Mark My Words: A Robust Multilingual Model for Punctuation in Text and Speech Transcripts

Sidharth Pulipaka[1] **Ashwin Sankar**[1,2] **Raj Dabre**[1,2,3*]
[1]Nilekani Centre at AI4Bharat [2]Indian Institute of Technology, Madras
[3]Indian Institute of Technology, Bombay

🤗 ai4bharat/Cadence

Abstract

Punctuation plays a vital role in structuring meaning, yet current models often struggle to restore it accurately in transcripts of spontaneous speech, especially in the presence of disfluencies such as false starts and backtracking. These limitations hinder the performance of downstream tasks like translation, text-to-speech, summarization, etc. where sentence boundaries are critical for preserving quality. In this work, we introduce Cadence, a generalist punctuation restoration model adapted from a pretrained large language model. Cadence is designed to handle both clean written text and highly spontaneous spoken transcripts. It surpasses the previous state-of-the-art in performance while expanding support from 14 to all 22 Scheduled Languages of India and English. We conduct a comprehensive analysis of model behavior across punctuation types and language families, identifying persistent challenges under domain shift and with rare punctuation marks. Our findings demonstrate the efficacy of utilizing pretrained language models for multilingual punctuation restoration and highlight Cadence's practical value for low-resource NLP pipelines at scale.

1 Introduction

Punctuation plays a vital role in written language, offering syntactic structure, semantic clarity, and pragmatic cues such as pauses, emphasis, and sentence boundaries. However, text generated by Automatic Speech Recognition (ASR) systems or large-scale web crawls often lacks punctuation (Bhogale et al., 2025). This absence significantly impairs both human readability and the effectiveness of downstream NLP tasks like Machine Translation (MT), Text Summarization, and Sentiment Analysis, posing a widespread challenge in processing raw textual data.

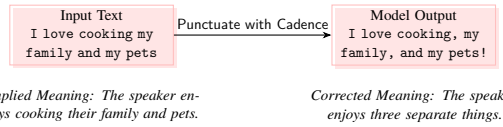

Figure 1: An example of critical disambiguation by our model, *Cadence*. By inserting serial commas, the model correctly interprets the input as a list of three distinct items, rather than a single, disturbing hobby.

While punctuation restoration has progressed for high-resource languages like English (Guhr et al., 2021), Indic languages face substantial hurdles. These include scarcity of annotated corpora, especially for low-resource languages, and linguistic complexity with diverse scripts, grammars, and unique marks like the Devanagari purna virama. Prior efforts are often limited in language or punctuation scope, or use non-scalable, language-specific models, hindering cross-lingual generalization, particularly for under-represented languages (Gupta et al., 2022).

To bridge this gap for Indic languages, we first construct a large, diverse multilingual punctuation corpus covering both written and ASR-transcribed text, aggregating multiple sources, including Sangraha-verified (Khan et al., 2024), IndicVoices (Javed et al., 2024), translated Cosmopedia (Ben Allal et al., 2024), and IndicCorp-v2 (Doddapaneni et al., 2023). Secondly, we adapt the GEMMA3-1B-PRETRAIN model (Team et al., 2025) for punctuation restoration by converting it into a bidirectional transformer using a Masked Next Token Prediction (MNTP) objective (BehnamGhader et al., 2024), which we call Cadence. Thirdly, we evaluate the model's performance on crucial and frequently used punctuations as well as rare punctuations. Moreover, we establish Cadence as the state-of-the-art tool for punctuation restoration across both written and spoken contexts in English and 22 Indian languages;

*Corresponding Author: raj.dabre@cse.iitm.ac.in

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 1758–1776

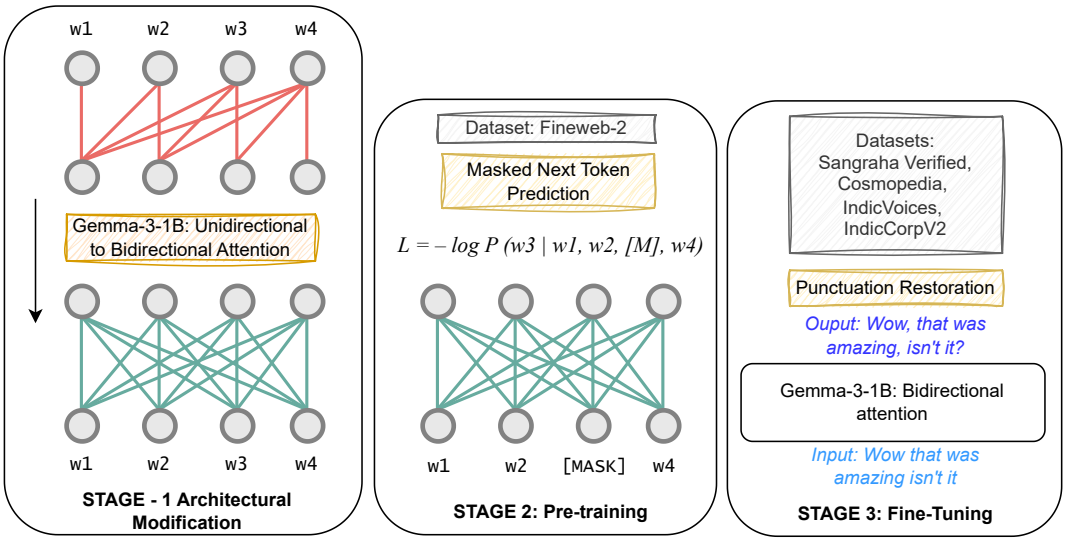

Figure 2: Overview of our training methodology. *Stage-1:* Modify causal attention to bidirectional attention. *Stage-2:* Pre-train with Masked Next Token Prediction Objective. *Stage-3:* Train for punctuation restoration, as a token-level classification task. Figure inspired from BehnamGhader et al., 2024.

