Metody przetwarzania języka naturalnego

Laboratorium 2 – Rozpoznawanie nazwanych encji (NER)

mgr inż. Aleksandra Kwiatkowska

1. Wprowadzenie

1.1. Cel instrukcji

Niniejsza instrukcja laboratoryjna ma na celu wprowadzenie w zagadnienie rozpoznawania nazwanych encji (Named Entity Recognition - NER), które stanowi jedno z fundamentalnych zadań w dziedzinie przetwarzania języka naturalnego. Zostaną przedstawione różnorodne podejścia do identyfikacji i klasyfikacji encji w tekście, począwszy od prostych metod słownikowych, poprzez systemy regułowe, aż po zaawansowane techniki wykorzystujące uczenie maszynowe i głębokie sieci neuronowe. Umiejętność implementacji i wykorzystania systemów NER jest kluczowa w tworzeniu nowoczesnych aplikacji analizujących tekst, takich jak systemy ekstrakcji informacji, chatboty, wyszukiwarki semantyczne czy narzędzia do automatycznego przetwarzania dokumentów biznesowych i prawnych.

1.2. Rozpoznawanie nazwanych encji

Rozpoznawanie nazwanych encji to proces automatycznej identyfikacji i klasyfikacji kluczowych elementów informacyjnych w tekście nieustrukturyzowanym. Zadanie to polega na lokalizacji fragmentów tekstu odnoszących się do konkretnych bytów rzeczywistego świata oraz przypisaniu im odpowiednich kategorii semantycznych. Typowe kategorie encji obejmują:

- osoby (PERSON),
- organizacje (ORGANIZATION),
- lokalizacje (LOCATION),
- daty (DATE),
- kwoty pieniężne (MONEY),
- wartości procentowe (PERCENT).

System NER analizując zdanie „Jan Kowalski pracuje w Google od 2020 roku w Warszawie", powinien rozpoznać „Jan Kowalski" jako osobę, „Google" jako organizację, „2020 roku" jako datę, a „Warszawie" jako lokalizację.

Proces rozpoznawania encji składa się zazwyczaj z dwóch głównych etapów:

- segmentacji, czyli wyznaczenia granic encji w tekście
- klasyfikacji, czyli przypisania zidentyfikowanym fragmentom odpowiednich typów. Złożoność tego zadania wynika z wieloznaczności języka naturalnego, różnorodności form zapisu tych samych encji oraz kontekstowej natury wielu wyrażeń. Na przykład słowo "Wisła" może oznaczać rzekę (lokalizacja) lub klub sportowy (organizacja), a jego prawidłowa interpretacja wymaga analizy kontekstu.

Systemy rozpoznawania nazwanych encji znajdują szerokie zastosowanie w różnorodnych dziedzinach informatyki stosowanej i biznesu. W systemach wyszukiwania informacji NER umożliwia indeksowanie i przeszukiwanie dokumentów na podstawie występujących w nich encji, co znacząco poprawia precyzję i relewantność wyników. W analizie mediów społecznościowych pozwala na automatyczne śledzenie wzmianek o konkretnych osobach, firmach czy produktach, co jest nieocenione w monitorowaniu reputacji i analizie sentymentu. W dziedzinie medycyny NER wykorzystuje się do ekstrakcji nazw leków, chorób, objawów i procedur medycznych z dokumentacji klinicznej, co wspiera systemy wspomagania decyzji medycznych.

1.3. Podejścia do rozpoznawania nazwanych encji

- *Metoda słownikowa* stanowi najprostsze i historycznie najwcześniejsze podejście do rozpoznawania nazwanych encji. Opiera się ona na wykorzystaniu przygotowanych wcześniej list (słowników) zawierających znane encje wraz z ich typami. System przeszukuje analizowany tekst w poszukiwaniu dokładnych dopasowań lub częściowych zgodności z wpisami w słownikach. Choć metoda ta charakteryzuje się wysoką precyzją dla encji znajdujących się w słownikach, jej głównym ograniczeniem jest niska kompletność - system nie jest w stanie rozpoznać encji nieobecnych w bazie.
  Tworzenie i utrzymywanie słowników wymaga znacznego nakładu pracy, szczególnie w dynamicznie zmieniających się domenach. Słowniki muszą być regularnie aktualizowane o nowe encje, co w przypadku niektórych kategorii, takich jak nazwy firm czy produktów, może być procesem ciągłym. Dodatkowo, metoda

