UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

NATURAL LANGUAGE PROCESSING

Language Identification

MAJOR: INFORMATION AND COMMUNICATION TECHNOLOGY

Group 8

Team members

BA12-062 Nguyễn Đăng Duy

BA12-110 Vũ Thành Long

BA12-050 Nguyễn Minh Đức

BA12-138 Nguyễn Trọng Nghĩa

BA12-139 Nguyễn Quang Ngọc

BA12-150 Phạm Đức Phương

February, 2025

Table of Content

I. Introduction 3
II. Project goal 3
III. Target Audience or Customers 4
IV. Methodology 5
1. Dataset 5
2. Data preprocessing 5
3. Multinomial Naive Bayes 6
V. Expected Result 7
VI. References 8

I. Introduction

Finding the language of a given text input is the goal of language detection, a basic problem in natural language processing (NLP). Being able to automatically detect languages has grown to become increasingly important as a result of the constantly expanding amount of language data generated internationally across many platforms and industries. The goal of this project is to develop an effective and reliable language detection system by combining machine learning algorithms, preprocessing methods, and performance measurements.

The project uses a labeled dataset containing text samples in different languages such as English, Hindi, French, Spanish, German, etc. By implementing Naive Bayes classifiers, the system can predict the language of the input text such as character n-grams, frequencies or phonetic patterns based on the text samples learned from the training data. In addition, the model is designed to handle short text inputs and special symbols in some languages.

The goal of our project is to create a system that can identify language on many different input text sets, including mixed languages and short texts. In addition, this project also has great potential in language detection for practical applications such as translation tools, search engines, or AI-powered chatbots. With its extremely user-friendly and easy functionality to predict the language of custom inputs, this system has demonstrated the practicality and effectiveness of using machine learning for language detection.

II. Project goal

This project's main goal is to create a dependable and effective language detection system that can correctly determine the language of a given text input. The project uses strong preprocessing and machine learning approaches to accomplish the following specific goals:

- **Accurate Language Detection:** Create a program that can identify languages from a variety of input formats, such as short texts or single texts. Accuracy is the top priority in order to guarantee the prediction is as accurate as feasible.
- **Application Integration:** Provide a language detection tool that works in unison with current systems and applications, including search engines, translation platforms, and conversational AI chatbots. This objective centers on the model's practical implementation across a range of sectors and real-world use scenarios.
- **Wide Language Coverage:** Incorporate support for multiple languages, including major global languages (English, French, Spanish) as well as regional and less

commonly used languages (Malayalam, Kannada, Turkish). This ensures inclusivity and applicability across diverse linguistic needs.

In addition to developing a useful language identification system, this project hopes to help solve practical issues with managing multilingual text data. By achieving these objectives, the system serves as a base for improving multilingual engagement and communication in the current worldwide digital environment.

III. Target Audience or Customers

The Language Detection System is a versatile tool with a wide range of potential users and applications across various industries. Its target audience and customers include:

- **Mobile App Developers:** Incorporating language identification into messaging apps, social media platforms, or smart keyboards allows developers to offer features like multilingual suggestions, automated language change, and customized user experiences.
- **Translation and Localization Services:** By determining the languages of incoming text input, the system can be used by translation and localization companies to automate the preprocessing step, increasing workflow accuracy and efficiency in multilingual projects.
- **Government Agencies:** Language detection can be used to manage public information across multiple language regions, manage international communication, and identify harmful content, among other public safety and compliance applications.
- **Social Media Platforms:** Social media organizations can utilize language detection to better context analysis in several languages, personalize data provided by users according to language preferences, and improve user content regulation.
- **E-commerce Platforms:** By determining consumer language preferences, adjusting product recommendations, and enabling multilingual customer care, online shops and marketplaces can include language recognition to improve their offerings.
- **AI-Developers:** Language detection can be used as a basic module to provide multilingual capabilities in virtual assistants for researchers and developers working on conversational AI.
- **Educational Platforms:** Online applications can use this system to deliver multilingual educational content to improve accessibility for learners across different language backgrounds.

These kinds of consumers are our target market; by satisfying their requirements, our language detection systems have the ability to revolutionize various user experiences, support process automation, and increase productivity while managing foreign text data across many industries.