Assamese, Bengali, Bodo, Dogri, Gujarati, Hindi, Kannada, Kashmiri, Konkani, Maithili, Malayalam, Manipuri, Marathi, Nepali, Odia, Punjabi, Sanskrit, Santali, Sindhi, Tamil, Telugu, and Urdu. Finally, we evaluate Cadence across two downstream tasks, namely: Machine Translation and Speech Translation.

Cadence establishes a new state-of-the-art (SOTA) performance across multiple Indic languages and domains, surpassing existing baselines with an average Macro F1 improvement of 77% while also introducing support for 11 new languages. It demonstrates significant gains in translation quality, on average improving the BLEU score by 1.75 and the chrF++ score by 1.55 for Machine Translation. For Speech Translation, it achieves an even greater enhancement with an average increase of 2.4 in the BLEU score and 2.85 in the chrF++ score. To foster further advancements and support downstream applications, we release all code, models and other artifacts publicly under permissive licenses.

2 Related Work

Punctuation Restoration in Machine Translation and Speech Translation: Punctuation restoration (PR) is a crucial preprocessing step for machine translation (MT) and speech translation (ST). In MT, punctuation provides essential segmentation and syntactic cues vital for translation quality (Vandeghinste et al., 2018); its absence degrades translations. The impact is greater in ST, where unpunctuated Automatic Speech Recognition (ASR) transcripts hinder segmentation crucial for real-time systems and data alignment (Javed et al., 2024; Sankar et al., 2025).

Punctuation Restoration for Indic Languages: Indic languages present unique PR challenges due to linguistic diversity and specific punctuation conventions. Early efforts were often monolingual, limiting scalability and cross-lingual transfer (Tripathy and Samal, 2022; Gupta et al., 2022). Gupta et al. (2022) introduced IndicPunct, a suite of monolingual transformer-based models for 14 Indian languages. While effective on formal text, IndicPunct faced limitations with spontaneous speech transcripts and a restricted punctuation set. These shortcomings highlight the need for more robust, generalist models for Indic languages, especially for spontaneous speech.

Resources and Models for Punctuation Restoration: While punctuation restoration has shifted from BERT-based classifiers (Gupta et al., 2022; Guhr et al., 2021) to LLMs (Sankar et al., 2025), a critical gap remains for Indic languages. Most models are trained on formal web text (Penedo et al., 2024; Doddapaneni et al., 2023) and perform poorly on disfluent spontaneous speech, a problem exacerbated by the lack of large, punctuated spoken corpora. Consequently, non-autoregressive models that generalize across written and spoken domains while handling large punctuation sets are rare, a

void that Cadence; a scalable, multilingual, LLM-based model; is designed specifically to address.

3 Methodology

Our approach to developing a robust multilingual punctuation restoration model hinges on two core pillars: a comprehensive data strategy designed to encompass linguistic diversity and varied text styles, and a multi-stage model training and adaptation process.

3.1 Data Strategy for Multilingual Punctuation Restoration

The foundation of our methodology lies in the careful curation and utilization of diverse textual data for both pre-training and fine-tuning phases.

Pre-training Data Corpus: For the initial continual pre-training phase, we leverage large-scale, high-quality multilingual web corpora. These resources are selected for their broad linguistic coverage, providing the model with exposure to a wide array of languages and writing styles. This extensive, general-domain data helps in building foundational representations that are adaptable to various downstream tasks and linguistic contexts.

Fine-tuning Data Amalgamation: We construct a large, diverse fine-tuning dataset by aggregating text from various corpora spanning multiple domains (e.g., news, literature, web text, extempore text) and styles. This dataset combines formal written text with unstructured spoken language transcripts, which features disfluencies, repetitions, and false starts, to expose the model to a wide range of punctuation patterns, improving its robustness and ability to generalize across real-world inputs.

3.2 Model Training and Adaptation

The model undergoes a multi-stage training process, starting from a pre-trained foundation, followed by continual pre-training for domain and multilingual adaptation, and culminating in task-specific fine-tuning.

3.2.1 Model Architecture Adaptation

We begin with a foundation pre-trained transformer-based language model. Standard autoregressive language models are typically designed for unidirectional text generation, processing context only from preceding tokens. However, for sequence tagging tasks like punctuation restoration, where understanding the surrounding context is crucial, bidirectional information flow (accessing both preceding and succeeding tokens) is highly beneficial. Therefore, we adapt the model's attention mechanism to be fully bidirectional. This modification allows each token to attend to all other tokens in the input sequence, enabling a richer contextual understanding necessary for accurate punctuation prediction during subsequent training stages.

