

Politechnika Krakowska  
Wydział Informatyki i Telekomunikacji

# Tłumaczenie tekstu z wczytanego zdjęcia przy pomocy biblioteki FastText

---

Projekt z przedmiotu  
Przetwarzanie Języka Naturalnego

Jakub Szostak  
Grzegorz Świstak  
Hubert Tomana

## Spis treści

1. Abstrakt.....	3
2. Wstęp.....	3
a. Cel projektu.....	3
b. Zakres.....	3
c. Metodyka.....	3
3. Część teoretyczna.....	3
4. Część praktyczna .....	4
5. Podsumowanie.....	6
6. Bibliografia.....	6

## **1. Abstrakt**

Idea naszego projektu opiera się na wykorzystaniu biblioteki fastText do rozpoznania użytego języka w wybranych tekście. Dodatkowo uznaliśmy, że idealnym rozszerzeniem tego tematu będzie zaimplementowanie dodawania zdjęcia, z którego dany tekst będzie sczytywany, a następnie tłumaczenie go na wybrany przez użytkownika język. W wyniku tego powstał w pełni funkcjonalny, niezależny moduł, który może stać się cennym dodatkiem w przyszłych projektach..

## **2. Wstęp**

### **2.1 Cel projektu**

Celem naszego projektu było stworzenie programu do tłumaczenia tekstu znajdującego się na wczytanym zdjęciu oraz zapoznanie się z użytą w tym celu biblioteką fastText.

### **2.2 Zakres**

Zakres naszego projektu obejmuje zaimplementowanie wczytania zdjęcia do środowiska Google Colab. Następnie zapisanie w zmiennej tekstu znajdującego się na nim, by kolejno rozpoznać język, w jakim ów zdanie jest zapisane oraz przetłumaczyć je na język wybrany przez użytkownika. Po zakończeniu pracy nad projektem, stworzenie prezentacji oraz raportu.

### **2.3 Metodyka**

Projekt nasz napisany jest w języku Python, przy pomocy środowiska Google Colab. Opiera się na użyciu trzech bibliotek :

- EasyOcr – rozpoznanie tekstu na zdjęciu oraz zapisanie go do zmiennej
- FastText – rozpoznanie użytego języka w zdaniu
- GoogleTrans – przetłumaczenie go na język wybrany przez użytkownika

## **3. Część teoretyczna**

### **3.1 Sczytanie tekstu ze zdjęcia**

Za sczytanie tekstu z obrazka w naszym projekcie odpowiada narzędzie EasyOCR, posiadające swój pakiet pythonowy. Narzędzie to najlepiej radzi sobie z tekstem pokroju paragonów, ulotek czy wydrukowanego tekstu. Ze względu na ograniczenie narzędzia, możliwe jest tylko skonfigurowanie w sposób umożliwiający odczytywanie znaków z języków o tym samym alfabecie, przy czym domyślnie zawsze jest w stanie odczytać alfabet używany w języku angielskim.

### **3.2 Rozpoznanie języka**

Rozpoznanie języka zostało stworzone przy pomocy biblioteki fastText. Zawiera ona model sieci neuronowej zdolnej do rozpoznawania 176 języków (dokładna lista dostępna pod adresem

<https://fasttext.cc/docs/en/language-identification.html>). Proces rozpoznawania opiera się na modelach n-gramów. Oznacza to, że do analizy semantycznej wysyłane są fragmenty zdań zawierających ilość słów równej wartości n. Domyślnie ustawioną wartością w tej bibliotece jest n=3. Dodatkowo, stosowane jest osadzanie podstów (subword embeddings), czyli rozdzielanie słów na często występujące podśłowa. Znacząco pomaga to w rozpoznawaniu słów rzadko występujących lub nie pojawiających się wcale w korpusie, jak i w językach opartych o

### 3.3 Tłumaczenie tekstu

GoogleTrans to biblioteka, umożliwiająca korzystanie z prostego interfejsu API używającego Google Translate. Jest to wrapper wokół funkcji dostępnych na stronie, który przyjmując jako argumenty tekst oraz poszczególne języki, przeprowadza operacje tłumaczenia i wczytuje wynik do zmiennej.