słownikowa ma trudności z obsługą wieloznaczności, ta sama nazwa może reprezentować różne typy encji w zależności od kontekstu. Mimo tych ograniczeń, podejście słownikowe jest nadal szeroko stosowane jako element hybrydowych systemów NER, gdzie służy do wstępnej identyfikacji kandydatów na encje lub jako źródło cech dla bardziej zaawansowanych metod.

```
slownik_osob = ["Adam Mickiewicz", "Maria Skłodowska-Curie", "Jan Kochanowski"]

text = "Maria Skłodowska-Curie urodziła się w Warszawie, ale Adam Mickiewicz mieszkał w Krakowie."
def dictionary_based_ner(text, people_dict):
    entities = []

        for person in people_dict:
            if person in text:
                start_idx = text.index(person)
                end_idx = start_idx + len(person)
                entities.append((person, "PERSON", start_idx, end_idx))
     return entities
```

- *Podejście oparte na regułach* wykorzystują ręcznie tworzone reguły lingwistyczne do identyfikacji encji na podstawie ich charakterystycznych cech. Reguły te mogą uwzględniać różnorodne aspekty tekstu, takie jak kapitalizacja, kontekst występowania, struktury składniowe czy wzorce morfologiczne. Na przykład, reguła może określać, że sekwencja słów rozpoczynających się wielką literą, poprzedzona słowem „prezydent", prawdopodobnie reprezentuje nazwisko osoby. Systemy regułowe często wykorzystują wyrażenia regularne do definiowania wzorców oraz parsery do analizy struktury składniowej zdań.
  Zaletą podejścia regułowego jest jego interpretowalność, można dokładnie prześledzić, dlaczego system podjął konkretną decyzję. Reguły mogą być precyzyjnie dostosowane do specyfiki danej domeny lub języka, co pozwala na osiągnięcie wysokiej skuteczności w wąskich zastosowaniach. Głównym wyzwaniem jest jednak pracochłonność tworzenia kompleksowego zestawu reguł oraz trudność w obsłudze wszystkich możliwych przypadków i wyjątków. Systemy regułowe często zawodzą przy przetwarzaniu tekstów o nieoczekiwanej strukturze lub stylu.

```
text = "Pani Nowak mieszka w Krakowie, a Pan Kowalski pracuje w Gdańsku od 2024 roku."

rules = {
    "PERSON": r"\b(?:Pan|Pani)\s[A-Z][a-ząćęłńóśźż]+\b",
    "DATE": r"\b\d{4}\b"
}
def rule_based_ner(text, rules):
    entities = []

    for entity_type, pattern in rules.items():
        for match in re.finditer(pattern, text):
            entity = match.group()
            entities.append((entity, entity_type))
    return entities
```

- *Podejście oparte na uczeniu maszynowym* zrewolucjonizowało dziedzinę NER, umożliwiając automatyczne uczenie się wzorców identyfikacji encji z oznaczonych danych treningowych. Klasyczne algorytmy ML, takie jak ukryte modele Markowa (HMM), warunkowe pola losowe (CRF) czy maszyny wektorów nośnych (SVM), traktują NER jako problem etykietowania sekwencji. Modele te wykorzystują bogaty zestaw cech ekstraktowanych z tekstu, obejmujących cechy leksykalne (same słowa), morfologiczne (końcówki, przedrostki), syntaktyczne (części mowy, zależności składniowe) oraz kontekstowe (słowa sąsiednie, pozycja w zdaniu).
  Proces tworzenia systemu NER opartego na uczeniu maszynowym rozpoczyna się od przygotowania korpusu treningowego z ręcznie oznaczonymi encjami. Następnie definiuje się zestaw cech istotnych dla danego zadania i trenuje wybrany model na przygotowanych danych. Kluczową zaletą tego podejścia jest zdolność do generalizacji, czyli model może nauczyć się rozpoznawać encje niewystępujące w danych treningowych, jeśli mają podobne cechy do znanych przykładów. Modele ML potrafią również lepiej radzić sobie z niejednoznacznością i szumem w danych niż metody regułowe.