3.2.2 Continual Pre-training for Enhanced Representation

To further adapt the bidirectionally-modified model for the nuances of the diverse linguistic landscape it will encounter and to better prepare it for the sequence tagging nature of the punctuation restoration task, we perform a dedicated phase of continual pre-training.

Masked Next Token Prediction: We employ a Masked Next Token Prediction (MNTP) objective, adapted from BehnamGhader et al. (2024). In this setup, after masking a random subset of input tokens, the model's task is to predict a masked token at position $i + 1$ using only the representation of its preceding, unmasked token i. Although the representation for token i is built using bi-directional attention over the entire unmasked sequence, the predictive objective remains strictly local. This design compels the model to learn strong local dependencies, making it particularly effective for tasks like punctuation prediction where the immediately preceding word is the most salient cue.

Curriculum Learning for Multilingual Adaptation: Given the significant variation in data availability (ranging from high-resource to low-resource languages) and the diverse linguistic characteristics across the target languages, we adopt a curriculum learning strategy during continual pre-training. This staged approach gradually introduces linguistic complexity to the model:

1. Foundation Phase: Training initially focuses on a high-resource language (a language with abundant available training data) to establish robust foundational representations.

2. High-to-Mid Resource: The model is then exposed to a group of mid-to-high-resource languages. These are languages for which substantial amounts of training data are available, though typically less than the initial high-resource language. This phase allows the model to begin generalizing across related linguistic structures and benefit from these larger datasets.

3. Low Resource: Subsequently, lower-resource

languages are introduced. These are languages characterized by comparatively limited availability of training data. This step encourages knowledge transfer from the more data-rich languages learned in previous phases, which is critical for achieving good performance on languages with scarce data.

4. Consolidation Phase: Finally, the model is trained on a mixture of data from all supported languages. This phase aims to consolidate learning across the entire linguistic spectrum and mitigate potential catastrophic forgetting of earlier-learned languages or features.

This progressive exposure helps the model to incrementally adapt to increasing linguistic diversity while maintaining training stability and fostering cross-lingual transfer.

3.2.3 Task-Specific Fine-tuning for Punctuation Restoration

Following the continual pre-training phase, the adapted language model is fine-tuned specifically for the punctuation restoration task using the amalgamated dataset described earlier.

Task Formulation: We frame punctuation restoration as a token-level sequence classification problem. For each token in an input unpunctuated sequence, the model's objective is to predict the punctuation mark that should follow it. If no punctuation is appropriate after a token, it predicts a special label, indicating the absence of punctuation.

Model Head: To adapt the pre-trained model for this classification task, its original language modeling head is replaced with a new task-specific head. This consists of a linear classification layer that takes the final hidden state representation of each input token and outputs logits over the set of possible punctuation classes.

Punctuation Label Space: The model is trained to predict a comprehensive set of punctuation marks, including standard punctuation marks, as well as script-specific punctuation found within the target language and frequently occurring combinations of punctuation marks to capture more complex typographic conventions.

Sampling Strategy for Data Imbalance: Recognizing the data imbalance across languages in our fine-tuning corpus, we employ a weighted sampling strategy. This technique ensures adequate representation for low-resource languages by oversampling their data, thereby promoting balanced learning and robust cross-lingual performance (Arivazhagan et al., 2019).

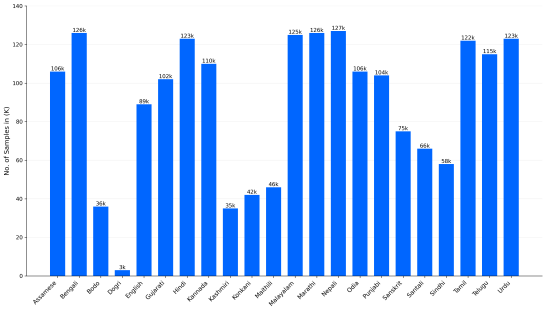

Figure 3: Statistics of our training corpus, showing the number of entries available for each supported language, represented in thousands.

Punctuation Mark	Instances	Punctuation Mark	Instances
.	8,916k	۔	1,537k
,	12,777k	؟	72k
?	531k	."	89k
-	2,949k	).	66k
;	308k	),	118k
_	183k	“,	10k
!	675k	“.	10k
’	1,720k	?”	41k
…	28k	”?	578
“	1,057k	!”	100k
।	10,002k	”।	14k
(	1,697k	،	875k
)	1,235k	۔	203k
:	1,159k	॥	420k
،	377	।।	86k

Table 1: Breakdown of supported punctuation marks and the number of instances for each in our training corpus, represented in thousands.

4 Experimental Setup

In this section, we describe the datasets, training procedures, and metrics used to evaluate Cadence.

4.1 Datasets

The development of Cadence relies on carefully curated datasets for both its continual pre-training and task-specific fine-tuning phases, ensuring broad linguistic coverage and exposure to diverse text styles.

Pretraining Dataset We source pretraining data from the Indic subset of FineWeb-2 (Penedo et al., 2024). This high-quality, multilingual web corpus provides broad coverage across the Indian linguistic landscape, a result of its web-scale collection and rigorous filtering.