## 4. Część praktyczna

Pierwszym krokiem było pobranie odpowiednich bibliotek oraz ich wczytanie :

```
!pip install googletrans==4.0.0-rc1
!pip install fasttext
!pip install easyocr

from googletrans import Translator
from google.colab import drive
import fasttext
from google.colab import files
from PIL import Image
import os
import easyocr
```

Następnie, łącząc się z dyskiem Google, wczytaliśmy znajdujący się tam model do rozpoznawania języka oraz utworzyliśmy instancję Readera z biblioteki easyOcr, zdolnego do rozpoznawania alfabetu łacińskiego oraz cyrylicy :

```
drive.mount('/content/drive')
model_path = '/content/drive/MyDrive/Colab Notebooks/model.bin'
```

```
model = fasttext.load_model(model_path)
reader = easyocr.Reader(['en', 'ru'])
```

Utworzyliśmy również funkcję do wykorzystania biblioteki GoogleTrans. Przyjmuje ona trzy argumenty :

- Text - tekst będący obiektem do tłumaczenia
- Source\_language – język, w którym został napisany dany tekst
- Target\_language – język, do którego chcemy przetłumaczyć dany fragment.

W funkcji zainicjowaliśmy instancję Translatora i, wywołując funkcję tłumaczenia, zwróciliśmy jej wynik :

```
def translate_text(text, source_language='pl' ,target_language='en'):  
    translator = Translator()  
    translation = translator.translate(text, src=source_language, dest=target_language)  
    return translation.text
```

By przetworzyć zdjęcie, zapisaliśmy je w pamięci podręcznej :

```
uploaded = files.upload()  
  
if len(uploaded) > 0:  
    image_path = list(uploaded.keys())[0]
```

I wywołaliśmy funkcję *readtext* obiektu Reader, zapisując wynik tekstowy w zmiennej *wynik* :

```
result = reader.readtext(image_path)  
for detection in result:  
    wynik = wynik + detection[1] + ' '
```


Zmienną *wynik* przekazaliśmy do wczytanego modelu, wywołując metodę *predict*. Jej wynik zapisany jest w postaci *\_\_label\_\_xx*, gdzie *xx* to skrót danego języka, dlatego, wywołując metodę *split*, uzyskaliśmy sam ów skrót :

```
predicted_language = model.predict(wynik)[0][0]  
language = predicted_language.split('__label__')[1]
```

By następnie został on przekazany do utworzonej przez nas funkcji *translate\_text*, w wyniku której otrzymaliśmy oczekiwany wynik : przetłumaczony tekst z wczytanego obrazu. Jako język wybraliśmy język polski :

```
translated_text = translate_text(wynik, source_language=language ,target_language='en')  
print(f"Original text: {wynik}")  
print(f"Translated text: {translated_text}")
```

Wynik działania programu :



```
Original text: Je sais nager depuis que je suis petite:  
Translated text: I can swim since I was little:
```

## 5. Podsumowanie

Nasz projekt bardzo dobrze spełnia swoje założenia. Poprawnie wczytuje zdjęcie oraz znajduje na nim fragment tekstowy, a następnie rozpoznaje jego język i tłumaczy na inny, wybrany przez użytkownika. Może on być bardzo cennym dodatkiem w aplikacjach mobilnych, a nawet podstawą jednej z nich. Posiada również opcje dalszego rozwoju, takie jak polepszenie sieci konwolucyjnej do sczytywania wiadomości tekstowej czy stworzenie własnego tłumacza tekstu, który, w przeciwieństwie do Google Translate, bierze pod uwagę również sentyment danego zdania.

## 6. Bibliografia

- <https://fasttext.cc/>
- <https://github.com/JaidedAI/EasyOCR>
- <https://pypi.org/project/googletrans/>
- <https://medium.com/mlearning-ai/tesseract-vs-keras-ocr-vs-easyocr-ec8500b9455b>
- „Enriching Word Vectors with Subword Information” Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov