для підготовки даних навчання, наведено нижче:

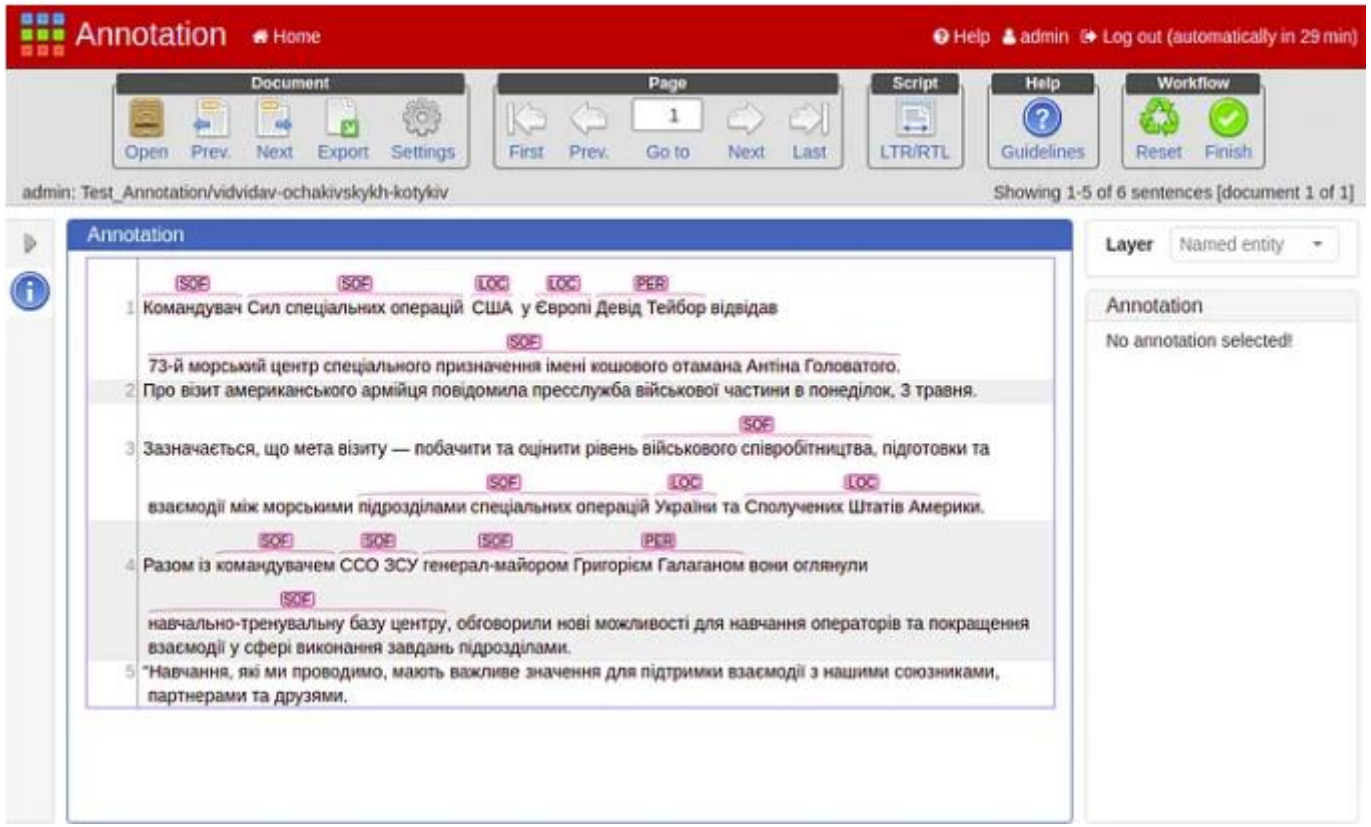
```
Командувач Сил спеціальних операцій США у Європі Девід Тейбор
відвідав 73-й морський центр спеціального призначення імені кошового
отамана Антіна Головатого.

Про візит американського армійця повідомила пресслужба військової
частини в понеділок, 3 травня.

Зазначається, що мета візиту — побачити та оцінити рівень військового
співробітництва, підготовки та взаємодії між морськими підрозділами
спеціальних операцій України та Сполучених Штатів Америки.

Разом із командувачем ССО ЗСУ генерал-майором Григорієм Галаганом
вони оглянули навчально-тренувальну базу центру, обговорили нові
можливості для навчання операторів та покращення взаємодії у сфері
виконання завдань підрозділами.

“Навчання, які ми проводимо, мають важливе значення для підтримки
взаємодії з нашими союзниками, партнерами та друзями. Цей навчальний
центр надає нам можливість тренуватися разом і вчитися один у одного
в стратегічно важливому регіоні”, - цитує Тейбора пресслужба 73-го
морського центру.
```

- Отже, тепер давайте подивимось, як створити нову сутність. Для цього виконайте наступні кроки:
- 1. Перейдіть на вкладку “Tagsets”.
- 2. Виберіть Named Entity зі списку наборів тегів.
- 3. Для створення нового тегу, наприклад “SOF”, натисніть «Create» в розділі «Tags».



- Таким же чином ви можете створити свою кастомну сутність.

- Тепер в меню проекту виберіть Анотація. З’явиться нове спливаюче вікно, виберіть документ, який ви хочете анотувати звідти.



- Як тільки ви закінчите з анотацією, натисніть «Export», виберіть «WebAnno TSV v3.2» у спливаючому вікні та експорту його.

