Hybrid Language Identification using DistilBERT, FastText, and N-gram Filtering

Oscar Pastural, Louis Gauthier, Clément Florval
CentraleSupélec, Université Paris-Saclay

Abstract

This work presents a hybrid solution for automatic language identification that combines fine-tuned DistilBERT, the open-source *cis-lmu/glotlid* FastText model, and an innovative n-gram filtering mechanism. Our approach, designed for a Kaggle competition, achieves a final ensemble accuracy of 88.5% by averaging the probabilities from DistilBERT and FastText and applying a trigram-based filter. We also explored alternative ensemble strategies and experimented with large language models (LLMs) via API; however, due to practical constraints and inefficiencies in local testing, only the open-source models were used.

1 Introduction

Automatic language identification is a crucial component in multilingual information processing, impacting applications from digital content moderation to translation. Although transformer-based models like DistilBERT have achieved notable success, distinguishing between closely related languages remains challenging, especially for short or noisy texts. Traditional statistical methods (e.g., FastText) provide strong baselines but often lack the necessary contextual depth.

Our work integrates the strengths of deep contextual embeddings and efficient statistical models. In addition, we initially explored leveraging state-of-the-art large language models (LLMs) via OpenAI's GPT-4 API to label samples. While GPT-4 performed well in preliminary tests, attempts to replicate these results locally using models such as DeepSeek R1 Distilled Qwen 1.5B, Qwen 2.5 1.5B, and Aya-101 proved unsatisfactory due to poor performance, slow inference, and limited language coverage. Consequently, we focused on an open-source pipeline that remains both effective and efficient.

2 Solution

Our solution comprises three main components:

2.1 DistilBERT Classifier

We fine-tuned a multilingual DistilBERT model (*distilbert-base-multilingual-cased*) on the provided dataset. Training was conducted on Google Colab using A100 GPUs, with each epoch taking roughly 30 minutes. Over 5 epochs (a total of 2 hours and 30 minutes), the model learned to output probability distributions over language labels.

2.2 FastText Predictions

To complement DistilBERT, we employed the *cis-lmu/glotlid* FastText model. Owing to its training on over 2,000 languages, FastText was expected to help with languages underrepresented in our training data. We also attempted a threshold-based strategy—switching to FastText predictions when DistilBERT's top probability fell below 0.67—but this approach only achieved 87.39% accuracy, making it less effective than our ensemble.

2.3 N-gram Filtering

A key innovation in our method is the n-gram filtering mechanism. We constructed language-specific n-gram vocabularies (unigrams to 4-grams) from the training data. After extensive experimentation, we determined that requiring at least 30% of trigrams to be present in a language's vocabulary provided the best trade-off: it reduced the candidate set by 53% (leaving 47% of candidates) while misfiltering only 1.8% of samples. This filtering increased DistilBERT's accuracy from 86.93% to 87.85%.

2.4 Ensemble Approaches

Our intuition was that FastText's broad language coverage could compensate for cases where DistilBERT struggled. By averaging the probability outputs of DistilBERT and FastText and applying

the n-gram filter, we achieved our best accuracy of 88.5%. We further investigated a group-based selection strategy that used tuples of top predictions from both models to dynamically choose the best candidate; however, this method yielded only marginal gains (from 87.46% to 87.82%), confirming that simple averaging with robust filtering was most effective.

3 Results and Analysis

Our experiments were structured into several key stages:

3.1 Baseline Performance

The fine-tuned DistilBERT model achieved a baseline accuracy of approximately 86.9% on the evaluation set. Independently, FastText performed around 10% lower but added valuable diversity to our predictions.

3.2 Impact of N-gram Filtering

The n-gram filter, with a 30% trigram threshold, improved DistilBERT's accuracy from 86.93% to 87.85% by reducing false positives while preserving the true labels in 98.2% of cases.

3.3 Ensemble Approaches

Averaging the probabilities from DistilBERT and FastText, coupled with n-gram filtering, resulted in the best overall accuracy of 88.5%. Alternative strategies, including threshold-based switching (accuracy of 87.39%) and group-based selection (accuracy improved marginally from 87.46% to 87.82%), were explored but did not outperform the simple ensemble.

3.4 LLM Experimentation

We initially tested large language models using OpenAI's GPT-4 API to assess their potential for language labeling. Although the API demonstrated strong performance, local tests with models like DeepSeek R1 Distilled Qwen 1.5B, Qwen 2.5 1.5B, and Aya-101 were not competitive in terms of speed, language coverage, or accuracy. Given these limitations and competition constraints on external API usage, we did not incorporate LLMs into our final pipeline.

3.5 Overall Findings

The experiments highlight that:

- The n-gram filtering mechanism effectively reduces false positives with minimal impact on true labels.
- Averaging DistilBERT and FastText probability distributions, together with filtering, yields the highest accuracy (88.5%).
- Alternative ensemble strategies offer only marginal improvements and add complexity.
- While LLM-based labeling shows promise, its practical limitations render it unsuitable for our efficient, open-source solution.

4 Conclusion

Our hybrid approach, combining deep contextual modeling with robust statistical techniques and an innovative n-gram filtering mechanism, has proven effective for automatic language identification. The final ensemble achieves 88.5% accuracy and demonstrates robustness against noisy and ambiguous inputs. Future work may investigate adaptive thresholding or the integration of additional linguistic features to further enhance performance.

4.1 References

References

Marta Bañón, Jaume Zaragoza-Bernabeu, Gema Ramírez-Sánchez, and Sergio Ortiz-Rojas. 2024. Fastspell: the langid magic spell. *arXiv preprint arXiv:2404.08345*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Amir Windisch, Matthias Müller, and Alexander Fraser. 2023. Glotlid: Language identification for low-resource languages. *arXiv preprint arXiv:2310.16248*.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. *arXiv preprint arXiv:2402.07827*.