

# Using large language models to provide effective embeddings for native language identification

Roxane Auger   Olivier Boeffard   Jean-François Bonastre

AMIAD, Pôle Recherche

Palaiseau, France

firstname.lastname@polytechnique.edu

## Abstract

Native Language Identification (NLI) consists in predicting a person’s native language based on their second language production. In the field of text-based NLI, approaches based on Large Language Models (LLM) outperform previous approaches based on statistical classifiers trained on linguistic features. However, the unstructured nature of LLM textual output does not support the direct application of usual classifiers. To address this limitation, we explore the use of LLM to generate effective latent vector representations (embeddings) for NLI. Our experiments, across ten models ranging from 0.1B to 405B parameters, show that embeddings gathered from a frozen LLM coupled with a linear classifier match the performance of fine-tuned generative LLM. Furthermore, layer-depth analysis reveals that the first 40% of an LLM’s layers are sufficient to achieve full representational utility for NLI.

## 1 Introduction

Native language identification (NLI) is a natural language processing (NLP) task consisting in predicting a person’s native language ( $L_1$ ) based on their second language ( $L_2$ ) production (Odlin, 1989). In this context,  $L_1$  refers to any language acquired from birth, while  $L_2$  refers to any subsequent language learned later in life.

Because native language often correlates with demographic traits like nationality and ethnicity (Goswami et al., 2026), NLI serves as a key tool in forensics (Malmasi and Dras, 2015), author profiling (Uluslu et al., 2024), and threat actor attribution. Moreover, NLI allows the teaching of a second language to be adapted to the difficulties specific to a native language (Goswami et al., 2024). Most NLI use cases are globally applicable: e-learning platforms address a highly diverse student audience, while cyber forensics must attribute responsibility to malicious actors from anywhere in the world.

Consequently, adapting NLI to these applications requires automated systems capable of handling both high volumes of real-time queries and a wide range of languages.

Large language models (LLM) are autoregressive architectures with billions of parameters, pre-trained on massive amounts of data (Naveed et al., 2024). Their emergence marks a paradigm shift for NLP tasks, which can now be performed using natural language instructions and textual responses (Radford et al., 2019). When applied to text-based NLI, LLM (Zhang and Salle, 2023) outperform earlier approaches based on statistical classifiers trained on linguistic features (Markov, 2018).

Because of the unstructured nature of generative outputs, text generator systems do not support the direct application of usual classifiers. Extracting the latent vector representations (embeddings) of LLM offers an alternative way of leveraging the power of LLM while maintaining compatibility. This hypothesis is supported by recent experiments demonstrating that LLM can produce effective embeddings for diverse NLP tasks (BehnamGhader et al., 2024).

In line with this hypothesis, this study evaluates the ability of LLM to produce effective embeddings for text-based NLI. To this end, we develop a processing pipeline that couples an LLM with a task-driven linear classifier.<sup>1</sup> We conduct a comparative performance evaluation across ten models, ranging from 0.1 to 405 billion parameters. Our investigation addresses two primary research questions:

- Do the latent vector representations of LLM yield effective embeddings for NLI tasks?
- How does the embedding quality vary when extracted from a given hidden layer?

<sup>1</sup>Our code is available at <https://github.com/AMIAD-Research/embed-train-predict>

## 2 Related work

Early approaches for text-based NLI relied on statistical classifiers like logistic regression (Tetreault et al., 2012) and support vector machines (Markov, 2018) trained on linguistic features. These methods have been outperformed by text generator approaches relying on modern LLM (Zhang and Salle, 2023). LLM performance is evidenced by the fine-tuned open-source LLM Gemma (7B) and Llama 3.1 (8B) accuracy score of 90%<sup>2</sup> (Ng and Markov, 2025; Nicholls and Alperin, 2025) and the fine-tuned proprietary model GPT-4 accuracy score of 92%<sup>2</sup> (Goswami et al., 2025).

Previous embedding-based NLI approaches have paired classifiers with encoder-only models. Despite allowing for a fully optimized pipeline, such systems have not demonstrated a performance improvement over text generator approaches. This performance gap is evidenced by the fine-tuned encoder-only model BERT accuracy score of 77%<sup>2</sup> (Steinbakken and Gambäck, 2020).

## 3 NLI systems

For this study, we developed two families of NLI systems: text generator and text embedder. Their configuration details are provided in Appendices A and B, respectively, and their inference pipelines are illustrated in Figure 1.

For **text generator (TG)** systems, we provide both the instructions relative to the NLI task and the text to classify to the LLM through the prompt. We then extract the classification label from the generated text using a regular expression filter.

