ConLID: Supervised Contrastive Learning for Low-Resource Language Identification

Negar Foroutan[1]*, **Jakhongir Saydaliev**[1]*, **Grace Kim**[2], **Antoine Bosselut**[1]
[1]EPFL [2]The University of Texas at Austin
Correspondence: {negar.foroutan, antoine.bosselut}@epfl.ch

Abstract

Language identification (LID) is a critical step in curating multilingual LLM pretraining corpora from web crawls. While many studies on LID model training focus on collecting diverse training data to improve performance, low-resource languages – often limited to single-domain data, such as the Bible – continue to perform poorly. To resolve these imbalance and bias issues, we propose a novel supervised contrastive learning (SCL) approach to learn domain-invariant representations for low-resource languages. We show that our approach improves LID performance on out-of-domain data for low-resource languages by 3.2 percentage points, while maintaining its performance for the high-resource languages.[1]

1 Introduction

Language identification (LID) is a fundamental preprocessing task in natural language processing (NLP). In recent years, LID has been a crucial step in the creation of large-scale textual pretraining corpora. LID models are used to label heterogeneous, multilingual document sources in web crawls. Subsequently, these labels are used to either filter non-target language content (Rae et al., 2022; Dubey et al., 2024; Li et al., 2024) or leverage language identification for effective data scheduling in multilingual pretraining (Conneau et al., 2019; de Gibert et al., 2024; Imani et al., 2023; Laurençon et al., 2023; Nguyen et al., 2023).

The application of these LID systems to web crawls requires scalable methods that can be efficiently applied to large document collections (Penedo et al., 2025). Consequently, current approaches to LID (Kargaran et al., 2023; Team et al., 2022) predominantly use simple models (Joulin et al., 2017) trained with cross-entropy (CE) loss. While these supervised approaches have been sufficient to practically solve LID for high-resource standard languages (McNamee, 2005), low-resource languages and dialects remain difficult to identify and categorize accurately for two reasons. First, data in these languages is often scarce or even mislabeled (Roy et al., 2022; Yu et al., 2022; Joshi et al., 2020a), leading to class imbalances during training (Kargaran et al., 2023). Second, available data is often concentrated in specific domains, such as religious texts (*i.e.*, Bible translations; Agić and Vulić, 2019; Imani et al., 2023; Ebrahimi and Kann, 2021). This domain entanglement yields models trained on narrow datasets that often fail to generalize effectively to diverse types of text.

To mitigate these limitations, we propose a novel

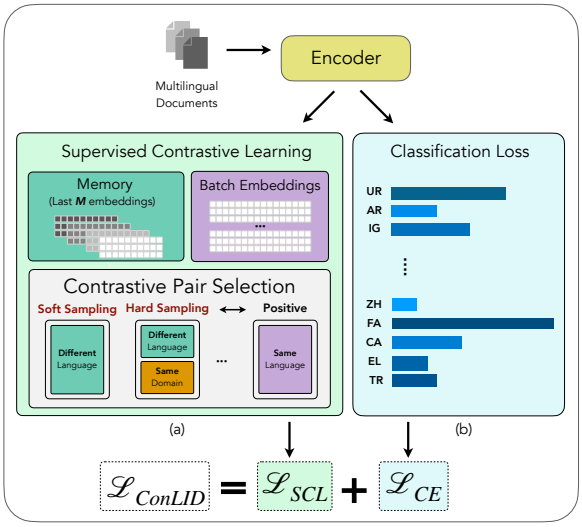

Figure 1: Overview of training ConLID. Each sentence is first processed by an encoder that generates a sentence representation. This representation is then passed to two components: (a) a Supervised Contrastive Learning (SCL) module, and (b) a feed-forward neural network that serves as the classification head. The final loss used for training is a combination of the classification loss and the SCL loss.

*Equal contribution
[1]Our code and model are available at https://github.com/epfl-nlp/ConLID.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6693–6708
March 24-29, 2026

approach that employs supervised contrastive learning (SCL; Khosla et al., 2020) for LID. Our SCL objective explicitly encourages representations of texts from the same language to cluster together while pushing those from different languages further apart in the embedding space. Combined with a classic CE loss, our dual-objective framework promotes learning less biased language representations that are robust to in-language domain shifts, enabling better generalization across diverse text types. A critical component of our SCL training method is a *memory bank* (Tan et al., 2022) from which we scalably sample more diverse positive and negative examples, effectively increasing our training batch size while keeping memory requirements constant. This augmentation enhances our contrastive learning approach, yielding stable training and discriminative language representations. To mitigate the second challenge, we employ a novel *hard negative* mining scheme for SCL training, where data sources in our training set are labeled with domain information and negative samples are drawn from different languages in the same domain to learn language-specific, domain-invariant representations during training.

We evaluate our approach on three benchmark datasets: `GlotLID-C` (Kargaran et al., 2023), FLORES-200 (Team et al., 2022), and UDHR,[2] and demonstrate that our approach leads to improvements over state-of-the-art LID methods, particularly in out-of-domain evaluations. Specifically, our best SCL-based approaches achieve a 3.2 percentage points improvement over CE-based models for low-resource languages, and a 5.4 percentage points improvement for languages with data from diverse domains, indicating enhanced domain generalization.

Moreover, we conduct an in-depth analysis of underperforming languages in out-of-domain evaluations and observe that misclassifications predominantly occur among linguistically related languages, highlighting potential challenges in LID for closely related language pairs. Finally, we assess our approach in a real-world scenario by evaluating it on FineWeb-2 (Penedo et al., 2025), a large-scale multilingual pretraining corpus. We compare its performance against state-of-the-art LID systems for low-resource languages, providing insights into the practical significance of LID systems.

[2] https://huggingface.co/datasets/cis-lmu/udhr-lid

Our key contributions are as follows: We introduce the first application of supervised contrastive learning (SCL) for domain generalization in LID. We provide a comprehensive analysis of misclassified languages in out-of-domain settings, shedding light on their linguistic and data-related characteristics. These contributions advance the robustness of language identification, particularly for low-resource and domain-diverse scenarios, paving the way for more reliable multilingual NLP systems.

2 Related Work

2.1 Language Identification

LID tools are often used to collect multilingual corpora (Dunn, 2020; Penedo et al., 2025; Kargaran et al., 2023), and there have been extensive work on developing LID tools focusing on covering as diverse languages as possible (Kargaran et al., 2023; Adebara et al., 2022; Dunn and Edwards-Brown, 2024; Burchell et al., 2023; Team et al., 2022). While many of LID models' architectures are based on FastText (Kargaran et al., 2023; Dunn and Edwards-Brown, 2024; Burchell et al., 2023; Team et al., 2022), which fails to learn domain-invariant representation of languages, some models are based on Transformer (Adebara et al., 2022; Caswell et al., 2020; Sindane and Marivate, 2024), which is less efficient for real-world scenarios.

2.2 Contrastive Learning

Contrastive Learning (CL) has been widely used in Computer Vision for learning generalizable representations in a self-supervised setting, without relying on labels (Chen et al., 2020). Supervised Contrastive Learning (SCL) extends this idea to leverage labeled data, grouping same-class examples while separating different-class ones (Khosla et al., 2020). It is shown that the performance of SCL is largely affected by the batch size, as the contrasting examples are selected within the batch to compute the contrastive loss (Khosla et al., 2020; Gunel et al., 2021; Tan et al., 2022).

