PolyLingua: Margin-based Inter-class Transformer for Robust Cross-domain Language Detection

Ali Lotfi Rezaabad, Bikram Khanal, Shashwat Chaurasia, Lu Zeng, Dezhi Hong, Hossein Beshashati, Thomas Butler, Megan Ganji

Alexa, Amazon
{alotfi,ganjmegj}@amazon.com

Abstract

Language identification is a crucial first step in multilingual systems such as chatbots and virtual assistants, enabling linguistically and culturally accurate user experiences. Errors at this stage can cascade into downstream failures, setting a high bar for accuracy. Yet, existing language identification tools struggle with key cases–such as music requests where the song title and user language differ. Open-source tools like LangDetect, FastText are fast but less accurate, while large language models, though effective, are often too costly for low-latency or low-resource settings. We introduce PolyLingua, a lightweight Transformer-based model for in-domain language detection and fine-grained language classification. It employs a two-level contrastive learning framework combining instance-level separation and class-level alignment with adaptive margins, yielding compact and well-separated embeddings even for closely related languages. Evaluated on two challenging datasets—Amazon Massive (multilingual digital assistant utterances) and a Song dataset (music requests with frequent code-switching)—PolyLingua achieves 99.25% F1 and 98.15% F1, respectively, surpassing Sonnet 3.5 while using 10× fewer parameters, making it ideal for compute- and latency-constrained environments.

Introduction

Identifying a customer's language quickly and accurately is critical for delivering a more personal and culturally respectful experience (Wu et al. 2025; Tedeschi et al. 2021). This need is especially acute in tools such as virtual assistants, chatbots, and customer-service centers that interact with speakers of many languages. Detecting the language at the earliest point in the interaction enables a personalized experience by routing the user to the most appropriate content or support channel exactly when it is needed, thereby reducing friction, increasing satisfaction, and improving overall engagement and conversion rates (Jauhiainen et al. 2019b). Moreover, early language detection allows downstream language models to be prompted in the user's language and cultural context, making the generated responses more helpful and respectful (Deroy and Maity 2024).

Commercial APIs and rule-based approaches (Nakatani 2010) are widely used for language detection because of their speed and simplicity. However, they frequently struggle with short utterances or when the text contains names or entities from another language (Joulin et al. 2017). At the same time, more precise approaches rely on large, highly resource-intensive models that are not ideal for systems with limited memory or latency. Despite its importance, language detection is frequently overlooked when developing better model architectures (Jauhiainen et al. 2019a). Although large language models (LLMs) often achieve higher accuracy, they are computationally more expensive. In contrast, encoder-only models are especially well-suited for task-specific problems such as sentiment analysis and named-entity recognition (Zaratiana et al. 2024), as demonstrated by recent work on ModernBERT (Warner et al. 2025).

In this paper, we frame the problem as multi-class classification because out-of-domain (OOD) languages can misleadingly resemble supported languages and be classified with high confidence. Therefore, we introduce POLYLINGUA, a lightweight yet effective multi-task model for language identification (Aghajanyan et al. 2021; Aribandi et al. 2022; Liu et al. 2021). PolyLingua performs two tasks simultaneously: (1) in-domain detection to determine whether a given utterance falls under a set of supported languages, serving as a quick gating mechanism to route the request, and (2) fine-grained language classification within those supported languages to enable precise customized experiences. Both tasks are trained end-to-end with a shared encoder, allowing the model to efficiently learn common representations while supporting task-specific objectives. This shared architecture leverages the complementary information between the two tasks to enhance generalization, while reducing the number of parameters and minimize computational overhead.

PolyLingua employs a two-level contrastive learning strategy (Wu et al. 2018; Wang, Liu, and Yu 2021). At the instance level, it utilizes a margin-based supervised contrastive loss for separating embeddings from confusable language pairs. At the class level, the model introduces a class-aware, margin-based separation loss that encourages each utterance to stay close to the mean embedding of its own language class while maintaining a minimum margin from those of

Personalization in the Era of Large Foundation Models, Workshop @ AAAI 2026.

other classes, particularly for analogous or confusable language pairs. This dual contrastive signal helps the model in forming tight, well-separated clusters in the embedding space, increasing both accuracy and generalization (Chen et al. 2020).

Finally, we conduct thorough assessments of PolyLingua against a variety of baselines, such as commercial APIs, rule-based tools, open-source multilingual models, and LLMs (Anthropic 2023) employed for language detection. Our experiments show that PolyLingua matches the performance of LLMs while using far fewer computational resources, making it a practical choice for real-world applications that need both accuracy and low latency. In summary, our contributions are:

- We propose PolyLingua, a lightweight, end-to-end multitask model that leverages a shared encoder to jointly perform in-domain detection and fine-grained language classification, avoiding the risk of confidently misclassifying OOD utterances as supported languages.
- We create a two-level contrastive training objective by combining instance-level supervision with a novel class-level separation loss that improves robustness to ambiguous languages.
- We present a comprehensive empirical evaluation of different methods using a variety of datasets. Results show that PolyLingua delivers high performance at a low computational cost.

Related Works

Recent research on multi-task learning has shown that it is capable of significantly improving sample efficiency and generalizing in low-resource and multilingual settings (Sanh et al. 2022; Aribandi et al. 2022). In parallel, contrastive learning has become a popular framework for training models with more structured and discriminative embeddings. Beyond SimCLR (Chen et al. 2020), efforts like SupCon (Khosla et al. 2020)and SPCL (Wang et al. 2022) introduce supervised contrastive mechanisms that improve performance, robustness, and generalization.