Fine-tuning Dataset: To adapt our model for the punctuation task, we constructed a multilingual training corpus by aggregating data from four diverse sources, each contributing complementary strengths. **Sangraha-Verified (S)** provides high-

quality, accurately punctuated formal text (Khan et al., 2024); **IndicVoices-ST (IV)** offers punctuated transcripts of spontaneous speech, capturing spoken language patterns (Sankar et al., 2025); the **Translated Cosmopedia (C)** dataset introduces syntactically varied, structured knowledge content (Ben Allal et al., 2024); and **IndicCorp-v2 (IC)** contributes wide-domain natural language text with rich punctuation usage (Doddapaneni et al., 2023). This combination ensures broad linguistic coverage and stylistic diversity.

By combining these sources, we create a robust training corpus. Figure 3 provides a detailed breakdown of this aggregated dataset, showing the approximate number of training instances per language and Table 1 shows the counts for each of the 30 supported punctuation labels in the overall training set.

4.2 Training Details

In this section we elucidate the training details including model architecture, pretraining and finetuning details and evaluation setup.

4.2.1 Model Architecture

We adopt GEMMA3-1B-PRETRAIN (Team et al., 2025) as our base model. Although Gemma is originally designed as a causal decoder for text generation, punctuation restoration benefits from access to bidirectional context. We replace its causal attention mechanism with bidirectional attention to attend to both left and right contexts.

4.2.2 Continual Pretraining

To adapt the modified GEMMA3-1B-PRETRAIN model to the multilingual and low-resource nature of the task, we perform continual pretraining (CPT) using the MNTP objective and a curriculum over languages.

Curriculum Learning Strategy: To manage linguistic diversity and data imbalance, we employ a four-phase curriculum (Dabre et al., 2019) with progressively adjusted masking ratios. We first continually train the model on English (0.30 ratio), then adapted to 13 high- and mid-resource Indic languages (Hindi, Telugu, Tamil, Bengali, Malayalam, Marathi, Kannada, Gujarati, Assamese, Oriya, Punjabi, Sindhi, Urdu) with a 0.25 ratio. Subsequently, low-resource languages including Bodo, Dogri, Konkani, Kashmiri, Maithili, Manipuri, Nepali, Sanskrit, and Santali are introduced (0.15 ratio) to encourage generalization. In the final phase (last 10% of steps), all 23 languages are trained on jointly (0.25 ratio) to consolidate learning and mitigate catastrophic forgetting.

4.2.3 Finetuning

After the pretraining stage, the model is fine-tuned for supervised punctuation restoration, framed as token-level sequence tagging. For each input token, it predicts a subsequent punctuation mark or an O (Outside) label if no punctuation follows. To achieve this, we modify the output vocabulary of GEMMA3-1B-PRETRAIN to handle the token-level sequence tagging task, training it to predict one of 30 punctuation classes or an O (Outside) label for each token. This comprehensive label space (detailed in Table 1) includes standard English, Indic-specific, Arabic-script (Urdu) marks, and frequent multi-character combinations, enabling fine-grained modeling of diverse writing styles.

We train Cadence using the AdamW optimizer (Loshchilov and Hutter, 2019) with a max learning rate of $2 \times e^{-4}$ with 10% of the training steps as warmup followed by a cosine decay to $1 \times e^{-6}$, with an effective batch size of 64.

4.3 Evaluation

Test Set: Our *intrinsic evaluation* set comprises held-out samples from IndicCorp-v2 (IC), Sangraha-Verified (S), a translated Cosmopedia dataset (C), and BPCC (Gala et al., 2023). For spontaneous speech transcripts from the synthetically augmented IndicVoices-ST (IV) dataset, using Gemini-2.5-Flash (Comanici et al., 2025), we apply an additional programmatic quality filter (refer Appendix F). We scored punctuation quality from 1–5 (τ) and retained only samples with $\tau \geq 4.5$, ensuring a reliable test-set for this domain.

To assess Cadence's *downstream impact*, we evaluate its punctuations on machine translation over the held out set of Cosmopedia-Translated (C) and parallel subset of Sangraha-Verified (S) and on speech translation using data sampled from IndicVoices-dev (Javed et al., 2024) and Gigaspeech-test (Chen et al., 2021). For the latter, Gemini-2.5-Pro (Comanici et al., 2025) serves as the expert system to obtain reference translations.