For **text embedder (TE)** systems, the model computes a single forward pass over the text to classify. Embeddings are extracted from the final hidden layer and reduced to a fixed-size vector using a mean pooling strategy (Reimers and Gurevych, 2019; Tao et al., 2025). The NLI task is then performed via a Linear Discriminant Analysis (LDA) (Zhao et al., 2024). The linear classifier is trained independently for each combination of model and corpus, using only the training subset.

LDA is linear, deterministic, and computed analytically from class statistics, which isolates embedding quality, our object of study, from classifier engineering and removes any per-model hyperparameter tuning. Trained identically for every model, corpus, and transformer layer, it leaves the

<sup>2</sup>The results were obtained using the TOEFL11 corpus (Blanchard et al., 2013)

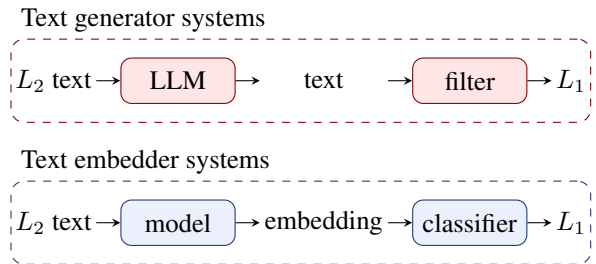


Figure 1: Illustration of NLI inference pipeline for text generator and text embedder systems. Both systems are provided with a text written in a second language ( $L_2$ ) from which they predict the author’s native language ( $L_1$ ).

source of the representations as the only varying factor, and it can only reveal the separability already present in the embedding space.

The models used for our text generator and text embedder systems are frozen, relying exclusively on their pre-trained knowledge without task-specific fine-tuning. This behavior introduces an asymmetry between the two systems, as the text embedder system requires a classifier trained on the target data. To fill this gap, we compare our results with fine-tuned systems from prior work. In the **fine-tuned text generator (TG-FT)** category, we select GPT-4 (Goswami et al., 2025), Mistral (7B) v0.1, and Llama 3 (8B) (Ng and Markov, 2025). In the **fine-tuned text embedder (TE-FT)** category, we select RoBERTa and XLM-R (Goswami et al., 2025).

## 4 Data

We evaluate our NLI systems on two academic corpora: TOEFL11 and CLC-FCE. By nature, the vocabulary, structure, and topics of academic production differ from natural conversation, limiting their representativeness for certain real-world NLI applications. Moreover, the two corpora vary in both scale and language distribution, providing distinct evaluation environments.

The **TOEFL11** (Blanchard et al., 2013) corpus contains 12,100 essays written by candidates taking the Test of English as a Foreign Language (TOEFL). The corpus is balanced with 1,100 texts for each of its eleven represented native languages. Following the original distribution guidelines, the corpus is partitioned into three subsets: 9,900 texts for training and 1,100 texts each for validation and testing.

| LLM                                      | TOEFL11               | CLC-FCE        |
|------------------------------------------|-----------------------|----------------|
| <i>Fine-tuned text generator (TG-FT)</i> |                       |                |
| GPT-4 (Goswami et al., 2025)             | 92                    | <b>83</b>      |
| Mistral (7B) v0.1 (Ng and Markov, 2025)  | 89.8                  | -              |
| Llama 3 (8B) (Ng and Markov, 2025)       | 85.3                  | -              |
| <i>Text generator (TG)</i>               |                       |                |
| Mistral (7B) v0.1                        | 17.4 $\pm$ 2.2        | 15.0 $\pm$ 5.9 |
| Llama 3 (8B)                             | 54.5 $\pm$ 2.9        | 47.1 $\pm$ 8.2 |
| Mistral Small (24B) 2501                 | 67.8 $\pm$ 2.8        | 58.6 $\pm$ 8.1 |
| Llama 3.1 (70B)                          | 85.1 $\pm$ 2.1        | 70.0 $\pm$ 7.5 |
| Mistral Large (123B) 2411                | 86.1 $\pm$ 2.0        | 75.0 $\pm$ 7.1 |
| Llama 3.1 (405B)                         | 90.8 $\pm$ 1.7        | 79.3 $\pm$ 6.7 |
| <i>Text embedder (TE)</i>                |                       |                |
| Mistral (7B) v0.1                        | 83.9 $\pm$ 2.2        | 60.7 $\pm$ 8.0 |
| Llama 3 (8B)                             | 87.1 $\pm$ 2.0        | 65.7 $\pm$ 7.8 |
| Mistral Small (24B) 2501                 | 88.9 $\pm$ 1.9        | 72.1 $\pm$ 7.4 |
| Llama 3.1 (70B)                          | 88.5 $\pm$ 1.9        | 80.7 $\pm$ 6.5 |
| Mistral Large (123B) 2411                | 90.8 $\pm$ 1.7        | 82.1 $\pm$ 6.3 |
| Llama 3.1 (405B)                         | <b>93.4</b> $\pm$ 1.5 | 82.1 $\pm$ 6.3 |

