

MixLID: Token-Level Language Identification in Mixed-Language Text

Seongtae Hong, Youngjoon Jang, Seungyoon Lee, Jungseob Lee, Heuiseok Lim*

Department of Computer Science and Engineering, Korea University

ghdchlwlsl23@korea.ac.kr

Abstract

Processing multilingual text requires both identifying a broad range of languages and distinguishing the language of each token in inputs containing multiple languages. We introduce MixLID, a token-level language identification model supporting 111 language–script combinations. MixLID follows a two-stage training process: it first learns to distinguish individual languages using the language labels of monolingual examples, then continues training on mixed-language data. The second stage uses concatenated examples from existing code-switching data alongside augmented examples created through word substitution based on correspondences between English source words and loanwords. For monolingual identification, MixLID outperforms all baselines in both accuracy and macro F1 on four benchmarks within the language coverage shared with each baseline. For mixed-language identification, it outperforms the baselines in identifying the language of individual tokens on every evaluated language pair. MixLID achieves high performance across a broad range of languages in both monolingual and mixed-language settings. The model is publicly available at <https://huggingface.co/nlpai-lab/MixLID>.

1 Introduction

Language identification (LID) determines the language used in a text, providing language information for constructing and using multilingual datasets (Burchell et al., 2025). In large-scale corpus construction, LID is used to categorize text by language and filter for a target language (Wenzek et al., 2020; Nguyen et al., 2024). Errors in language identification can lead to the inclusion of text in other languages or the exclusion of target-language text, affecting the composition and quality of language-specific training data (Penedo et al., 2025; Kreutzer et al., 2022). Beyond data curation,

LID is also used to evaluate whether generated text is in the requested language (Chollampatt et al., 2025). In reinforcement learning, language matching is incorporated into the reward to encourage models to respond in the target language (Liu et al., 2026; Zhang et al., 2026).

Treating an input as a single unit for language identification yields one language label for the entire text. Real-world text, however, may exhibit code-switching across sentences or within a sentence through the insertion of words or expressions from another language (Elfardy and Diab, 2012; Burchell et al., 2024a). A single label cannot distinguish the languages used in different spans of such an input (Kargaran et al., 2024). This motivates token-level identification in addition to identifying the predominant language of the input (Nguyen and Doğruöz, 2013; Solorio et al., 2014).

Prior work has expanded the language coverage of datasets and models to include low-resource languages (Burchell et al., 2023; Kargaran et al., 2023; Adebara et al., 2022). For token-level LID in mixed-language inputs, supervised sequence labeling approaches use naturally occurring code-switched data with token-level language annotations to learn from word and character features and context (Chit-taranjan et al., 2014; Samih et al., 2016). To address the scarcity of token-level annotations, synthetic data have also been constructed by randomly concatenating or mixing phrases from different languages and used for training alongside monolingual and naturally occurring code-switched data (Zhang et al., 2018). Training for mixed-language inputs thus requires considering both how expressions from different languages are combined and how monolingual and mixed-language data are used to train the model.

We introduce MixLID, a token-level language identification model supporting 111 language–script combinations. The model first learns to distinguish individual languages using the language label

*Corresponding authors

of each monolingual example as supervision for its tokens. Training then continues on inputs formed by concatenating code-switched examples and inputs augmented through word substitutions based on English source words and their corresponding loanwords, to identify the language of individual tokens in mixed-language contexts. For monolingual identification, MixLID outperforms each baseline in accuracy and macro F1 on all four benchmarks, with each comparison restricted to the language–script combinations supported by both models. For mixed-language identification, it outperforms the baselines on every evaluated language pair. Analysis of the training stages shows that subsequent mixed-language training improves the identification of individual tokens while largely preserving performance on monolingual inputs.

2 MixLID

MixLID is trained in two stages to identify the language of each token in both monolingual and multilingual inputs. It first learns to distinguish languages from monolingual examples in a multilingual corpus, then continues training on existing and synthetic examples containing multiple languages.

2.1 Training Data Construction

Monolingual Data We use monolingual examples with sentence-level language labels to construct token-level training data. Specifically, we use OpenLID-v3 (Fedorova et al., 2026) data covering 111 language–script combinations and assign each sentence’s language label to its constituent words. However, the language label of a sentence may not be appropriate for all expressions it contains. We therefore assign separate labels to numbers, punctuation, and symbols. Abbreviations found in the lists¹ receive a language label that covers the entire expression, including periods. Abbreviations in the English list receive the English label regardless of the source language, whereas those found only in the source-language list retain the source-language label. We further apply a heuristic filter based on English letter patterns to examples in non-Latin scripts to reduce incorrect assignment of the source-language label to expressions from other languages, while retaining these abbreviations.

Mixed-Language Data To identify the language of each token in mixed-language contexts, we

use both code-switched data and data augmented through word substitution. We concatenate examples from the ACS Corpus into longer training sequences while preserving their token-level language labels. For augmentation, we apply word substitutions to sentences and documents from Smol (Caswell et al., 2025) and passages from Bebele (Bandarkar et al., 2024), retaining each augmented text as a separate training example. We construct the substitution dictionary from Wiktionary etymological data extracted with Wikitext (Ylonen, 2022), retaining direct borrowing relationships between English source words and loanwords in other languages. We apply these dictionary mappings to each text independently, replacing English source words with corresponding loanwords or vice versa. Each replacement is assigned the label of the selected target language, while the remaining expressions are assigned the language label of the original text by default. All data and substitution candidates are restricted to supported languages. We retain the monolingual labeling rules for abbreviations, numbers, punctuation, and symbols, but omit the exclusion rules based on Latin character patterns to preserve mixed-language inputs. Detailed preprocessing and labeling rules for both data types are provided in Appendix B.

