Classfication de langages par ANLP

João PEDRO REGAZZI FERREIRA DA SILVA
Guilherme MERTENS DE ANDRADE
Lucas TRAMONTE
Quentin LEMBOULAS

Abstract

This project aims to automatically detect the language of text extracts. To achieve this, we implemented a text processing pipeline comprising tokenization of text data and trained a BERT model to predict the language of each text. Our approach using the XLM-RoBERTa model achieved an accuracy of 88% on the challenge test set.

1 Introduction

Language detection is a critical task in natural language processing (NLP), enabling the automatic identification of a text's language. This capability is essential for applications such as multilingual content management and automated translation.
In this project, we address the challenge of detecting the language of short text extracts. Our approach involves developing a text processing pipeline and employing the XLM-RoBERTa model to achieve high accuracy in language identification. The goal is to create an efficient and robust solution tailored to handle the nuances of brief, multilingual texts.

2 Solution selected : XLM-RoBERTA

The RoBERTa model was selected because of its performance. In the following part, we will present how we implemented the model and adapted it to the challenge.

2.1 XLM-RoBERTa Overview

XLM-RoBERTa is well-suited for our language classification task due to its multilingual training on 100 languages using a masked language modeling (MLM) approach. Unlike previous models, it does not require explicit language identifiers, allowing it to generalize across languages efficiently. Its large-scale training on 2.5TB of CommonCrawl data enables it to capture rich cross-lingual representations, making it highly effective in distinguishing languages. Additionally, since we are leveraging a pre-trained model, we benefit from its strong performance on cross-lingual benchmarks without the need for extensive labeled data or additional training from scratch. Conneau et al. (2020)

2.2 Data Preprocessing

The first step in data preprocessing is the encoding, using a `LabelEncoder` to transform textual categories into numerical values.

Then we performed Text tokenization using XLM-Roberta's tokenizer, which transforms the texts into sequences of tokens. Truncation and padding are applied to ensure that all sequences have a uniform length, which is essential for batch processing during model training.

The calculation of class weights is essential to manage category imbalance. This process involves counting the number of occurrences of each class and then inverting the frequencies to give more importance to under-represented classes. The weights are then normalized and transferred to the training device, either CPU or GPU, to be used during the training process..

2.3 Model Training

The model training process begins with the initialization of a pre-trained XLM-Roberta model, configured to handle the number of categories present in the dataset. Then a customized dataset is constructed using a `DataLoader`.

The AdamW optimizer is employed, along with a linear scheduler that includes a warm-up phase. This combination helps stabilize the learning process and prevents abrupt changes in the learning rate, leading to more effective training.

The CrossEntropyLoss function is used as the

loss function, with class weighting applied to mitigate data imbalance. This approach ensures that the model does not become biased towards the majority classes, leading to better overall performance.

Mixed precision training is implemented using `torch.cuda.amp` with `GradScaler`. This technique accelerates GPU training by utilizing lower precision data types while minimizing numerical errors, resulting in faster and more efficient training.

Eventually the model is saved whenever a significant improvement in loss is detected. This ensures that the best-performing version of the model is preserved for future use and evaluation.

3 Results and analysis

In this section, we compare our models' performance and discuss their outcomes. As shown in Table 1, XLM-RoBERTa achieves the highest accuracy of 88%, confirming its superiority over traditional feature-based methods. Accordingly, all subsequent results in this section use XLM-RoBERTa, as it was chosen for the final submission.

Model	Kaggle Accuracy
XLM-RoBERTa	0.883
TF-IDF with LR	0.711

Table 1: Results for each one of the tested solutions.

Figure 1 compares the top-10 languages in the training set and in our predictions, showing nearly identical proportions. Waray has the largest discrepancy, with only 0.1% proportion difference. Overall, this indicates a balanced distribution between train and test sets and a strong model fit, further supported by our 88% accuracy.

Figure 2 presents the word-count distribution in both train and test sets, showing that 78% of the texts contain fewer than 30 words, with an average length of about 24. This indicates that most samples are relatively short. Despite the limited context such brevity provides, our model still performs effectively, demonstrating its robustness in handling concise inputs.

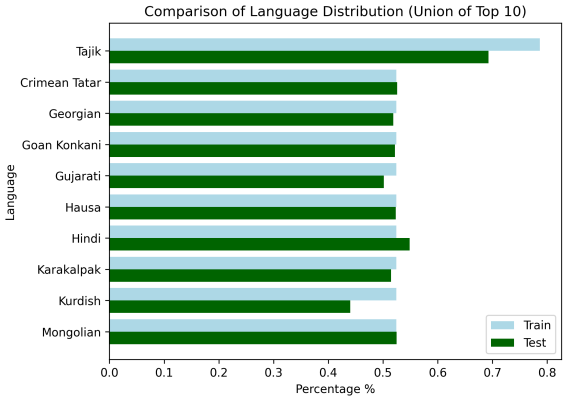

Figure 1: Top-10 language distribution.

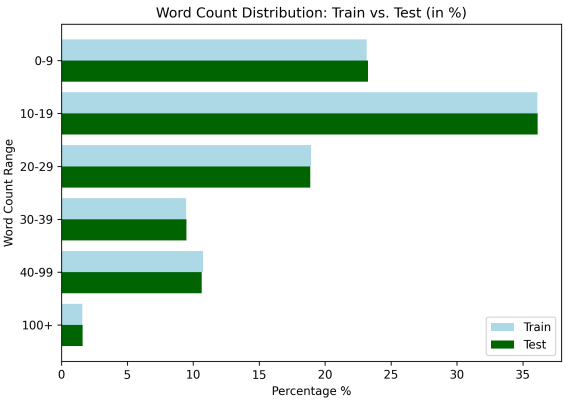

Figure 2: Word-count bins distribution.

4 Conclusion

In this work, we addressed the challenge of detecting languages from short text extracts, leveraging XLM-RoBERTa's multilingual capabilities. Our model achieved an accuracy of 88%, demonstrating the robustness of transformer-based approaches even on relatively brief inputs. Analysis of language and word-count distributions confirmed that our data was well-balanced and that concise texts did not hinder performance. Overall, this validates the practicality of our method for tasks requiring language identification across diverse samples.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.