In NLP, while the previous works have applied SCL for in-domain (Gunel et al., 2021) and out-of-domain (Tan et al., 2022; Khondaker et al., 2023) sentence-level classification tasks, those works (1) focus on the cases with only a few class labels (typically, less than 10), which allows for a good use of SCL, and (2) using Transformer-based models. To the best of our knowledge, we are the first to apply SCL to the LID task with significantly larger label

classes (~2,000) and using a simple linear classifier instead of a Transformer-based model.

3 Methodology

In the following, we provide a detailed description of our base model architecture (§3.1), the principles of supervised contrastive learning (SCL; §3.2), our positive/negative pair selection approach (§3.3), and an optimized training strategy for the SCL objective, particularly suited for scenarios involving a large number of classes (§3.4).

3.1 Base Classifier

We train the base model following the methodology employed by FastText (Joulin et al., 2017). Given an input sentence, each word in the sentence is decomposed into character-level n-grams. The sentence representation is, then, obtained by computing the average of the vectors corresponding to these n-grams, along with the word-level embedding. The training process proceeds as follows:

Initially, the training dataset is partitioned into mini-batches. For a sampled mini-batch of size B, we define $S = \{x_i, y_i\}_{i=1}^{B}$, where x_i represents the input text, and y_i denotes its corresponding language class. Each input text x_i is represented by an embedding, $\mathbf{z}_i$, using an encoder function $\mathbf{z}_i = E(x_i)$, where E corresponds to the encoder. The resulting embeddings $\mathbf{z}_i$ are subsequently passed through a feed-forward neural network f, which serves as the classification head. The model is trained using the cross-entropy (CE) loss, formulated as:

$$\mathcal{L}_{CE} = -\frac{1}{B}\sum_{i=1}^{B} y_i \log\big(f(\mathbf{z}_i)\big) \tag{1}$$

3.2 Supervised Contrastive Learning

The Supervised Contrastive Learning (SCL) objective minimizes the distance between representations of positive pairs (of the same language) while maximizing the separation between negative pairs (of different languages) to enhance the discriminative capability of learned hidden representations. Formally, given a mini-batch S of size B, the SCL objective is computed as follows:

$$\mathcal{L}_{SCL} = -\frac{1}{B}\sum_{i=1}^{B}\left(\log\Big(\sum_{\mathbf{z}_p\in P(i)} e^{\mathbf{z}_i\cdot\mathbf{z}_p/\tau}\Big) - \log\Big(\sum_{\mathbf{z}_p\in P(i)} e^{\mathbf{z}_i\cdot\mathbf{z}_p/\tau} + \sum_{\mathbf{z}_n\in N(i)} e^{\mathbf{z}_i\cdot\mathbf{z}_n/\tau}\Big)\right) \tag{2}$$

where $\mathbf{z}_i$ corresponds to the representation of a single data sample, $\mathbf{z}_p$ are positive samples drawn from a positive pair set $P(i) \equiv \{z_j \in S, y_j = y_i\}$ (*i.e.*, all samples in S that share the same label as $\mathbf{z}_i$), $\mathbf{z}_n$ are negative samples drawn from the negative pair set $N(i)$ (defined according to §3.3), and τ is the temperature. The final combined loss function is then:

$$\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{SCL} \tag{3}$$

3.3 Negative Pairs Selection

We select negative pairs for training (Eq. 2) using two strategies:

Soft selection. Given a batch $S = \{x_i, y_i\}_{i=1}^{B}$, we select the negative pairs as samples with different labels ($N(i) \equiv \{x_j \in S, y_j \neq y_i\}$).

Hard selection. Given a batch $S = \{x_i, y_i; s_i, d_i\}_{i=1}^{B}$, where s_i and d_i denote the script and domain of x_i, negative pairs are chosen as follows: for each sample x_i in the batch, we select negative pair samples with a different label y, the same script s, and the same domain d. If fewer than K such negatives exist, we relax the conditions in the following order: (2) different label y and same script s, (3) different label y and same domain d, (4) different label y only (*i.e.*, equivalent to Soft Selection). The algorithm of hard selection method is given in Appendix C.

3.4 Memory Bank

The performance of Supervised Contrastive Learning (SCL) is highly sensitive to batch size, as both positive and negative pairs are sampled within each mini-batch. So, large and diverse batches are required to increase the effectiveness of SCL where the ideal case is having all labels in each batch.

In most studies employing SCL, the number of classes is relatively small (typically fewer than 10), allowing for batch sizes in the range of 64 to 512. However, in our case, the number of classes is substantially larger (*i.e.*, 2,099), surpassing the feasible batch sizes that can be accommodated by contemporary GPUs. To mitigate this limitation, and following the work by Tan et al. (2022), we integrate a *memory bank*, that stores the last M data samples in memory in addition to the current batch (B), allowing each data sample in the batch to draw the negative and positive samples from $B+M$ samples. This increases the number of possible negative and positive samples making the contrastive loss more

representative. If the number of stored examples exceeds M, only the most recent M examples are retained, while older examples are discarded. Then, $\mathcal{L}_{SCL}$ is computed using $M + B$ examples.

4 Experimental Setup

We adopt the same text representation approach as FastText, utilizing character n-grams and word embeddings while following the same hyperparameter settings established by (Team et al., 2022).

4.1 Training

Dataset. We train our model using the `GlotLID-C` dataset (Kargaran et al., 2023) covering 2,099 languages. As no official splits were released by the authors, we partition the dataset into training (85%) and testing (15%) subsets. To address the imbalance between high- and low-resource languages, we down-sample high-resource languages in the training set to a maximum of 100k sentences per language. This down-sampling reduces the total number of training sentences from 306M to 62M, impacting 301 high-resource languages. We denote this down-sampled version of the training dataset as `GlotLID-C-train-down`. We report the detailed data distribution in Appendix A.

Models. We train the following LID models using `GlotLID-C-train-down`:
(a) **LID$_{\text{CE}}$** - The model trained only with the cross-entropy objective, as defined in Equation (1).
(b) **LID$_{\text{SCL}}$** - The model trained with the combined objective in Equation (3), without employing a memory bank.
(c) **ConLID-S** - The model trained with the combined objective in Equation (3), employing a memory bank with soft negative pair selection, as explained in §3.3.
(d) **ConLID-H** - The model trained with the combined objective in Equation (3), employing a memory bank with hard negative pair selection, as explained in §3.3.
We set the memory bank size to $M = 2048$ for both soft selection (M_S) and hard selection (M_H), with a minimum number of negatives $K = 1024$ for hard selection. In our experiments, we use a batch size of $B = 128$. The details of our hyperparameter tuning process are provided in Appendix B.2.

To further improve inference performance, we employ an ensemble approach that integrates the predictions of LID$_{\text{CE}}$ and ConLID-S by selecting the maximum their probabilities, denoted as **ConLID-S+LID$_{\text{CE}}$**. We report the ensembling method results for ConLID-S only as we did not observe significant differences when using ConLID-H.

Baselines. In our experiments, we implemented LID$_{\text{CE}}$ as a replication of GlotLID-M (Kargaran et al., 2023), as official data splits for GlotLID-M were not openly released (hindering a non-contaminated evaluation on `GlotLID-C-test`). However, to enable a full comparison with prior methods, we also report the performance of the open-weight GlotLID-M, the previous SOTA model on UDHR, and **GlotLID-M+ConLID-S**, our ensemble of GlotLID-M and ConLID-S by taking the sum of their predicted distributions. Technical details can be found in Appendix B.1.