2.2 Learning Objective

We align the labels in the constructed data with subword tokens using character offsets in the input text. When a word is split into multiple subwords, all resulting subwords are assigned the same language label as the original word. A multilingual encoder produces a contextual representation for each subword, and a classification head predicts its label. Let $x = (x_1, \dots, x_T)$ denote the input subword sequence and $y_i \in \mathcal{Y}$ the label associated with each labeled subword x_i . The set $\mathcal{I}$ contains the positions of labeled subwords used for training. The label set $\mathcal{Y}$ includes the supported language labels and separate labels for numbers, punctuation, and symbols, and is shared across both training stages. Let $p_\theta(c \mid x, i)$ denote the predicted probability of label $c \in \mathcal{Y}$ at position i , where θ comprises the parameters of the encoder and classification head. We train the model to predict the corresponding token labels by minimizing the cross-entropy loss:

$$\mathcal{L}(\theta; x, y) = -\frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \log p_\theta(y_i \mid x, i). \quad (1)$$

¹We use the abbreviation lists provided by NLTK Punkt.

Benchmark	Metric	OpenLID-v3		GlotLID-v3		NLLB-LID		fastText lid.176		CLD3		CommonLingua		CONLID		Polyglot Tagger	
		Baseline	MixLID	Baseline	MixLID	Baseline	MixLID	Baseline	MixLID	Baseline	MixLID	Baseline	MixLID	Baseline	MixLID	Baseline	MixLID
FLORES+	#L	110		110		106		70		74		96		110		70	
	Acc	99.19	99.77	99.38	99.77	98.97	99.75	95.59	99.94	97.37	99.91	95.40	99.89	98.89	99.77	99.46	99.91
	Macro F1	99.42	99.76	99.57	99.76	99.15	99.75	95.47	99.94	97.31	99.91	96.18	99.89	99.32	99.76	99.58	99.92
WiLI-2018	#L	82		82		80		70		70		81		82		67	
	Acc	97.58	98.43	97.72	98.41	97.52	98.37	95.88	98.91	95.76	98.59	97.05	98.41	97.64	98.41	98.58	98.84
	Macro F1	98.16	98.58	98.21	98.58	97.98	98.54	96.34	98.98	96.34	98.71	97.82	98.58	98.20	98.58	98.88	98.92
UDHR	#L	101		101		98		67		72		91		101		67	
	Acc	98.82	99.38	98.14	99.38	98.36	99.34	94.99	99.87	97.60	99.84	93.45	99.80	97.10	99.38	99.33	99.84
	Macro F1	98.75	99.32	99.14	99.32	98.40	99.29	95.30	99.86	97.23	99.84	94.63	99.81	98.36	99.32	99.54	99.84
CommonLID	#L	58		58		54		39		45		53		58		41	
	Acc	94.04	96.95	94.56	96.75	93.93	96.66	95.93	97.74	90.28	97.25	87.07	96.81	90.43	96.75	95.42	97.74
	Macro F1	81.85	85.79	83.47	85.80	83.28	86.28	84.96	90.17	75.72	88.51	78.30	86.50	78.56	85.80	85.07	88.51

Table 1: Sentence-level language identification performance (%) on the language-script intersection between MixLID and each baseline. #L denotes the intersection size. Bold indicates the higher score in each paired comparison.

Training	Dataset	Source	Augmentation	#L	#Samples
Mono	OpenLID-v3	Mono	—	111	123.4M
Mixed	ACS Corpus	CS	concatenation	10	509.8K
	SmolSent	Mono	Substitution	87	46.8K
	SmolDoc	Mono	Substitution	92	20.7K
	Belebele	Mono	Substitution	111	56.6K

Table 2: Training data composition and statistics.

The same objective is used in both stages, with training on mixed-language data continuing from the model trained on monolingual data.

3 Experiments

3.1 Setup

Training We use mmbert-small (Marone et al., 2025) as the backbone of MixLID with a 114-class token classification head for 111 language-script combinations and three auxiliary labels (NUM, PUNCT, and SYM). Table 2 summarizes the training data. Training proceeds in two stages, beginning with approximately 123.4M monolingual examples from OpenLID-v3. The resulting model is then further trained on approximately 634K examples constructed from the ACS Corpus, Smol, and Belebele for mixed-language identification. ACS examples are shuffled and concatenated in groups of up to four. For Smol and Belebele, loanword-based substitution is applied independently to each eligible expression with a probability of 70%. Detailed training settings for both stages and hardware specifications are provided in the Appendix B.