As multilingual user bases expand, services such as virtual assistants, chatbots, and support platforms should accurately detect a user's language to deliver personalized and culturally appropriate responses. Large models are sometimes too heavy or expensive for real-time use, and many commercial or rule-based solutions lack robustness. There are several approaches in the literature for language identification. For example, FastText (Joulin et al. 2017), langdetect (Nakatani 2010), and XLM-RoBERTa (Conneau et al. 2020a) are popular options. However, these methods often perform poorly on short or less structured text containing code-switching languages. While prompt-based LLMs offer higher accuracy (Touvron et al. 2023; Anthropic 2023), their computational overhead limits deployment in latency-sensitive or resource-constrained environments (Wan et al. 2023; Bai et al. 2024). In this paper, we propose PolyLingua, a model that balances speed and performance for efficient and real-world uses.

Methodology

Let assume the training dataset consist of utterance and label tuples $\{(u_i, y_i)\}_{i=1}^{N}$, where u_i is the input utterance, $y_i \in \mathcal{Y}$ is the language label (e.g., English, Spanish, etc.), and $d_i \in \{0, 1\}$ is a binary label indicating whether the utterance belongs to a supported in-domain language $d_i = 1$ (in-domain) or $d_i = 0$ (out-of-domain). Note that d_i can be directly inferred from y_i, as the set of in-domain languages $\mathcal{Y}_{\text{in}} \subset \mathcal{Y}$ is known. Leveraging this, we include d_i during training to enable effective multitask supervision. Below, we propose PolyLingua, a multi-task learning framework that learns more robust and separable representations to enhance language identification task.

Architecture

Contrastive learning is an approach that brings samples in the same class closer and pushes ones across classes further in the embedding space (Khosla et al. 2020) (see 1 for the objective). Leveraging this, we propose PolyLingua, a model based on two-level supervised contrastive objectives with adjustable inter-class margins, which improves performance on both in-domain detection and fine-grained language identification. The model is built based on a shared transformer encoder that converts the input utterance u_i into a representation. Based on this, we incorporate three components in this encoder: (1) an in-domain detection head that performs binary classification over d_i; (2) a language identification head that classifies among the in-domain language classes $y_i \in \mathcal{Y}_{\text{in}}$; and (3) a projection head that converts the encoder outputs into a L_2-normalized embedding space (see Figure 1). This projection is used to calculate a contrastive loss that pushes utterances from different classes away while encouraging utterances with the same language label to be near together in the embedding space.

Training Objectives

PolyLingua employs a two-level supervised contrastive loss at the instance level. The loss for the projected embeddings $\{z_i\}$ pushes apart different-label tuples with class-awareness that promoting proximity between same-label pairs.

$$\mathcal{L}_{\text{instance}} = \sum_{i \in \mathcal{I}} \frac{1}{|P(i)|} \sum_{p \in P(i)} -\log \left(\frac{\exp(z_i^\top z_p / \tau)}{\sum_{a \in A(i)} \exp(z_i^\top z_a / \tau)} \right), \tag{1}$$

where $\tau > 0$ is the temperature, $\mathcal{I}$ represents the set of all anchor instances, and $P(i)$ represents the set of indices corresponding to positive utterances for instance i, encouraging proximity between pairs with the same label. $A(i)$ stands for the set of all candidate samples used in the batch except i.

In addition to instance-level contrastive learning, we enforce separation at the class level by encouraging each embedding z_i to be close to the mean embedding of its class and far from those of other classes. Let c_y denote the centroid (mean embedding) of class y, computed as:

$$c_y = \frac{1}{|A_y|} \sum_{i \in A_y} z_i, \tag{2}$$

where $A_y = \{i \in \mathcal{I} : y_i = y\}$ is the set of samples belonging to class y in the batch. We then define a class-level contrastive loss:

$$\mathcal{L}_{\text{class}} = \sum_{i \in \mathcal{I}} -\log \left(\frac{\exp(z_i^\top c_{y_i}/\tau)}{\sum_{y \in \mathcal{Y}_{\text{in}}} \exp(z_i^\top c_y/\tau - \delta_{y_i,y})} \right), \quad (3)$$

where $\delta_{y_i,y} \geq 0$ is an adaptive margin penalty between classes y_i and y.

Adaptive margins for linguistically similar classes: A key contribution of our approach is the introduction of class-pair-specific adaptive margins that reflect linguistic similarity between languages. Linguistically similar languages (e.g., Portuguese and Spanish, which share substantial vocabulary and grammatical structures, or French and Spanish with considerable lexical overlap) are more prone to confusion. To enforce stronger separation between such confusing pairs, we assign larger margins:

$$\delta_{y_i,y} = \begin{cases} 0 & \text{if } y_i = y \\ \delta_{\text{high}} & \text{if } y_i \neq y \text{ and } (y_i, y) \in \mathcal{P}_{\text{confusing}} \\ \delta_{\text{low}} & \text{otherwise} \end{cases} \quad (4)$$

where $\delta_{\text{high}} > \delta_{\text{low}} \geq 0$, $\mathcal{P}_{\text{confusing}}$ is the set of confusing language pairs identified based on linguistic family relationships and empirical confusion patterns. The margin penalty $\delta_{y_i,y}$ reduces the similarity score for negative class centroids in the denominator of Equation 3, making $\exp(z_i^\top c_y/\tau - \delta_{y_i,y})$ smaller for negative classes. This effectively decreases their contribution to the denominator, making it harder for the model to assign high probabilities to incorrect classes, particularly those linguistically similar to the true class. During training, this mechanism pushes each embedding closer to its class centroid while enforcing larger separations from confusing classes, resulting in well-separated and discriminative language clusters in the embedding space.