Table 1: Accuracy score (%) and Wilson 95% confidence interval half-widths of LLM-based text generator and text embedder systems on the TOEFL11 and CLC-FCE corpora. A dash marks a score not reported by the cited work.

The **CLC-FCE**<sup>3</sup> (Yannakoudakis et al., 2011) corpus contains 1,244 essays written by candidates taking the Cambridge ESOL First Certificate in English (FCE). To address class imbalance, the corpus was downsampled to include 63 annotated texts for each of the fourteen native languages selected from the sixteen represented. We partition the corpus into three subsets: 602 texts for training and 140 texts each for validation and testing.

## 5 Selected models

We evaluate two transformer model architectures: instruction-tuned LLM and encoder-only models. LLM enable evaluation across both text generator and text embedder systems. By opting for instruction-tuned variants, we ensure that the models can effectively process the task instructions within our prompts. By selecting encoder-only models, we allow performance comparison with text embedder systems using dedicated embedding architectures. Full details regarding model configurations are provided in Appendix C.

Following the work of Ng and Markov (2025), we select the LLM Mistral (7B) v0.1 (Jiang et al., 2023) and Llama 3 (8B) (AI@Meta, 2024). To evaluate performance across a diverse range of LLM

scales, we include Mistral Small (24B) 2501 (MistralAI, 2025), Llama 3.1 (70B) (AI@Meta, 2024), Mistral Large (123B) 2411 (MistralAI, 2024), and Llama 3.1 (405B) (MetaAI, 2024). Following the work of Goswami et al. (2025), we select the encoder-only English model RoBERTa (Liu et al., 2019) and the multilingual XLM-R (Conneau et al., 2020), including *base* and *large* variants of each encoder-only model.

## 6 Results

As shown in Table 1, the Wilson 95% confidence interval half-widths for the CLC-FCE corpus exceed 5%. This wide margin is a direct consequence of the limited sample size of the corpus.

Llama 3.1 (405B) achieves our highest TG performance, yielding an accuracy of 90.8%  $\pm$ 1.7 on TOEFL11 and 79.3%  $\pm$ 6.7 on CLC-FCE. Within the 95% confidence interval, these results are comparable to GPT-4 (TG-FT), which achieves 92% and 83% accuracy, respectively. Notably, Llama 3.1 (405B) (TE) establishes our highest overall performance score across both corpora, matching GPT-4 (TG-FT) performance with an accuracy of 93.4%  $\pm$ 1.5 on TOEFL11 and 82.1%  $\pm$ 6.3 on CLC-FCE.

On TOEFL11, the errors of our TE systems are concentrated on two language pairs, Hindi/Telugu and Japanese/Korean. For Llama 3.1 (405B) (TE),

<sup>3</sup>Downloaded from: <https://s3-eu-west-1.amazonaws.com/ilexir-website-media/fce-released-dataset.zip>

23% of the Telugu essays are labeled Hindi and 9% of the Korean ones are labeled Japanese. The same pairs dominate the errors reported by previous NLI systems (Ng and Markov, 2025; Nicholls and Alperin, 2025). Per-language confusion matrices for the Llama 3 (8B), Mistral Small (24B) 2501 and Llama 3.1 (405B) TG and TE systems are provided in Appendix D.

Mistral (7B) v0.1 (TG) reaches less than 20% accuracy on both corpora. This failure comes from instruction following rather than from output formatting. The model overproduces the out-of-domain class "eng", which mirrors the language of the input text ( $L_2$ ) and which the prompt explicitly forbids. It predicts "eng" for 65.5% of the TOEFL11 test essays and 51.4% of the CLC-FCE ones. The other LLM return no invalid or out-of-domain label, so their TG performance gap reflects genuine misclassifications.

For a given LLM, TG systems are consistently outperformed by TE systems, while the performance differences between TG-FT and TE systems remain marginal (89.8% for Mistral (7B) v0.1 (TG-FT) and 85.3% for Llama 3 (8B) (TG-FT) compared to  $83.9\% \pm 2.2$  for Mistral (7B) v0.1 (TE) and  $87.1\% \pm 2.0$  for Llama 3 (8B) (TE)).