- *Podejście oparte na głębokim uczeniu* reprezentuje najnowocześniejsze podejście do rozpoznawania nazwanych encji, osiągając obecnie najlepsze wyniki na większości benchmarków. Architektura typowego systemu DL dla NER opiera się na wielowarstwowych sieciach rekurencyjnych (RNN), szczególnie LSTM (*Long Short-Term Memory*) lub GRU (*Gated Recurrent Units*), często w konfiguracji dwukierunkowej (BiLSTM). Sieci te są w stanie automatycznie uczyć się złożonych reprezentacji tekstu, uwzględniających długodystansowe zależności i subtelne wzorce kontekstowe.
  Przełomowym momentem w rozwoju metod głębokiego uczenia dla NER było wprowadzenie wstępnie wytrenowanych modeli językowych, takich jak BERT, GPT czy RoBERTa. Modele te, wytrenowane na ogromnych korpusach tekstowych, zawierają bogate reprezentacje języka, które można dostosować do konkretnego zadania NER poprzez proces fine-tuningu. Wykorzystanie transferu wiedzy z modeli pretrenowanych znacząco poprawia skuteczność systemów NER, szczególnie w przypadku ograniczonej ilości danych treningowych dla konkretnej domeny. Dodatkowo, modele transformerowe wprowadzają mechanizm uwagi (attention), który pozwala systemowi skupić się na najbardziej istotnych częściach kontekstu przy podejmowaniu decyzji o klasyfikacji encji.

1.4. Rola kapitalizacji w rozpoznawaniu encji

Kapitalizacja stanowi jeden z najważniejszych elementów wykorzystywanych w systemach rozpoznawania nazwanych encji, szczególnie w językach używających alfabetu łacińskiego. W większości języków europejskich istnieją ustalone konwencje dotyczące użycia wielkich liter, które często sygnalizują obecność nazw własnych. Analiza wzorców kapitalizacji pozwala na wstępną identyfikację potencjalnych encji bez konieczności głębokiej analizy semantycznej tekstu. System analizujący kapitalizację może szybko wyodrębnić kandydatów na encje, co znacząco przyspiesza cały proces NER.
Wykorzystanie kapitalizacji jako cechy w systemach NER wymaga jednak uwzględnienia kontekstu występowania wielkiej litery. Początek zdania naturalnie rozpoczyna się wielką literą, co nie świadczy o obecności encji. Podobnie, w niektórych tekstach, szczególnie w mediach społecznościowych czy komunikatorach, użytkownicy mogą stosować niekonwencjonalną kapitalizację dla podkreślenia emocji czy zwrócenia uwagi. Systemy NER muszą zatem implementować mechanizmy rozróżniające między kapitalizacją znaczącą a przypadkową czy stylistyczną.

1.5. SpaCy

SpaCy to jedna z najpopularniejszych bibliotek do przetwarzania języka naturalnego, oferująca wydajne i dokładne narzędzia do różnorodnych zadań NLP, w tym rozpoznawania nazwanych encji. Biblioteka została zaprojektowana z myślą o zastosowaniach produkcyjnych, oferując wysoką wydajność przetwarzania oraz łatwość integracji z aplikacjami. SpaCy dostarcza wstępnie wytrenowane modele dla wielu języków, w tym języka polskiego, które można wykorzystać bezpośrednio lub dostosować do własnych potrzeb poprzez dodatkowy trening.
Architektura spaCy opiera się na potokach przetwarzania (pipelines), gdzie tekst przechodzi przez szereg komponentów wykonujących kolejne zadania: tokenizację, tagowanie części mowy, parsowanie zależności, lemmatyzację oraz rozpoznawanie encji. Każdy komponent może wykorzystywać wyniki poprzednich etapów, co pozwala na bardziej kontekstową i dokładną analizę. Model NER w spaCy wykorzystuje zaawansowane techniki głębokiego uczenia, w tym sieci neuronowe z mechanizmami uwagi, co zapewnia wysoką skuteczność rozpoznawania encji.

Aby rozpocząć pracę ze spaCy dla języka polskiego, należy najpierw zainstalować bibliotekę oraz odpowiedni model językowy. SpaCy oferuje kilka modeli dla języka polskiego różniących się rozmiarem i dokładnością.
- model podstawowy (sm - small) jest kompaktowy i szybki,
- model średni (md - medium) oferuje lepszą dokładność,
- model duży (lg - large) zapewnia najlepsze wyniki kosztem większego rozmiaru i wolniejszego przetwarzania.

