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

Anonymous ACL submission

Abstract

Native Language Identification (NLI) consists of predicting a person's native language based on their second language production. In the field of text-based NLI, Large Language Models (LLM) based approaches outperform previous approaches based on statistical classifiers trained on linguistic features. But the unstructured nature of LLM's textual output does not support the direct application of usual classifiers. To address this limitation, we are exploring the use of LLM to generate effective latent vector representations (embeddings) for NLI. Our experiments, across ten models ranging from 125M to 405B parameters, show that embeddings gathered from a frozen LLM coupled with a linear classifier bring the same 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 of 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 of the 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, pretrained 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 LLM's latent vector representations (embeddings) offers an alternative to 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 a LLM with a task-driven linear classifier. We conduct a comparative performance evaluation across ten models, ranging from 125 million 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?

2 Previous works

While early approaches for text-based NLI relied on statistical classifiers like logistic regres-

sion (Tetreault et al., 2012) and support vector machines (Markov, 2018) trained on linguistic features, they 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 (7B) accuracy score of 90%[1] (Ng and Markov, 2025; Nicholls and Alperin, 2025) and the fine-tuned proprietary model GPT-4 accuracy score of 92%[1] (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%[1] (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). 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.

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 behaviour introduces 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).

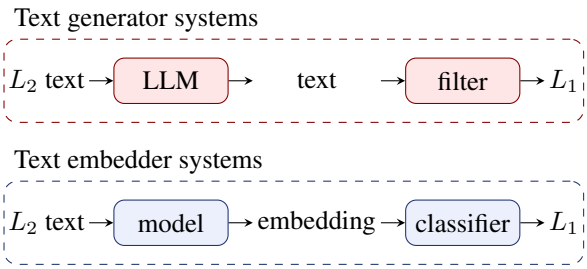


Figure 1: Illustration of NLI inference pipeline for text generator and text embedder systems. Both systems are provided with a text written in a second language (L_2) from which they predict the author's native language (L_1).

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.

The **CLC-FCE**[2] (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 through the sixteen represented. We partition the corpus into three subsets: 602 texts for training and 140 texts each for validation and testing.

[1] The results were obtained using the TOEFL11 corpus (Blanchard et al., 2013)

[2] Downloaded from: https://s3-eu-west-1.amazonaws.com/ilexir-website-media/fce-released-dataset.zip

LLM	TOEFL11	CLC-FCE
Fine-tuned text generator (TG-FT)		
GPT-4 (Goswami et al., 2025)	92	**83**
Mistral (7B) v0.1 (Ng and Markov, 2025)	89.8	-
Llama 3 (8B) (Ng and Markov, 2025)	85.3	-
Text generator (TG)		
Mistral (7B) v0.1	17.4 ±2.4	15.0 ±6.8
Llama 3 (8B)	54.5 ±2.9	47.1 ±8.2
Mistral Small (24B) 2501	67.8 ±2.7	58.6 ±7.8
Llama 3.1 (70B)	85.1 ±2.0	70.0 ±7.0
Mistral Large (123B) 2411	86.1 ±1.9	75.0 ±6.4
Llama 3.1 (405B)	90.8 ±1.6	79.3 ±5.9
Text embedder (TE)		
Mistral (7B) v0.1	83.9 ±2.1	60.7 ±7.7
Llama 3 (8B)	87.1 ±1.9	65.7 ±7.3
Mistral Small (24B) 2501	88.9 ±1.7	72.1 ±6.8
Llama 3.1 (70B)	88.5 ±1.8	80.7 ±5.7
Mistral Large (123B) 2411	90.8 ±1.6	82.1 ±5.5
Llama 3.1 (405B)	**93.4** ±1.3	82.1 ±5.5

Table 1: Accuracy score (%) and 95% confidence intervals according to Wilson's method of LLM-based text generator and text embedder systems on the TOEFL11 and CLC-FCE corpora.

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 models RoBERTa (Liu et al., 2019) and the multilingual XLM-R (Conneau et al., 2019), including *base* and *large* variants of each encoder-only model.

6 Results

As shown in Table 1, the 95% Wilson's confidence intervals 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 the accuracy of 90.8% ±1.6 on TOEFL11 and 79.3% ±5.9 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 the accuracy of 93.4% ±1.3 on TOEFL11 and 82.1% ±5.5 on CLC-FCE.

Mistral (7B) v0.1 (TG) brings very low performance with less than 20% accuracy on both corpora. An analysis of the LLM's text output shows an overproduction of the out-of-domain class "eng", which mirrors the language of the input text (L_2).

For a given LLM, TG systems are consistently outperformed by TE systems, where the performance differences between TG-FT and TE systems remain marginal (89.8% for Mistral (7B) v0.1 (TG-FT) and 85.3% Llama 3 (8B) (TG-FT) compared to 83.9% ±2.1 for Mistral (7B) v0.1 (TE) and 87.1% ±1.9 for Llama 3 (8B) (TE)).