We also compare ConLID-S against two other publicly available LID models: AfroLID (Adebara et al., 2022) and NLLB-LID (Team et al., 2022).

4.2 Evaluation

We evaluate our models on the `GlotLID-C-test` (Kargaran et al., 2023), FLORES-200 (Team et al., 2022), and UDHR (Universal Declaration of Human Rights) datasets.

`GlotLID-C-test`. This dataset is comprised of 15% of the examples in `GlotLID-C` (described above, §4.1), which contains 2,099 languages.

FLORES-200. This dataset comprises 842 articles sourced from English-language Wikimedia projects. Each sentence in the dataset has been translated into 204 unique language-script combinations, covering 196 distinct languages. These translations have been manually verified by human annotators to ensure quality and accuracy. Since the FLORES-200 development set is incorporated into `GlotLID-C`, and following prior work, we adopt the dev-test subset (containing 1,012 sentences) as our evaluation set for `FLORES-200`. Our analysis focuses on 199 languages that overlap between FLORES-200 and `GlotLID-C`.

UDHR. The original UDHR dataset contains over 500 translations of the Universal Declaration of Human Rights. Kargaran et al. (2023) curated and refined this dataset, reducing it to 369 languages. During our experiments, we identified discrepancies in seven languages (dip_Latn, pcd_Latn, aii_Syrc, guu_Latn, mto_Latn, chj_Latn,

Method	GlotLID-C test (2099 languages)		UDHR (360 languages)		FLORES-200 (199 languages)	
	F1↑	FPR↓	F1↑	FPR↓	F1↑	FPR↓
LID_{CE}	$.9858_{6e\text{-}4}$	$.0000068_{1.2e\text{-}7}$	$.8925_{19e\text{-}4}$	$.0002222_{34e\text{-}7}$	$.9650_{3e\text{-}4}$	$.0001142_{9.9e\text{-}7}$
LID_{SCL}	$.9833_{1e\text{-}4}$	$.0000075_{0.2e\text{-}7}$	$.8938_{19e\text{-}4}$	$\underline{.0002095_{18e\text{-}7}}$	$.9617_{3e\text{-}4}$	$.0001182_{25e\text{-}7}$
ConLID-S	$.9860_{2e\text{-}4}$	$.0000067_{0.3e\text{-}7}$	$\underline{.9012_{21e\text{-}4}}$	$.0002170_{28e\text{-}7}$	$.9652_{3e\text{-}4}$	$.0001177_{20e\text{-}7}$
ConLID-H	$\underline{.9862_{1e\text{-}4}}$	$\underline{.0000067_{0.2e\text{-}7}}$	$.8961_{20e\text{-}4}$	$.0002211_{34e\text{-}7}$	$.9651_{3e\text{-}4}$	$.0001174_{14e\text{-}7}$
$\text{ConLID-S+LID}_{\text{CE}}$	$\mathbf{.9868_{1e\text{-}4}}$	$\mathbf{.0000065_{0.4e\text{-}7}}$	$.8998_{15e\text{-}4}$	$.0002180_{33e\text{-}7}$	$.9655_{2e\text{-}4}$	$.0001171_{19e\text{-}7}$
GlotLID-M	-	-	.8919	.0002139	$\underline{.9696}$	$\underline{.0001071}$
GlotLID-M+ConLID-S	-	-	$\mathbf{.9060_{15e\text{-}4}}$	$\mathbf{.0002082_{33e\text{-}7}}$	$\mathbf{.9716_{2e\text{-}4}}$	$\mathbf{.0001025_{19e\text{-}7}}$

Table 1: Performance of LID models on `GlotLID-C-test`, UDHR, and FLORES-200 with 2,099, 360, and 199 languages respectively, when trained using `GlotLID-C-train-down` dataset. Results are averaged over five runs and each subscript indicates the standard deviation of five runs. The best performance among the 5 methods is in **bold**, while the second best performance is underlined. No results are reported for GlotLID-M and GlotLID-M+ConLID-S on `GlotLID-C-test` as they were trained on splits that include parts of `GlotLID-C-test` in the training data.

auc_Latn) due to character misalignments between `GlotLID-C` and UDHR, and two languages (taj_Deva, tzm_Tfng) that had only one example each. We exclude these nine languages, resulting in a final UDHR dataset comprising 360 languages. UDHR serves as an out-of-domain evaluation dataset and is not included in `GlotLID-C-train-down`.

FineWeb-2. This dataset is a multilingual pretraining corpus encompassing ∼1,800 languages (Penedo et al., 2025). The LID process in the FineWeb-2 pipeline is conducted using GlotLID-M (Kargaran et al., 2023), a model capable of recognizing 2102 languages. We use this dataset to evaluate our model in large-scale real-world scenarios, which provides insights into the practical significance of our approach for LID for web crawls.

Metrics. Following previous work (Team et al., 2022; Kargaran et al., 2023), we report the F1 score and the false positive rate (FPR) as evaluation metrics. The F1 score combines precision and recall, both critical metrics: precision reflects the accuracy of classification decisions, while recall ensures minimal data loss. FPR, defined as $\text{FPR} = \frac{\text{FP}}{\text{FP+TN}}$, where FP represents false positives and TN represents true negatives, evaluates the impact of false positives. This is particularly important in our scenario, where the negative class is significantly larger, amplifying the effects of even a low false positive rate.

5 Experimental Analysis

Table 1 presents the overall results for `GlotLID-C-test`, UDHR, and FLORES-200. We observe no significant improvement when directly applying the SCL objective (LID_{SCL}) compared to the cross-entropy baseline (LID_{CE}). However, when incorporating a memory bank to increase the number of contrastive negatives beyond the number of classes, we achieve performance gains across all three evaluation datasets. Notably, this improvement is most significant for UDHR (out-of-domain), indicating a better generalization of the contrastive learning approach.

For in-domain evaluation (`GlotLID-C-test` and FLORES-200), the highest F1 score among our methods is an ensemble approach that combines LID_{CE} and ConLID-S with a memory bank of M_H=2048 and K=1024, suggesting that LID_{CE} already performs well for certain languages (as we will show in §5.1), while $\text{ConLID-S+LID}_{\text{CE}}$ balances the performance across all labels.

When comparing to prior work, we observe that GlotLID-M tends to outperform our methods on FLORES-200, but underperforms them on UDHR, which is slightly more out of distribution. The ensembled method (GlotLID-M+ConLID-S), however, achieves the best score across UDHR and FLORES-200, indicating the methods are complementary and could be combined for optimal accuracy. No comparison is made on `GlotLID-C-test` as GlotLID-M was trained on different subsets of `GlotLID-C`, meaning the evaluation would be contaminated (same for GlotLID-M+ConLID-S).

Table 4 presents a performance comparison with AfroLID and NLLB-LID, evaluated on the subset of languages covered by all models. Results indicate that ConLID-S consistently surpasses both models across all three benchmark datasets.

Resource Level	$\|L\|$	LID_{CE}	$\Delta\text{LID}_{\text{SCL}}$	ΔConLID-S	ΔConLID-S+LID_{CE}
High-resource	330	0.9065	0	0.0066	0.0066
Low-resource	30	0.7380	0.0168	0.0323	0.0154

Table 2: F1 score change on UDHR of the LID models with SCL stratified by language resource level. LID_{CE} is used as a baseline, and subtracted from each model's score to compute the Δ score, *e.g.*, $\Delta\text{LID}_{\text{SCL}}$= LID_{SCL}-LID_{CE}. $|L|$ refers to the number of languages.

