ANLP Kaggle Competition 2025 Report

Paul-Alexandre MARENGHI
paul-alexandre.marenghi@student-cs.fr

Ambroise MARTIN ROUVILLE
ambroise.martin-roualle-de-rouville@student-cs.fr

Clément VERON
clement.veron@student-cs.fr

Edouard SEGUIER
edouard.seguier@student-cs.fr

Théo MICHEL
theo.michel@student-cs.fr

Abstract

Language Identification (LID) is essential in NLP for tasks like machine translation and content filtering. In this project, we compare different approaches, from TF-IDF with Logistic Regression and Naïve Bayes to FastText and DistilBERT fine-tuning. We evaluate their accuracy, efficiency, and robustness, particularly for short texts and closely related languages. Our results highlight the trade-offs between traditional statistical methods and deep learning, showing that while classical models provide strong baselines, transformer-based approaches achieve higher accuracy in complex multilingual scenarios.

1 Introduction

Language classification is a fundamental task in NLP, enabling applications such as machine translation, multilingual search, and automated content moderation. As multilingual data becomes increasingly prevalent, the need for efficient and accurate classification methods continues to grow.

In this project, we investigate different language classification approaches, ranging from traditional statistical methods to deep learning-based techniques. We begin with TF-IDF representations, leveraging Logistic Regression and Multinomial Naïve Bayes, which are widely used for their simplicity and efficiency. We then explore FastText, which incorporates n-gram embeddings and a linear classifier to capture subword information. Finally, we fine-tune DistilBERT, a transformer-based model, to evaluate its effectiveness in learning contextual representations.

Our analysis compares these methods in terms of accuracy, computational efficiency, and robustness, particularly when handling short texts and closely related languages. By systematically evaluating these techniques, we provide insights into their strengths, trade-offs, and applicability to real-world multilingual NLP tasks.

2 Solution

2.1 EDA

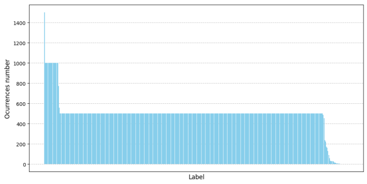

Figure 1: Label occurences distribution in train set.

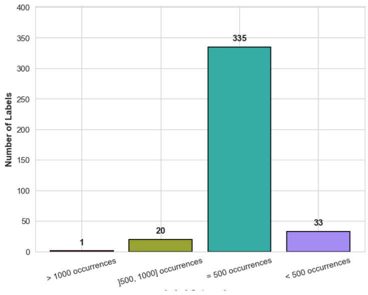

Figure 2: Number of labels per occurences range in train set.

We observe four main groups of labels based on their occurrence count in the training set. It is therefore crucial to account for this imbalance during training to optimize the generalization of our results.

We decide to use a sampler to address the class imbalance by either oversampling underrepresented labels or undersampling overrepresented ones, ensuring a more balanced training process and improved model generalization.

2.2 Tested Approaches

We experimented with multiple paradigms for language classification, ranging from classical statistical methods to deep learning-based approaches. Below, we briefly describe each method:

2.2.1 TF-IDF + Logistic Regression

TF-IDF (Term Frequency-Inverse Document Frequency) transforms text into a weighted word-frequency representation, emphasizing important words while reducing the influence of common ones. A logistic regression classifier is then applied to these features, providing a simple yet effective linear model for classification.

2.2.2 TF-IDF + Multinomial Naïve Bayes

Similar to the previous approach, this method also uses TF-IDF for feature extraction.

To further improve performance, it takes punctuation into account forced by a smaller ngram range. Other TF-IDF settings have been tested (removing accents, punctuation, coupling it with other features such as average word lenght) but have not resulted in better performance.

Additionnaly, classification is performed using Multinomial Naïve Bayes, a probabilistic model that assumes feature independence and is well-suited for text classification tasks. It makes use of *sample_weight* param to deal with the imbalance in the training set.

On a final note, only 1/3 of the training set has been used to train the model due to resource constraints. This enhances a large margin for improvement.

2.2.3 FastText: N-grams + Linear Classifier

FastText by Facebook uses subword n-grams for word representation and a hierarchical softmax, good for morphologically rich languages and handling out-of-vocabulary words.

Data Preparation The dataset was preprocessed to fit FastText's format (text-label pairs formatted as `__label__<label> <text>`). Class imbalance was addressed via manual balancing, negative sampling, and one-vs-all loss.

Model Training A supervised FastText model was trained using autotune for hyperparameter optimization, exploring learning rates, window sizes, and n-gram ranges. The final model achieved **0.845** precision on the validation set.

Zero-Shot Benchmarking with GlotLID GlotLID, a pretrained FastText-based model, achieved **0.75** accuracy in a zero-shot setting, underscoring the need for domain-specific fine-tuning.

Approach	Accuracy
GlotLID (zero-shot)	0.751
FastText (no class balance)	0.831
FastText (with class balance)	0.845

Table 1: Accuracy of FastText approaches on the validation set.

2.2.4 DistilBERT Fine-Tuning

Data Preparation The data was preprocessed by removing missing values and filtering small classes (fewer than two instances). Texts were tokenized using the multilingual DistilBERT tokenizer with a maximum length of 128 tokens.

Model Training We used DistilBERT-base-multilingual-cased as our base model, a compact multilingual version of BERT with 97% of performance while being 40% lighter. We fine-tuned it on our dataset with a linear classification layer and a learning rate of 2e-5 to leverage its deep contextual embeddings, capturing complex linguistic patterns to improve classification accuracy. Training was conducted over 3 epochs with a batch size of 16. The final model achieved a precision of **0.8542** on the validation set.

Optimizations for Inference For the inference phase, we implemented batch processing with a batch size of 32 to accelerate predictions and minimizing memory usage.

3 Results

Approach	Accuracy
TF-IDF + Logistic Regression	0.77161
TF-IDF + Multinomial Naïve Bayes	0.79020
FastText: N-grams + Linear Classifier	0.84178
DistilBERT Fine-Tuning	**0.85042**

Table 2: Accuracy of different language classification approaches on the kaggle test set.

To sum up, we compared four language classification methods, from simple TF-IDF with Logistic Regression to more advanced approaches like FastText (0.84178) and fine-tuned DistilBERT. The results show a clear trend: while traditional methods are efficient, transformer-based deep learning models perform best. FastText offers a middle ground, balancing efficiency with improved accuracy. However, models like DistilBERT, despite their higher computational cost, achieve superior performance.

Based on our results, it is highly likely that fine-tuning DistilBERT for more epochs would have yielded even better results.

4 References

https://huggingface.co/facebook/fasttext-language-identification

https://huggingface.co/cis-lmu/glotlid

https://huggingface.co/dinalzein/xlm-roberta-base-finetuned-language-identification

https://huggingface.co/DunnBC22/distilbert-base-multilingual-cased-language_detection