Kaggle competition NLP CS 2025 report

Alexandre Humblot, Théo Cavina, Théo Putegnat, Racim Menasria

Abstract

Classifying the language of sentences is the first step for a translation task. We aim to solve this problem using natural language processing methods and deep learning. Using XLM-RoBERTa, we manage to reach 88% accuracy on a dataset including 389 languages.

1 Introduction

Our work has been conducted in the context of a Kaggle competition. The objective was to develop a text classifier that would recognize the language of a sentence amongst 389 possible labels. To solve this problem, we tried different approaches with increasing complexity and results. With our best method, we managed to classify correctly 88% of the sentences. We will focus the description of our work on the best scoring method, but here are the different approaches we tried :

1. Our first approach was to use langdetect, while it only choose among 55 labels, it reached 12% accuracy. It is possible to increase the number of detected language, however, we felt that the way langdetect was working (with n-gram and probabilities computation) would not allow us to reach high score.

2. During this exploration phase, we also tried to implement then finetune fasttext, here again, it only manage to choose between a reduced number of languages (176). However, we tried to fine-tune it this time, and it brought us to 70% accuracy.

3. For the next step, we considered training specialist model on every alphabet and then ask the right model to predict the language. Unfortunately, this strategy did not provide us with any satisfactory results

4. We ended up using XLM-R Roberta, a transformer-based language model. We will describe more precisely our methodology and how we manage to reach 88% accuracy in the next section.

2 Solution

2.1 Data exploration

The first step of our NLP project was to know our data. Here is a quick summary of the important things in our dataset :

- We have 390 possible labels. Amongst them, some languages use several alphabet and other are from fictional universe.

- The label repartition in the training set is not quite uniform. Indeed, the vast majority of labels (339) have 500 entries while 24 (7% of the dataset) have 35 entries or less. As we will see, these "rare languages" will be challenging to deal with.

2.2 Model

To solve the language detection problem, we used **XLM-RoBERTa**. It is a transformer-based model built on the robust RoBERTa architecture and pre-trained on an extensive multilingual corpus covering over 100 languages. This extensive pre-training enables the model to capture rich, cross-lingual semantic and syntactic representations, making it highly effective for classifying texts. Its multi-head attention mechanism allows the model to focus on different contextual aspects simultaneously, which is crucial for discerning subtle differences between languages that may share similar lexical or syntactic features. Additionally, the use of sub-word tokenization via the original RoBERTa tokenizer ensures that even rare or previously unseen words are decomposed into meaningful units, thereby preserving essential semantic information across diverse vocabularies. By fine-tuning XLM-RoBERTa on our specific language detection task, we leverage

both its robust pre-trained representations and its adaptability, resulting in a model that performs well even on languages that were underrepresented during the initial pre-training phase. The fully detailed training parameters used for training are available in the code.

2.3 Fine-tuning

Adapting RoBERTa to our task has been done by fine-tuning the pre-trained XLM-R model for classification with our data. We split the training data into 2 training and validation sets to be able to compute the performance of our model.

2.4 Preliminary results

Adopting this method, we reach 88% accuracy. However, the precision of the model is highly dependent on the class and on the number of examples per class. Fig.1 shows that the worst performances occur with classes with fewer examples. One of the issues we are facing is that the languages with fewer than 10 examples are rare, and the pre-training phase of the model we are using most certainly did not provide knowledge regarding these "rare languages".

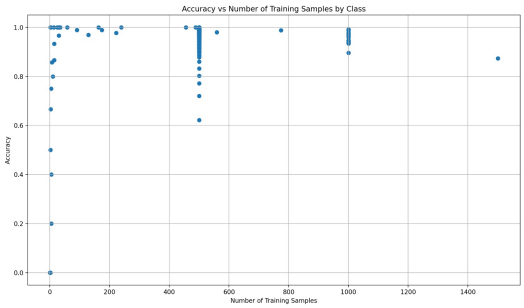

Figure 1: Accuracy depending on the number of entries

2.5 Data augmentation

Since we observed that classes with few training samples were numerous, we wanted to apply some kind of data augmentation. We identified the rare language and found an open online data language archive (http://www.language-archives.org/). This allowed us to add data to the rare languages.

Unfortunately, when trained using the augmented dataset, our model did not show signs of improvement and kept getting an accuracy around 88%.

2.6 Transductive learing

Additionally, we explore transductive inference, a technique that refines model predictions by incorporating knowledge about the distribution of the test set. Unlike standard inference, which classifies each input independently, transductive inference adjusts predictions to align with expected label distributions, potentially mitigating biases introduced during training.

3 Results and Analysis

3.1 Summary of results

Tab.1 displays a summary of our results.

Model	Accuracy
langdetect	12%
fasttext fine-tuned	70%
XML-R finetuned	88%
XML-R w/ data augmentation	88%
XML-R w/ transductive learning	88%

Table 1: Summary of the accuracy depending on the model

3.2 Analysis

We expected that RoBERTa performs better than the other model we used. However, the accuracy stagnation with the application of data augmentation techniques is surprising. Indeed, the rare language seemed to perform badly and since the test set is more homogeneous than the training set, we hoped for greater improvement. This may be the consequence of RoBERTa training. The rare languages might be too different from the languages seen during pre-training.

Another possible explanation is the similarity between languages. If we take the example of Kuanyama, our model reaches 0% accuracy on the language. With a bit of research, we learn that Kuanyama and Ndonga are inter-comprehensive and might share a lot. Also, we see that our dataset contains only 1 example of Kuanyama and 100 examples of Ndonga. If these languages are alike, it is very likely that the model does not differentiate between them. This tendency can even be emphasized if we consider languages that the model did not see during the pre-training phase (probably the case for Kuanyama).