- Тепер давайте почнемо кодування для створення остаточних форматованих користувальницьких навчальних даних уформаті Spacy для навчання користувальницької моделі розпізнавання іменованих сутностей (NER) з використанням Spacy.

```
# Підготовка тренувальних даних в форматі Spacy
TRAIN_DATA = []
ent_list = []
from web_anno_tsv import open_web_anno_tsv

tsv = '/content/vidvidav-ochakivskykh-kotykiv.tsv'

with open_web_anno_tsv(tsv) as f:
    for i, sentence in enumerate(f):
        #print(f"Sentence {i}:", sentence.text)
        ent_list_sen = []
        for j, annotation in enumerate(sentence.annotations):

ent_list_sen.append((annotation.start,annotation.stop,annotation.labe
l))
        ent_list.append(ent_list_sen)
        ent_dic = {}
        ent_dic['entities'] = ent_list[-1]
        # Підготуйте підсумкові дані навчання
        TRAIN_DATA.append([sentence.text,ent_dic])
```

```
TRAIN_DATA
```

```
[['Командувач Сил спеціальних операцій США у Європі Девід Тейбор відвідав 73-й
  {'entities': [(0, 10, 'SOF'),
    (11, 35, 'SOF'),
    (36, 39, 'LOC'),
    (42, 48, 'LOC'),
    (49, 61, 'PER'),
    (71, 157, 'SOF')]}],
 ['Про візит американського армійця повідомила прессслужба військової частини в
  {'entities': []}],
 ['Зазначається, що мета візиту – побачити та оцінити рівень військового співро
  {'entities': [(58, 85, 'SOF'),
    (125, 158, 'SOF'),
    (159, 166, 'LOC'),
    (170, 195, 'LOC')]}],
 ['Разом із командувачем ССО ЗСУ генерал-майором Григорієм Галаганом вони оглян
  {'entities': [(9, 21, 'SOF'),
    (22, 29, 'SOF'),
    (30, 45, 'SOF'),
    (46, 65, 'PER'),
    (80, 113, 'SOF')]}],
 ['"Навчання, які ми проводимо, мають важливе значення для підтримки взаємодії
  {'entities': []}],
 ['Цей навчальний центр надає нам можливість тренуватися разом і вчитися один у
  {'entities': []}]]
```

- Тепер давайте спробуємо навчити нову свіжу модель NER, використовуючи підготовлені призначені для користувача дані NER. Визначте змінні, необхідні для обробки навчальної моделі.

```
model = None
model_dir=Path("model_ner")
n_iter=100

if model is not None:
    nlp = spacy.load(model)
    print("Loaded model '%s'" % model)
else:
    nlp = spacy.blank('uk')
    print("Created blank 'uk' model")

#Потім завантажте порожню модель для процесу, що виконує дію NER, і
налаштуйте конвеєр тільки з NER за допомогою функції create_pipe.

if 'ner' not in nlp.pipe_names:
    ner = nlp.create_pipe('ner')
    nlp.add_pipe(ner, last=True)
else:
    ner = nlp.get_pipe('ner')

for _, annotations in TRAIN_DATA:
    for ent in annotations.get('entities'):
        ner.add_label(ent[2])

other_pipes = [pipe for pipe in nlp.pipe_names if pipe != 'ner']
with nlp.disable_pipes(*other_pipes):  # only train NER
    optimizer = nlp.begin_training()
    for itn in range(n_iter):
        random.shuffle(TRAIN_DATA)
        losses = {}
        for text, annotations in tqdm(TRAIN_DATA):
            nlp.update(
                [text],
                [annotations],
                drop=0.5,
                sgd=optimizer,
                losses=losses)
        print(losses)
```

Щоб протестувати навчену модель,

```
for text, _ in train_data:
        doc = nlp(text)
        print("Entities", [(ent.text, ent.label_) for ent in doc.ents])
        print("Tokens", [(t.text, t.ent_type_, t.ent_iob) for t in doc])
```

```
Entities [('Командувач', 'SOF'), ('Сил спеціальних операцій', 'SOF'), ('США', 'LOC'), ('Європі', 'LOC'), ('Девід Тейбор', 'PER'),
Tokens [('Командувач', 'SOF', 3), ('Сил', 'SOF', 3), ('спеціальних', 'SOF', 1), ('операцій', 'SOF', 1), ('США', 'LOC', 3), ('у', '
Entities [('командувачем', 'SOF'), ('ССО ЗСУ', 'SOF'), ('генерал-майором', 'SOF'), ('Григорієм Галаганом', 'PER'), ('навчально-тре
Tokens [('Разом', '', 2), ('із', '', 2), ('командувачем', 'SOF', 3), ('ССО', 'SOF', 3), ('ЗСУ', 'SOF', 1), ('генерал', 'SOF', 3),
Entities []
Tokens [('"', '', 2), ('Навчання', '', 2), (',', '', 2), ('які', '', 2), ('ми', '', 2), ('проводимо', '', 2), (',', '', 2), ('мают
Entities []
Tokens [('Цей', '', 2), ('навчальний', '', 2), ('центр', '', 2), ('надає', '', 2), ('нам', '', 2), ('можливість', '', 2), ('тренув
Entities [('військового співробітництва', 'SOF'), ('підрозділами спеціальних операцій', 'SOF'), ('України', 'LOC'), ('Сполучених Ш
Tokens [('Зазначається', '', 2), (',', '', 2), ('що', '', 2), ('мета', '', 2), ('візиту', '', 2), ('–', '', 2), ('побачити', '', 2
Entities []
Tokens [('Про', '', 2), ('візит', '', 2), ('американського', '', 2), ('армійця', '', 2), ('повідомила', '', 2), ('прессслужба', '',
```

Нарешті, збережіть модель на свій шлях, який зберігається в змінної model_dir.

```
if model_dir is not None:
    model_dir = Path(model_dir)
    if not model_dir.exists():
        model_dir.mkdir()
    nlp.to_disk(model_dir)
    print("Saved model to", model_dir)

model = spacy.load(model_dir)
```