In our TE systems the LLM is never updated. We only fit an LDA head on embeddings obtained with a single forward pass per text, and inference does not require autoregressive decoding. Embedding the whole TOEFL11 corpus with Mistral (7B) v0.1 takes 14 minutes on a single A100 and fitting the LDA head takes 72 seconds of CPU time. In comparison, the QLoRA fine-tuning of the open-source TG-FT baselines is part of a reported total of roughly 120 A100 hours for five models on two corpora, or approximately 12 A100 hours for a single model on a single corpus (Ng and Markov, 2025).<sup>4</sup> End-to-end computation times on H100 are reported in Appendix E.

For systems based on encoder-only models, Table 2 also shows marginal performance differences between TE and TE-FT systems (71% for RoBERTa (0.1B) (TE-FT) and 63% for XLM-R (0.3B) (TE-FT) compared to  $70.5\% \pm 2.7$  for RoBERTa (0.1B) (TE) and  $68.6\% \pm 2.7$  for XLM-R (0.3B) (TE)).

Upgrading to the large variant of each encoder-only model yields no significant performance im-

| Encoder-only                            | TOEFL11        |
|-----------------------------------------|----------------|
| <i>Fine-tuned text embedder (TE-FT)</i> |                |
| RoBERTa (0.1B) (2025)                   | 71             |
| XLM-R (0.3B) (2025)                     | 63             |
| <i>Text embedder (TE)</i>               |                |
| RoBERTa (0.1B)                          | $70.5 \pm 2.7$ |
| XLM-R (0.3B)                            | $68.6 \pm 2.7$ |
| RoBERTa Large (0.4B)                    | $72.7 \pm 2.6$ |
| XLM-R Large (0.6B)                      | $70.7 \pm 2.7$ |

Table 2: Accuracy score (%) and Wilson 95% confidence interval half-widths of text embedder systems based on encoder-only models on the TOEFL11 corpus.

provements over the corresponding base version.

## 7 NLI performance for text embedder systems across layer depth

While the previous section explores text-embedder system performance where embeddings are extracted from the final hidden layer, this section explores performance where the embeddings are gathered from any of the model’s hidden layers.

Figure 2 highlights distinct patterns between encoder-only models and LLM. For TE systems based on encoder-only models, the accuracy remains almost constant across all layers. In contrast, TE systems based on LLM yield lower performance in layers below 40% depth than in deeper layers, where accuracy stabilizes.

Selecting the best layer instead of the last layer leaves the two confidence intervals overlapping for all ten TE systems. The best layer is not itself identifiable: its score and the second-best one differ by less than the confidence interval half-width. On TOEFL11, Llama 3.1 (405B) moves from  $93.4\% \pm 1.5$  to  $95.0\% \pm 1.3$ , an interval entirely above the GPT-4 (TG-FT) score of 92%.

## 8 Conclusion

The primary aim of this study was to determine whether the latent vector representations of LLM are effective for NLI tasks. With an accuracy score of 95% using an LLM-based text embedder, our experiments demonstrate that the answer is yes. Specifically, our comparative analysis reveals that using frozen LLM to produce vector representations of text combined with a linear classifier matches the accuracy of fine-tuned generative LLM. Consequently, these findings challenge the necessity of computationally expensive fine-tuning,

<sup>4</sup>We divide by five models and two corpora the total computation time reported by Ng and Markov (2025).