Evaluation For sentence-level evaluation, we compare MixLID with OpenLID-v3 (Fedorova et al., 2026), GlotLID-v3 (Kargaran et al., 2023), NLLB-LID (Costa-Jussà et al., 2022), fastText lid.176 (Joulin et al., 2017), CLD3², Common-

Lingua (PleIAs, 2026), CONLID (Foroutan et al., 2026), and Polyglot Tagger³ on four benchmarks: FLORES+ (Burchell et al., 2024b), WiLI-2018 (Thoma, 2018), UDHR (Kargaran et al., 2023)⁴, and CommonLID (Suarez et al., 2026). Each comparison is restricted to languages jointly supported by MixLID and the baseline. MixLID and Polyglot Tagger determine the input language by majority voting over token predictions. We report overall accuracy and macro F1, the mean of per-language F1 scores. For token-level evaluation, we compare against MaskLID (Kargaran et al., 2024), Polyglot Tagger, and CLD3 on ACS (Sterner and Teufel, 2025) and the LID subset of CodeMixBench (Yang and Chai, 2025). CLD3 provides language predictions over text spans, which we align with the reference tokens. We retain language combinations supported by four models and evaluate the same eligible tokens. Token accuracy and macro F1 are computed per example and averaged within each language combination.

3.2 Results

Monolingual LID at the Sentence Level Table 1 shows that MixLID outperforms all eight baselines in both accuracy and macro F1 across the four benchmarks. This advantage holds over both sentence-level identification models and Polyglot Tagger, which supports token-level identification. Although MixLID is trained to predict language labels for individual tokens, these results show that its predictions also support accurate identification of the predominant language of an input. The higher accuracy and macro F1 indicate an advantage in

³<https://huggingface.co/polyglot-tagger/language-identification>

⁴<https://huggingface.co/datasets/cis-lmu/udhr-lid>

²<https://github.com/google/cld3>

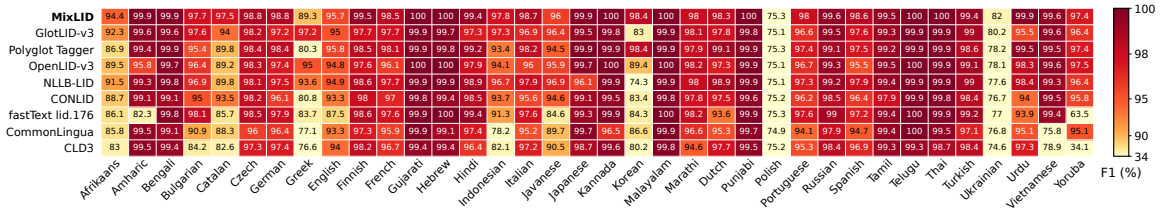


Figure 1: Language-wise F1 scores for the 36 languages shared by all models and all four sentence-level benchmarks. Each cell shows the average score across these datasets.

	Pair	CLD3	MaskLID	Polyglot	MixLID
ACS	da-en	56.1 / 32.6	38.7 / 20.7	62.5 / 37.9	83.3 / 73.5
	de-en	56.4 / 33.9	35.9 / 22.9	78.2 / 62.0	98.0 / 96.9
	es-en	48.2 / 28.2	22.9 / 15.3	74.6 / 56.9	96.1 / 94.1
	fr-en	54.8 / 32.1	24.4 / 15.5	71.0 / 47.7	93.1 / 90.4
	id-en	31.7 / 18.4	19.3 / 12.1	38.0 / 24.2	92.1 / 89.1
	it-en	56.3 / 33.3	29.6 / 19.3	74.3 / 55.7	95.6 / 93.6
	nl-en	52.2 / 30.9	35.8 / 21.9	66.1 / 45.1	92.6 / 90.0
	sv-en	56.6 / 32.9	41.7 / 24.9	71.7 / 48.7	92.3 / 87.1
	tr-de	58.1 / 35.1	53.1 / 37.2	86.6 / 80.5	93.9 / 81.5
	tr-en	44.6 / 25.8	31.2 / 18.9	69.9 / 45.6	91.6 / 87.4
CodeMixBench	zh-en	86.2 / 69.5	37.5 / 25.2	94.7 / 89.2	97.0 / 93.0
	Avg	54.6 / 33.9	33.6 / 21.3	71.6 / 53.9	93.2 / 88.8
	de-en	86.3 / 61.1	55.3 / 29.2	90.6 / 65.8	95.2 / 82.8
	es-en	55.9 / 47.3	25.8 / 17.6	82.3 / 71.2	92.8 / 88.7
	zh-en	80.5 / 46.0	61.6 / 31.0	83.9 / 55.3	95.5 / 90.3
	Avg	74.2 / 51.5	47.6 / 25.9	85.6 / 64.1	94.5 / 87.3

Table 3: Token-level LID results (Acc / Macro F1).

both overall prediction accuracy and performance averaged across languages. The results demonstrate consistent performance advantages across evaluation data from diverse sources, including web text.

Mixed-Language LID at the Token Level Table 3 shows that MixLID outperforms MaskLID, Polyglot Tagger, and CLD3 in both accuracy and macro F1 across all evaluated subsets of ACS and CodeMixBench. Averaged across language pairs, MixLID achieves an accuracy of 93.23% and a macro F1 of 88.78% on ACS, and 94.50% and 87.29%, respectively, on CodeMixBench. These results indicate that MixLID can distinguish expressions from different languages within the same context, beyond identifying the predominant language of an input. Its advantage across the evaluated language combinations suggests that this ability is not limited to a particular language pair.