For systems based on encoder-only models, Ta-

Encoder-only	TOEFL11
Fine-tuned text embedder (TE-FT)	
RoBERTa (125M) (2025)	71
XLM-R (125M) (2025)	63
Text embedder (TE)	
RoBERTa (125M)	70.5 ±2.6
XLM-R (125M)	68.6 ±2.7
RoBERTa Large (355M)	72.7 ±2.5
XLM-R Large (355M)	70.7 ±2.6

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

ble 2 also shows marginal performance differences between TE and TE-FT systems (71% for RoBERTa (125M) (TE-FT) and 63% for XLM-R (125M) (TE-FT) compared to 70.5% ±2.6 for RoBERTa (125M) (TE) and 68.6% ±2.7 for XLM-R (125M) (TE)).

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

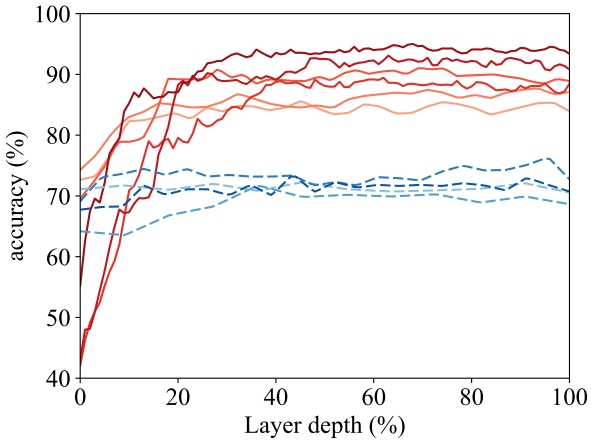
Figure 2 highlights distinct patterns between encoder-only models and LLM. For TE systems based on encoder-only models, the accuracy remains almost constant across all layers. In contrast, TE systems based on LLM yield lower performance in layers below 40% depth than deeper layers, where accuracy stabilizes.

Selecting the best layer instead of the last layer only brings minor performance improvements. When leveraging its optimal hidden layer, our best model Llama 3.1 (405B) achieves the accuracy score of 95.1% ±1.1 on the TOEFL11 corpus, surpassing the GPT-4 (TG-FT) performance. The best layer in terms of performance for each TE system is provided in Appendix D.

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

Furthermore, our layer depth analysis states 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 behaviour requires further study, this insight suggests a cost-saving opportunity: leveraging only the first layers for NLI instead of the full model.



Model	Number of layers
LLM	
Llama 3.1 (405B)	126
Mistral Large (123B) 2411	88
Llama 3.1 (70B)	80
Mistral Small (24B) 2501	40
Llama 3 (8B)	32
Mistral (7B) v0.1	32
Encoder-only	
XLM-R Large (355M)	24
RoBERTa Large (355M)	24
XLM-R (125M)	12
RoBERTa (125M)	12

Figure 2: Accuracy score (%) of text embedder systems on the TOEFL11 corpus based on the model's transformer layer depth (%) used for embedding extraction.

Limitations

While frozen LLM-based text embedders offer a high-performance and cost-effective alternative to fine-tuned generative systems for NLI, several limitations apply. First, potential data leakage within these models makes it difficult to isolate whether high accuracy stems from true capacity or prior data exposure. Second, due to our experimental protocol, the linear classifier is restricted to a fixed set of labels and cannot classify data from unknown native languages. Finally, we do not evaluate whether the discovered 40% layer depth stability threshold 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

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. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

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, Lélio 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.

Ilia 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 Ilia 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 (NLPAI).

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, 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 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

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.

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.

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.

C Models

Selected open-source models for our experiments cover a large range of model sizes. Table 4 provides the number of parameters, the number of layers and the vector dimensions of the selected models supplemented by information regarding the computational resources used.

D Best layer performance for text embedder systems

Table 3 provides the best layer performance for our text embedder systems on TOELF11 corpus.

Model	TOEFL11
RoBERTa	72.3 ±2.6
RoBERTa Large	76.4 ±2.4
XLM-R	71.7 ±2.6
XLM-R Large	73.5 ±2.5
Mistral (7B) v0.1	85.6 ±1.9
Llama 3 (8B)	87.7 ±1.8
Mistral Small (24B) 2501	91.1 ±1.5
Llama 3.1 (70B)	89.5 ±1.7
Mistral Large (123B) 2411	93.3 ±1.3
Llama 3.1 (405B)	**95.1** ±1.1

Table 3: Best layer accuracy score (%) and 95% confidence intervals of text embedder systems on the TOEFL11 corpus.

Model name	Number of parameters	Number of layers	Vector dimensions	Number of H100 used
LLM				
Mistral-7B-Instruct-v0.1	7B	32	4 096	1
Meta-Llama-3-8B-Instruct	8B	32	4 096	1
Mistral-Small-24B-Instruct-2501	24B	40	5 120	1
Llama-3.1-70B-Instruct	70B	80	8 192	4
Mistral-Large-Instruct-2411	123B	88	12 288	4
Llama-3.1-405B-Instruct	405B	126	16 384	16
Encoder-only				
roberta-base	125M	12	512	1
roberta-large	355M	24	1024	1
xlm-roberta-base	125M	12	512	1
xlm-roberta-large	355M	24	1024	1

Table 4: Specifications of open-source models selected for our experiments.