Instalacja w terminalu modelu podstawowego:
```
pip install spacy
```

python -m spacy download pl_core_news_sm

```
1  import spacy
2
3  nlp = spacy.load("pl_core_news_sm")
4
5  print("Rozpoznawane typy encji:", nlp.get_pipe("ner").labels)
6
7  tekst = """Google zostało założone 4 września 1998 roku w Menlo Park w Kalifornii przez
   Larryego Pagea i Sergeya Brina, a w 2015 roku stało się częścią spółki-matki Alphabet Inc.
   z siedzibą w Mountain View. Jest godzina 23:59."""
8
9  doc = nlp(tekst)
10
11 print("\nZnalezione encje:")
12 for ent in doc.ents:
13     print(f"{ent.text} | {ent.label_}")
```

Listing 1. Przykład użycia NER z biblioteką spaCy

Listing 1 pokazuje jak przy pomocy spaCy rozpoznawać encje nazwane (osoby, organizacje, miejsca, daty, czas) w tekście po polsku. W linii 1 zostaje zaimportowana biblioteka *spacy*, a następnie zostaje załadowany polski model językowy *pl_core_news_sm* do zmiennej *nlp* (linia 3). Ten model zawiera m.in. komponent NER. Następnie, w linii 5 wyciągany jest ten komponent przez *nlp.get_pipe("ner")* i wypisywane są jego etykiety (*.labels*) — to po prostu lista typów encji, jakie model potrafi rozpoznawać.
Dalej, w linii 7, tworzona jest zmienna *tekst* z kilkoma zdaniami o Google z nazwami własnymi. Ten tekst jest przepuszczony przez potok (linia 9), co uruchamia wszystkie komponenty modelu i zwraca obiekt *doc* z analizą. Najważniejsze tutaj jest pole *doc.ents* - to lista znalezionych encji. Pętla przechodzi po tych encjach i dla każdej wypisuje się dokładny fragment z oryginalnego tekstu oraz etykietę typu (linie 12-13).
Wynik uruchomienia:

```
Rozpoznawane typy encji: ('date', 'geogName', 'orgName', 'persName', 'placeName', 'time')

Znalezione encje:
Google | orgName
4 września 1998 roku | date
Menlo Park | geogName
Kalifornii | placeName
Larryego Pagea | orgName
Sergeya Brina | persName
2015 roku | date
Alphabet Inc. | persName
Mountain View | persName
godzina 23:59 | time
```

Zadania do wykonania

Zadanie 1: Implementacja prostego systemu NER opartego na regułach

Należy stworzyć system rozpoznawania encji wykorzystujący reguły i wyrażenia regularne. System powinien rozpoznawać:

- Osoby: wzorzec „Pan/Pani + Imię + Nazwisko” lub sekwencje słów z wielkiej litery.
- Daty: różne formaty dat (DD.MM.YYYY, DD-MM-YYYY, słowne).
- Kwoty pieniężne: liczby z walutami (PLN, EUR, USD, zł).
- Adresy email i strony WWW.

```
def rozpoznaj_encje_reguly(tekst):
    """
    Implementacja prostego systemu NER opartego na regułach.
    Zwraca słownik z listami znalezionych encji według typu.
    """
    # TODO: Implementacja
    pass

# Test na tekście:
tekst_test = """
Pan Jan Kowalski z firmy ABC Sp. z o.o. przesłał fakturę na kwotę 5000 PLN.
Spotkanie odbędzie się 15.03.2024 w siedzibie Fundacji Rozwoju.
Kontakt: jan.kowalski@firma.pl lub www.firma.pl
Pani Anna Nowak z Uniwersytet Warszawski przedstawiła raport.
Wczoraj otrzymaliśmy przelew na 2500,50 zł.
Stowarzyszenie Pomocy Dzieciom organizuje zbiórkę 10 grudnia 2024 r.
```

Zadanie 2: Analiza porównawcza metod rozpoznawania encji

Porównać skuteczność trzech metod NER na tym samym zestawie tekstów:

1. Metoda słownikowa (przygotować słowniki dla każdego typu encji)
2. Metoda oparta na kapitalizacji
3. SpaCy

Dla każdej metody należy:

- Zaimplementować rozpoznawanie encji
- Stworzyć raport porównawczy

```
teksty_testowe = [
    """Prezes POL Jan Kowalski spotkał się z przedstawicielami Microsoft w Warszawie. Rozmowy dotyczyły współpracy w zakresie cyfryzacji.""",
    """Anna Nowak z Uniwersytet Warszawski przedstawiła wyniki badań podczas konferencji w Krakowie. Google i Apple sponsorowały wydarzenie.""",
    """Ministerstwo Finansów ogłosiło nowy program wsparcia dla firm. Adam Kowalski z NBP skomentował decyzję podczas konferencji prasowej."""
]
```