4 Analysis

4.1 Language-wise Performance

To examine whether the overall performance advantage of MixLID extends to individual languages, we compare performance on the 36 languages shared by all models and all four benchmarks, as shown in Figure 1. MixLID achieves the highest F1 on 26 of these languages. Its mean F1 is 97.38%, exceeding the strongest baseline, GlotLID-v3, by

Benchmark	Stage 1		Stage 2		Stage 1+2	
	Acc	Macro F1	Acc	Macro F1	Acc	Macro F1
FLORES+	99.67	99.64	97.39	97.21	99.77	99.76
WiLI-2018	98.52	98.62	94.05	94.74	98.43	98.58
UDHR	98.69	98.34	96.32	95.86	99.38	99.32
CommonLID	93.64	81.42	90.91	74.67	96.95	85.79
ACS	53.75	34.31	87.32	80.78	93.23	88.78
CodeMixBench	77.36	54.41	78.55	69.65	94.50	87.29

Table 4: Effect of training stages on LID performance at the sentence and token levels across six benchmarks.

0.99 percentage points. It outperforms GlotLID-v3 and Polyglot Tagger on 33 and 31 languages, respectively, and achieves F1 scores above 95% on 32 of the 36 languages. These results indicate that the overall performance advantage of MixLID is reflected across a broad range of languages.

4.2 Effect of Training Stages

Table 4 compares training on monolingual data alone (Stage 1), mixed-language data alone (Stage 2), and both stages in sequence (Stage 1+2). Stage1 outperforms Stage2 on all four sentence-level benchmarks, whereas Stage2 performs better at token identification in mixed-language inputs. The model trained through both stages outperforms either individual stage on the mixed-language benchmarks ACS and CodeMixBench. It also surpasses Stage1 in sentence-level identification, except for a slight decrease on WiLI-2018. These results suggest that the two stages serve complementary roles, improving mixed-language identification while largely preserving monolingual performance.

5 Conclusion

We introduced MixLID, a token-level language identification model covering 111 language labels and designed to distinguish languages within mixed-language inputs. Trained on monolingual data followed by existing and augmented mixed-language data, MixLID outperformed the baselines in both monolingual and mixed-language identification. Our analysis showed that mixed-language training improved individual token identification

while largely preserving performance on monolingual inputs. We hope MixLID will serve as a useful resource for research and applications requiring language identification within multilingual text.

Limitations

The linguistic scope of this study is limited to the 111 language labels supported by MixLID and does not encompass the full range of linguistic diversity. In addition, dictionary-based substitution and automatic labeling may not fully account for contextual usage and linguistic ambiguity, potentially introducing unnatural expressions or labeling errors into the training data. Finally, mixed-language evaluation covers a limited set of code-switching language combinations, and generalization to a wider range of combinations requires further validation.

Ethical Considerations

This study uses publicly available datasets and lexical resources to develop a model for language identification in text. The loanword-based augmented data are constructed for training token-level language identification and do not represent the actual language use or code-switching practices of any particular linguistic community. To ensure transparency in data use and experimentation, we document the resource sources, data construction and preprocessing methods, and evaluation procedures.

References

- Ife Adebare, AbdelRahim Elmadany, Muhammad Abdul-Mageed, and Alcides Inciarte. 2022. Afrolid: A neural language identification tool for african languages. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1958–1981.
- Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabza. 2024. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 749–775.
- Laurie Burchell, Alexandra Birch, Nikolay Bogoychev, and Kenneth Heafield. 2023. An open dataset and model for language identification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 865–879.
- Laurie Burchell, Alexandra Birch, Robert Thompson, and Kenneth Heafield. 2024a. Code-switched language identification is harder than you think. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 646–658.
- Laurie Burchell, Ona de Gibert, Nikolay Arefyev, Mikko Aulamo, Marta Bañón, Pinzhen Chen, Mariia Fedorova, Liane Guillou, Barry Haddow, Jan Hajic, et al. 2025. An expanded massive multilingual dataset for high-performance language technologies (hplt). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17452–17485.
- Laurie Burchell, Jean Maillard, Antonios Anastasopoulos, Christian Federmann, Philipp Koehn, and Skyler Wang. 2024b. Findings of the wmt 2024 shared task of the open language data initiative. In *Proceedings of the Ninth Conference on Machine Translation*, pages 110–117.
- Isaac Caswell, Elizabeth Nielsen, Jiaming Luo, Colin Cherry, Geza Kovacs, Hadar Shemtov, Partha Talukdar, Dinesh Tewari, Moussa Doumbouya, Djibrila Diane, et al. 2025. Smol: Professionally translated parallel data for 115 under-represented languages. In *Proceedings of the Tenth Conference on Machine Translation*, pages 1103–1123.
- Gokul Chittaranjan, Yogarshi Vyas, Kalika Bali, and Monojit Choudhury. 2014. Word-level language identification using crf: Code-switching shared task report of msr india system. In *Proceedings of the first workshop on computational approaches to code switching*, pages 73–79.
- Shamil Chollampatt, Minh Quang Pham, Sathish Reddy Indurthi, and Marco Turchi. 2025. Cross-lingual evaluation of multilingual text generation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7766–7777.
- Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.
- Heba Elfardy and Mona Diab. 2012. Token level identification of linguistic code switching. In *Proceedings of COLING 2012: posters*, pages 287–296.
- Mariia Fedorova, Nikolay Arefyev, Maja Buljan, Jindřich Helcl, Stephan Oepen, Egil Rønningstad, and Yves Scherrer. 2026. Openlid-v3: Improving the precision of closely related language identification—an experience report. In *Proceedings of the 13th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 275–292.
- Negar Foroutan, Jakhongir Saydaliev, Grace Kim, and Antoine Bosselut. 2026. Conlid: Supervised contrastive learning for low-resource language identification. In *Proceedings of the 19th Conference of the*