Train-set Domains	$\|L\|$	LID_{CE}	$\Delta\text{LID}_{\text{SCL}}$	ΔConLID-S	ΔConLID-S+LID_{CE}
Random	7	0.8157	0.0109	0.0541	0.0329
Bible	95	0.7821	-0.0077	0.0190	0.0149

Table 3: F1 score change of the LID models' with SCL by GlotLID-C train domains. LID_{CE} is used as a baseline, and subtracted from each model's score to compute the absolute Δ score, *e.g.* $\Delta\text{LID}_{\text{SCL}}$= LID_{SCL}-LID_{CE}. $|L|$ refers to the number of languages.

5.1 Performance on UDHR Dataset

We present a detailed analysis of the LID models on the UDHR dataset to systematically examine the specific aspects in which the SCL objective enhances performance and the conditions under which it may lead to performance degradation.

Resource Analysis. As the performance of a language model is strongly influenced by the amount of training data available, we categorize languages into high- and low-resource groups based on the number of training examples available in `GlotLID-C-train-down`. Languages with fewer than 10k examples are designated as **low-resource**, while those with 10k or more examples are categorized as **high-resource**.

For each group of languages, we compute the average F1 score difference between SCL-based models (LID_{SCL}, ConLID-S, ConLID-S+LID_{CE}) and LID_{CE}. The results are presented in Table 2. Our analysis indicates that the model ConLID-S yields the most notable improvements for low-resource languages, with F1 score gain of 3.23 percentage points, validating the importance of memory bank (§3.4). In contrast, the performance gain for high-resource languages is more modest, at 0.66 percentage points. We further extend this analysis in Appendix D.3, where we present detailed results based on the low- and high-resource categorization defined by Joshi et al. (2020b).

Domain Analysis. The `GlotLID-C` dataset comprises five distinct textual domains: **Bible** Translations, **Literature** (*i.e.*, stories & books), **Politics** (*i.e.*, government documents), **Multi-domain** (News and Wikipedia articles), and **Random** (machine translation datasets, chat logs, and web-scraped data). To evaluate the impact of SCL across different training data domains, we analyze languages trained exclusively on a single domain. While the distribution of `GlotLID-C-train-down` is shown in Appendix A, among the languages in UDHR, 7 languages have been exclusively trained on **Random**, while 95 on **Bible**. Therefore, we compute the F1 score difference between SCL-based models (LID_{SCL}, ConLID-S, ConLID-S+LID_{CE}) and LID_{CE} for random and bible domains and present the results in Table 3. Our results indicate that the inclusion of a memory bank leads to performance improvements across both domains. Notably, the **Random** datasets exhibit a more significant gain, with improvements of up to 5.41 percentage points. We hypothesize that this is due to the broader range of subdomains covered by **Random**, which facilitate learning more generalized language representations. In contrast, the **Bible** domain is limited to religious texts, making it less conducive to generalization.

Script Analysis. We conduct a similar analysis on the scripts of the languages and present the results in Table 5. While the UDHR dataset comprises 31 scripts, we report only those for which at least one of the metrics $\Delta\text{LID}_{\text{SCL}}$, ΔConLID-S, ΔConLID-S+LID_{CE} is nonzero. As observed, ConLID-S exhibits an improvement of up to 10.9 percentage points (on *Java*), while some high-resource scripts show non-significant improvement (*e.g.*, *Hani*, *Cyrillic*). We attribute this primarily to the resource level of the languages associated with each script, which aligns with our findings from §5.1, particularly regarding the behavior of low-resource languages. A breakdown analysis is given in the Appendix D.2.

Method	GlotLID-C test		UDHR		FLORES-200	
	F1↑	FPR↓	F1↑	FPR↓	F1↑	FPR↓
ConLID-S[a]	**.9860**	**.0000058**	**.9012**	**.0002170**	**.9652**	**.0001177**
AfroLID[a]	.8609	.0002883	.7605	.0012766	.7618	.0011196
ConLID-S[b]	**.9841**	**.0000058**	**.9639**	**.0002170**	**.9861**	**.0001177**
NLLB-LID[b]	.9335	.0002380	.9405	.0003649	.9614	.0002708

Table 4: Comparison of ConLID-S with AfroLID and NLLB-LID systems. [a]Evaluated on 398/82/44 languages for GlotLID-C test/UDHR/FLORES-200 respectively. [b]Evaluated on 207/162/182 languages for GlotLID-C test/UDHR/FLORES-200 respectively.

Script	LID_{CE}	$\Delta\text{LID}_{\text{SCL}}$	ΔConLID-S	$\Delta\text{ConLID-S+LID}_{\text{CE}}$
Hani	0.5678	-0.0265	0.0020	0.0011
Tifinagh	0.7158	0.0134	0.0134	0.0134
Arabic	0.8686	0.0277	0.0678	0.0325
Latin	0.8803	0.0012	0.0092	0.0079
Java	0.8829	0.0923	0.1090	0.1007
Tibetan	0.9411	0.0253	0.0084	0.0084
Cyrillic	0.9731	0.0053	0.0011	0.0030

Table 5: F1 score change of the LID models' with SCL by language script. LID_{CE} is used as a baseline, and subtracted from each model's score to compute the absolute Δ score, *e.g.* $\Delta\text{LID}_{\text{SCL}} = \text{LID}_{\text{SCL}} - \text{LID}_{\text{CE}}$.

5.2 Out-of-Domain Generalization

As noted `GlotLID-C-train-down` contains the following 5 domains: Bible, Literature, Politics, Multi-domain, Random. To further evaluate the out-of-domain generalization capabilities of our approach, we conducted the following experiment: we retained only the Bible domain from `GlotLID-C-train-down` and excluded the others. Using this single-domain dataset, we trained two models — one with SCL (ConLID-S-B) and one with CE (LID_{CE} − B). We then evaluated both models on all five domains of `GlotLID-C`, as well as on Flores-200 and UDHR. The resulting macro F1-scores are presented in Table 18.

The results demonstrate that even when training is limited to a single domain (*i.e.*, Bible), our method improves generalization to previously unseen domains. Although the gains are modest under single-domain training, the advantage becomes substantially more pronounced when the model is trained on multiple domains.

5.3 FineWeb-2 Analysis

To assess the scalability and real-world applicability of our approach as an LID tool for multilingual web crawls, we evaluate its performance on the FineWeb-2 corpus.

Prediction Agreement. We compare ConLID-S with GlotLID-M, a state-of-the-art LID method, using a subset[3] of the FineWeb-2 corpus that has been labeled by GlotLID-M. To quantify their similarity, we compute the agreement score between their predictions and report the results for high- and low-resource languages in Table 6. The agreement score is defined as follows:

$$\text{Agreement} = \frac{|\text{Pred.}_{\text{ConLID-S}} \cap \text{Pred.}_{\text{GlotLID-M}}|}{\text{\# of documents}}$$

Resource Level	$\lvert L\rvert$	$\lvert D\rvert$	Macro Agreement	Micro Agreement
High	1396	4.5B	0.9004	0.9064
Low	273	2.9M	0.6708	0.5861

Table 6: Agreement between ConLID-S and GlotLID-M on FineWeb-2 filtered dataset. $|L|$ refers to the number of languages, while $|D|$ refers to the total number of documents in each group. The agreement scores have been calculated using 10–10K documents per language.