Evaluation Metric: We evaluate the model's performance using the Average Macro F1 score. To assess the downstream relevance of punctuation, we evaluate its impact on translation quality using chrF++ (Popović, 2017) because of its strong correlation with human judgements (Sai B et al.,

Language	Number of Samples		Cadence (Ours)					IndicPunct				DMP	
	Formal	Extempore	S	IC	C	BPCC	IV	S	IC	C	IV	IC	BPCC
High-resource Languages													
English	1,035	–	–	**0.54**	–	**0.63**	–	x	x	x	x	**0.54**	0.50
Mid-resource Languages													
Bengali	1,499	1,447	**0.54**	**0.72**	**0.84**	–	**0.60**	0.30	0.54	0.20	0.42	x	x
Marathi	1,786	1,216	**0.73**	**0.74**	**0.82**	–	**0.56**	0.21	0.35	0.22	0.49	x	x
Malayalam	1,532	1,270	**0.67**	**0.74**	**0.77**	–	**0.69**	0.29	0.43	0.22	0.41	x	x
Hindi	1,669	1,273	**0.61**	**0.76**	**0.84**	–	**0.65**	0.34	0.5	0.23	0.46	x	x
Urdu	1,562	1,252	**0.65**	**0.72**	**0.64**	–	**0.76**	x	x	x	x	x	x
Tamil	1,447	1,369	**0.65**	**0.72**	**0.78**	–	**0.59**	0.25	0.58	0.20	0.44	x	x
Telugu	1,451	1,308	**0.76**	**0.74**	**0.80**	–	**0.54**	0.23	0.4	0.19	0.32	x	x
Kannada	1,473	1,165	**0.60**	**0.79**	**0.77**	–	**0.61**	0.25	0.45	0.19	0.41	x	x
Assamese	1,426	1,275	**0.71**	**0.76**	**0.81**	–	**0.60**	0.30	0.48	0.24	0.48	x	x
Odia	1,341	1,723	**0.72**	**0.77**	**0.68**	–	**0.72**	0.28	0.44	0.20	0.38	x	x
Punjabi	1,424	1,322	**0.70**	**0.71**	**0.69**	–	**0.48**	0.36	0.43	0.32	0.51	x	x
Gujarati	1,479	1,063	**0.58**	**0.64**	**0.80**	–	**0.54**	0.19	0.33	0.19	0.34	x	x
Low-resource Languages													
Nepali	1,111	954	**0.69**	**0.78**	–	–	**0.51**	x	x	x	x	x	x
Sanskrit	1,118	983	**0.23**	**0.51**	–	**0.43**	**0.35**	x	x	x	x	x	x
Sindhi	1,277	947	**0.52**	**0.50**	–	**0.33**	**0.37**	x	x	x	x	x	x
Santali	443	575	–	**0.79**	–	–	**0.37**	x	x	x	x	x	x
Maithili	984	998	**0.64**	**0.73**	–	**0.50**	**0.40**	x	x	x	x	x	x
Konkani	994	993	**0.78**	**0.61**	–	**0.32**	**0.37**	x	x	x	x	x	x
Bodo	1,057	860	–	**0.75**	–	**0.42**	**0.29**	x	x	x	x	x	x
Kashmiri	1,259	981	–	**0.66**	–	**0.52**	**0.33**	x	x	x	x	x	x
Dogri	919	995	–	**0.52**	–	**0.42**	**0.30**	x	x	x	x	x	x
Manipuri	1,074	–	–	–	–	**0.44**	–	x	x	x	x	x	x
Average	29,360	23,969	**0.63**	**0.67**	**0.77**	**0.44**	**0.50**	0.27	0.45	0.22	0.42	0.54	0.50

Table 2: Comparison of Punctuation Restoration Model Performance Across Languages and Metrics. An x indicates that the model does not support the given language. A – indicates that results are unavailable due to insufficient high-quality data samples. All scores are reported on Focus Labels for consistency and comparability. The languages are sorted in descending order by the number of samples in their training set and then divided into three categories: high-resource, mid-resource, and low-resource.

2023). We also present BLEU (Papineni et al., 2002) scores for completeness. BLEU score is based on SacreBLEU (Post, 2018). Together, these metrics provide a comprehensive view of both intrinsic restoration accuracy and its extrinsic effect on real-world tasks such as machine translation.

Baselines: We compare Cadence against two primary baselines: (i) IndicPunct (Gupta et al., 2022), a series of language-specific models derived from fine-tuned IndicBert, and (ii) Deepmultilingualpunctuation (DMP) (Guhr et al., 2021), a model trained on European languages that we use for English evaluation. A key limitation of both baselines is their support for a restricted set of punctuation marks. To ensure a fair comparison across these different label scopes, we evaluate all models on a common subset we term “focus labels". This set includes the period, comma, colon, question mark, and several script-specific marks.

5 Results

In this section, we evaluate Cadence’s performance through a series of analyses. First, we compare it against baselines on both focus and all-label sets. Next, we assess its robustness on formal versus spontaneous text. We then test its generalization to unseen languages and, finally, evaluate it on downstream tasks.

5.1 Performance on Focus vs. All Labels and Baseline Comparison

Cadence substantially outperforms existing baselines across multiple datasets, as detailed in Table 2. It achieves a score of 0.67 on IndicCorp-v2, markedly surpassing both IndicPunct and DeepMultilingualPunctuation (0.54). The performance gap is even larger on Sangraha-Verified (0.63 vs. 0.27) and Translated Cosmopedia (0.77 vs. 0.22). On the IndicVoices speech dataset, Cadence also