- European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6693–6708.
- Javier Iranzo-Sánchez, Parnia Bahar, Alejandro Pérez-González-de Martos, and Mattia Antonino Di Gangi. 2025. Word-level language identification using encoder-only and decoder-only models. In *Proceedings of the 8th International Conference on Natural Language and Speech Processing (ICNLSP-2025)*, pages 82–86.
- Tommi Jauregi, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2019. Automatic language identification in texts: A survey. *Journal of Artificial Intelligence Research*, 65:675–782.
- Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomáš Mikolov. 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*, pages 427–431.
- Amir Hossein Kargaran, Ayyoob Imani, François Yvon, and Hinrich Schuetze. 2023. Glotlid: Language identification for low-resource languages. *arXiv preprint arXiv:2310.16248*.
- Amir Hossein Kargaran, François Yvon, and Hinrich Schütze. 2024. Masklid: Code-switching language identification through iterative masking. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 459–469.
- Ben King and Steven Abney. 2013. Labeling the languages of words in mixed-language documents using weakly supervised methods. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1110–1119.
- Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan Van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, et al. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.
- Yifeng Liu, Siqi Ouyang, Lei Li, et al. 2026. Mending the holes: Mitigating reward hacking in reinforcement learning for multilingual translation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 33699–33723.
- Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. 2025. *mmbert: A modern multilingual encoder with annealed language learning*. Preprint, arXiv:2509.06888.
- Dong Nguyen and A Seza Doğruöz. 2013. Word level language identification in online multilingual communication. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 857–862.
- Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A Rossi, and Thien Huu Nguyen. 2024. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4226–4237.
- Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. 2025. Fineweb2: One pipeline to scale them all—adapting pre-training data processing to every language. *arXiv preprint arXiv:2506.20920*.
- PleIAs. 2026. *Commonlingua: Byte-level language identification for 334 languages*.
- Younes Samih, Suraj Maharjan, Mohammed Attia, Laura Kallmeyer, and Thamar Solorio. 2016. Multilingual code-switching identification via lstm recurrent neural networks. In *Proceedings of the second workshop on computational approaches to code switching*, pages 50–59.
- Edoardo Signoroni, Emma Bednářková, and Pavel Rychlý. 2026. Rebelöt: Datasets and token-level language identification for lombard-italian-english code-mixing. In *SIGUL 2026 Joint Workshop with ELE, EURALI, and DCLRL: Towards Inclusivity and Equality: Language Resources and Technologies for Under-Resourced and Endangered Languages@ LREC 2026*, page 253.
- Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, et al. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the first workshop on computational approaches to code switching*, pages 62–72.
- Igor Sterner. 2024. Multilingual identification of english code-switching. In *Proceedings of the Eleventh Workshop on NLP for Similar Languages, Varieties, and Dialects (VarDial 2024)*, pages 163–173.
- Igor Sterner and Simone Teufel. 2023. Tongueswitcher: Fine-grained identification of german-english code-switching. In *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, pages 1–13.
- Igor Sterner and Simone Teufel. 2025. Minimal pair-based evaluation of code-switching. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18575–18598.
- Pedro Ortiz Suarez, Laurie Burchell, Catherine Arnett, Rafael Mosquera, Sara Hincapié Monsalve, Thom

- Vaughan, Damian Stewart, Malte Ostendorff, Idris Abdulmumin, Vukosi Marivate, et al. 2026. Commonlid: Re-evaluating state-of-the-art language identification performance on web data. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33063–33080.
- Martin Thoma. 2018. The wili benchmark dataset for written language identification. *arXiv preprint arXiv:1801.07779*.
- Rob Van Der Goot. 2025. Identifying open challenges in language identification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18207–18227.
- Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the twelfth language resources and evaluation conference*, pages 4003–4012.
- Yilun Yang and Yekun Chai. 2025. Codemixbench: Evaluating code-mixing capabilities of llms across 18 languages. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2139–2169.
- Tatu Ylonen. 2022. Wiktextextract: Wiktionary as machine-readable structured data. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1317–1325.
- Xue Zhang, Yunlong Liang, Fandong Meng, Songming Zhang, Kaiyu Huang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2026. Think natively: Unlocking multilingual reasoning with consistency-enhanced reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20766–20783.
- Yuan Zhang, Jason Riesa, Dan Gillick, Anton Bakalov, Jason Baldridge, and David Weiss. 2018. A fast, compact, accurate model for language identification of codemixed text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 328–337.

A Related Work

Research on language identification at the sentence and document levels has explored statistical classification using character and word features as well as neural approaches (Jauhiainen et al., 2019). More recent work has expanded language coverage through the collection, curation, and release of training data, addressing diverse scripts and low-resource languages (Burchell et al., 2023; Kargaran et al., 2023; Adebara et al., 2022). Efforts

to better distinguish closely related languages and handle nonlinguistic noise have included enriching training data, merging selected language varieties, and treating noise as a separate class (Fedorova et al., 2026). Evaluations on web data have shown that high scores on established benchmarks do not necessarily translate into comparable performance in domains containing noise and diverse language varieties (Suarez et al., 2026). A systematic comparison of factors including input length, training data size, and domain has likewise identified cross-domain generalization as a major challenge (Van Der Goot, 2025). To address domain differences, supervised contrastive learning has been used to learn language representations and improve identification of low-resource languages on out-of-domain data (Foroutan et al., 2026).

