

Narzędzie do rozpoznawania języka oparte na bibliotece fastText



```
from fasttext import load_model  
import tkinter as tk
```

Importujemy potrzebne biblioteki, aby program działał poprawnie:

- **load_model** - jest to funkcja z biblioteki Fasttext, która ładuje wcześniej trenowany model do pamięci
- **tkinter** - biblioteka służąca do stworzenia interfejsu graficznego

```
def identify_language(text):  
    model = load_model('lid.176.bin')  
  
    result = model.predict(text, threshold=0.8)  
  
    if result[0]:  
        lang_code = result[0][0].split('__label__')[1]  
        return lang_code  
    else:  
        return "Nie rozpoznano języka"
```

- Funkcja **identify_language** jest funkcją, która przyjmuje argument **text**.
- **text** - argument, który jest tekstem, którego język mamy rozpoznać
- **model = load_model('lid.176.bin')** - model **'lid.176.bin'** jest ładowany do pamięci. Jest to model z zestawu modeli FastText umożliwiający rozpoznawanie języków
- **result = model.predict(text, threshold=0.8)** - metoda predict jest wywoływana na tekście. Argument **threshold=0.8** określa minimalny poziom pewności prawidłowego wyniku

```
if result[0]:  
    lang_code = result[0][0].split('__label__')[1]  
    return lang_code  
else:  
    return "Nie rozpoznano języka"
```

- Jeśli w wynik zawiera jakiekolwiek informacje oznacza to, że model zidentyfikował język.
- `lang_code = result[0][0].split('__label__')[1]` - result zwraca wynik w postaci:

```
(( '__label__pl', ), array([0.97729325]))
```

- W tej linijce musimy wyodrębnić kod języka 'pl' z ciągu '__label__pl'.
- Jeśli model nie zidentyfikował języka to zwracana jest o tym informacja.

```
class LanguageIdentifierApp(tk.Tk):
    def __init__(self):
        super().__init__() # Inicjalizacja okna aplikacji
        self.title("Identyfikator Języka") # Ustawienie tytułu okna
        self.geometry("400x150") # Ustawienie rozmiaru okna
        self.text_entry = tk.Entry(self, width=50) # Tworzenie pola tekstowego
        self.text_entry.pack(pady=10) # Dodanie pola tekstowego do okna
        self.identify_button = tk.Button(self, text="Zidentyfikuj język", command=self.identify_language) # Tworzenie przycisku
        self.identify_button.pack() # Dodanie przycisku do okna
        self.result_label = tk.Label(self, text="", font=("Helvetica", 12)) # Tworzenie etykiety do wyświetlania języka
        self.result_label.pack(pady=20) # Dodanie etykiety do okna

    def identify_language(self):
        text = self.text_entry.get() # Pobieranie tekstu z pola tekstowego
        language = identify_language(text) # Wywołanie funkcji identyfikującej język
        self.result_label.config(text=f"Język: {language}") # Wyświetlanie języka w oknie
```

Konstruktor klasy **LanguageIdentifierApp** tworzy interfejs użytkownika z polem tekstowym do wprowadzania tekstu, przyciskiem do identyfikacji języka oraz etykietą do wyświetlania wyniku

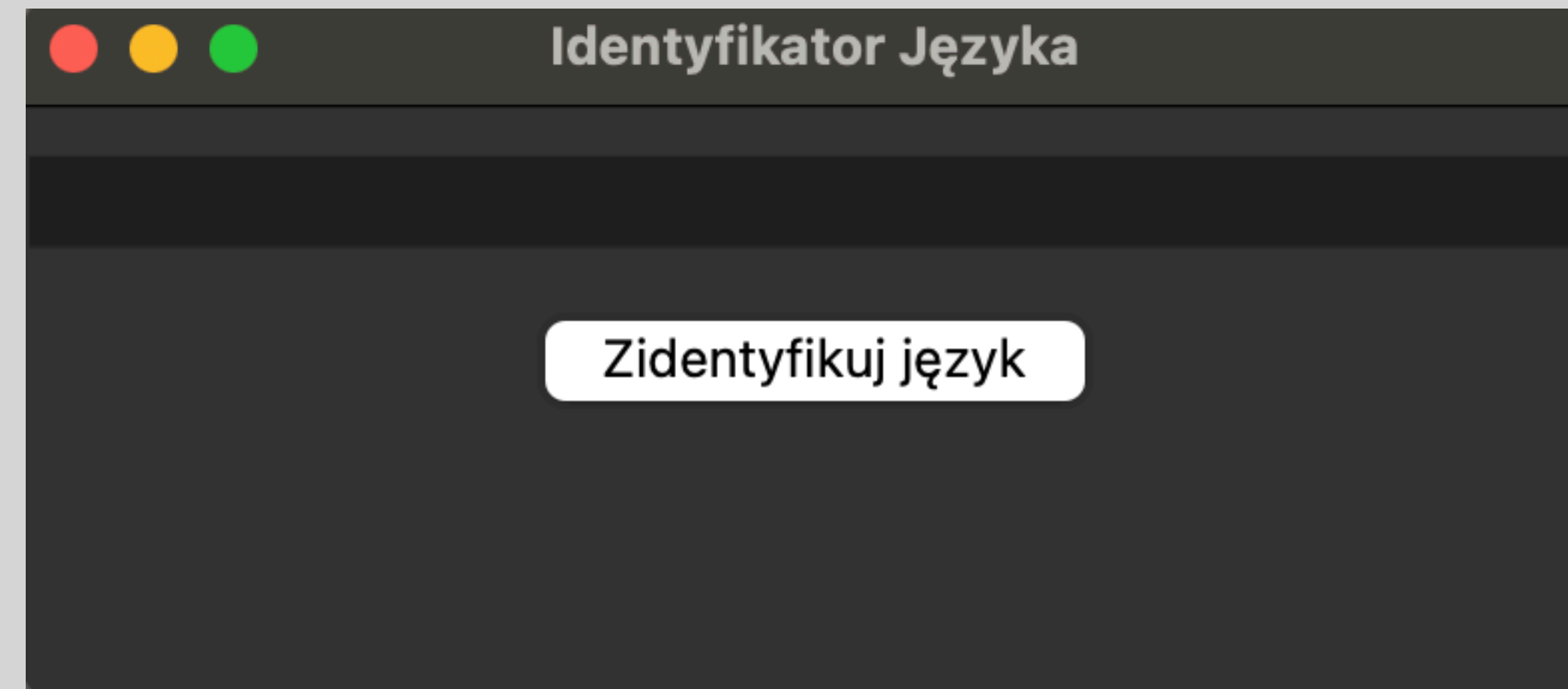

```
if __name__ == "__main__":  
    app = LanguageIdentifierApp()  
    app.mainloop()
```