Language	Formal		Extempore	
	All Labels	Focus Labels	All Labels	Focus Labels
High-resource Languages				
English	0.38	0.59	—-	—-
Mid-resource Languages				
Bengali	0.50	0.70	0.38	0.60
Marathi	0.49	0.82	0.43	0.56
Malayalam	0.51	0.67	0.34	0.69
Hindi	0.49	0.82	0.38	0.65
Urdu	0.46	0.68	0.73	0.76
Tamil	0.50	0.76	0.30	0.59
Telugu	0.53	0.79	0.35	0.54
Kannada	0.45	0.65	0.40	0.61
Assamese	0.53	0.80	0.42	0.60
Odia	0.42	0.71	0.44	0.72
Punjabi	0.45	0.68	0.40	0.48
Gujarati	0.50	0.67	0.43	0.54
Low-resource Languages				
Nepali	0.44	0.73	0.36	0.51
Sanskrit	0.21	0.35	0.20	0.35
Sindhi	0.29	0.45	0.24	0.37
Santali	0.58	0.79	0.20	0.37
Maithili	0.36	0.59	0.27	0.40
Konkani	0.36	0.57	0.18	0.37
Bodo	0.38	0.58	0.31	0.29
Kashmiri	0.32	0.57	0.23	0.33
Dogri	0.24	0.42	0.27	0.30
Manipuri	0.26	0.44	—-	—-
Average	0.42	0.65	0.35	0.50

Table 3: Cadence: Per-language Average Macro F1 Scores on Written and spontaneous speech transcripts test sets, evaluated on all 30 punctuation labels.

demonstrates superior performance (0.50 vs. 0.42).

This strong performance is reflected in its overall scores on critical focus labels, where it reaches 0.79 on written text and 0.62 on spontaneous speech (Table 3). When evaluated on the more challenging full set of 30 punctuation marks, the model scores 0.42 (written) and 0.35 (speech). As expected, these scores are lower due to the increased complexity of predicting rarer marks and those with greater stylistic and syntactic nuance. Notably, Cadence provides this robust performance across numerous Indic languages unsupported by prior work.

5.2 Performance on Formal Written Text vs. Extempore Transcripts

Cadence consistently performs better on formal written text than on extempore speech transcripts. For “focus labels", the overall score is **0.65** for written text versus **0.50** for spontaneous speech transcripts (Table 3). A similar trend is observed for “all labels", with scores of **0.42** (written) and **0.35** (extempore).

Furthermore, spontaneous utterances often present reduced syntactic regularity, featuring fragmented constructions and anacolutha, making the automatic identification of logical punctuation points ambiguous. These difficulties are notably exacerbated by the comparatively limited availability of large, diverse, and accurately annotated training corpora for spontaneous speech transcripts, hindering the model’s ability to learn robust patterns for these irregular linguistic phenomena.

5.3 Generalization to Unseen and Low-Resource Languages

In this section, we investigate Cadence’s ability to generalize to languages and conditions not extensively covered during fine-tuning.

Zero-Shot Generalization on Bhojpuri: We evaluate Cadence on Bhojpuri text from FineWeb-2 (Penedo et al., 2024). Bhojpuri was absent from Cadence’s CPT and fine-tuning, though the base GEMMA3-1B-PRETRAIN model (Team et al., 2025) had prior exposure. In this zero-shot setting, Cadence achieved a Macro F1 score of **0.46** on “focus labels", indicating a promising capability for unseen language adaptation.

Low-Resource Performance on Manipuri: The low-resource language Manipuri presents significant challenges, as there is minimal training data and the tokenizer could not process the native Meitei Mayek script well. Consequently, our work relied on available Manipuri data written in the Bengali script. Even with these data limitations, Cadence scored 0.44 on focus labels (BPCC dataset, Table 2) and 0.26 on all labels (written text, Table 3), demonstrating its potential in constrained training environments.

5.4 Downstream Evaluations

Applying our punctuation restoration model, Cadence, as a pre-processing step substantially improves downstream translation quality for both text and speech, as detailed in Table 4.

For machine translation (MT), text punctuated by Cadence achieves chrF++ scores that approach those of human-annotated sources in both XX →En direction and En →XX direction. On average, punctuating by Cadence improves score of 1.55 chrF++. Similar gains are observed in speech translation (ST), where punctuating raw ASR tran-