Word-level identification in mixed-language text has been formulated as sequence labeling that accounts for context, with weak supervision derived from monolingual data (King and Abney, 2013). Synthetic mixed-language data have also been used to train token classifiers supporting multiple languages (Zhang et al., 2018). Research has explored identification using lexicons, morphological analysis, and contextual embeddings, as well as comparisons between encoder-based classification and decoder-based generation (Sternier and Teufel, 2023; Iranzo-Sánchez et al., 2025). Other work has harmonized annotations across language-pair datasets to train token-level identification of English versus other languages and examined transfer to language pairs unseen during training (Sternier, 2024). Token identification models have also been fine-tuned on authentic code-switched data and synthetic examples constructed by inserting contiguous spans from another language into sentences (Signoroni et al., 2026). An openly available model similarly fine-tunes a pre-trained encoder to identify the language of each token using synthetic data formed by concatenating sentences in different languages. Other approaches predict the set of languages present in a mixed-language input through sentence-level multilabel classification, or iteratively mask language-specific features to detect additional languages with an existing identifier (Burchell et al., 2024a; Kargaran et al., 2024). Our work focuses on two-stage training from monolingual to mixed-language data, using both existing code-switched data and augmented data based on loanword correspondences for mixed-language training. This approach sup-

ports token-level language identification in mixed-language inputs across a broad range of languages, while aggregating the predictions also enables identification of the language of a monolingual input.

B Training Details

B.1 Language Inventory

Table 5 lists the languages supported by MixLID and the 111 language labels used for training. We add three labels, NUM, PUNCT, and SYM, for numbers, punctuation, and symbols, respectively, yielding 114 output classes. These labels are assigned to the corresponding expressions instead of a language label. The same label inventory is used in both training stages.

B.2 Shared Labeling and Token Alignment

We distinguish numbers, punctuation, and symbols using Unicode categories, while treating decimal points and commas separating groups of three digits as part of a numeric expression. For example, 3.14 and 1,000 each receive a single NUM label, whereas 3,14 is split into NUM, PUNCT, and NUM. Percent and per mille signs and their variants are labeled SYM regardless of their Unicode categories. Abbreviations matching the NLTK Punkt lists receive a language label over the entire expression, including periods. Matches in the English list take precedence and receive the English label; those found only in the list for the original language retain that language label.

Apostrophes within words receive the same language label as the word, whereas standalone quotation marks are labeled PUNCT. Ordinary spaces are left unlabeled, while newlines and tabs in mixed-language data are labeled PUNCT. The resulting labeled spans are aligned with subwords using character offsets. When a subword overlaps multiple spans, we assign the label of the span with the greatest character overlap, excluding ordinary spaces. Ties are resolved in favor of the earlier span.

B.3 Monolingual Data Preparation

To reduce the assignment of a sentence’s language label to expressions from other languages, we apply a filter based on English letter patterns to monolingual examples in non-Latin scripts. Except for expressions in the abbreviation lists, examples containing a sequence of three or more English letters are excluded. This filter is not applied to examples in Latin scripts or to mixed-language data.

B.4 Mixed-Language Data Preparation

Code-Switched Data Preparation To train token-level language identification on longer inputs containing multiple languages, we construct training sequences by concatenating existing code-switched examples. We use the ACS Corpus, which provides token-level language labels. Before concatenation, we remove anonymized user mentions and relabel tokens without a specific language label as numbers, punctuation, or symbols. Examples containing any remaining tokens that cannot be relabeled are excluded. We shuffle the cleaned examples and concatenate up to four with a single space between them, preserving their language labels and using each example once.

Loanword-Based Augmentation To construct code-switched data, we apply loanword-based substitution to datasets containing monolingual examples in multiple languages: sentences and documents from Smol and passages from Belebele. The substitution dictionary includes entries for borrowing, adapted borrowing, unadapted borrowing, learned borrowing, semi-learned borrowing, and orthographic borrowing. We exclude entries whose forms are identical to their English source words or differ only in case, as well as affix forms with a leading or trailing hyphen. For reverse substitution, we also exclude loanwords associated with multiple English source words. Within each input, we select the longest non-overlapping matches at word boundaries and independently substitute each match with a probability of 70%. When an English source word has candidates in multiple languages, we select a candidate language uniformly at random. We do not perform contextual word sense disambiguation or infer inflected forms, and we retain examples even when no substitution is applied.

B.5 Training Configuration

Table 6 reports the hyperparameters for each training stage. The second stage is initialized from the model trained in the first stage, with the optimizer and learning rate scheduler reinitialized. Both stages update all model parameters using four NVIDIA RTX A6000 GPUs.