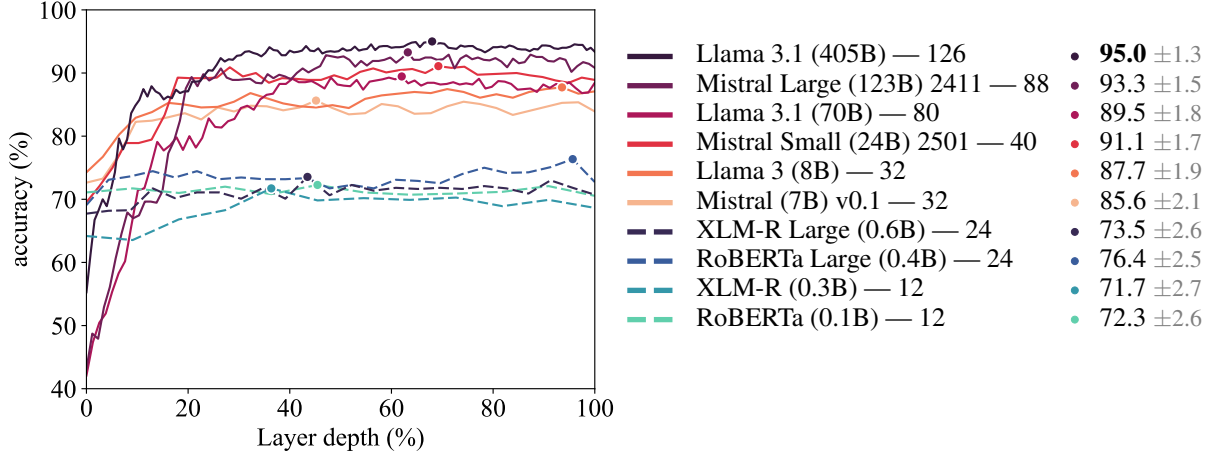


Figure 2: Accuracy score (%) of text embedder systems on the TOEFL11 corpus based on the model’s transformer layer depth (%) used for embedding extraction. Solid lines denote LLM and dashed lines denote encoder-only models. Each legend entry reports the model name, its number of transformer layers, and its best layer accuracy score (%) with the Wilson 95% confidence interval half-width. The dot marks the layer reaching that score on the corresponding curve.

positioning frozen LLM text embedders as a high-performance and cost-effective alternative.

Furthermore, our layer depth analysis reveals an architectural divergence: while keeping only the first layer of encoder-only models is enough for NLI, LLM require a minimum depth of 40% to achieve their full representational utility. Even if the mechanism driving this behavior requires further study, this insight suggests a cost-saving opportunity: leveraging only the first layers for NLI instead of the full model.

## Limitations

While frozen LLM-based text embedders match the accuracy of fine-tuned generative systems for NLI at a lower cost, several limitations apply.

First, potential data leakage within these models makes it difficult to isolate whether high accuracy stems from genuine generalization or from prior exposure to the evaluation corpora. Our results argue against a leak of essay and label pairs. Such a leak would allow direct label retrieval, so zero-shot TG systems would match their TE counterparts, which is not what we observe. Answering this question requires further experiments, such as testing our systems on essays written after the training cutoff of the models.

Second, our comparison with fine-tuned systems relies on accuracy scores reported in prior work, which we do not reproduce under our own protocol. Third, the linear classifier operates on a closed set of labels and cannot assign a native language

unseen during training, whereas a text generator system can receive new labels through its prompt. Finally, we do not evaluate whether the 40% minimum layer depth we report for LLM-based text embedder systems holds for alternative architectures, such as Mixture-of-Experts (MoE) models.

## Ethical considerations

The identification of a person’s native language raises ethical concerns related to privacy. Since writing style constitutes a biometric indicator, NLI algorithms can partially compromise the anonymity of a text. This research strictly adheres to all terms and conditions outlined in the ETS Corpus of Non-Native Written English (LDC2014T06) User License Agreement and the CLC FCE Dataset Licence Agreement. This publication was written with the assistance of artificial intelligence tools for English language rephrasing and stylistic support.

## Acknowledgments

This project was provided with computer and storage resources by GENCI at IDRIS thanks to the grant 2026-AD011017021 on the supercomputer Jean Zay’s A100 and H100 partitions and the grant 2026-R1001016831 on the supercomputer DALIA’s GB200 partitions. We thank the anonymous reviewers for their feedback, the researchers at AMIAD for their support during the writing process, and Nathan Cassereau for his help with code development.