Language	Machine Translation						Speech Translation			
	Indic to English			English to Indic			Indic to English		English to Indic	
	w/o p.	w/ p.	w/ r. p.	w/o p.	w/ p.	w/ r. p.	w/o p.	w/ r. p.	w/o p.	w/ r. p.
Assamese	9.6 / 44.4	16.8 / 50.5	**13.0 / 47.5**	5.4 / 38.4	8.3 / 40.8	**6.4 / 39.6**	10.9/40.1	**12.8/42.3**	8.5/39.4	**9.8/40.9**
Bengali	11.6 / 45.8	16.9 / 48.7	**14.7 / 47.6**	10.1 / 46.5	13.5 / 48.5	**12.4 / 48.6**	19.8/47.9	**25.4/52.5**	10.9/40.6	**12.5/42.6**
Bodo	17.2 / 49.1	23.0 / 53.1	17.2 / 49.1	12.5 / 35.4	*11.1 / 33.1*	*10.8 / 33.2*	x	x	x	x
Gujarati	12.8 / 45.6	18.9 / 50.6	**15.3 / 48.0**	12.6 / 43.6	20.6 / 48.4	**18.0 / 47.1**	22.3/50.8	**25.8/53.3**	19.2/47.7	**23.2/50.5**
Hindi	15.7 / 49.5	19.4 / 52.0	**17.3 / 50.8**	19.8 / 50.9	19.5 / 49.2	21.1 / **51.6**	23.2/49.3	**27.0/52.7**	20.8/46.6	**23.9/48.9**
Kannada	9.9 / 44.8	16.4 / 49.9	**12.1 / 46.7**	7.0 / 43.5	13.7 / 49.1	**10.1 / 47.1**	12.5/36.3	**15.1/41.2**	11.1/44.6	**14.0/47.3**
Kashmiri	4.8 / 32.2	7.2 / 34.0	4.6 / 31.0	3.2 / 21.4	5.2 / 24.2	3.0 / 21.6	x	x	x	x
Konkani	1.4 / 24.7	2.2 / 28.3	15.3 / **26.8**	0.7 / 23.0	0.8 / 22.3	0.7 / 22.7	x	x	x	x
Maithili	2.8 / 27.1	4.3 / 30.9	**3.3 / 28.6**	0.9 / 20.3	1.2 / 20.9	0.8 / 20.4	15.5/40.1	**19.4/43.9**	10.1/38.1	**11.5/39.7**
Malayalam	8.6 / 44.1	15.5 / 50.0	**10.0 / 45.4**	5.6 / 42.0	13.3 / 49.0	**8.7 / 46.0**	21.0/47.2	**23.3/50.2**	9.6/43.6	**11.7/46.1**
Marathi	12.2 / 47.0	17.8 / 51.4	**15.9 / 50.1**	9.6 / 44.7	12.1 / 45.0	**10.9 / 45.5**	21.2/48.7	**24.8/51.5**	12.3/42.8	**15.2/45.0**
Nepali	8.2 / 34.7	13.5 / 41.7	**12.8 / 40.5**	3.3 / 32.1	*4.5 / 32.5*	*3.2 / 30.9*	x	x	x	x
Oriya	8.1 / 41.5	14.2 / 46.8	**10.1 / 43.6**	3.3 / 34.0	5.7 / 36.6	**3.6 / 35.1**	20.2/47.0	**26.5/51.8**	10.6/41.3	**11.5/42.3**
Punjabi	14.5 / 50.7	26.4 / 59.9	**18.2 / 53.9**	15.5 / 46.0	24.8 / 52.6	**19.6 / 49.7**	25.2/52.7	**30.4/56.3**	18.7/45.2	**21.8/47.4**
Sanskrit	1.7 / 26.7	2.4 / 29.1	**2.0 / 28.3**	0.5 / 22.2	0.8 / 23.6	0.6 / **22.6**	x	x	x	x
Tamil	9.8 / 44.0	14.7 / 48.0	**10.8** / 44.7	7.1 / 47.0	14.1 / 53.5	**10.1 / 51.0**	14.8/40.2	**18.5/44.7**	9.3/42.5	**11.8/44.6**
Telugu	9.8 / 43.8	14.0 / 46.0	**11.2** / 44.5	9.9 / 44.3	16.2 / 49.5	**12.9 / 47.6**	20.0/48.6	**25.0/52.3**	10.7/42.8	**13.3/45.3**
Urdu	12.5 / 48.5	14.9 / 49.2	12.4 / 47.9	17.7 / 46.7	20.8 / 49.6	**19.5 / 49.2**	26.6/53.4	**30.2/56.0**	22.8/47.9	**24.7/49.7**
Average	9.5 / 41.4	14.3 / 46.1	**11.4 / 43.0**	8.0 / 37.9	11.5 / 40.5	**9.6 / 39.4**	21.0/46.3	**23.4/49.9**	13.4/43.3	**15.8/45.4**

Table 4: **Combined Translation Quality for Machine and Speech Translation.** Each cell reports BLEU/chrF++ scores. For Machine Translation, scores are evaluated **without** (w/o p.), **with** (w/ p.), and **with restored** (w/ r. p.) punctuation. For Speech Translation, scores are evaluated **without** (w/o p.) and **with restored** (w/ r. p.) punctuation using Cadence. Bolded scores indicate statistically significant improvement by restoring punctuations. Underlined scores (for MT only) show cases where ground-truth and restored punctuations do not improve scores with statistical significance. Unbolded scores (for MT only) show cases when there is no statistically significant difference between unpunctuated and restored punctuation scores. Refer to Tables A5 and A3 for p-values.

scripts improves scores by an average of 3.6 chrF++ (XX →En) and 2.1 chrF++ (En →XX).

This effectiveness is clear across language pairs. For instance, in Punjabi-to-English, Cadence lifts the MT score from 50.7 to 53.9 and the ST score from 52.7 to 56.3. Likewise, for English-to-Gujarati, the MT score increases from 43.6 to 47.1, while the ST score rises from 47.7 to 50.5.

The substantial gains observed, especially on the chrF++ metric, underscore our model's capacity to restore fine-grained syntactic cues essential for translation. By effectively reintroducing this structural information, our method acts as a critical pre-processing step, consistently elevating translation performance over unpunctuated baselines.