C Evaluation Details

C.1 Baseline Models

The baselines include models that predict a single language for an entire input and models that

Language	Label	Language	Label	Language	Label	Language	Label
Afrikaans	afr_Latn	Hausa	hau_Latn	Marathi	mar_Deva	Spanish	spa_Latn
Amharic	amh_Ethi	Hebrew	heb_Hebr	Modern Greek	ell_Grek	Standard Estonian	ekk_Latn
Arabic	ara_Arab	Hindi	hin_Deva	Nepali	npi_Deva	Standard Latvian	lvs_Latn
Armenian	hye_Armn	Hungarian	hun_Latn	Nigerian Fulfulde	fuv_Latn	Standard Malay	zsm_Latn
Assamese	asm_Beng	Icelandic	isl_Latn	North Azerbaijani	azj_Latn	Sundanese	sun_Latn
Bambara	bam_Latn	Igbo	ibo_Latn	Northern Uzbek	uzn_Latn	Swahili	swh_Latn
Basque	eus_Latn	Iloko	ilo_Latn	Norwegian Bokmål	nob_Latn	Swati	ssw_Latn
Bengali	ben_Beng	Indonesian	ind_Latn	Nyanja	nya_Latn	Swedish	swe_Latn
Bulgarian	bul_Cyrl	Italian	ita_Latn	Odia	ory_Orya	Tajik	tgk_Cyrl
Burmese	mya_Mymr	Japanese	jpn_Jpan	Panjabi	pan_Guru	Tamil	tam_Taml
Catalan	cat_Latn	Javanese	jav_Latn	Paraguayan Guaraní	gug_Latn	Telugu	tel_Telu
Cebuano	ceb_Latn	Kabuverdianu	kea_Latn	Pedi	nso_Latn	Thai	tha_Thai
Central Khmer	khm_Khmr	Kachin	kac_Latn	Persian	fas_Arab	Tibetan	bod_Tibt
Central Kurdish	ckb_Arab	Kannada	kan_Knda	Plateau Malagasy	plt_Latn	Tigrinya	tir_Ethi
Croatian	hrv_Latn	Kazakh	kaz_Cyrl	Polish	pol_Latn	Tosk Albanian	als_Latn
Czech	ces_Latn	Kinyarwanda	kin_Latn	Portuguese	por_Latn	Tsonga	tso_Latn
Danish	dan_Latn	Kirghiz	kir_Cyrl	Romanian	ron_Latn	Tswana	tsn_Latn
Dutch	nld_Latn	Korean	kor_Hang	Russian	rus_Cyrl	Turkish	tur_Latn
English	eng_Latn	Lao	lao_Lao	Serbian	srp_Cyrl	Ukrainian	ukr_Cyrl
Filipino	fil_Latn	Lingala	lin_Latn	Shan	shn_Mymr	Urdu	urd_Arab
Finnish	fin_Latn	Lithuanian	lit_Latn	Shona	sna_Latn	Vietnamese	vie_Latn
French	fra_Latn	Luo	luo_Latn	Sindhi	snd_Arab	Waray	war_Latn
Ganda	lug_Latn	Macedonian	mkd_Cyrl	Sinhala	sin_Sinh	West Central Oromo	gaz_Latn
Georgian	kat_Geor	Malayalam	mal_Mlym	Slovak	slk_Latn	Wolof	wol_Latn
German	deu_Latn	Maltese	mlt_Latn	Slovenian	slv_Latn	Xhosa	xho_Latn
Gujarati	guj_Gujr	Mandarin Chinese	cmn_Hans	Somali	som_Latn	Yoruba	yor_Latn
Haitian	hat_Latn	Mandarin Chinese	cmn_Hant	Southern Pashto	pbt_Arab	Zulu	zul_Latn
Halh Mongolian	khk_Cyrl	Maori	mri_Latn	Southern Sotho	sot_Latn		

Table 5: Supported languages and their training labels.

Setting	Monolingual	Mixed
Optimizer	AdamW	AdamW
LR schedule	Linear	Linear
Maximum length	8,192	8,192
Learning rate	5×10^{-5}	1×10^{-5}
Epochs	1	1
Batch size per GPU	512	128
Gradient accumulation	2	1
Effective batch size	4,096	512
Warmup ratio	0.05	0.05
Weight decay	0	0
Precision	BF16	BF16
Seed	42	42

Table 6: Training hyperparameters for both stages.

provide token-level or span-level predictions. The coverage figures below refer to the label inventories of the models used in our experiments and are distinct from the numbers of shared evaluation languages reported in the main tables.

- **OpenLID-v3** A fastText-based model trained on monolingual data from multiple sources, with a focus on improving identification precision for closely related languages. It provides 194 language labels and a non-language label.
- **GlottLID-v3** A fastText-based model designed to extend language identification coverage to low-resource languages. It provides 2,102 labels distinguishing languages and scripts.
- **NLLB-LID** A fastText-based language identification model used for data collection and filtering

in multilingual machine translation. It predicts the language of an entire input, and the model used in our experiments provides 218 output labels.

- **fastText lid.176** A general-purpose language identification model trained on Wikipedia, Tatoeba, and SETimes. It supports 176 languages and predicts the language of an input text.
- **CommonLingua** A lightweight byte-level model designed for language identification in large corpora. Its training data include diverse documents and low-resource languages, and it provides 334 language labels.
- **CONLID** A language identification model that uses contrastive learning and supports a broad range of languages. The model used in our experiments provides 2,099 output labels.
- **CLD3** A neural language identification model using character n -gram features, with 109 language and script labels. It provides predictions for both entire inputs and text spans and is therefore included in both evaluations.
- **MaskLID** A method that applies iterative masking to an existing language identification model to identify expressions in different languages within mixed-language inputs, without additional training. We use GlottLID-v3, with its 2,102 labels, as the underlying model.