## References

- AI@Meta. 2024. [Llama 3 model card](#).
- Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. [LLM2Vec: Large language models are secretly powerful text encoders](#). *Preprint*, arXiv:2404.05961.
- Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. 2013. [TOEFL11: A corpus of non-native English](#). *ETS Research Report Series*. Publisher: Educational Testing Service ERIC Number: EJ1109982.
- 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](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Dhiman Goswami, Jai Kruthunz Naveen Kumar, and Sanchari Das. 2026. [NLP privacy risk identification in social media \(NLP-PRISM\): A survey](#). *Preprint*, arXiv:2602.15866.
- Dhiman Goswami, Sharanya Thilagan, Kai North, Shervin Malmasi, and Marcos Zampieri. 2024. [Native language identification in texts: A survey](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3149–3160, Mexico City, Mexico. Association for Computational Linguistics.
- Dhiman Goswami, Marcos Zampieri, Kai North, Shervin Malmasi, and Antonios Anastasopoulos. 2025. [Multilingual native language identification with large language models](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, pages 193–199, Albuquerque, USA. Association for Computational Linguistics.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. [Mistral 7B](#). *Preprint*, arXiv:2310.06825.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692.
- Shervin Malmasi and Mark Dras. 2015. [Multilingual native language identification](#). *Natural Language Engineering*, 23:1–53.
- Iliia Markov. 2018. [Automatic Native Language Identification](#). Ph.D. thesis, Instituto Politécnico Nacional.
- MetaAI. 2024. [Meta-Llama-3.1-405B-Instruct](#). Hugging Face model card.
- MistralAI. 2024. [Mistral-Large-Instruct-2411](#). Hugging Face model card.
- MistralAI. 2025. [Mistral-Small-24B-Instruct-2501 model card](#).
- Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2024. [A comprehensive overview of large language models](#). *Preprint*, arXiv:2307.06435.
- Yee Man Ng and Iliia Markov. 2025. [Leveraging open-source large language models for native language identification](#). In *Proceedings of the 12th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 20–28, Abu Dhabi, UAE. Association for Computational Linguistics.
- Robin Nicholls and Kenneth Alperin. 2025. [Cross-genre native language identification with open-source large language models](#). In *Proceedings of the 2nd LUHME Workshop*, pages 103–108, Bologna, Italy. UP - Universidade do Porto (<https://doi.org/10.21747/978-989-9193-73-4/lan2>), LIACC - Laboratório de Inteligência Artificial e Ciência de Computadores da Universidade do Porto, CLUP - Centro de Linguística da Universidade do Porto, UEF - The University of Eastern Finland and UAH - Universidad de Alcalá.
- Terence Odlin. 1989. [Language Transfer: Cross-Linguistic Influence in Language Learning](#), 1 edition. Cambridge University Press.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. [Language models are unsupervised multitask learners](#). *OpenAI blog*, 1(8):9.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stian Steinbakken and Björn Gambäck. 2020. [Native-language identification with attention](#). In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 261–271. Indian Institute of Technology Patna, Patna, India. NLP Association of India (NLP AI).

Chongyang Tao, Tao Shen, Shen Gao, Junshuo Zhang, Zhen Li, Kai Hua, Wenpeng Hu, Zhengwei Tao, and Shuai Ma. 2025. [LLMs are also effective embedding models: An in-depth overview](#). *Preprint*, arXiv:2412.12591.

Joel Tetreault, Daniel Blanchard, Aoife Cahill, and Martin Chodorow. 2012. [Native tongues, lost and found: Resources and empirical evaluations in native language identification](#). In *Proceedings of COLING 2012*, pages 2585–2602, Mumbai, India. The COLING 2012 Organizing Committee.

Ahmet Yavuz Uluslu, Gerold Schneider, and Can Yildizli. 2024. [Native language identification improves authorship attribution](#). In *Proceedings of the 7th International Conference on Natural Language and Speech Processing (ICNLSP 2024)*, pages 297–303, Trento. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A New Dataset and Method for Automatically Grading ESOL Texts. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Wei Zhang and Alexandre Salle. 2023. [Native language identification with large language models](#). *arXiv preprint*.

Shuping Zhao, Bob Zhang, Jian Yang, Jianhang Zhou, and Yong Xu. 2024. [Linear discriminant analysis](#). *Nature Reviews Methods Primers*, page 70.

## A Text generator systems

For the text generator systems on the TOEFL11 corpus, inference is conducted using the system and user prompts illustrated in Figures 3 and 4, respectively. These prompts are adapted from the work of Zhang and Salle (2023), where native language labels follow the ISO 639-3 standard identifiers, and the `< TEXT >` tag is populated with the text to classify. For the CLC-FCE corpus experiments,

You are a forensic linguistics expert that reads English texts written by non-native authors to classify the native language of the author as one of the following iso 639-3 code:

"ara": Arabic  
 "zho": Chinese  
 "fra": French  
 "deu": German  
 "hin": Hindi  
 "ita": Italian  
 "jpn": Japanese  
 "kor": Korean  
 "spa": Spanish  
 "tel": Telugu  
 "tur": Turkish

Use clues such as spelling errors, word choice, syntactic patterns, and grammatical errors to decide on the native language of the author. DO NOT USE ANY OTHER CLASS. IMPORTANT: Do not classify any input as "eng" (English). English is an invalid choice.

Valid output formats:

Class: "ara"  
 Class: "zho"  
 Class: "fra"  
 Class: "deu"  
 Class: "hin"  
 Class: "ita"  
 Class: "jpn"  
 Class: "kor"  
 Class: "spa"  
 Class: "tel"  
 Class: "tur"

You ONLY respond in JSON files. The expected output from you must be a json format, strictly: {"native\_lang": The chosen class, ara, zho, fra, deu, hin, ita, jpn, kor, spa, tel, tur}

Figure 3: System prompt for text generator systems inference.