Combined objective: The final training objective combines both contrastive losses with the standard cross-entropy losses for in-domain detection and language identification:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{indomain}} + \lambda_2 \mathcal{L}_{\text{langid}} + \lambda_3 (\mathcal{L}_{\text{instance}} + \mathcal{L}_{\text{class}}), \quad (5)$$

where $\mathcal{L}_{\text{indomain}}$ and $\mathcal{L}_{\text{langid}}$ are cross-entropy losses for the domain detection and language identification heads respectively, and λ_1, λ_2 are weighting hyperparameters that balance the contribution of each loss component.

Positive pairs: To generate positive pairs for contrastive learning, we utilized data augmentation techniques that maintain the original semantics. An original utterance and its augmented variant produced using the following techniques make up each positive pair:

- **Random Deletion**: Randomly removing tokens (with probability of $p = 0.15$).
- **Adding Noise**: Injects character-level perturbations typing errors and noisy user inputs or ASR transcriptions.
- **Dynamic Entity Replacement**: Using a multilingual named entity recognition model (Tedeschi et al. 2021) to locate entities in the utterance with high confidence. Then in each epoch, we replaced another randomly chosen entity of the same type. This increases cross-lingual diversity while preserving semantic consistency.

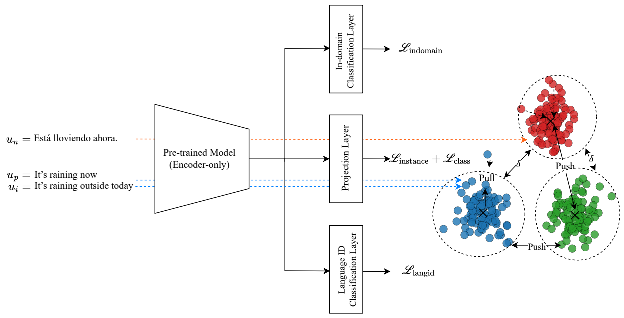

u_n = Está lloviendo ahora.
u_p = It's raining now
u_i = It's raining outside today
Pre-trained Model (Encoder-only)
In-domain Classification Layer
Projection Layer
Language ID Classification Layer
$\mathscr{L}_{\text{indomain}}$
$\mathscr{L}_{\text{instance}} + \mathscr{L}_{\text{class}}$
$\mathscr{L}_{\text{langid}}$
Pull
Push
Push
δ
δ

Figure 1: Given u_i and a positive u_p from the same class, and a negative u_n from a different language, PolyLingua aligns u_i with u_p while pushing it away from u_n, using a shared encoder and class-aware margin-based contrastive losses.

Further information on data augmentation and hyperparameter tuning is provided in Appendix A.

