

Narzędzie do rozpoznawania języka w Python z wykorzystaniem biblioteki fastText

Projekt na przedmiot Przetwarzanie Języka Naturalnego

Autorzy:

Paweł Wójcik

Paweł Światłoń

Michał Warchoń

Jan Książek

Abstrakt

Problem wykrycia języka, w którym został napisany tekst, polega na analizie cech charakterystycznych różnych języków, takich jak słownictwo, składnia czy alfabet. Wyzwaniem jest dokładne rozpoznanie języka w przypadku krótkich tekstów, wielojęzycznych dokumentów oraz podobieństw między niektórymi językami. Dodatkowo, slang, regionalizmy i błędy gramatyczne mogą utrudniać poprawne przypisanie tekstu do właściwego języka.

Cel

Celem tego projektu jest stworzenie efektywnego narzędzia do automatycznego wykrywania języka tekstu przy użyciu biblioteki fastText w języku Python. Narzędzie to ma za zadanie szybko i precyzyjnie identyfikować język, w którym napisany jest dany fragment tekstu, co może znaleźć zastosowanie w wielu dziedzinach, takich jak analiza treści w mediach społecznościowych, przetwarzanie dokumentów czy systemy tłumaczeń automatycznych.

Zakres

Projekt obejmuje implementację programu w Pythonie, który wykorzystuje pretrenowane modele fastText do rozpoznawania języka tekstu. Zakres prac obejmuje:

1. Przygotowanie środowiska pracy i instalację niezbędnych bibliotek.
2. Integrację modelu fastText z kodem Pythona.
3. Testowanie i walidację rozwiązania na różnych zestawach danych tekstowych.
4. Optymalizację kodu pod kątem szybkości działania i precyzji detekcji.

Metodyka

Metodyka realizacji projektu obejmuje kilka kluczowych etapów:

1. **Instalacja i konfiguracja fastText:** Zainstalowanie biblioteki fastText oraz pobranie odpowiednich pretrenowanych modeli językowych.
2. **Przygotowanie danych testowych:** Zebranie i wstępne przetworzenie danych tekstowych w różnych językach, które posłużą do testowania i walidacji programu.
3. **Implementacja rozwiązania:** Napisanie skryptu w Pythonie, który za pomocą fastText będzie analizował teksty i identyfikował język.
4. **Testowanie i walidacja:** Przeprowadzenie testów na różnych zbiorach danych w celu oceny dokładności i wydajności programu, a następnie wprowadzenie ewentualnych usprawnień.
5. **Dokumentacja i analiza wyników:** Sporządzenie dokumentacji technicznej oraz analiza wyników uzyskanych w trakcie testowania, aby ocenić skuteczność rozwiązania i zidentyfikować możliwe obszary do dalszej optymalizacji.

Część teoretyczna

1. Modele fastText

FastText to biblioteka opracowana przez Facebook AI Research (FAIR) do efektywnego uczenia maszynowego na dużych zbiorach danych tekstowych. Jest wykorzystywana zarówno do klasyfikacji tekstu, jak i do uczenia reprezentacji słów (word embeddings).

- **Klasyfikacja tekstu:** FastText może być używany do klasyfikacji tekstu poprzez trenowanie modeli na oznaczonych danych. W przypadku rozpoznawania języka, model uczy się z danych zawierających zdania oznaczone etykietami języków. [1]
- **Word Embeddings:** FastText reprezentuje słowa jako wektory w przestrzeni wielowymiarowej, co pozwala na uchwycenie semantycznych i syntaktycznych podobieństw między słowami. Dodatkowo, FastText bierze pod uwagę n-gramy znaków, co pozwala lepiej radzić sobie z rzadkimi i nieznanymi słowami. [2]

2. Przetwarzanie języka naturalnego

Przetwarzanie języka naturalnego to dziedzina sztucznej inteligencji, która zajmuje się interakcją między komputerami a ludzkim językiem. W programie wykorzystano kilka kluczowych technik NLP:

- **Tokenizacja zdań:** Proces podziału tekstu na pojedyncze zdania. W programie użyto `sent_tokenize` z biblioteki NLTK, aby podzielić tekst na zdania przed analizą językową. [3]
- **Przetwarzanie wstępne:** Proces przygotowania tekstu do analizy, który może obejmować czyszczenie tekstu, usuwanie znaków specjalnych, normalizację i inne kroki. W programie przewidziano miejsce na dodatkowe kroki przetwarzania wstępnego.

Część praktyczna

Rdzeniem programu jest skrypt wykonany w języku python, który importuje biblioteki potrzebne wstępnego przetwarzania i rozpoznawania języka tekstu. Są to *fasttext*, *pandas* oraz *nlTK*. Skrypt został podzielony na logiczne części w postaci funkcji. Każda odpowiada za wyraźnie wyodrębnione zadanie. Oto ich opis:

- `Preprocess_text` - funkcja odpowiedzialna za wstępne przetworzenie tekstu poddawanego analizie.
- `Train_custom_model` - funkcja która trenuje własny model fastText. Odczytuje dane z plików TSV zawierające dane w różnych językach. Następnie zapisuje przetworzone dane do pliku `fasttext_data1.txt` w formacie zgodnym z fastText. Potem trenuje model i zapisuje go do pliku `custom_model.bin`.

- Predict_language - rozpoznaje tekst na podstawie modelu fastText i przetwarza wyniki predykcji.
- Predict_language_segments - tokenizuje tekst na zdania, przewiduje język dla każdego zdania osobno.
- Index - wysyła html do przeglądarki z formularzem do uzupełnienia tekstu, który potem będzie analizowany. Po przesłaniu formularza wykona się analiza tekstu i zostanie zwrócona predykcja wraz z poziomem ufności co do przewidzianego rozwiązania.

Podsumowanie

Program służy do automatycznego rozpoznawania języka, w którym napisany jest tekst, wykorzystując bibliotekę fastText oraz framework Flask. Program wykonuje następujące kroki: ładowanie modeli, przetwarzanie tekstu, predykcja języka. Program jest efektywnym narzędziem do analizy języka tekstu, łącząc zaawansowane techniki NLP i uczenia maszynowego z łatwym w użyciu interfejsem webowym.

Bibliografia

- <https://fasttext.cc/docs/en/supervised-tutorial.html> [1]
- <https://fasttext.cc/docs/en/python-module.html> [2]
- <https://www.nltk.org/book/ch07.html> [3]