**Project Report: Language Detection using Python**

## JOSHUA ISAAC, URK21S1165

**1. Introduction**

**1.1 Project Overview**

Language detection is a crucial task in natural language processing (NLP) that involves identifying the language in which a given text is written. This is essential for various applications, including content filtering, language-based routing in chatbots, and pre-processing in multilingual NLP pipelines. This project utilizes Python's langdetect library to detect the language of input text and the langcodes library to map language codes to human-readable names.

**1.2 Objectives**

The primary objectives of this project are:

* To implement a simple language detection tool in Python.
* To map language codes to full language names for enhanced readability.
* To handle errors for robust language detection.

**2. Methodology**

**2.1 Libraries and Tools**

* **langdetect**: A Python library for language detection, based on Google’s language-detection library.
* **langcodes**: Used to convert language codes (e.g., 'en' for English) to their full language names.

**2.2 Workflow**

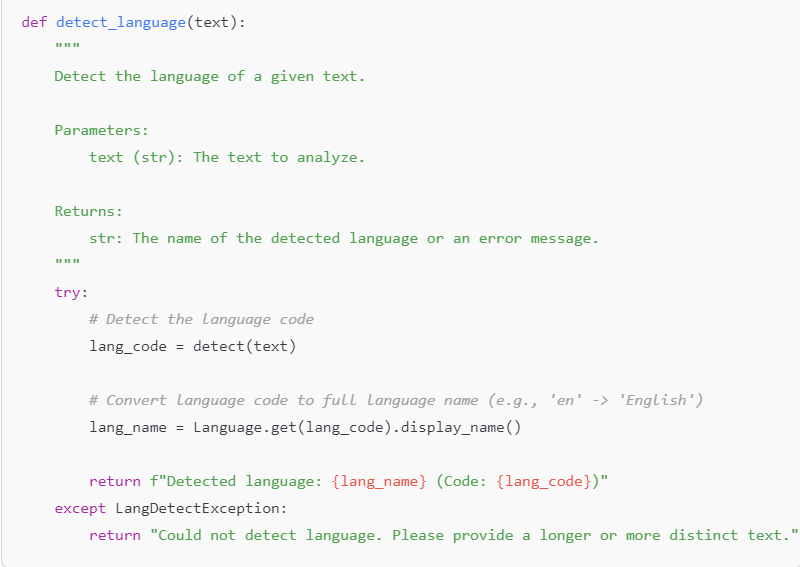
1. **Language Detection**: The script detects the language code of a text input using langdetect.
2. **Language Mapping**: The detected code is then mapped to a full language name using langcodes.
3. **Error Handling**: The script manages exceptions, especially for cases where the input text is too short to determine a language reliably.

**3. Code Explanation**

**3.1 detect\_language Function**

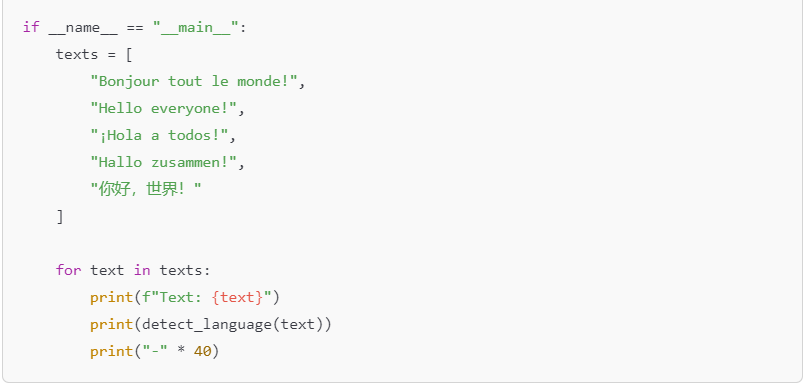
The detect\_language function is the core of the project, performing the following tasks:**Inputs**: A text string for language detection.

* **Detection**: Uses langdetect.detect() to determine the language code.
* **Mapping**: Uses langcodes.Language.get() to convert the code to a full name.
* **Error Handling**: Captures LangDetectException to provide a user-friendly error message.

****

**3.2 Main Script**

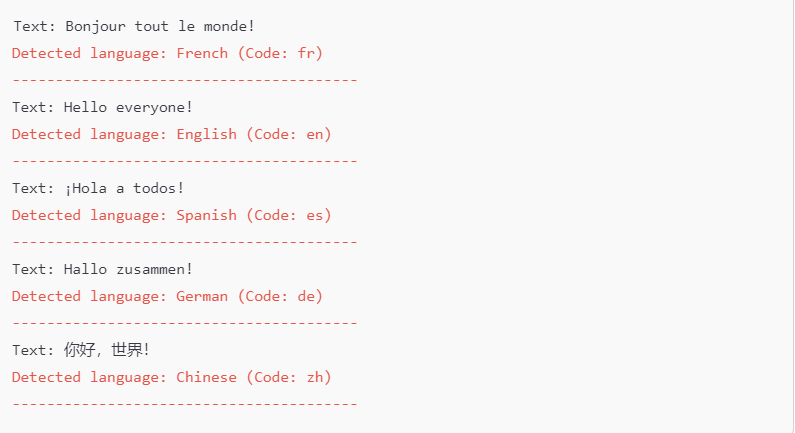
The main script executes a set of example texts, calling detect\_language on each and printing the results.



**4. Results and Observations**

**4.1 Sample Output**

Upon running the script, the following output is generated:



**4.2 Observations**

* The detect\_language function successfully identifies languages with a high degree of accuracy for the sample texts.
* Shorter or ambiguous sentences may sometimes yield errors or incorrect results, highlighting the importance of using sufficiently distinct text for detection.

**5. Potential Applications**

* **Content Filtering**: Automatically filtering content based on language (e.g., restricting specific language content on websites).
* **Language-Specific Routing**: Redirecting users to language-appropriate pages or support agents.
* **Pre-Processing for NLP**: Identifying languages before applying language-specific models in multilingual NLP applications.
* **Social Media Analysis**: Determining language trends and categorizing content based on language.

**6. Limitations**

**6.1 Dependency on Text Length and Uniqueness**

* The model struggles with very short or ambiguous text, which may lead to incorrect or non-detectable results.

**6.2 Limited Language Detection**

* Although langdetect supports many languages, it may not handle regional dialects or lesser-known languages effectively.

**6.3 Computational Dependency**

* The performance may vary based on computational resources, especially for processing large volumes of text in real-time applications.

**7. Future Work**

**7.1 Model Fine-Tuning**

* Integrate a more advanced model, such as a transformer-based model, for improved language detection accuracy.

**7.2 Enhanced Language Coverage**

* Incorporate support for regional dialects and more languages by expanding beyond langdetect.

**7.3 Integration with Larger Applications**

* This script can be further developed into a language-aware module that integrates with larger NLP workflows, including machine translation and sentiment analysis pipelines.

**8. Conclusion**

The Language Detection Project demonstrates a simple yet effective approach to identifying text language with Python. By utilizing langdetect and langcodes, the script achieves reliable language detection for common languages and can be adapted to support multilingual applications. Although it has limitations, the project provides a strong foundation for further development in multilingual NLP tasks.

THANK YOU