

Narzędzie do rozpoznawania języka oparte na bibliotece fastText

Autorzy:

Dawid Laska

Jakub Kurc

Jan Kołek

Projekt dotyczy stworzenia programu do automatycznego rozpoznawania języka na podstawie wprowadzonego tekstu. W tym celu wykorzystano gotowy model **FastText**¹, który potrafi identyfikować 176 języków, oraz model opracowany przez autorów projektu, zaprojektowany do rozpoznawania pięciu języków: angielskiego, polskiego, hiszpańskiego, francuskiego i niemieckiego. Modele te działają równolegle, co umożliwia porównanie ich precyzji oraz czułości w klasyfikowaniu tekstu do odpowiednich etykiet językowych.

Celem projektu jest stworzenie skutecznego i wydajnego rozwiązania, które umożliwi automatyczne identyfikowanie języka w tekście. Rozwiązanie to ma istotne zastosowanie w wielu obszarach, takich jak tłumaczenia, analiza nastrojów czy personalizacja treści. Automatyczne rozpoznawanie języka pozwala na szybkie i precyzyjne przetwarzanie tekstu w zależności od jego języka, co znacząco wspiera efektywność procesów komunikacyjnych i analizy tekstu.

W ramach tego projektu zespół opracował kompleksowe rozwiązanie umożliwiające identyfikację języka w tekście. Kluczowe elementy tego systemu zostaną teraz zaprezentowane.

Stworzyliśmy aplikację, która pozwala użytkownikowi na interakcję za pośrednictwem graficznego interfejsu użytkownika (rys. 1). Dzięki aplikacji użytkownik może wprowadzić dowolny tekst, a system automatycznie określa jego język i wyświetla wyniki. Interfejs został zaprojektowany przy użyciu **Flask**² – lekkiego frameworka do tworzenia aplikacji webowych w języku Python.

W części opartej na Flask, po wejściu na stronę główną renderowany jest szablon strony **index.html**. Kiedy użytkownik wypełni formularz i wyśle dane za pomocą metody „POST”, wprowadzony tekst jest analizowany przy użyciu załadowanych modeli. Wynik analizy przedstawia język, który najbardziej pasuje do wprowadzonego tekstu.

¹ <https://fasttext.cc/docs/en/language-identification.html>

² <https://flask.palletsprojects.com/en/stable/>

Language Identifier

Enter text:

☐ Use Fasttext model?

Identify Language

english

3

Interfejs graficzny zaimplementowaliśmy przy pomocy HTML i CSS.

³ Wygląd interfejsu graficznego

Aplikacja pozwala na wybranie modelu, który będzie odpowiedzialny za rozpoznanie języka.

Domyślnym modelem jest model opracowany przez zespół, natomiast po wybraniu opcji “Use FastText Model”, modelem odpowiedzialny za rozpoznanie będzie gotowy model FastText.

Language Identifier

Enter text:

Guten Morgen, wo ist meine Lieblingstasse? Ich kann es nicht finden

☐ Use Fasttext model?

Identify Language

german

4

Language Identifier

Enter text:

Guten Morgen, wo ist meine Lieblingstasse? Ich kann es nicht finden

☒ Use Fasttext model?

Identify Language

de

5

⁴ Działanie aplikacji z modelem domyślnym, opracowanym przez zespół

⁵ Działanie aplikacji z gotowym modelem FastText

Część odpowiedzialna za tłumaczenie powstała przy użyciu biblioteki fastText . W celu jej wykorzystania należy zainstalować moduł przy użyciu polecenia “pip install fasttext”.

Poniżej prezentujemy dane w formacie odpowiednim do trenowania modelu przez bibliotekę.

```
_label_english "Why do I have to wake up so early?" complained the student.  
_label_english "I don't know how I can get all of this done before the deadline!" wondered the office worker.  
_label_english "This cake tastes amazing!" exclaimed the chef.  
_label_english "Has anyone seen my keys?" asked the desperate driver.  
_label_english "I've always dreamed of going to Paris!" said the young girl.  
_label_french "Pourquoi dois-je me lever si tôt?" se plaignit l'étudiant.  
_label_french "Je ne sais pas comment je vais tout terminer avant la date limite!" se demanda l'employé.  
_label_french "Ce gâteau a un goût incroyable!" s'exclama le chef.  
_label_french "Quelqu'un a vu mes clés?" demanda le conducteur désespéré.  
_label_french "J'ai toujours rêvé d'aller à Paris!" dit la jeune fille.  
_label_german "Warum muss ich so früh aufstehen?" beschwerte sich der Student.  
_label_german "Ich weiß nicht, wie ich das alles vor der Deadline schaffen soll!" fragte sich die Büroangestellte.  
_label_german "Dieser Kuchen schmeckt fantastisch!" rief der Koch aus.  
_label_german "Hat jemand meine Schlüssel gesehen?" fragte der verzweifelte Fahrer.  
_label_german "Ich habe immer davon geträumt, nach Paris zu reisen!" sagte das junge Mädchen.  
_label_spanish "¿Por qué tengo que levantarme tan temprano?" se quejó el estudiante.  
_label_spanish "No sé cómo voy a terminar todo esto antes de la fecha límite!" se preguntó la trabajadora.  
_label_spanish "¡Este pastel sabe increíble!" exclamó el chef.  
_label_spanish "¿Alguien ha visto mis llaves?" preguntó el conductor desesperado.
```