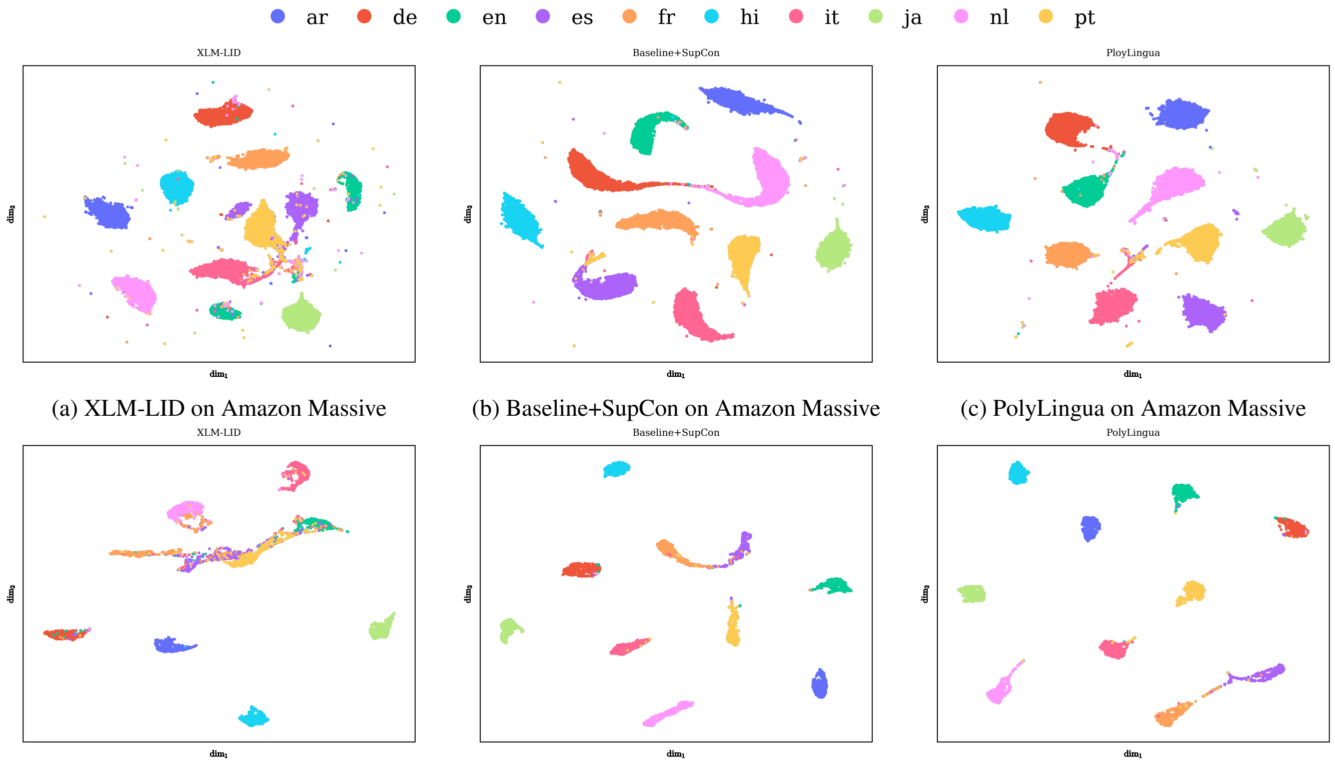
Experiments

Model Setup

We train our model in a multitask setup consisting of two classification objectives. We use a single backbone with two classification heads for each objective. As a result, sharing a common base model for both tasks simplifies deployment and lowers latency during inference. Additionally, we used a projection head with a margin-based, class-aware loss to help the model group examples from the same language together and push apart different ones. This made it easier to tell apart similar languages like Portuguese, Spanish, and French. For experiments, we employed the multilingual MiniLM encoder (Gu et al. 2023) due to its efficiency and robust performance across a wide range of languages. The encoder processes each utterance and produces the embeddings. Lastly, the model is trained end-to-end using the joint objective presented in 5.

Datasets

Amazon Massive We use the Amazon Massive dataset, which covers 52 languages on natural language understanding (NLU) tasks. In our experiments, we only focus on the utterances and their language labels to build a dataset for both in-domain detection and language classification. For this, we pick 10 in-domain target languages for classification: English, Spanish, French, Arabic, Hindi, Dutch, German, Italian, Portuguese, and Japanese, which are the key languages for our international businesses. Therefore, utterances from the remaining languages are considered out-of-domain (OOD) and are used for the in-domain detection task. To preserve class balance as the original dataset, we



(a) XLM-LID on Amazon Massive

(b) Baseline+SupCon on Amazon Massive

(c) PolyLingua on Amazon Massive

(d) XLM-LID on Song Dataset

(e) Baseline+SupCon on Song Dataset

(f) PolyLingua on Song Dataset

Figure 2: UMAP projections of utterance embeddings from Amazon Massive (top row) and the Song dataset (bottom row). Each point represents an utterance and is colored by its true language label. On Amazon Massive, PolyLingua forms more compact and well-separated clusters than XLM-LID and Baseline+SupCon, especially for similar languages such as French, Portuguese, and Spanish, due to its margin-based inter-class separation. On the Song dataset, which includes noisy utterances with diverse and multilingual artist and song entities, PolyLingua again shows clearer clustering, demonstrating robustness to intra-class variation and entity-induced noise.

subsample the out-of-domain utterances such that the out-domain examples constitute 40% of the total training set size. Additionally, we follow the same approach to create the test set from the Amazon Massive test split. More information about the dataset has been provided in Appendix C.

Synthetic Dataset To test our model's ability to handle real and less structured inputs, we also constructed a difficult synthetic dataset containing a lot of entities that mimics real-world conversations, such as music requests with artist and song names. For each in-domain language, we created 50 different diverse templates with placeholders like *[SONG_NAME]* and *[ARTIST_NAME]*. Eventually, we filled these placeholders with actual names from the Million Song Dataset (Bertin-Mahieux et al. 2011). As a result, the resulting dataset covers a broad range of entities; spanning several languages, genres, and scripts (such as non-standard capitalization and accented characters). This made the utterances complex and often ambiguous. Moreover, these templates reflect informal grammar, code-switching, and incomplete structures common in real-world utterances. We eventually created 10,000 of these utterances per language that many language identification techniques currently in use find challenging to tag. See Appendix C for more information.

Base Models

We tested our proposed model against a variety of language detection methods and baselines: (1) **Baseline**: A multi-task model trained with just cross-entropy (CE) loss for ablation analysis, using the same architecture as our proposed method; (2) **Baseline+SupCon**: A multi-task model trained with the same architecture, using cross-entropy loss for both tasks along with a supervised contrastive loss (Conneau et al. 2020b), allowing us to evaluate the contribution of the proposed loss functions; (3) **FastText (LID.176.bin)**: An efficient, open-source language identifier trained on Wikipedia text (Joulin et al. 2017); (4) **Langdetect**: A lightweight, rule-based language detection tool (Nakatani 2010); (5) **XLM-LID**: A multilingual transformer model fine-tuned for language identification, available at *papluca/xlm-roberta-base-language-detection*; and (6) **Sonnet 3.5 (via Amazon Bedrock)**: A powerful large language model prompted with a carefully crafted instruction for language detection. See Appendix B for additional details about the prompt.

Table 1: Comparison of models performance on the Amazon Massive and Song datasets. Metrics reported: In-domain Accuracy (InAcc), Precision (Prec), Recall, Top-1 Accuracy (Top-1), Top-5 Accuracy (Top-5), and F1 Score, all in percentage (%).