Text to classify: "`< TEXT >`". Classify the text above as one of the following iso codes: ara, zho, fra, deu, hin, ita, jpn, kor, spa, tel, tur. Do not output any other class - do NOT choose "eng" (English). What is the closest native language of the author of this English text from the given list?

Figure 4: User prompt for text generator systems inference.

the prompt structure remains identical, with only the language targets updated to match the corpus' labels: Catalan (cat), German (deu), Greek (ell), French (fra), Italian (ita), Japanese (jpn), Korean (kor), Polish (pol), Portuguese (por), Russian (rus), Spanish (spa), Thai (tha), Turkish (tur), and Chinese (zho). During inference, the predicted label is

| Hugging Face model name                   | Number of parameters | Number of layers | Vector dimensions |
|-------------------------------------------|----------------------|------------------|-------------------|
| <i>LLM</i>                                |                      |                  |                   |
| mistralai/Mistral-7B-Instruct-v0.1        | 7B                   | 32               | 4 096             |
| meta-llama/Meta-Llama-3-8B-Instruct       | 8B                   | 32               | 4 096             |
| mistralai/Mistral-Small-24B-Instruct-2501 | 24B                  | 40               | 5 120             |
| meta-llama/Llama-3.1-70B-Instruct         | 70B                  | 80               | 8 192             |
| mistralai/Mistral-Large-Instruct-2411     | 123B                 | 88               | 12 288            |
| meta-llama/Llama-3.1-405B-Instruct        | 405B                 | 126              | 16 384            |
| <i>Encoder-only</i>                       |                      |                  |                   |
| FacebookAI/roberta-base                   | 0.1B                 | 12               | 768               |
| FacebookAI/roberta-large                  | 0.4B                 | 24               | 1024              |
| FacebookAI/xlm-roberta-base               | 0.3B                 | 12               | 768               |
| FacebookAI/xlm-roberta-large              | 0.6B                 | 24               | 1024              |

Table 3: Specifications of open-source models selected for our experiments.

extracted from the LLM output using the regular expression "`(\w{3})`".

## B Text embedder systems

For the text embedder systems, embeddings are extracted during inference via the `hidden_states` attribute using the Hugging Face Transformers library (Wolf et al., 2020). To compress the variable-length sequence of token vectors into a single fixed-size representation  $u$ , we apply a mean pooling strategy that averages the output vectors across all  $n$  tokens:

$$u = \frac{1}{n} \sum_{i=1}^n t_i$$

Our Linear Discriminant Analysis (LDA) classifier utilizes the `LinearDiscriminantAnalysis` implementation from Scikit-learn (Pedregosa et al., 2011), configured with a singular value decomposition (SVD) solver. The dimensionality of the reduced feature space is determined by  $\min(n_{classes} - 1, n_{features})$ , yielding 10 components for the TOEFL11 corpus and 13 components for the CLC-FCE corpus. In all experiments, classifiers are trained and evaluated on features extracted from the matching model and corpus pair.

## C Models

Selected open-source models for our experiments cover a large range of model sizes. Table 3 provides the Hugging Face model name, the number of parameters, the number of layers, and the vector dimensions of the selected models.

## D Confusion Matrices

Figure 5 provides the confusion matrices of the Llama 3 (8B), Mistral Small (24B) 2501 and Llama 3.1 (405B) text generator and text embedder systems on the TOEFL11 corpus. Each row is normalized to sum to 100.

## E Computation time

Table 4 provides the computation time of our text embedder systems on the TOEFL11 corpus, measured on the H100 partition of the Jean Zay super-computer.

| Model                     | H100 | Time    |
|---------------------------|------|---------|
| <i>LLM</i>                |      |         |
| Mistral (7B) v0.1         | 1    | 0:10:01 |
| Llama 3 (8B)              | 1    | 0:09:32 |
| Mistral Small (24B) 2501  | 1    | 0:16:13 |
| Llama 3.1 (70B)           | 4    | 0:44:04 |
| Mistral Large (123B) 2411 | 4    | 1:10:10 |
| Llama 3.1 (405B)          | 16   | 1:50:33 |
| <i>Encoder-only</i>       |      |         |
| RoBERTa (0.1B)            | 1    | 0:02:43 |
| RoBERTa Large (0.4B)      | 1    | 0:04:19 |
| XML-R (0.3B)              | 1    | 0:02:45 |
| XML-R Large (0.6B)        | 1    | 0:04:24 |

Table 4: Wall-clock computation time (H:mm:ss) of text embedder systems on the TOEFL11 corpus, covering the loading of the weights, the embedding of the 12,100 texts, and the fitting of the LDA head, on the reported number of H100.