IV. Methodology

1. Dataset

Our dataset is a CSV file taken from Kaggle. It contains around 10000 items. Each item contains the text and the language of that text. The texts are written in different ways, including punctuation, special characters, … accurately describing the reality, increasing the complexity of the classification task.

This is some samples:

Nature, in the broadest sense, is the natural, physical, material world or universe.	English
A qualquer momento.	Portuguese
La substance universelle se compose ainsi aussi bien du corps que de l'esprit.	French
A causa della sua natura aperta, vandalismi e imprecisioni sono problemi riscontrabili in Wikipedia.	Italian

Currently, in the initial testing phase, our dataset only has 17 languages: English, Malayalam, Hindi, Tamil, Kannada, French, Spanish, Portuguese, Italian, Russian, Swedish, Dutch, Arabic, Turkish, German, Danish, Greek. We will research and add more languages in the future.

2. Data preprocessing

a. Clean the data

We remove unnecessary elements from the data such as numbers, special characters like punctuation, parentheses..., and convert to lowercase. Removing special characters and numbers helps reduce noise in the processing and converting letters to lowercase ensures that similar words are treated the same.

b. Text Vectorization Methods for Language Detection

CountVectorizer converts the words in a text into a matrix of numbers, where each row represents a text, and each column represents a unique word in the entire data set. The values in the matrix represent the number of times the word appears in the text.

Easy to implement for language problems: CountVectorizer is suitable for simple problems such as language detection, because some languages have specific vocabulary (e.g. Vietnamese uses accents, Japanese uses Hiragana/Katakana/Kanji, English does not use accents).

The disadvantages is that common words like "the", "is", "and" in English may appear a lot but do not provide much information to distinguish the language. This can reduce the effectiveness of the model. That is why we use TfidfVectorizer.

TfidfVectorizer will measure the number of occurrences of words, thereby finding common words and reducing their reliability. At the same time, it also highlights words that are unique or characteristic of a specific language, improving model accuracy.

c. Label Encoding

Convert labels to numbers to make the model easier to process.

d. Splitting data

We split the data into 2 parts:

- 80% for training
- 20% for testing

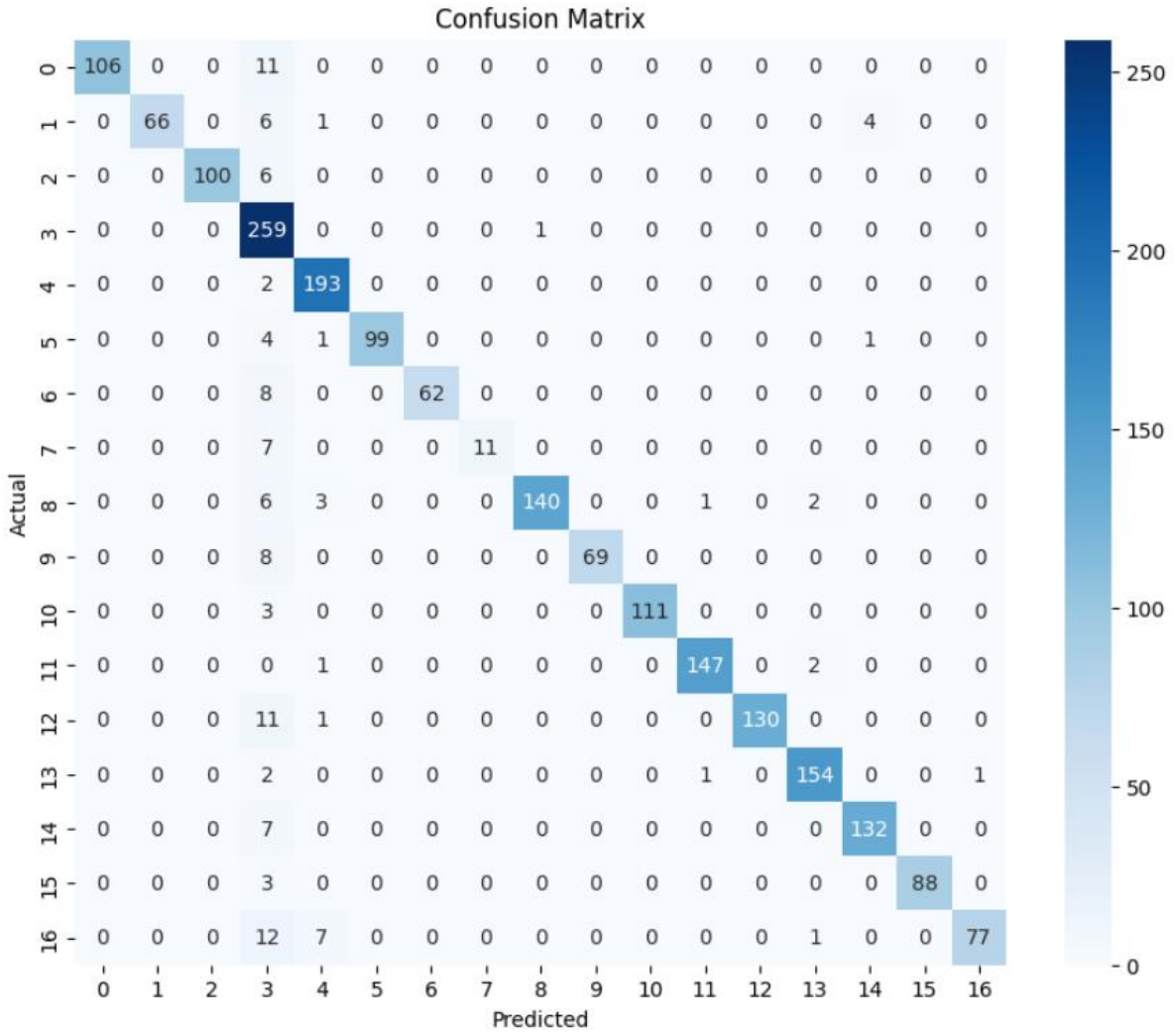
3. Multinomial Naive Bayes

Multinomial Naive Bayes (MNB) is a machine learning algorithm commonly used for text classification tasks, including language detection. It is based on the **Naive Bayes** theorem, which assumes conditional independence between features (words in this context). MNB specifically works well with discrete data, such as word counts or term frequencies, making it ideal for natural language processing tasks where the input features are vectors generated from text data.

Model	**Advantages**	**Disadvantages**
Multinomial Naive Bayes	- Efficient for text data. - Simple to implement.	- Assumes conditional independence. - Struggles with sparse data.
Logistic Regression	- Handles non-linear relationships. - Provides better interpretability for feature weights.	- Computationally more expensive. - May overfit without regularization.

Support Vector Machines (SVM)	- Works well with high-dimensional data. - Strong generalization performance.	- Computationally intensive for large datasets. - Requires careful tuning of parameters.
Random Forest	- Handles non-linear data well. - Resistant to overfitting.	- Computationally expensive for text data. - Requires significant memory.

V. Expected Result



1. Accuracy:

$$\text{Accuracy} = \frac{total\ correct\ predictions}{total\ predictions} = \frac{1894}{2000} \approx 94.7\%$$

2. **The best label:** Label 3 (259 correct predictions) very accurate and low error.
3. **The worst label:** Label 12 had the most errors, with 12 instances being mistaken for other classes.
    - Some common errors:
        - Label 12 (Predicted = 16, Actual = 12): 12 instances being mistaken.
        - Label 4 (Predicted = 3, Actual = 4): 2 instances error.
4. **Summary**
    - The system is capable of recognizing languages with high accuracy (94.7%), even in cases where the text contains multiple languages and it is a long text.
    - The system can process input text at high speed while still ensuring accuracy, suitable for high-demand situations. In addition, it can still be expanded to many other languages in the future.

VI. References

Dataset:

https://www.kaggle.com/datasets/basilb2s/language-detection/data?fbclid=IwY2xjawH91oBleHRuA2FlbQIxMAABHcNqY6rypv8hTDN_CZgA0SOWjUdcyq8v0pdjPXNyOrAJ84emaOEGXK6Wow_aem_h-wnEGsGPZV54c-MTXUpeg

Source code:

https://colab.research.google.com/drive/1ideVWjMWZQ54EJkvIxOhVws6h6yS__b4?fbclid=IwY2xjawH_zbFleHRuA2FlbQIxMAABHfmOeu60g8NTt6a0fYdUOFEjhfcPgB9J3HuACv4j0y5fKG7DdT42VEFGg_aem_i9O5B_-To59yUvt5Ok87bw