

Narzędzie do Rozpoznawania Języka oparte na Bibliotece fastText

Autorzy

Bartłomiej Milecki, Mariusz Gamrat, Szymon Jura

1. Abstrakt

Niniejszy projekt skupia się na implementacji narzędzia do rozpoznawania języka, które zostało oparte na bibliotece fastText. W tym projekcie użyte zostały dwa modele. Jeden wstępnie zdefiniowany, drugi wytrenowany na podstawie dostarczonych danych.

2. Wstęp

2.1 Cel

Celem projektu było stworzenie narzędzia, które jest w stanie rozpoznać język tekstu na podstawie modeli generowanych przez bibliotekę fastText. Rozpoznawanie języka jest kluczowym elementem w wielu dziedzinach, takich jak tłumaczenie maszynowe, analiza sentymentu czy przetwarzanie języka naturalnego. Stworzenie efektywnego narzędzia do identyfikacji języka może przyczynić się do poprawy jakości i efektywności tych systemów.

2.2 Zakres

Zakres projektu obejmował implementację dwóch metod rozpoznawania języka: jednej korzystającej z predefiniowanego modelu dostarczanego przez fastText, a drugiej wykorzystującej model wytrenowany na podstawie dostarczonych danych. Zdecydowano się na dwa podejścia, aby zobaczyć, jak różnią się wyniki i jakie są mocne strony każdego z nich.

2.3 Metodyka

Projekt został zrealizowany za pomocą języka Python oraz biblioteki fastText, która jest często stosowana w przetwarzaniu języka naturalnego. Dodatkowo, do obsługi kodów języka, użyto biblioteki iso_language_codes, która umożliwia tłumaczenie kodów języka ISO na pełne nazwy języków.

I Część Teoretyczna

Biblioteka fastText

fastText to biblioteka do przetwarzania języka naturalnego, która umożliwia tworzenie modeli uczenia maszynowego. Została stworzona przez Facebook's AI Research (FAIR) lab i jest szczególnie skuteczna w rozpoznawaniu języków. Biblioteka ta umożliwia tworzenie zarówno modeli uczenia nadzorowanego, jak i nienadzorowanego, co czyni ją bardzo uniwersalnym narzędziem do przetwarzania języka naturalnego.

Uczenie nadzorowane

W projekcie wykorzystano metodę uczenia nadzorowanego, która polega na trenowaniu modelu na danych, dla których znamy oczekiwane wyniki. Model jest następnie w stanie przewidywać wyniki dla nowych, nieznanych danych. Jest to powszechna metoda w uczeniu maszynowym, która pozwala na precyzyjne prognozowanie wyników na podstawie wcześniej nauczonych wzorców.

II Część Praktyczna

3. Implementacja

Projekt składa się z jednej funkcji, detectLanguage, która przyjmuje dwa argumenty: opcję modelu oraz tekst do analizy. Opcja "1" korzysta z predefiniowanego modelu fastText, natomiast opcja "2" korzysta z modelu wytrenowanego na dostarczonych danych. Wybór między dwoma modelami daje użytkownikowi większą elastyczność w zastosowaniu narzędzia, umożliwiając mu wybór najbardziej odpowiedniego dla danego zadania.

```
def detectLanguage(option ,text):  
    if(option == "1"):  
        #załadowanie gotowego modelu  
        model = fasttext.load_model("lid.176.bin")  
  
        #analiza języka  
        predictions = model.predict(text, k=3)  
        print(predictions)  
  
    if(option == "2"):  
        #trening na podstawie własnych danych  
        model = fasttext.train_supervised("./training_data.txt",  
                                          epoch = 132,  
                                          lr=0.5,  
                                          dim = 200,  
                                          ws = 5,  
                                          loss="softmax")  
  
        #analiza języka  
        predictions = model.predict(text, k=3)  
        print(predictions)
```

