# Language Detection App Using FastText

**Authors:** [Alejandro Barreiro], [Helin Salduz], [José Francisco Brothers R]

Abstract

This project demonstrates the development of a language detection application using the FastText library. The tool can predict the language of a given text with high accuracy, leveraging a pre-trained model. The application also includes functionality for scraping text from web pages and loading text from files, making it versatile for real-world use. This report outlines the aim, scope, and methodology of the project, along with the theoretical background and practical implementation.

Introduction

The primary aim of this project is to create an application capable of accurately detecting the language of a given text. This includes the ability to process text entered by the user, text loaded from web pages, and text imported from files. The application uses FastText, a library known for its efficiency and accuracy in text classification tasks.

Scope

Developing a user-friendly graphical user interface (GUI) using Tkinter.

Integrating the FastText library for language detection.

Implementing web scraping functionality to extract text from web pages.

Allowing users to load text from local files for language detection.

Providing a top-three language prediction with confidence scores.

Methodology

To achieve the project's aims, the following methodology and tools were employed:

**Programming Language:** Python, due to its simplicity and extensive libraries.

**Library for Language Detection:** FastText, a library developed by Facebook's AI Research (FAIR) lab.

**GUI Framework:** Tkinter, a standard Python interface to the Tk GUI toolkit.

**Web Scraping Tools:** BeautifulSoup and urllib for extracting text content from web pages.

**Theoretical Part**

FastText Overview

FastText is a library for efficient learning of word representations and sentence classification. It is designed to handle large datasets efficiently and can train on millions of words in seconds. FastText uses n-gram features to improve performance on rare words, making it a powerful tool for language detection.

Language Detection Model

The pre-trained language identification model provided by FastText can predict 176 different languages. It works by using character n-grams and word embeddings to create a highly accurate classifier. The model is pre-trained on large datasets and can be used out-of-the-box for language identification tasks.

Web Scraping

Web scraping involves extracting data from websites. For this project, BeautifulSoup is used to parse HTML content, and urllib is used to fetch web pages. These tools allow us to extract text from web pages, which can then be processed by the FastText model for language detection.

**Practical Part**

The GUI for the language detection app is built using Tkinter. It includes:

A text entry widget for users to input text or URLs.

Buttons to trigger language detection, load web pages, and load files.

A label to display the results of the language detection.

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Functionality

**Text Detection:** Users can input text directly into the application, and upon clicking the "Detect Language" button, the top three language predictions with confidence scores are displayed.

**Web Page Loading:** Users can input a URL, and the application will scrape text content from the page. This text is then used for language detection.

**File Loading**: Users can load text from a local file, which is then displayed in the text entry widget for language detection.

Results

The application was tested with various text inputs in different languages. The FastText model provided accurate predictions with high confidence scores. The web scraping and file loading functionalities worked seamlessly, allowing for versatile input methods.

Summary

The project achieved its aim of creating a language detection application using FastText. The application can process user input, web pages, and text files, making it a versatile tool for language detection. The GUI is user-friendly, and the FastText model delivers accurate predictions. Future improvements could include adding support for more input formats and enhancing the user interface.

Bibliography

Facebook AI Research (FAIR). FastText Library. Retrieved from https://fasttext.cc

Tkinter Documentation. Retrieved from https://docs.python.org/3/library/tkinter.html

BeautifulSoup Documentation. Retrieved from https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Urllib Documentation. Retrieved from https://docs.python.org/3/library/urllib.html