In §5.1, we demonstrated that the incorporation of SCL was particularly beneficial for low-resource languages. To illustrate its effect, Table 6 presents the agreement scores for different language groups in FineWeb-2. As expected, the agreement is high for high-resource languages, but much lower for low-resource languages, with a micro agreement rate of ∼58.61%. Since the dataset lacks ground-truth labels, it is not possible to directly determine

[3]10 – 10k documents per language.

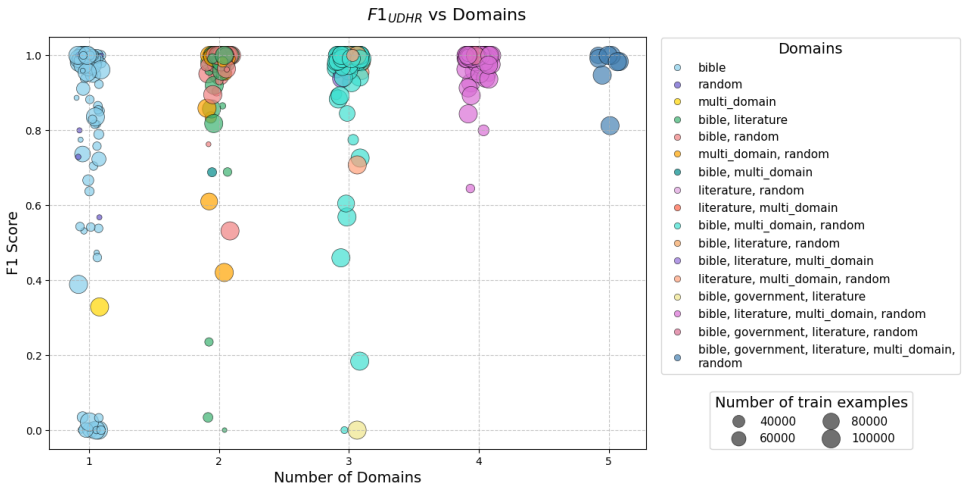

Figure 2: Performance of ConLID-S on the UDHR dataset based on training data domains. The more domains included in the training data, the higher performance the model shows for a given language. The size of the training data is not always positively correlated with model's performance, particularly in cases where languages are represented in limited domains.

which model is correct in cases of disagreement. However, as shown earlier in §5.1, incorporation of SCL outperforms the CE-based method, including GlotLID-M, on low-resource languages, suggesting that a low agreement score in these languages likely reflects ConLID-S making the correct predictions where the GlotLID-M fails. Importantly, even small differences are practically meaningful: a 1% improvement in low-resource language performance corresponds to ∼28,000 documents. This highlights the real-world impact of even minor improvements in low-resource LID models.

6 Under-performing languages

Using the ConLID-S model, we conduct a detailed analysis of the characteristics of languages exhibiting low performance on the UDHR dataset (*i.e.*, languages for which models display limited domain generalization capabilities).

Language Analysis by Domain. We categorize languages based on domain availability and the number of sentences available in `GlotLID-C-train-down`. Figure 2 presents the corresponding F1 score performance on the UDHR dataset. We observe that models tend to exhibit improved generalization (*i.e.*, higher F1 scores on UDHR) in languages for which they had more diverse training domains. Models tend to generalize worse on languages for which training data was only available in one domain. Interestingly, while a majority of single-domain languages only had

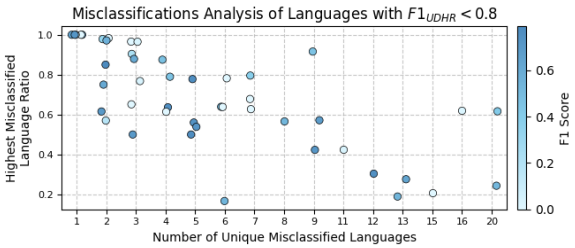

Figure 3: UDHR misclassification pairs (languages with F1 <0.8) using ConLID-S. Most misclassification for a language happens between less than 5 languages.

training data for the **Bible** domain, the model still achieves relatively high F1 scores for these languages, suggesting that training only on **Bible** data is not a sole factor contributing to low performance on UDHR. Finally, we observe that the size of the training data does not appear to be a sole determinant of low performance on UDHR either.

Language Misclassifications. To qualitatively assess these generalization issues, we now analyze patterns observed in misclassified language pairs. Among the 3448 misclassifications in the UDHR dataset, 99.6% occur within the same script, while only 0.4% involve different scripts. Most misclassifications also involve a target language being incorrectly identified as another high-resource language (see Appendix Table 16).

To better understand the diversity of languages involved in misclassifications, we examine the number of distinct languages a given language is misclassified as and whether there are consistent confu-

sion patterns with specific languages. For instance, the language "qug_Latn" (Chimborazo Highland Quichua), which belongs to the Quechuan language family and is spoken in Ecuador, is misclassified as 15 different languages. Notably, 60% of these cases involve "qup_Latn" (Southern Pastaza Quechua), another Quechuan language spoken in Ecuador, indicating a strong linguistic similarity that leads to frequent misclassification.

For languages in the UDHR dataset with an F1 score below 0.8, we visualize the misclassification trends in Figure 3. Specifically, we plot the ratio of the most frequently misclassified language to the total number of misclassifications for each language against the number of unique languages it is misclassified as.

Our analysis reveals that most languages are predominantly misclassified as only a small subset of the 2099 total languages (*i.e.*, *number of unique misclassified languages* < 10). This suggests that these languages share strong linguistic similarities with a limited set of other languages. Examples include Nyemba and Mbunda, both Bantu languages, and Northeastern and Southwestern Dinka, which exhibit significant linguistic overlap. Even languages with a high overall misclassification rate (*number of unique misclassified languages* > 10) tend to have one or a few dominant misclassification sources (*i.e.*, *highest misclassified language ratio*), as seen in the case of Chimborazo Highland Quichua and Southern Pastaza Quechua. More examples of such pairs are provided in Table 17.

7 Conclusion

In this work, we propose a novel approach that leverages supervised contrastive learning (SCL) for language identification (LID). The SCL objective explicitly encourages textual representations of the same language to form compact clusters while pushing apart representations of different languages in the embedding space. We evaluate our approach on three benchmark datasets: GlotLID, FLORES, and UDHR, demonstrating that incorporating SCL leads to considerable improvements over conventional cross-entropy (CE)-based LID methods. Specifically, our best SCL-based models achieve a 3.2 percentage points improvement over CE-based models for low-resource languages and a 6.8 percentage points improvement for languages with data from diverse domains, highlighting improved domain generalization, and underscoring the effectiveness of SCL in enhancing LID performance, particularly in challenging low-resource and cross-domain scenarios. Moreover, we also demonstrate that SCL and CE-based methods are complementary to each other and can be used together during inference, as in the case of GlotLID-M+ConLID-S.

Limitations

Contrastive learning has been shown to enhance model performance and generalization by capturing a broader range of factors of variation compared to traditional classification networks. However, its effectiveness relies on access to high-quality data from diverse domains, which poses a challenge in the development of Language Identification (LID) systems, where such data is often scarce.

While our training and in-domain test datasets encompass 2,099 languages, our out-of-domain evaluation dataset (UDHR) is limited to only 360 languages. This restriction constrains our ability to conduct a comprehensive analysis across the remaining languages. Additionally, the lack of evaluation dataset in other domains limits our ability to thoroughly assess method's generalization capacity.

References

Ife Adebara, 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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

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, Toronto, Canada. Association for Computational Linguistics.