4. Wyniki

Funkcja zwraca kod języka, który został rozpoznany dla danego tekstu. Kod ten jest następnie tłumaczony na pełną nazwę języka za pomocą biblioteki `iso_language_codes`. W przypadku predefiniowanego modelu, wynik jest generowany na podstawie dużej ilości danych, na których model był trenowany. Natomiast w przypadku modelu wytrenowanego na dostarczonych danych, wynik jest bardziej specyficzny i dostosowany do konkretnego zestawu danych.

```
What model do you want to use?
1-predefined
2-trained
3-trained on more data
1
Put in a phrase to analyse: J'ai envie de manger du poulet
Warning : `load_model` does not return WordVectorModel or SupervisedModel any more, but a `FastText` object which is very similar.
(('__label_fr', '__label_en', '__label_es'), array([9.99217510e-01, 2.87825183e-04, 2.13748746e-04]))
(venv) PS C:\Users\bartek\Desktop\projektPJN> python projekt.py

What model do you want to use?
1-predefined
2-trained
3-trained on more data
2
Put in a phrase to analyse: J'ai envie de manger du poulet
Read 0M words
Number of words: 1205
Number of labels: 9
Progress: 100.0% words/sec/thread: 303005 lr: 0.000000 avg.loss: 0.057454 ETA: 0h 0m 0s
(('__label_polish', '__label_german', '__label_italian'), array([0.85764331, 0.05778584, 0.04924709]))
(venv) PS C:\Users\bartek\Desktop\projektPJN> python projekt.py

What model do you want to use?
1-predefined
2-trained
3-trained on more data
3
Put in a phrase to analyse: J'ai envie de manger du poulet
Warning : `load_model` does not return WordVectorModel or SupervisedModel any more, but a `FastText` object which is very similar.
(('__label_french', '__label_portugal', '__label_spanish'), array([9.99989152e-01, 3.04011373e-05, 1.02709228e-05]))
(venv) PS C:\Users\bartek\Desktop\projektPJN>
```

```
What model do you want to use?
1-predefined
2-trained
3-trained on more data
1
Put in a phrase to analyse: tengo ganas de comer pollo
Warning : `load_model` does not return WordVectorModel or SupervisedModel any more, but a `FastText` object which is very similar.
(('__label_es', '__label_pt', '__label_gl'), array([0.93851745, 0.01211088, 0.00934522]))
(venv) PS C:\Users\bartek\Desktop\projektPJN> python projekt.py

What model do you want to use?
1-predefined
2-trained
3-trained on more data
2
Put in a phrase to analyse: tengo ganas de comer pollo
Read 0M words
Number of words: 1205
Number of labels: 9
Progress: 100.0% words/sec/thread: 440046 lr: 0.000000 avg.loss: 0.034476 ETA: 0h 0m 0s
(('__label_polish', '__label_portuguese', '__label_italian'), array([0.98300308, 0.01317898, 0.00345691]))
(venv) PS C:\Users\bartek\Desktop\projektPJN> python projekt.py

What model do you want to use?
1-predefined
2-trained
3-trained on more data
3
Put in a phrase to analyse: tengo ganas de comer pollo
Warning : `load_model` does not return WordVectorModel or SupervisedModel any more, but a `FastText` object which is very similar.
(('__label_spanish', '__label_french', '__label_portugal'), array([9.81117487e-01, 1.83755141e-02, 5.34785853e-04]))
```

```

What model do you want to use?
1-predefined
2-trained
3-trained on more data
1
Put in a phrase to analyse: Eu sinto vontade de ter frango
Warning : `load_model` does not return WordVectorModel or SupervisedModel any more, but a `FastText` object which is very similar.
((('__label_pt', '__label_gl', '__label_en'), array([0.97770739, 0.0078512 , 0.0035214 ])))
(venv) PS C:\Users\bartek\Desktop\projektPJN> python projekt.py

What model do you want to use?
1-predefined
2-trained
3-trained on more data
2
Put in a phrase to analyse: Eu sinto vontade de ter frango
Read 0M words
Number of words: 1205
Number of labels: 9
Progress: 100.0% words/sec/thread: 266224 lr: 0.000000 avg.loss: 0.062779 ETA: 0h 0m 0s
((__label_polish', '__label_portuguese', '__label_italian'), array([0.94829994, 0.03148298, 0.0199365 ]))
(venv) PS C:\Users\bartek\Desktop\projektPJN> python projekt.py

What model do you want to use?
1-predefined
2-trained
3-trained on more data
3
Put in a phrase to analyse: Eu sinto vontade de ter frango
Warning : `load_model` does not return WordVectorModel or SupervisedModel any more, but a `FastText` object which is very similar.
((__label_portugal', '__label_french', '__label_spanish'), array([9.94146705e-01, 5.54801291e-03, 3.35084042e-04]))

```

5. Podsumowanie

Celem projektu było stworzenie narzędzia do rozpoznawania języka za pomocą biblioteki fastText. Cel ten został osiągnięty poprzez implementację dwóch metod: jednej korzystającej z predefiniowanego modelu, a drugiej korzystającej z modelu wytrenowanego na dostarczonych danych. Oba modele okazały się skuteczne w swoich specyficznych zastosowaniach, co pokazuje, że wybór odpowiedniego modelu zależy od konkretnego zadania.

Bibliografia

1. Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of Tricks for Efficient Text Classification. ArXiv, abs/1607.01759.
2. FastText Documentation: <https://fasttext.cc/docs/en/support.html>
3. Python Documentation: <https://docs.python.org/3/>
4. iso_language_codes Documentation: <https://pypi.org/project/iso-language-codes/1.0>.