Model	**Amazon Massive**						**Song Dataset**					
	InAcc	Prec	Recall	Top-1	Top-5	F1	InAcc	Prec	Recall	Top-1	Top-5	F1
Langdetect	95.08	79.17	79.29	79.29	–	78.60	90.25	71.33	40.13	40.13	–	41.18
FastText	97.74	94.56	92.95	92.95	–	93.20	95.44	89.60	79.44	79.44	–	81.53
XLM-LID	82.59	97.43	96.02	96.02	98.64	96.65	91.45	87.02	82.18	82.18	90.06	81.73
Baseline	**99.58**	99.20	99.20	99.20	**99.52**	99.20	96.24	95.41	91.83	91.83	98.84	93.47
Baseline+SupCon	99.27	99.24	99.23	99.23	99.47	99.23	97.82	95.79	93.15	93.15	96.60	94.31
Sonnet 3.5.	99.67	97.90	97.77	97.77	–	97.79	98.95	96.07	95.03	95.03	–	95.52
PolyLingua	99.51	**99.25**	**99.25**	**99.25**	99.49	**99.25**	**99.71**	**98.29**	**98.06**	**98.06**	**99.73**	**98.15**

These baselines include a wide variety of simple rule-based tools, open-source models, commercial APIs, and large language models, enabling us to compare our approach from different angles.

Results

The results in Table 1 compare multiple language identification models evaluated on the two datasets. Conventional tools such as Langdetect and FastText show reasonable performance, with FastText outperforming Langdetect notably in both precision and recall on the Amazon dataset. XLM-LID shows high recall but has significantly lower in-domain accuracy (82.59%). Our two ablation models—Baseline and Baseline+SupCon—show consistent improvement over open-source and commercial tools, with top-5 accuracy exceeding 99% in both variants, suggesting that supervised contrastive learning contributes positively to model robustness. For additional comparative examples of PolyLingua versus Sonnet 3.5, see Table 5 in the Appendix C.

Our proposed model, PolyLingua, achieves the best overall performance across both datasets. On Amazon Massive, it has a top-1 accuracy of 99.25% and the highest F1 score of 99.25%. More importantly, the improvements are even more prominent on the more challenging Song Dataset. Additionally, Polylingua surpasses all baselines with an F1 score of 98.15%, indicating strong generalization across unseen and noisier utterances. Compared to Sonnet 3.5, for which we constructed a strong tailored prompt (see Appendix C), PolyLingua delivers higher accuracy and recall, particularly in top-5 metrics (99.73% vs. 96.60%). Overall, these findings show that PolyLingua performs better than LLM, commercial APIs, and supervised baseline. This highlights the efficacy of our end-to-end training. Also, we report on average latency per single inference in Table 2, with full benchmarking details provided in Appendix A.

In Figure 3, left plot displays the confusion matrix for PolyLingua, indicating high true positive rates and low confusion. On the other side, the center and right plots visualize the differential impact of PolyLingua over the Baseline+SupCon and Baseline models, respectively. Blue on diagonal cells show gains in true positives, while red off-diagonal cells show reduced misclassification. These results demonstrate that our two-level margin-based contrastive loss significantly enhances language separation and reduces confusion, especially among closely related languages such as Spanish, Portuguese, French and English. Arrows in the caption highlight improvements.

Table 2: Average latency per inference call for different models.

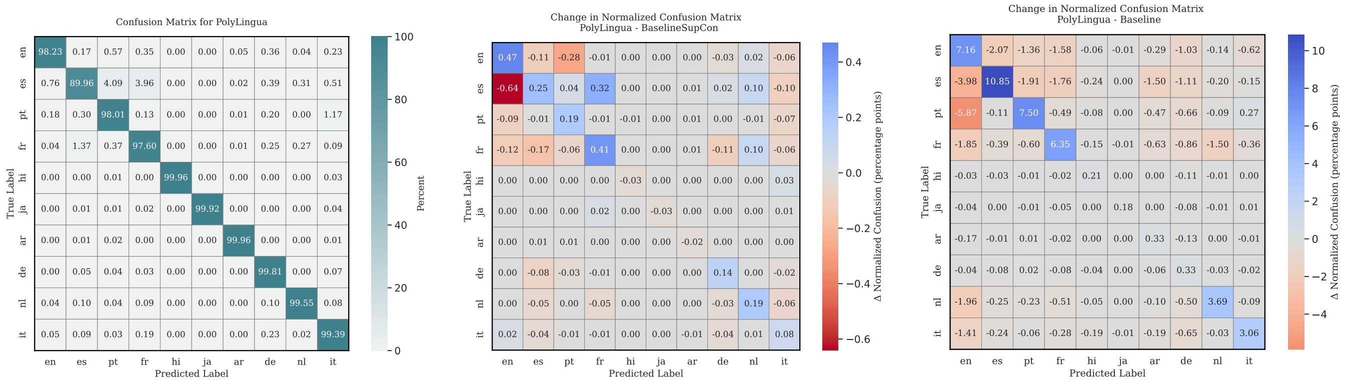
Model	**Latency (ms)**
Langdetect	8.53
FastText	0.01
XLM-LID	7.29
Sonnet 3.5	1373.31
PolyLingua	**7.19**

Table 3: Performance comparison on Amazon Massive dataset.