Isaac Caswell, Theresa Breiner, Daan van Esch, and Ankur Bapna. 2020. Language ID in the wild: Unexpected challenges on the path to a thousand-language web text corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*,

pages 6588–6608, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

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. In *Annual Meeting of the Association for Computational Linguistics*.

Ona de Gibert, Graeme Nail, Nikolay Arefyev, Marta Bañón, Jelmer van der Linde, Shaoxiong Ji, Jaume Zaragoza-Bernabeu, Mikko Aulamo, Gema Ramírez-Sánchez, Andrey Kutuzov, Sampo Pyysalo, Stephan Oepen, and Jörg Tiedemann. 2024. A new massive multilingual dataset for high-performance language technologies. *Preprint*, arXiv:2403.14009.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 516 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Jonathan Dunn. 2020. Mapping languages: the corpus of global language use. *Language Resources and Evaluation*, 54(4):999–1018.

Jonathan Dunn and Lane Edwards-Brown. 2024. Geographically-informed language identification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7672–7682, Torino, Italia. ELRA and ICCL.

Abteen Ebrahimi and Katharina Kann. 2021. How to adapt your pretrained multilingual model to 1600 languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4555–4567, Online. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. 2021. Supervised contrastive learning for pre-trained language model fine-tuning. In *International Conference on Learning Representations*.

Ayyoob Imani, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André Martins, François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1082–1117, Toronto, Canada. Association for Computational Linguistics.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020a. The State and Fate of Linguistic Diversity and Inclusion in the NLP World. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020b. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas 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, Valencia, Spain. Association for Computational Linguistics.

Amir Hossein Kargaran, Ayyoob Imani, François Yvon, and Hinrich Schuetze. 2023. GlotLID: Language identification for low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6155–6218, Singapore. Association for Computational Linguistics.

Amir Hossein Kargaran, François Yvon, and Hinrich Schuetze. 2024. Glotcc: An open broad-coverage commoncrawl corpus and pipeline for minority languages. In *Advances in Neural Information Processing Systems*.

Md Tawkat Islam Khondaker, Muhammad Abdulmageed, and Laks Lakshmanan, V.s. 2023. Cross-platform and cross-domain abusive language detection with supervised contrastive learning. In *The 7th Workshop on Online Abuse and Harms (WOAH)*, pages 96–112, Toronto, Canada. Association for Computational Linguistics.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral,

Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, Jörg Frohberg, Mario Šaško, Quentin Lhoest, Angelina McMillan-Major, Gerard Dupont, Stella Biderman, Anna Rogers, Loubna Ben allal, Francesco De Toni, and 35 others. 2023. The bigscience roots corpus: A 1.6tb composite multilingual dataset. *Preprint*, arXiv:2303.03915.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, and 40 others. 2024. Datacomp-lm: In search of the next generation of training sets for language models. *Preprint*, arXiv:2406.11794.

Paul McNamee. 2005. Language identification: a solved problem suitable for undergraduate instruction. *J. Comput. Sci. Coll.*, 20(3):94–101.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. *Preprint*, arXiv:2309.09400.

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*.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, and 61 others. 2022. Scaling language models: Methods, analysis & insights from training gopher. *Preprint*, arXiv:2112.11446.

Dwaipayan Roy, Sumit Bhatia, and Prateek Jain. 2022. Information asymmetry in Wikipedia across different languages: A statistical analysis. *Journal of the Association for Information Science and Technology*, 73(3):347–361.

Thapelo Andrew Sindane and Vukosi Marivate. 2024. From n-grams to pre-trained multilingual models for language identification. In *Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities*, pages 229–239, Miami, USA. Association for Computational Linguistics.

Qingyu Tan, Ruidan He, Lidong Bing, and Hwee Tou Ng. 2022. Domain generalization for text classification with memory-based supervised contrastive learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6916–6926, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Xinyan Yu, Trina Chatterjee, Akari Asai, Junjie Hu, and Eunsol Choi. 2022. Beyond Counting Datasets: A Survey of Multilingual Dataset Construction and Necessary Resources. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3725–3743, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Data Statistics

We report the `GlotLID-C-train-down` distribution by domains (Table 7), script (Table 8) and resource level (Table 9). As can be seen from Table 8, most languages contains either exclusively or inclusively Bible data.

Script	#Languages	Avg. #Data Samples per language
Java	2	50,118
Tibt	4	42,773
Hani	5	83,411
Tfng	5	22,801
Arab	39	46,203
Cyrl	71	42,553
Latn	1702	30,706

Table 7: `GlotLID-C-train-down` distribution by scripts. Only the scripts used in §5.1 are shown.

B Training Details

B.1 Comparison details to GlotLID-M

The comparison of technical details of ConLID-S, LID_{CE} and GlotLID-M can be found from Table 10.

B.2 Hyperparameters

We adopt the same text representation approach as FastText, utilizing character n-grams and word embeddings . Specifically, we set the following parameters: *minCount* = 1000, *minCountLabel* = 0, *wordNgrams* = 1, *bucket* = 10^6, *minn* = 2, *maxn* = 5, and *dim* = 256.

Domains	#Languages	Avg. #Data Samples per language
Bible, Government, Literature	1	100,000
Bible, Government, Literature, Random	1	100,000
Literature, Random	2	50,138
Literature, Multi-Domain	2	11,742
Bible, Literature, Multi-Domain	3	59,662
Literature	4	525
Bible, Government, Literature, Multi-Domain, Random	8	97,635
Literature, Multi-Domain, Random	8	44,277
Bible, Literature, Random	16	57,210
Bible, Multi-Domain	21	25,936
Multi-Domain	42	13,764
Multi-Domain, Random	53	32,901
Random	61	7,061
Bible, Literature, Multi-Domain, Random	89	93,826
Bible, Random	95	42,333
Bible, Multi-Domain, Random	121	76,753
Bible, Literature	157	35,128
UND	161	100,000
Bible	1254	22,874
Overall	2099	51,145

Table 8: `GlotLID-C-train-down` distribution by domains combination. Those with **Bible** data included are highlighted.

Resource Level	#Languages	Avg. #Data Samples per language
Low	398	3,959
High	1540	39,049

Table 9: `GlotLID-C-train-down` distribution by resource level. Those with UND labels have been excluded.

We train our models for a single epoch following Kargaran et al. (2024). We, also, found that additional epochs yield no noticeable improvement due to the rapid convergence. We use a batch size of 128 with a gradient accumulation step of 1, which is the maximum supported by our GPU configuration. Model training was conducted on 8 NVIDIA A100 GPUs (80 GB each) with 512 GB RAM. Optimization was performed using the AdamW optimizer with the following hyper-parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-06$, weight decay $\lambda = 0$, and a linear learning rate schedule.

Hyper-parameter tuning was conducted on the GlotLID-C-test, FLORES-200, and UDHR datasets, involving all 369 languages for UDHR. Macro F1 score served as the primary evaluation metric. Below, we summarize the tuning process and the selected hyper-parameters for down-sampling, contrastive temperature (τ), learning rate, memory bank sizes (M_S, M_H), and the minimum number of negatives (K).

GlotLID-C-train Down-sampling: We evaluated down-sampling high-resource languages using values 10K, 50K, 100K. The LID_{CE} model performed best with a cap of 100K, as shown in Table 11. Up-sampling, as proposed by Team et al. (2022), was also tested but yielded inferior results compared to down-sampling.