6

Dzięki odpowiedniemu przygotowaniu plików proces trenowania modelu można uruchomić za pomocą funkcji `fasttext.train_supervised`, dostarczając jej odpowiednie parametry. Parametry te były wielokrotnie modyfikowane, aby osiągnąć jak najlepsze wyniki. Funkcja ta wymaga ustawienia następujących parametrów:

- **input**: ścieżka do pliku z danymi treningowymi.
- **label_prefix**: prefiks dodawany przed etykietami klas w pliku z danymi.
- **epoch**: liczba epok treningowych (tu 100). Niska wartość prowadzi do niedouczenia modelu, natomiast zbyt duża może skutkować przeuczeniem, co oznacza, że model będzie zbyt dopasowany do danych treningowych i słabo generalizował na nowe dane, co może skutkować niedokładnymi wynikami w praktyce.
- **lr (learning rate)**: współczynnik uczenia (tu 0.1). Zbyt mały sprawia, że proces uczenia przebiega wolno, natomiast zbyt duży może pogorszyć zdolność modelu do generalizacji, prowadząc do słabszych wyników na danych testowych.

⁶ Fragment danych u
żytych do trenowania modelu

- **wordNgrams**: długość n-gramów używanych do reprezentacji słów. Krótkie n-gramy mogą obniżyć skuteczność detekcji języka, natomiast zbyt długie mogą zwiększyć wymiarowość wektorów, co z kolei komplikuje proces trenowania i może prowadzić do przeuczenia.
- **bucket**: liczba kubeków wykorzystywanych do haszowania. Zbyt mała liczba zwiększa ryzyko kolizji haszy (różne słowa mogą być traktowane jako jedno), co obniża jakość modelu. Z kolei zbyt duża wartość niepotrzebnie wydłuża czas treningu i zwiększa zapotrzebowanie na pamięć.
- **dim**: liczba wymiarów wektorowej reprezentacji słów (tu 720). Niewystarczająca liczba wymiarów ogranicza zdolność modelu do uchwycenia złożonych informacji o słowach, co obniża jakość klasyfikacji. Zbyt duża wartość może prowadzić do przeuczenia i problemów z generalizacją na nowe dane.
- **thread**: liczba wątków wykorzystywanych do treningu.
- **ws** określa szerokość okna kontekstowego, czyli liczbę słów z sąsiedztwa, które są brane pod uwagę podczas tworzenia reprezentacji danego słowa (tu 5).
- **loss** określa funkcję straty używaną podczas trenowania modelu (tu softmax).

Dostępne wartości:

- **softmax**: Używana w klasyfikacji wieloklasowej, gdzie każda próbka należy dokładnie do jednej z klas.
- **hs** (Hierarchical Softmax): Struktura drzewa do przyspieszenia obliczeń. Używana w dużych zbiorach danych, gdy liczba klas jest bardzo duża.
- **ns** (Negative Sampling): Funkcja strat stosowana w celu uproszczenia obliczeń, szczególnie w problemach z wieloma negatywnymi przykładami.
- **ova** (One-vs-All): Wykorzystuje podejście „jeden kontra reszta” do klasyfikacji wieloklasowej.

```
model_txt = fasttext.train_supervised('model.txt',  
                                     epoch=100,  
                                     lr=0.1,  
                                     dim=720,  
                                     ws=5,  
                                     loss='softmax')
```

Dane generowane przez funkcje odpowiedzialne za testowanie i rozpoznawanie języka są pobierane ze zmiennych poprzez odwołania do rekordów o odpowiadających im indeksach, a następnie przekazywane do funkcji renderującej szablon **index.html**.

Opracowane narzędzie do rozpoznawania języka, oparte na bibliotece FastText, okazało się skutecznym rozwiązaniem, które znajduje zastosowanie w wielu aplikacjach wymagających analizy języka naturalnego. Testy wykazały, że precyzja modelu wynosi około 96%, co jest znakomitą wynikiem, biorąc pod uwagę niewielki rozmiar zbioru danych użytego do trenowania modelu. W niektórych, szczególnych przypadkach model opracowany przez nasz zespół przewyższył dokładnością gotowy model dostarczany przez FastText.

Biblioteka FastText to wszechstronne narzędzie, które poza rozpoznawaniem języka może być z powodzeniem wykorzystywane w wielu innych obszarach przetwarzania języka naturalnego.

Repozytorium z kodem projektu dostępne jest pod adresem:

<https://github.com/dlaska888/FastText-LanguageRecognition>