NLP - dokumentacja wstępna

Porównanie różnych typów tagowania w zadaniu rozpoznawania jednostek nazewniczych.

Rafał Bosko
Paweł Bęza
Jakub Niewiński

Cel projektu **1**

Sposoby adnotacji **1**
IO Encoding 1
BIO Encoding 2
BMEWO Encoding 2
IOB Encoding - Inside–outside–beginning 2

Zbiory danych **2**
1. CoNLL 2003 2
2. BLURB 3

Modele **3**
1. SpaCy 3
2. BERT 3
3. Własny model 5

Cel projektu

Celem projektu jest porównanie modeli tagowania słów dla kilku wybranych zbiorów danych oraz sposobów adnotacji słów.

Sposoby adnotacji

Porównanie testów kodowania ma na celu sprawdzenie jak kolejne i bardziej rozbudowane sposoby kodowania wpływają na wydajność i skuteczność klasyfikacji słów.

IO Encoding

Jest to najprostsze, które oznacza każdy token jako znajdujący się w (I_X) w określonej klasie X lub nie należy do żadnej klasy (O). Wadą tego kodowania jest brak możliwości reprezentowania sąsiedztwa klas, ponieważ nie ma znacznika granicy.

BIO Encoding

Jest to kodowanie podobne do IO, lecz nieco bardziej złożone od poprzedniego, dzieli tagi in na początek encji (B_X) lub kontynuację encji (I_X).

BMEWO Encoding

Kodowanie BMEWO dodatkowo odróżnia tokeny end-of-entity (E_X) od tokenów mid-entity (M_X) i dodaje zupełnie nowy znacznik dla jednostek jednotokenowych (W_X).

https://aclanthology.org/W98-1118.pdf

IOB Encoding - Inside–outside–beginning

Prefiks I- przed tagiem wskazuje, że tag znajduje się w porcji. Znacznik O wskazuje, że token nie należy do żadnego fragmentu. Prefiks B- przed znacznikiem wskazuje, że znacznik jest początkiem porcji, która następuje bezpośrednio po innej porcji bez znaczników O między nimi. Jest używany tylko w tym przypadku: kiedy porcja występuje po znaczniku O, pierwszy token porcji przyjmuje prefiks I-. Przykładowe kodowanie IOB:

```
Alex I-PER
is O
going O
to O
Los I-LOC
Angeles I-LOC
in O
California I-LOC
```

Zbiory danych

1. CoNLL 2003

Zbiór danych zawiera otagowane dane w języku:

- angielskim - pochodzące z korpusu tekstu Reuters'a zawierającego wiadomości z okresu 09/1996 - 09/1997.
- niemieckim - pochodzące z korpusu tekstu *ECI* zawierającego wiadomości z niemieckiej gazety *Frankfurter Rundshau*.

Każde słowo posiada jedną z 4 kategorii:

1. osoba
2. lokalizacja
3. organizacja
4. inne

Powyższa informacja została zakodowana przy pomocy adnotacji typu IOB.

2. BLURB

Interesują nas tutaj podzbiory wykorzystujące kodowanie IOB (np. NCBI-diesease, BC5CDR-chem…). Podzbiory te zawierają otagowane zdania w języku angielskim z dziedzin biologii (konkretnie bakteriologii i chemii.

Modele

1. SpaCy

SpaCy to biblioteka typu open source do przetwarzania języka naturalnego w Pythonie. Została zaprojektowana do użytku produkcyjnego i pomaga tworzyć aplikacje, które przetwarzają i rozpoznają duże ilości tekstu.