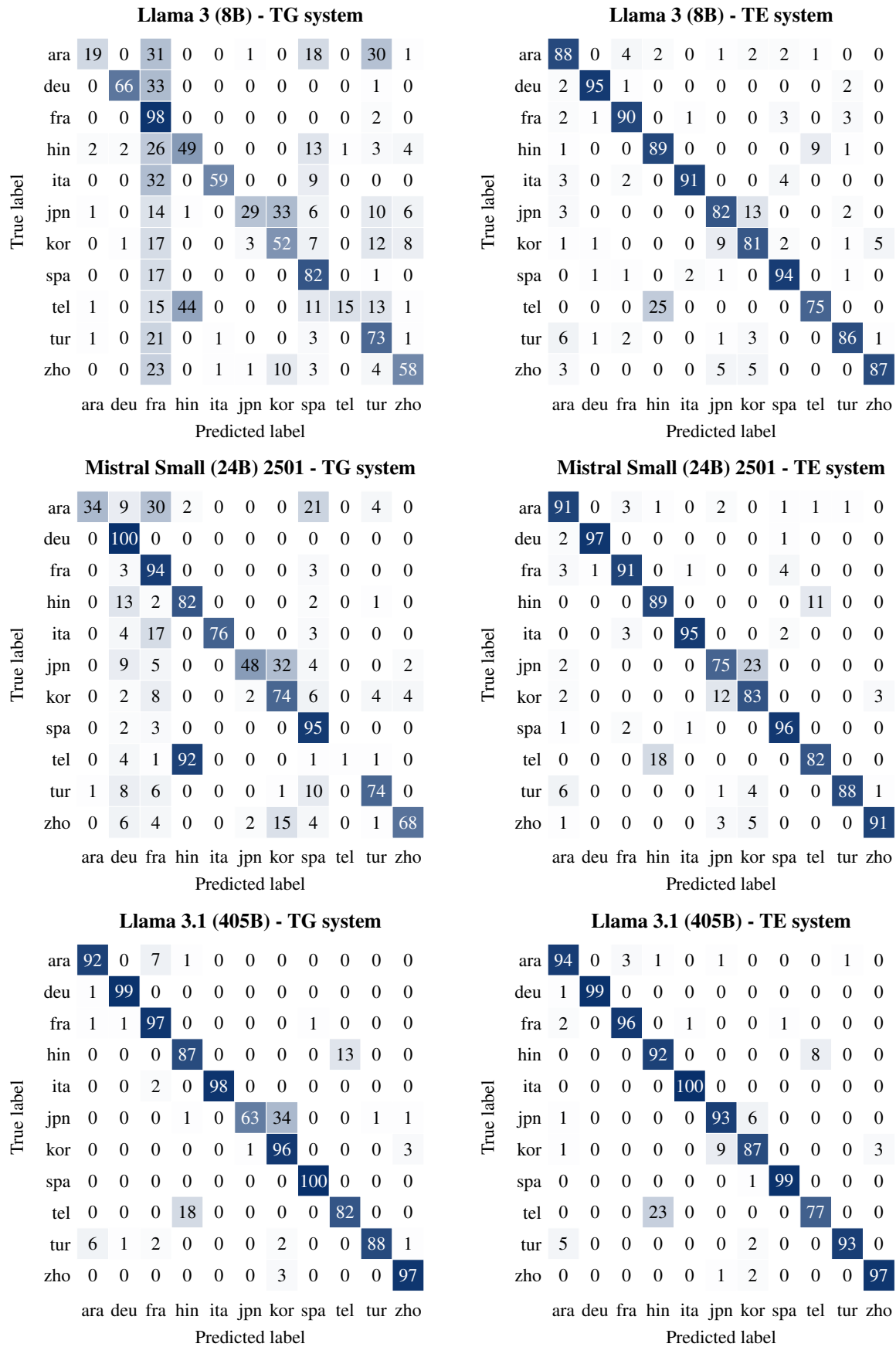


Figure 5: Confusion matrices of the Llama 3 (8B), Mistral Small (24B) 2501 and Llama 3.1 (405B) text generator (TG) and text embedder (TE) systems on the TOEFL11 corpus. Rows are true native languages and columns are predicted ones, both given as ISO 639-3 codes. Each row is normalized to sum to 100.