ANLP Kaggle Competition: Language Identification

Emile Joseph, Hippolyte Le Comte, Mariem Ammar, Augustin Savier, Paul Le Bolloch

Abstract

Language identification is a critical component of modern Natural Language Processing (NLP) pipelines, enabling multilingual applications to operate seamlessly across diverse linguistic landscapes. However, accurately classifying languages—especially low-resource or morphologically complex ones—remains a formidable challenge. This work presents our methodology for tackling a Kaggle competition on language identification, where the goal is to develop a robust and precise classifier capable of discerning the language of any given text.

1 Introduction

An in-depth analysis of the dataset revealed key insights that shaped our approach. The dataset comprises 390 distinct languages (both widely spoken languages and regional dialects) with highly imbalanced representation: while most languages have 500 instances, some are significantly underrepresented, with 24 languages having fewer than 50 training samples. However, a substantial 93.3% of the training data corresponds to languages with over 100 instances, suggesting that a well-optimized model should achieve high accuracy despite these disparities. Notably, the distribution in the test dataset follows a similar pattern, ensuring consistency between training and evaluation phases.

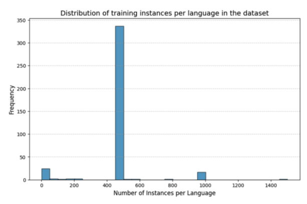

Figure 1: Language-wise Training Distribution

Most languages are represented by fewer than a million words, with nearly 10% of the dataset containing less than one hundred thousand words.

2 Solution

2.1 Preprocessing

We developed a text-cleaning function to remove URLs, special characters, punctuation, and extra spaces, while also converting text to lowercase. However, we chose to retain characters such as (), [], , "", and , as their removal negatively impacted performance by eliminating valuable contextual information.

2.2 Studied Approaches

2.2.1 TF-IDF with Character-Based Tokenization

This approach involves applying a TF-IDF vectorizer to our preprocessed texts, using `analyzer="char"` (character-based tokenization), which is better suited for language detection [1]. We set `ngram_range=(1,4)` to capture short character sequences and `max_features=200000` to limit the vocabulary size. We used GridSearch with MultinomialNB and SGDClassifier, suited for large datasets, to compare them. We observed that the performance of the SGDClassifier was significantly lower than that of the MultinomialNB. However, not all hyperparameters of MultinomialNB could be optimized with GridSearch due to long computation times. Through various tests, we empirically determined that the optimal number of `max_features` is 200,000.

2.2.2 Subword Tokenization with SentencePiece

To enhance our initial approach, we integrated SentencePiece, a simple, efficient, and language-independent subword tokenizer. This method aims to improve tokenization granularity and adaptability across different scripts and languages. We trained SentencePiece on our corpus with a vocabulary size of `60,000` [2].

2.2.3 Fine-Tuning Language Model

In an effort to enhance the performance for this task, we opted to fine-tune a pre-trained language model specifically for our text classification task. We selected XLM-RoBERTa,a multilingual model trained on 100 languages from the CC-100 corpus. With a tokenizer vocabulary size of 250,000 tokens, it was well-suited to represent the words in our dataset. XLM-RoBERTa builds on the RoBERTa

Preprocessing	Model	Accuracy (%)
TF-IDF	Multinomial NB	85.237
SentencePiece + TF-IDF	Multinomial NB	83.499
Tokenization (XLM-RoBERTa tokenizer)	XLM-RoBERTa finetuned	88.816

Table 1: Top results achieved on the Kaggle platform for the various explored approaches.

architecture and incorporates both Masked Language Modeling (MLM) and Translation Language Modeling (TLM) objectives during its pre-training. Unlike other multilingual models, it does not require language tensors in the input; instead, it determines the language directly from the tokens. [3].

We fine-tuned the model on our dataset for 10 epochs using the 80% training set, calculating accuracy at each epoch. The model's performance was also evaluated on both the 10% validation set and the 10% test set. This fine-tuning process improved the classification performance by leveraging XLM-RoBERTa's contextual representations.

3 Results & Analysis

To achieve these results with the Multinomial NB approach, we refined our preprocessing based on outcome analysis. We observed that our models struggled to identify certain languages, like Tajik (tgk), where sentences contained multiple alphabets. To address this, we implemented a method to identify the dominant alphabet in a sentence, as explained in the "Preprocessing" section.

We initially removed English words from non-English labeled sentences, but this negatively impacted performance, leaving some sentences almost wordless. Furthermore, when analyzing the final results, we found that languages like Bosnian (bos), written in Latin script, were not well recognized. A closer look revealed that Bosnian shares similarities with Serbian and Croatian but has subtle differences, such as "pravodnih organa" versus "pravosudnih organa." Since the Multinomial NB model relies on character sequences, it struggles to distinguish these nuances.

Some languages, like Serbian and Tajik, primarily use a specific alphabet, such as the Cyrillic script in Serbian. Including alphabet proportions in sentences could help the model better associate scripts with languages, addressing some classification challenges.

Finally, the difference in performance between Multinomial NB with and without SentencePiece likely results from suboptimal hyperparameter tuning during SentencePiece training on our corpus. In comparison, the XLM-RoBERTa model, after fine-tuning, showed a significant improvement in language classification accuracy. The model's ability to capture contextual information from the input tokens allowed it to better differentiate between similar languages, such as Bosnian, Serbian, and Croatian. Additionally, XLM-RoBERTa's multilingual capabilities helped it handle languages with mixed alphabets more effectively, overcoming some of the challenges seen with the Multinomial NB approach.

4 Conclusion

We achieve relatively strong results with our different approaches. Notably, we obtain performance nearly as good with Multinomial NB combined with extensive preprocessing as with a deep learning approach. However, while training Multinomial NB takes less than two minutes, the deep learning approach requires several hours of training. This is why using Multinomial NB can be a good compromise when training time is a constraint. However, if time is not an issue, the deep learning approach is preferable, as it achieves higher accuracy, which seems difficult to attain with Multinomial NB.

References

[1] João Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the First Instructional Conference on Machine Learning*, 2003.

[2] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, 2018.

[3] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. July 2019.

TEAM : Paul Le Bolloch . For more information, you can visit the GitHub repository at `https://github.com/paullebolloch/kaggle_nlp`.