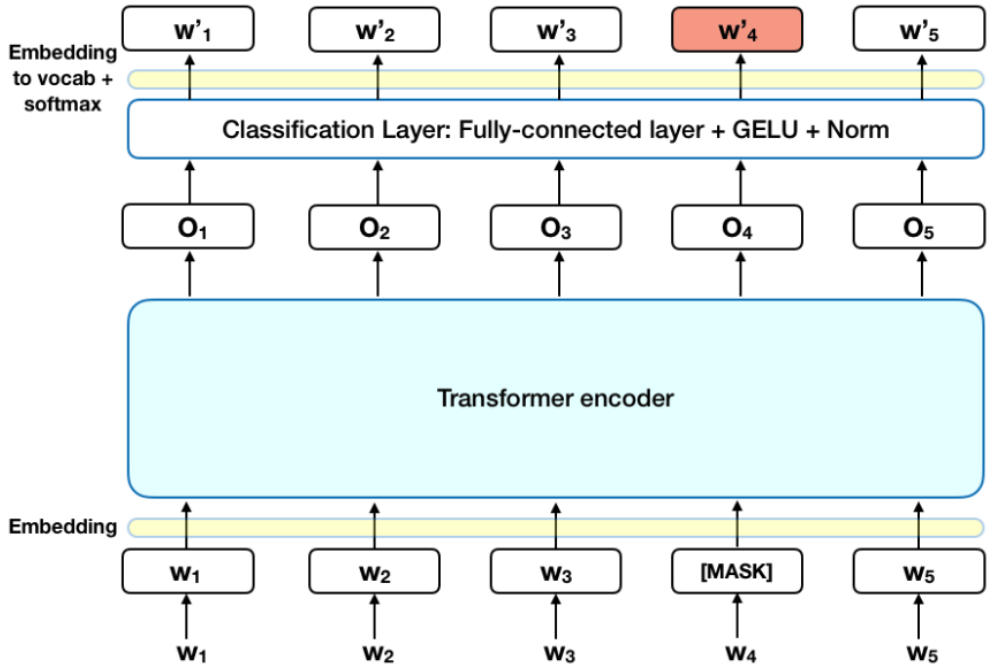
SpaCy zapewnia wyjątkowo wydajny system statystyczny dla NER w Pythonie, który może przypisywać etykiety do grup tokenów. Zapewnia domyślny model, który może rozpoznawać szeroki zakres klas, w tym osoby, organizacje, język, wydarzenia itp. Oprócz tych domyślnych, spaCy daje nam również możliwość dodawania dowolnych klas do modelu NER.

Implementacja będzie polegała na wykorzystaniu gotowego algorytmu z biblioteki SpaCy, dodaniem interesujących nas klas, a następnie ponownym uczeniu sieci. Wykorzystane do tego będą wewnętrzne funkcjonalności tej biblioteki.

2. BERT

Model BERT (ang. *Bidirectional Encoder Representations from Transformers*), jak sama nazwa wskazuje opiera swoje działanie na architekturze transformera, którego główną ideą jest wykorzystanie mechanizmu atencji. Jednak w przeciwieństwie do pierwowzoru BERT nie posiada dekodera, ponieważ jego zadaniem jest jedynie na stworzenie modelu językowego.

Na poniższym obrazku przedstawiono wysoko poziomowy schemat wspomnianej architektury:



Kolejną innowacją modelu BERT był sposób zdefiniowania celu predykcji. W przeciwieństwie do innych modeli, które za cel predykcji obierały kolejny wyraz w sekwencji model ten używał 2 technik:

- MLM (ang. *Masked Language - Modelling*) - 15% wyrazów przekazywanych na wejście do modelu zamieniane jest na token *[MASK]*. Następnie model próbuje przewidzieć zamaskowane wyrazy bazując na kontekście innych nie zamaskowanych wyrazów.
- NSP (ang. Next Sentence Prediction) - model otrzymuje pary zdań, na które składa się:
    - 50% par zdań występujących obok siebie
    - 50% par zdań w losowej odległości od siebie

  Model ma dodatkowo za zadanie przewidzieć czy drugie zdanie wsystępuję po zdaniu pierwszym.

Aby wytrenować model pod zadanie rozpoznawania jednostek nazewniczych można dodać na koniec modelu BERT warstwę klasyfikującą, która zwróci jedną z jednostek nazewniczych.

3. Własny model

Ostatnim modelem, który zostanie zaimplementowany będzie nasza własna rekurencyjna sieć neuronowa. Będzie oparta o komórki LSTM (Long short-term memory). Każda komórka przechowuje swój stan i wartość wyjściową, na których podstawie oblicza swoje wyjście. LSTM zostały opracowane, aby poradzić sobie z problemem zanikającego gradientu, który można napotkać podczas uczenia tradycyjnych RNN. Poniżej został przedstawiony rysunek przykładowej komórki LSTM. Dokładna struktura (liczba, wielkość i rodzaj warstw) zostanie dobrana przy procesie uczenia.