Effect of Contrastive Temperature (τ): We experimented with temperatures 0.02, 0.05, 0.1, 0.2, 0.4 (Table 12) and selected 0.05 based on empirical results and alignment with findings from Gao et al. (2021).

SCL Learning Rate: Initial experiments indicated that learning rates in the order of 10^{-3} achieved better convergence. We compared rates 1e-3, 2e-3, 4e-3 in Table 13 and selected 1e-3 as the optimal value.

Memory Bank Size for Soft Selection (M_S): We evaluated values 1024, 2048, 4096 for M_S and selected 2048 as the final configuration (Table 14).

Memory Bank Size (M_H) and Minimum Number of Negatives (K) for Hard Selection: Since M_H and K are interdependent, we tested combinations as shown in Table 15. The best performance was achieved with $M_H = 2048$ and $K = 1024$.

Algorithm 1 Hard Negative Sample Selection

Require: A minibatch $S = \{x_i, y_i; s_i, d_i\}_{i=1}^{B}$, and a number of minimum negatives K

for $i < B$ **do**
  $N(i) = \{z_j \in S, y_j \neq y_i, s_j = s_i, d_j = d_i\}$ {Condition 1}
  if $|N(i)| < K$ **then**
    $N(i) = \{z_j \in S, y_j \neq y_i, s_j = s_i\}$ {Condition 2}
  end if
  if $|N(i)| < K$ **then**
    $N(i) = \{z_j \in S, y_j \neq y_i, d_j = d_i\}$ {Condition 3}
  end if
  if $|N(i)| < K$ **then**
    $N(i) = \{z_j \in S, y_j \neq y_i\}$ {Condition 4}
  end if
end for
return $N(i)$ {negative pairs for each sample}

C Hard Selection Algorithm

The negatives pairs in the Hard selection method, as explained in §3.3, are sampled according to the Algorithm 1.

D Additional Results

D.1 Significance testing using bootstrapping

We additionally applied a t-test using bootstrapping as a significance testing method to more clearly

	GlotLID-M	LID_{CE}	ConLID-S
Train data	GlotLID-C(85%) (Upsampled)	GlotLID-C(85%) (Downsampled to 100K)	GlotLID-C(85%) (Downsampled to 100K)
Loss function	CE	CE	CE+SCL
Number of labels	2102	2099	2099
Train Module	FastText	PyTorch	PyTorch
Optimizer	SGD	ADAM	ADAM
#Epochs	2	1	1
Learning Rate	0.8	1e-3	1e-3

Table 10: Comparison of ConLID-S, LID_{CE} and GlotLID-M training configurations.

	Down-sampling max cap			
	10K	**50K**	**100K**	**Full**
GlotLID-C-test	0.9776	0.9855	0.9868	0.9855
UDHR	0.8592	0.8683	0.8688	0.8556
FLORES-200	0.9518	0.9615	0.9638	0.9621

Table 11: Macro F1 score of LID_{CE} using 10K, 50K, 100K to down-sample the high-resource languages, and using the full train dataset.

	Contrastive Temperature τ				
	0.02	**0.05**	**0.1**	**0.2**	**0.4**
GlotLID-C-test	0.9841	0.9864	0.9866	0.9851	0.9861
UDHR	0.8634	0.8687	0.8687	0.8617	0.8635
FLORES-200	0.9602	0.9625	0.9628	0.9615	0.9624

Table 12: Macro F1 score of LID_{SCL} using contrastive temperature, τ, of 0.02, 0.05, 0.1, 0.2, 0.4.

	Learning Rate		
	1e-3	**2e-3**	**4e-3**
GlotLID-C-test	0.9864	0.9858	0.9841
UDHR	0.8687	0.8631	0.8552
FLORES-200	0.9625	0.9622	0.9611

Table 13: Macro F1 score of LID_{SCL} trained with the learning rate values of 1e-3, 2e-3, 4e-3.

	$\mathbf{M_S}$		
	1024	**2048**	**4096**
GlotLID-C-test	0.9884	0.9883	0.9883
UDHR	0.8739	0.8742	0.8713
FLORES-200	0.9646	0.9649	0.9648

Table 14: Macro F1 score of ConLID-S with the values of 1024, 2048, 4096 for memory bank size, M_S.

		$\mathbf{M_H}$			
	K	**1024**	**2048**	**4096**	**8192**
GlotLID-C-test	512	0.9884	0.9882	0.9884	0.9882
	1024	-	0.9883	0.9885	0.9885
UDHR	512	0.8690	0.8731	0.8743	0.8700
	1024	-	0.8739	0.8716	0.8710
FLORES-200	512	0.9649	0.9649	0.9651	0.9653
	1024	-	0.9647	0.9648	0.9646

Table 15: Macro F1 score of ConLID-H with the values {1024, 2048, 4096, 8192} and {512, 1024} for memory bank size (M_H) and minimum number of negatives (K) respectively.

True → Pred	**# Misclassifications**
High → High	2182
Low → High	528
High → Low	440
Low → Low	281

Table 16: Number of misclassifications by data resources (from true to predicted language resource level) in UDHR dataset using ConLID-S

L_1	L_2	**# errors**
krl_Latn	olo_Latn	183 (100%)
bam_Latn	dyu_Latn	8 (100%)
hrv_Latn	hbs_Latn	45 (98%)
srp_Latn	hbs_Latn	29 (88%)
hak_Hani	wuu_Hani	43 (77%)
wuu_Hani	cmn_Hani	3 (75%)
qvn_Latn	quy_Latn	23 (64%)
yue_Hani	cmn_Hani	8 (62%)
zgh_Tfng	tzm_Tfng	14 (54%)
fuf_Latn	fuv_Latn	8 (27%)

Table 17: Examples of language pairs where the first language is frequently misclassified as the second.

Dataset	Domain Shift	Resource Level	$\lvert L_{\text{test}} \rvert$	LID_{CE} − B	ConLID-S-B
Bible	ID	High	1400	**0.9961**	**0.9961**
		Low	368	0.9883	**0.9885**
Random	OOD	High	202	0.5933	**0.6075**
		Low	104	0.6429	**0.6481**
Multi-Domain	OOD	High	152	0.7996	**0.8067**
		Low	90	**0.7514**	0.7444
Literature	OOD	High	119	0.8521	**0.8638**
		Low	51	0.7641	**0.7890**
Government	OOD	High	5	**0.9233**	0.9137
		Low	5	0.6263	**0.6358**
UDHR	OOD	High	246	0.8854	**0.8886**
		Low	83	0.7961	**0.8039**
FLORES-200	OOD	High	101	0.8938	**0.8969**
		Low	66	0.8367	**0.8375**

Table 18: Macro F1 scores of ConLID-S-B and LID_{CE} − B across in-domain and out-of-domain datasets.

	GlotLID-C Test	UDHR	FLORES-200
LID_{CE}	0.9874	0.6080	0.8598
ConLID-S	0.9878	0.6176	0.8616
GlotLID-M	0.9877	0.5771	0.8614
GlotLID-M+ConLID-S	0.9943	0.6201	0.8859

Table 19: Macro F1 score via bootstrapping with 1000 iterations (p<0.05)

demonstrate the effectiveness of SCL approach (Table 19). Based on the bootstrapped results, p-value is less than 0.05, and the SCL method still outperforms both the GlotLID-M and the cross-entropy baseline, LID_{CE}.

The relatively lower scores for the UDHR dataset can be attributed to the imbalance in the number of sentences across languages. Some over-represented languages exhibit lower performance, and their frequent selection during bootstrapping iterations negatively impacts the overall results.