Method	T-1↑	T-5↑	Inter↑	Intra↓	Ratio↑
Baseline+SupCon	93.98	95.02	0.042	0.042	9.72
XLM-LID	84.64	88.91	0.242	0.978	4.04
PolyLingua	**97.69**	**98.43**	**1.096**	**0.037**	**29.79**

Figure 2 presents 2D UMAP projections of model embeddings on the Amazon Massive and Song datasets for visualization. In both cases, each maker is an utterance colored by its language. We can see how well each model groups same-language utterances and separates different languages. Our model, PolyLingua, gives tighter and more separated clusters compared to XLM-LID and Baseline+SupCon. Indeed, it clearly separates confusing language pairs like French, Portuguese, and Spanish. On the Song dataset, which includes artist and song names with added noise, PolyLingua still forms clean language groups. This shows it can focus on language signals even when the input is noisy.

To further assess how well different approaches distinguish between similar and dissimilar utterances, we quantify the results in Figure 4. The cosine similarity for both positive and negative pairings can be seen in the top histograms.

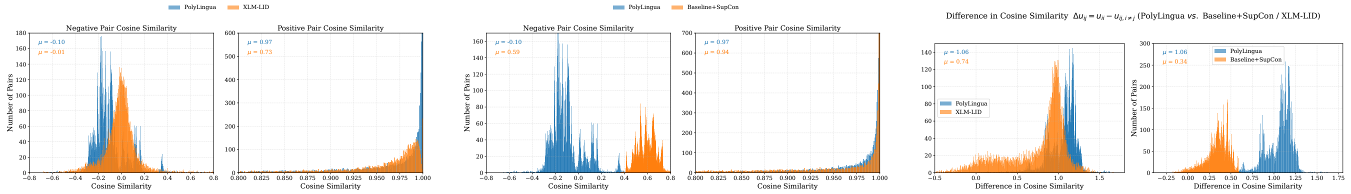


(a) PolyLingual Confusion Matrix

(b) Change in Confusion: PolyLingua – BaselineSupCon

(c) Change in Confusion: PolyLingua – Baseline

Figure 3: **Left**: Confusion matrix for the proposed PolyLingua model on 10 in-domain languages, demonstrating low misclassification. **Center**: Difference in the normalized confusion matrices between PolyLingua and the Baseline+SupCon model. **Right**: Difference between PolyLingua and the standard Baseline. Blue cells on the diagonal indicate improvements in true positive rates by PolyLingua, while red cells off the diagonal represent reductions in misclassification and confusion. Arrow pointer indicates the improvement in the performance in the Difference confusion matrix .



(a) Cosine similarity for positive and negative pairs:
PolyLingua vs XLM-LID.

(b) Cosine similarity for positive and negative pairs:
PolyLingua vs Baseline+SupCon.

(c) Difference in cosine similarity for positive vs. negative pairs: PolyLingua vs. XLM-LID/Baseline+SupCon

Figure 4: Comparison of cosine similarity distributions between PolyLingua and baseline models on the Amazon Massive dataset. (a) and (b) show positive and negative pair distributions for PolyLingua against XLM-LID and Baseline+SupCon, respectively. (c) shows difference histograms, indicating better class separation in PolyLingua.

Positive samples for PolyLingua are located closer together, while negative samples are broadly spaced compared to both Baseline+SupCon and XLM-LID. More importantly, the difference between the positive and negative pair similarities is displayed in the bottom plots. It is worth mentioning that better separation is indicated by a larger difference. Using this, the biggest difference is seen by PolyLingua, which demonstrates superior class boundary learning.

These findings demonstrate that PolyLingua creates clusters that are more robust and distinct than the baselines. Also, different approaches of assessing model performance are shown in Table 3, which uses a k-NN classifier on the learned embeddings to measure classification accuracy. This approach evaluates the capability of presented models to effectively capture representations without the need for a separate classifier head. With higher Top-1 and Top-5 accuracy, larger inter-class distance, and lower intra-class variation, PolyLingua performs better than Baseline and XLM-LID on all measures. Its high inter-to-intra ratio provides more evidence of its capacity to create small, distinct language clusters.

Conclusion

We introduced PolyLingua, a lightweight multi-task model for language identification that performs in-domain detection and fine-grained classification simultaneously. PolyLingua achieves outstanding accuracy and computational efficiency by utilizing a shared encoder and a two-level contrastive learning strategy. As part of this, we propose a novel two-level margin-based contrastive loss function that brings embeddings of the same class closer and explicitly enforces separation between different classes by introducing adjustable inter-class margins at the class level. By doing this, it promotes compact clusters and improves the model's performance to distinguish between closely related languages.

References

Aghajanyan, A.; Guu, K.; Roberts, A.; Chen, W.-t.; Choudhury, S.; and Zettlemoyer, L. 2021. Muppet: Massive Multi-task Representations with Pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Anthropic. 2023. Claude: Next-Generation AI Assistant. https://www.anthropic.com/index/introducing-claude. Accessed: 2025-05-04.

Aribandi, V.; Paranjape, A.; Shi, K.; and Kale, M. 2022. Ext5: Towards Extreme Multi-task Scaling for Transfer Learning. In *Proceedings of the 39th International Conference on Machine Learning*.

Bai, G.; Chai, Z.; Ling, C.; Wang, S.; Lu, J.; Zhang, N.; Shi, T.; Yu, Z.; Zhu, M.; Zhang, Y.; et al. 2024. Beyond efficiency: A systematic survey of resource-efficient large language models. *arXiv preprint arXiv:2401.00625*.

