# Narzędzie do rozpoznawania języka za pomocą biblioteki fastText

Autorzy:

Wiktor Kudzia

Bartosz Matras

Damian Madej

Celem projektu było zbudowanie narzędzia do rozpoznawania tekstu za pomocą języka python z użyciem biblioteki fastText. Narzędzie po odczytaniu inputu wprowadzonego przez użytkownika język w jakim został napisany tekst.

Aby zainstalować bibliotekę na naszym urządzeniu użyliśmy komendy:

***pip install fasttext***

Następnie napisaliśmy kod, który na podstawie modelu udostępnionego przez fastText: https://fasttext.cc/docs/en/language-identification.html, jest w stanie przewidzieć język podanego tekstu.

```python
import fasttext

# Ładowanie modelu FastText do wykrywania języka
try:
    model = fasttext.load_model('lid.176.bin')

except FileNotFoundError:
    print('Nie znaleziono pliku z modelem FastText')
    model = None

def detect_language(text):
  if model is None:
      return 'Brak dostępnego modelu FastText'

  # Predykcja języka dla podanego tekstu
  try:
      prediction = model.predict(text)
      print(prediction)
  except ValueError:
      return 'Niepoprawny tekst wejściowy'

  # Pobranie nazwy języka z predykcji
  language = prediction[0][0][9:]

  return language

# Przykładowe użycie narzędzia
text = input('Podaj tekst do sprawdzenia: ')
language = detect_language(text)
print(f'Język tekstu to: {language}')
```

Output dla tekstu w języku francuskim z wykorzystaniem modelu fastText:

```
Podaj tekst do sprawdzenia:
Voici un exemple de texte en français. Voici un autre exemple de texte en français.
(('__label__fr',), array([0.99600172]))
Język tekstu to: fr
```

Wynik programu pokazuje także w jakim stopniu program jest pewien języka (w tym wypadku 0.99600172). Gdzie jeden oznacza najwyższą dokładność.

Jak widać na powyższym zrzucie ekranu narzędzie wykrywa język poprawnie na dostarczonym przez fastText modelu. Następnie użyliśmy metody train_supervised, która zwróciła model na podstawie przykładowych wyrażeń w danych językach, podanych w pliku jako argument funkcji.

```python
import fasttext

# Ładowanie modelu FastText do wykrywania języka
try:
    model = fasttext.train_supervised('message.txt',
                                      epoch=50,
                                      lr=0.5,
                                      dim=100,
                                      ws=5,
                                      loss='softmax')
except FileNotFoundError:
    print('Nie znaleziono pliku z modelem FastText')
    model = None

def detect_language(text):
  if model is None:
      return 'Brak dostępnego modelu FastText'

  # Predykcja języka dla podanego tekstu
  try:
      prediction = model.predict(text)
      print(prediction)
  except ValueError:
      return 'Niepoprawny tekst wejściowy'

  # Pobranie nazwy języka z predykcji
  language = prediction[0][0][9:]

  return language

# Przykładowe użycie narzędzia
text = input('Podaj tekst do sprawdzenia: ')
language = detect_language(text)
print(f'Język tekstu to: {language}')
```

Metoda train_supervised posiada także takie fragmenty jak:

- "input" to argument, który określa źródło danych uczących do użycia do trenowania modelu. Może to być ścieżka do pliku z danymi uczącymi w formacie txt lub strumień danych uczących w formacie txt. Plik lub strumień muszą zawierać próbki tekstu z etykietami opisującymi klasę, do której należy dana próbka.

- "epoch" to argument, który określa liczbę epok trenowania modelu. Epoka to jedno przejście przez wszystkie próbki danych uczących. Wartość domyślna to 5. Im więcej epok trenowania, tym więcej czasu zajmie trenowanie modelu, ale może to również prowadzić do lepszych wyników.

- "lr" to argument, który określa współczynnik uczenia. Współczynnik uczenia to stała określająca, jak szybko model uczy się z danych uczących. Wartość domyślna to 1.0. Im większy współczynnik uczenia, tym szybsze uczenie się modelu, ale może to również prowadzić do większej chwiejności modelu.

- "dim" to argument, który określa wymiar wektora słowa. Wektor słowa to wektor numeryczny reprezentujący słowo w przestrzeni cech. Wartość domyślna to 100. Im większy wymiar wektora słowa, tym więcej informacji może być zawarte w wektorze, ale może to również prowadzić do większego zużycia pamięci i dłuższego czasu trenowania.

- "ws" to argument, który określa okno kontekstu. Okno kontekstu to liczba słów po lewej i prawej stronie danego słowa, które są brane pod uwagę podczas trenowania modelu. Wartość domyślna to 5. Im większe okno kontekstu, tym więcej informacji o kontekście słowa jest uwzględnianych podczas trenowania modelu, ale może to również prowadzić do większego zużycia pamięci i dłuższego czasu trenowania.

- "loss" to argument, który określa funkcję straty do użycia podczas trenowania modelu. Funkcja straty mierzy jakość modelu podczas trenowania i jest używana do aktualizowania wag modelu w celu lepszego dopasowania do danych uczących. Wartość domyślna to "softmax". Inne możliwe opcje to "hs" (hierarchiczny softmax), "ns" (negatywny sampling) lub "ova" (one-versus-all).

Po dogłębnej analizie stwierdziliśmy, że wartości wpisane na zrzucie ekranu kodu dają najlepszy stosunek czasu uczenia do dokładności wyniku.

```
Podaj tekst do sprawdzenia: siema
(('__label__polish',), array([0.90349704]))
Język tekstu to: polish
```

Narzędzie poprawnie wytypowało język z dokładnością ~0,9.

Poniżej umieszczamy fragment pliku „.txt” z przykładowymi wyrażeniami na podstawie których uczyliśmy program.

```
__label__polish "Nie mogę uwierzyć, ile pracy mam do zrobienia dzisiaj!" wykrzyknął prezydent Biden.
__label__polish "Nie mogę uwierzyć, jak ciężki jest ten test!" powiedziała studentka.
__label__polish "Czy naprawdę muszę iść na tę nudną konferencję?" zastanawiał się menedżer.
__label__polish "Nie mogę znaleźć mojego ulubionego kubka!" zmartwiła się pani domu.
__label__polish "Nie mogę uwierzyć, jak szybko upłynął ten rok!" stwierdziła nauczycielka.
__label__english "I can't believe how much work I have to do today!" President Biden exclaimed.
__label__english "I can't believe how difficult this test is!" said the student.
__label__english "Do I really have to go to this boring conference?" wondered the manager.
__label__english "I can't find my favorite mug!" lamented the housewife.
__label__english "I can't believe how quickly this year has gone by!" noted the teacher.
__label__english This is an example of English text.
__label__french "Je n'arrive pas à croire combien de travail j'ai à faire aujourd'hui!" s'exclama le président Biden.
__label__french "Je n'arrive pas à croire à quel point ce test est difficile!" a dit l'étudiant.
__label__french "Est-ce que je dois vraiment aller à cette conférence ennuyeuse?" s'est demandé le manager.
__label__french "Je ne trouve pas mon mug préféré!" a déploré la femme au foyer.
__label__french "Je n'arrive pas à croire à quelle vitesse cette année est passée!" a remarqué l'enseignante.
__label__french Voici un exemple de texte en français.
__label__german Hier ist ein Beispiel für deutschen Text.
__label__spanish Aquí hay un ejemplo de texto en español.
```

__label__[nazwa języka] oznacza etykietę języka, z którą dane wyrażenie jest identyfikowane. Etykieta jest wyświetlana jako wynikowy język wprowadzonego przez nas inputu.