Type	Name	URL
Model	OpenLID-v3	https://huggingface.co/HPLT/OpenLID-v3
	GlottID-v3	https://huggingface.co/cis-lmu/glottid
	NLLB-LID	https://huggingface.co/facebook/fasttext-language-identification
	fastText lid.176	https://fasttext.cc/docs/en/language-identification.html
	CommonLingua	https://huggingface.co/PlEIA/CLM/CommonLingua
	CONLID	https://huggingface.co/epfl-nlp/ConLID
	CLD3	https://github.com/google/cld3
	MaskLID	https://github.com/cisnlp/MaskLID
Benchmark	Polyglot Tagger	https://huggingface.co/polyglot-tagger/language-identification
	FLORES+	https://huggingface.co/datasets/openlanguage/flores_plus
	WiLI-2018	https://doi.org/10.5281/zenodo.841984
	UDHR	https://huggingface.co/datasets/cis-lmu/udhr-lid
	CommonLID	https://huggingface.co/datasets/commoncrawl/CommonLID
	ACS Benchmark	https://huggingface.co/datasets/igorsterner/acs-benchmark
	CodeMixBench	https://huggingface.co/datasets/CodeMixBench/CodeMixBench

Table 7: Sources of baseline models and evaluation benchmarks.

- **Polyglot Tagger** An XLM-R-based token classification model trained on synthetic data constructed by combining sentences in different languages. The checkpoint used in our experiments supports 105 languages. We use its token predictions directly for token-level evaluation and aggregate them for sentence-level evaluation.

C.2 Evaluation Benchmarks

Monolingual Benchmarks We use four benchmarks to evaluate the prediction of a single language for each input. FLORES+ contains parallel texts translated into multiple languages; we use the devtest split of v4.5. WiLI-2018 consists of paragraphs collected from Wikipedia, for which we use the official test split. UDHR is based on multilingual translations of the Universal Declaration of Human Rights; we use the cleaned version released with the GlottID study. CommonLID contains web text with human-annotated language labels. Each sentence, paragraph, or line is treated as one input, and each comparison is restricted to the languages jointly supported by the models.

Mixed-Language Benchmarks We use the ACS Benchmark and the LID subset of CodeMixBench to evaluate token-level language identification in mixed-language inputs. ACS provides observed code-switched sentences and manipulated variants. We evaluate both types of input across 11 language pairs in its test split, using the provided automatic language labels. For CodeMixBench, we evaluate German–English, Spanish–English, and Chinese–English, which are supported by all models in the comparison.

C.3 Evaluation Protocol and Metrics

Language Matching We map language codes to a common representation to reconcile differences in model label conventions. Sentence-level comparisons include only examples in languages supported by the benchmark and both models. Arabic and Persian varieties are mapped to the corresponding macrolanguage labels when these are provided by a model; support for one variety is not taken to imply support for other varieties. Script constraints are also applied when specified by the model. We use the script metadata provided by FLORES+ and UDHR, and infer the predominant script from the distribution of Unicode scripts in the input for WiLI-2018 and CommonLID. For fastText lid.176, CommonLingua, and Polyglot Tagger, we apply language-level coverage and the script constraints of the comparison model. Token-level evaluation includes only subsets for which MixLID, MaskLID, Polyglot Tagger, and CLD3 all support both languages in the pair.

Prediction Alignment For sentence-level evaluation, MixLID and Polyglot Tagger determine the input language by majority voting over subword language predictions, excluding non-language labels. Ties are resolved by lexicographic label order. For token-level evaluation, we use character offsets in the original text to align model output spans with the reference tokens. For each reference token, we sum the character overlap with predicted spans by language and select the language with the largest total overlap. CLD3 span offsets are converted from bytes to characters, and tokens with no corresponding prediction are treated as undetermined. We exclude tokens with non-language, mixed, or ambiguous reference labels, as well as

tokens consisting entirely of whitespace, numbers, punctuation, or symbols, consistently across models. Undetermined predictions and predictions outside the language pair are counted as errors for eligible tokens.

Sentence-Level Metrics We report accuracy and macro F1 to measure overall identification accuracy and performance across languages. Accuracy is the proportion of evaluation examples for which the language is correctly predicted.

$$\text{Acc} = \frac{\# \text{ correct samples}}{\# \text{ samples}} \quad (2)$$

For each language ℓ , F1 is computed from its true positives, false positives, and false negatives. Let $\mathcal{L}$ denote the set of shared languages included in the evaluation.

$$F_{1,\ell} = \frac{2 \text{ TP}_\ell}{2 \text{ TP}_\ell + \text{FP}_\ell + \text{FN}_\ell} \quad (3)$$

$$\text{MacroF1} = \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} F_{1,\ell} \quad (4)$$

Thus, accuracy measures the proportion of correct predictions across all evaluation examples, whereas macro F1 is the arithmetic mean of the F1 scores for individual languages.

Token-Level Metrics For mixed-language inputs, we first compute accuracy and macro F1 over the eligible tokens in each example, then average the scores across examples within each language pair. Examples with no eligible tokens are excluded. Let A_i and F_i denote the accuracy and macro F1 for example i , respectively, and let $\mathcal{L}_i$ be the union of its reference and predicted labels over eligible tokens.

$$A_i = \frac{\# \text{ correct tokens}}{\# \text{ tokens}} \quad (5)$$

$$F_i = \frac{1}{|\mathcal{L}_i|} \sum_{\ell \in \mathcal{L}_i} F_{1,\ell}^{(i)} \quad (6)$$

Here, $F_{1,\ell}^{(i)}$ is computed using only the eligible tokens in example i . Avg in the main tables is the arithmetic mean of the language-pair scores within each benchmark. All scores are reported as percentages.