Bertin-Mahieux, T.; Ellis, D. P.; Whitman, B.; and Lamere, P. 2011. The Million Song Dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*.

Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; and Stoyanov, V. 2020a. Unsupervised Cross-lingual Representation Learning at Scale. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8440–8451. Online: Association for Computational Linguistics.

Conneau, A.; Khandelwal, K.; Goyal, N.; and et al. 2020b. Unsupervised Cross-lingual Representation Learning at Scale. In *ACL*.

Deroy, A.; and Maity, S. 2024. Prompt Engineering Using GPT for Word-Level Code-Mixed Language Identification in Low-Resource Dravidian Languages. *arXiv preprint arXiv:2411.04025*.

Gu, Y.; Dong, L.; Wei, F.; and Huang, M. 2023. MiniLLM: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*.

Jauhiainen, T.; Lui, M.; Zampieri, M.; Baldwin, T.; and Lindén, K. 2019a. Automatic language identification in texts: a survey. *J. Artif. Int. Res.*, 65(1): 675–682.

Jauhiainen, T.; Lui, M.; Zampieri, M.; Lindén, K.; and Jauhiainen, H. 2019b. Automatic language identification in texts: A survey. *Journal of Artificial Intelligence Research*, 65: 675–782.

Joulin, A.; Grave, E.; Bojanowski, P.; and Mikolov, T. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 427–431.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised Contrastive Learning. In *NeurIPS*.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2021. Pre-train Prompt Tune: Towards Efficient Foundation Model Tuning via Prompt Transfer. In *Advances in Neural Information Processing Systems*.

Nakatani, S. 2010. Langdetect: Language detection library ported from Google's language-detection library. https://github.com/shuyo/language-detection.

Sanh, V.; Webson, A.; Raffel, C.; and et al. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *ICLR*.

Tedeschi, S.; Maiorca, V.; Campolungo, N.; Cecconi, F.; and Navigli, R. 2021. WikiNEuRal: Combined Neural and Knowledge-based Silver Data Creation for Multilingual NER. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2521–2533. Punta Cana, Dominican Republic: Association for Computational Linguistics.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wan, Z.; Wang, X.; Liu, C.; Alam, S.; Zheng, Y.; Liu, J.; Qu, Z.; Yan, S.; Zhu, Y.; Zhang, Q.; et al. 2023. Efficient large language models: A survey. *arXiv preprint arXiv:2312.03863*.

Wang, X.; Liu, Z.; and Yu, S. X. 2021. Unsupervised Feature Learning by Cross-Level Instance-Group Discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12586–12595.

Wang, Y.; Chen, W.; Zhuang, B.; and et al. 2022. SPCL: Supervised Prototypical Contrastive Learning for Few-shot Recognition. In *CVPR*.

Warner, B.; Chaffin, A.; Clavié, B.; Weller, O.; Hallström, O.; Taghadouini, S.; Gallagher, A.; Biswas, R.; Ladhak, F.; Aarsen, T.; Adams, G. T.; Howard, J.; and Poli, I. 2025. Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2526–2547. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Wu, J.; Yang, S.; Zhan, R.; Yuan, Y.; Chao, L. S.; and Wong, D. F. 2025. A Survey on LLM-Generated Text Detection: Necessity, Methods, and Future Directions. *Computational Linguistics*, 51(1): 275–338.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised Feature Learning via Non-Parametric Instance Discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zaratiana, U.; Tomeh, N.; Holat, P.; and Charnois, T. 2024. GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Proceedings of the 2024 Conference* *of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 5364–5376. Mexico City, Mexico: Association for Computational Linguistics.

Appendix A: Implementation Details

Parameters

We fine-tune a lightweight multilingual transformer, MiniLM-L12-H384, in a multi-task learning setup that performs binary in-domain detection as well as multi-class language identification. Notably, the suggested technique is model-agnostic and can be used with any other pretrained encoder. Please see Table 4 for more details. Hyperparameters, including $\lambda_1, \lambda_2, \cdots, \delta_{\text{high}}$ and others, were optimized using random search.

Latency Evaluation Setup

To evaluate inference efficiency, we measured the average latency for each model over 200 independent single-inference runs. For transformer-based models like XLM-LID and PolyLingua, the measurements were conducted on an A10 GPU. For Sonnet 3.5, latency was calculated using Amazon Bedrock API calls made from an EC2 instance. For lightweight baselines such as FastText and Langdetect, latency was computed by their python package directly on the same EC2 instance. All measurements reflect end-to-end latency per input utterance.

Augmentation

Appendix B: Prompt for Sonnet 3.5

We employ the following prompt to assess the language identification capabilities of proprietary models, like Claude Sonnet 3.5. By specifically telling the model to disregard named entities like artist or brand names, the prompt aims to isolate the linguistic structure of each utterance. To guarantee deterministic results, we set the temperature to 0.0 and use the Bedrock API to invoke the model. The model is expected to return only the ISO 639-1 code of the primary language enclosed in `<language>...</language>` tags.