D.2 Script Analysis Breakdown

We showed in §5.1 that while low-resource scripts have shown significant improvement over the baseline using ConLID-S, some high-resource scripts had more modest improvement. We show the breakdown of the script analysis by the resource level in Table 20. We attribute the performance gains primarily to the resource level of the languages associated with each script.

While the number of low-resource languages is typically smaller, those within a given script tend to show greater improvements in performance compared to their high-resource counterparts. Additionally, the number of languages per script varies significantly. For instance, the Latin script encompasses 282 languages in this table, whereas the substantial improvement seen in the Java script is based on a single language. This discrepancy in language count per script also contributes to the observed performance variance.

D.3 Resource level categorization

Languages can be categorized as low- or high-resource based on either the availability of large amounts of unlabeled data (*e.g.*, pretraining data collected from the web) or the quantity of labeled data available for specific tasks. In many cases, these two criteria coincide. However, in this study, our primary goal is to evaluate the effectiveness of our approach under conditions where labeled data for training an LID system is scarce. For this reason, we adopt the second criterion — the amount of labeled training data — as the basis for our categorization. Specifically, we classify languages with fewer than 10,000 labeled examples as low-resource, and those with 10,000 or more labeled examples as high-resource, as described in Section 5.1. We find this definition particularly meaningful because the performance of language models is strongly influenced by the amount of labeled data available for training. To provide a complementary perspective, we also analyze our results using the categorization proposed by Joshi et al. (2020b). Table 21 presents the performance of our method on the UDHR dataset according to this scheme. Labels 0 through 5 follow the definitions from Joshi et al. (2020b), and we introduce an additional label, -1, for languages not included in their original categorization (low-resource and under-represented languages). As shown, most of the performance gains occur within these labeled groups, highlighting the robustness of our approach across different categorization schemes.

Script	Resource Level	#UDHR Languages	LID_{CE}	ΔLID_{SCL}	ΔConLID-S	ΔConLID-S+LID_{CE}
Arabic	Low	5	0.5176	0.1275	0.2824	0.1419
Arabic	High	1	0.9388	0.0078	0.0249	0.0106
Cyrillic	Low	5	0.9525	0.0072	0.0019	0.0038
Cyrillic	High	24	0.9877	0.0039	0.0005	0.0023
Hani	High	4	0.7570	-0.0468	-0.0088	0.0014
Java	Low	1	0.8829	0.0923	0.1090	0.1007
Latin	Low	20	0.8394	0.0039	0.0128	0.0108
Latin	High	262	0.9560	-0.0037	0.0026	0.0025
Tifinagh	Low	1	0.7158	0.0134	0.0134	0.0134
Tibetan	High	2	0.9411	0.0253	0.0084	0.0084

Table 20: F1 score change of the LID models' with SCL by language script broken down by the resource level. LID_{CE} is used as a baseline, and subtracted from each model's score to compute the absolute Δ score, *e.g.* ΔLID_{SCL}= LID_{SCL}-LID_{CE}.

Language Class	$\lvert L \rvert$	LID_{CE}	ConLID-S
-1	135	0.8183	0.0204
0	93	0.9059	0.0004
1	76	0.9554	0.0036
2	13	0.9686	-0.0024
3	20	0.9830	0.0067
4	17	0.9311	-0.0020
5	6	0.9812	0.0008

Table 21: The performance of the models on UDHR dataset based on the categorization low- and high-resource languages of Joshi et al. (2020b).

D.4 UND labels recovery

Kargaran et al. (2023) introduced the UND_XXX labels due to the nature of GlotLID-M, which is based on the FastText model. This model relies on hashed n-grams rather than storing the n-grams themselves to save memory and facilitate efficient embedding lookups. Consequently, GlotLID-M always produces a prediction, even for n-grams that were not observed during training.

In our approach, we adopted a similar method but instead opted to store all n-grams encountered during training. This allows us to differentiate between previously seen n-grams and those unseen during training, which we classify as UNK-ngram during inference. To assess the effectiveness of this approach, we applied our model (ConLID-S) to the FineWeb-2 dataset, specifically focusing on instances labeled as UND_XXX. We then apply a five-step filtering process to extract "clean data" from these samples:

1. model's prediction label script is the same as UND_XXX script
2. model's prediction probability is above 95%
3. a document contains less than 5% UNK-ngrams
4. a document does not contain any space-separated letters (*e.g.*, *A B C D E ...*)
5. a document contains more than one word

The proportion of recovered clean data using these filtering criteria is illustrated in Figure 4. We report results for nine scripts where script agreement exceeds 5%. As shown, the data remains largely non-recoverable. A high level of script disagreement is likely attributable to noisy or meaningless data, where predictions from both GlotLID-M and ConLID-S approach Randomness. However, we were able to recover a limited number of data points for certain scripts, including Bengali (Beng, ∼3%, 1,429 samples), Gothic (Goth, ∼2%, 113 samples), Hebrew (Hebr, ∼2%, 4,256 samples), and Cyrillic (Cyrl, ∼1%, 10,208 samples). The recovered data primarily consists of short sentences or repetitive word patterns, yet they still align with the predicted language. For instance, the recovered Cyrillic data includes the following sentences in Russian:

- ну и я ушла (nu i ya ushla) – *well, I left too*
- ну и с пятницей ну и, ежели чо (nu i s pyatnitsey nu i, yezheli cho) – *well, happy Friday, and, if anything*
- Народ в ЖЖ флешмобится (Narod v ZhZh fleshmobitsya) – *people on LiveJournal are doing a flash mob.*

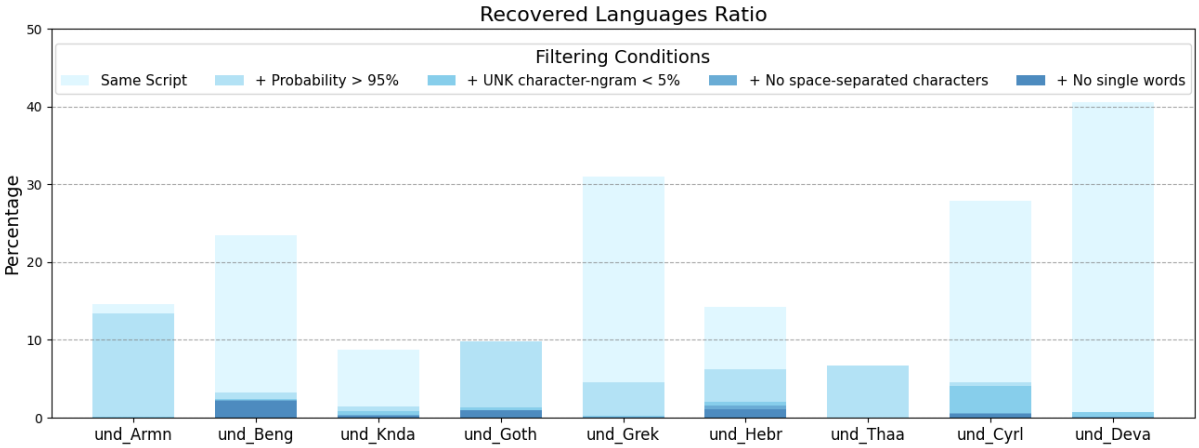

Figure 4: Ratio of the recovered text by applying 5 levels of filtering on FineWeb-2 UND_XXX dataset. The prediction is carried out by ConLID-S model.