W funkcji main tworzona jest instancja aplikacji a następnie jest ona uruchamiana

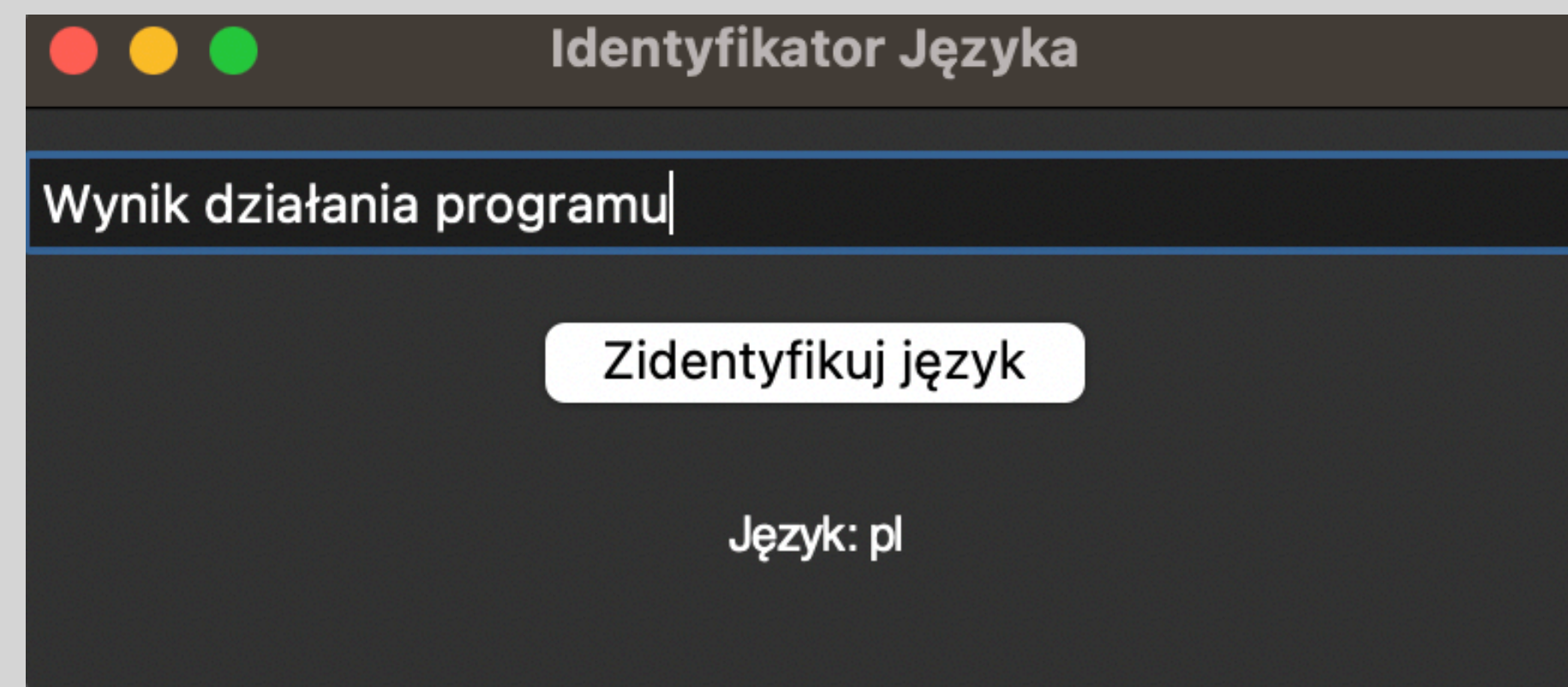
Uruchomienie programu

- Aby móc korzystać z programu należy zainstalować bibliotekę fasttext
pip3 install fasttext
- Należy również pobrać plik lid.176.bin poprzez poleceni wnet
wget <https://dl.fbaipublicfiles.com/fasttext/supervised-models/lid.176.bin>

Wynik działania programu



Po uruchomieniu programu pokazuję się okno z polem tekstowym, należy tam wpisać zdanie, którego język chcemy zidentyfikować a następnie kliknąć przycisk 'Zidentyfikuj język'



Dziękujemy za uwagę!

Szymon Twardowski
Mateusz Moneta