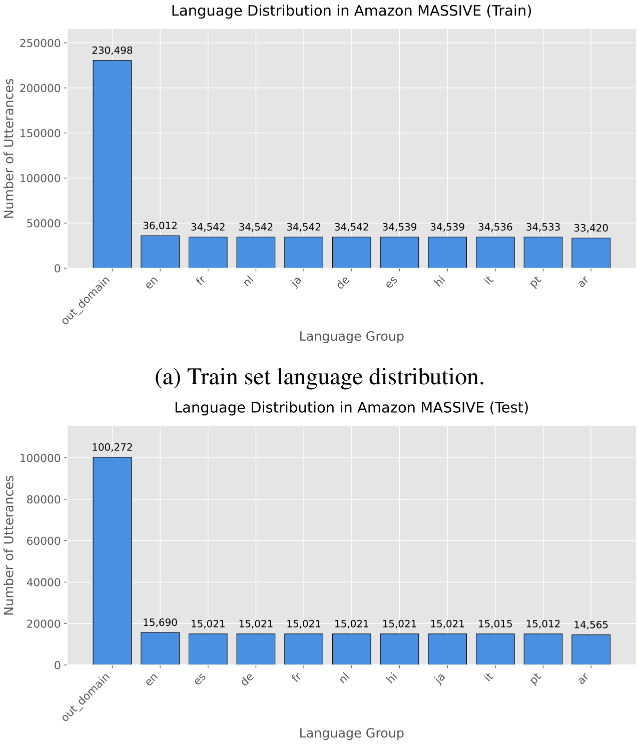
Prompt for Language Detection (Claude Sonnet 3.5)

Identify any named entities (titles, brand names, people, etc.). Ignore these named entities when determining the overall language. Focus on the remaining words and grammar. DO NOT OUTPUT ANY REASONING. Just output the ISO 639-1 code of the text's primary language between `<language>...</language>` tags.
Example: Reproduciendo "My Ghoulish Figure" de Conan O'Brien Needs A Friend en Amazon Music.
`<language>es</language>`
`<text>utterances</text>`

Appendix C: Datasets Statistics

Amazon Massive

In this section, we visualize the distribution of in-domain language labels and the number of words per utterance. The Amazon MASSIVE dataset exhibits a diverse range of languages and utterance lengths, which presents challenges for consistent language identification.



(a) Train set language distribution.

(b) Test set language distribution.

Figure 5: Distribution of language labels in the Amazon MASSIVE dataset. We consider ten languages as in-domain; all others are grouped under `out_domain`.

Table 4: Implementation Details

Parameters	**Values**
Pretrained Model	`Multilingual-MiniLM-L12-H384`
Max Input Length	256 tokens
Batch Size (per GPU)	150
Epochs	10
Optimizer	AdamW
Learning Rate	2e-5
Scheduler	CosineAnnealingLR ($T_{max} = 5$)
Temperature (τ)	0.07
Class Contrastive Loss Weight	20.0
Inter-class Margin Loss Weight	20.0
Contrastive Margin (δ_{high})	0.4 (for pairs of `es, pt, fr`)
Contrastive Margin (δ_{low})	0.0 (for other pairs)
Classification Loss Weight (λ_1, λ_2)	1.0 (each for in-domain and language ID)
Contrastive Loss Weight λ_3	0.1
Augmentations	Random deletion, typo noise, entity replacement

Table 5: sample code-mixed utterances from the Song Dataset where PolyLingua correctly identifies the primary language while Sonnet 3.5 misclassifies them. Utterances contain commands and song/artist names (highlighted in orange) mixing multiple languages.

Utterance	**Label**	**PolyLingua**	**Sonnet 3.5**
reproduz Looking Thru Bloodshot Eyes do The Casualties	pt	pt	en
toca por favor G2 K1 do Tara Putra	pt	pt	en
balance-moi Number One de Lightspeed Champion	fr	fr	en
je peux entendre Takaj Zhizn de Amsterdam Klezmer Band	fr	fr	nl
hey mets Sleepy Shores de Laurindo Almeida	fr	fr	pt

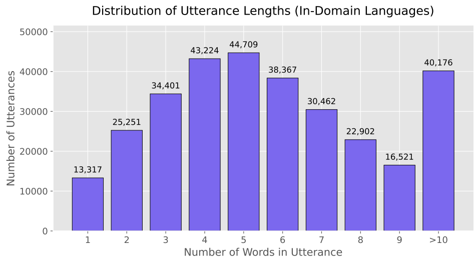


Figure 6: Distribution of utterance lengths (in words) for in-domain languages. Most utterances contain fewer than 7 words, emphasizing the need for robust modeling of short inputs.

Samples from Song Dataset

Utterance	**Label**
ich will Nautical Spirits_ Welcome to the Aquarium von Our Brother The Native hören	de
lass Penny Lane von Orlando Pops Orchestra laufen	de
oh spiel von Whole Wheat Bread	de
kannst du das Lied My Girlhood Among The Outlaws von Maria McKee spielen	de
could you put on Something Right by Julia Fordham	en
play This Room by Mikroboy feat Get Well Soon please	en
i want to listen to Cry now by Elliott	en
would you play Was wir alleine nicht schaffen by Xavier Naidoo	en
pon por favor Life In The Air Age de Be Bop Deluxe	es
tócanos When The Light Came Around de Florian Horwath	es
ponme por favor Fruitcake de Eraserheads	es
este pon People come people go de David Guetta Joachim Garraud Chris Willis	es
mets la chanson Cuando Te Vi de Vicentico	fr
j'aimerais écouter Oh mama oh papa de Ottavo Padiglione	fr
on écoute Third From The Sun de Chrome	fr
est-ce que tu pourrais mettre LÉternel féminin de Juliette Gréco	fr
mi fai sentire Mannequins di 108	it

Table 6: Some typical samples from the Song dataset