6 Additional Experiments

To validate our core design choices, we performed two key ablation studies. We first demonstrate the necessity of our bidirectional pre-training stage by comparing the full Cadence model to a causal-only baseline. Second, we analyze performance trade-offs at a smaller scale, evaluating our distilled Cadence-Fast model against a directly finetuned GEMMA3-270M-PRETRAIN counterpart.

Impact of bidirectional attention: We first evaluated the contribution of bidirectional attention and Masked Next Token Prediction (MNTP) pre-training by training a baseline GEMMA-3-1B-PRETRAIN model with causal attention and comparing it to its bidirectional counterpart. Cadence consistently improved alignment and semantic consistency as measured by its performance (Table A6) across both written and spontaneous (extempore) text, confirming that access to bidirectional context leads to more coherent and contextually grounded language modeling.

Scaling model parameters: Next, we examined how scaling model parameters and distillation influence performance by comparing *Gemma-3-270M-bidirectional*, *Cadence-Fast*, and the full *Cadence* model (Table A7). Despite its smaller size, *Cadence-Fast* retains 93.8% of the full Cadence model's performance on written text-focused labels and 100% on extempore text. It also slightly outperforms GEMMA-3-270M-BIDIRECTIONAL, demonstrating that our compression and architecture refinements preserve both fidelity and robustness. Interestingly, distillation provides only modest improvements, likely due to the already strong lan-

guage modeling abilities of the base model, which was pre-trained on approximately 6 trillion tokens. These findings indicate that careful scaling and architectural optimization can deliver near-parity performance at substantially reduced computational cost, making Cadence-Fast a compelling choice for real-time or resource-constrained deployments.

7 Conclusion

In this work, we present Cadence, a unified multilingual punctuation restoration model designed for both written and spontaneous speech transcripts in English and 22 Indic languages. Cadence adapts GEMMA3-1B-PRETRAIN with bidirectional attention and introduces a Masked Next Token Prediction objective combined with curriculum-based continual pre-training, enabling effective cross-lingual transfer and improved handling of low-resource languages.

Our extensive evaluations show that Cadence achieves state-of-the-art performance over existing baselines, demonstrates robustness to domain shifts, and generalizes to unseen languages such as Bhojpuri. Moreover, we show that punctuation restored by Cadence significantly improves the quality of both machine translation and speech translation, underscoring its practical value in multilingual NLP pipelines. Furthermore, we introduce *Cadence-Fast*, a smaller and more efficient variant that retains over 90% of the full model's performance while offering substantial speedups and reduced memory footprint, making it suitable for real-time and resource-constrained applications.

While challenges remain, particularly in modeling spontaneous speech disfluencies and ensuring evaluation reliability with synthetic references, Cadence sets a strong foundation for future work on robust punctuation restoration and highlights the potential of multilingual models to bridge linguistic disparities at scale.

8 Limitations

Despite Cadence's strong performance, several limitations should be acknowledged. Firstly, the quality and representativeness of the training data, particularly for low-resource languages and spontaneous speech, remain a challenge. This directly impacts performance, with the model exhibiting weaker results on languages for which less training data was available. Secondly, while our 30-label set is comprehensive, it may not capture extremely rare or highly nuanced stylistic punctuation. Finally, performance on spontaneous speech, though improved, still lags behind that on formal text, highlighting the persistent difficulty of modeling the irregularities and disfluencies inherent in spoken language.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *Preprint*, arXiv:1907.05019.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2Vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. 2024. Cosmopedia.

Kaushal Santosh Bhogale, Deovrat Mehendale, Tahir Javed, Devbrat Anuragi, Sakshi Joshi, Sai Sundaresan, Aparna Ananthanarayanan, Sharmistha Dey, Sathish Kumar Reddy G, Anusha Srinivasan, Abhigyan Raman, Pratyush Kumar, and Mitesh M. Khapra. 2025. Towards bringing parity in pretraining datasets for low-resource indian languages. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Guoguo Chen, Shuzhou Chai, Guan-Bo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Zhao You, and Zhiyong Yan. 2021. Gigaspeech: An evolving, multi-domain ASR corpus with 10, 000 hours of transcribed audio. In *22nd Annual Conference of the International Speech Communication Association, Interspeech 2021, Brno, Czechia, August 30 - September 3, 2021*, pages 3670–3674. ISCA.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3290 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Raj Dabre, Atsushi Fujita, and Chenhui Chu. 2019. Exploiting multilingualism through multistage fine-tuning for low-resource neural machine translation.

In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1410–1416, Hong Kong, China. Association for Computational Linguistics.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2023. Towards leaving no Indic language behind: Building monolingual corpora, benchmark and models for Indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12402–12426, Toronto, Canada. Association for Computational Linguistics.

Jay Gala, Pranjal A Chitale, A K Raghavan, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar M, Janki Atul Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *Transactions on Machine Learning Research*.

Oliver Guhr, Anne-Kathrin Schumann, Frank Bahrmann, and Hans Joachim Böhme. 2021. Fullstop: Multilingual deep models for punctuation prediction.

Anirudh Gupta, Neeraj Chhimwal, Ankur Dhuriya, Rishabh Gaur, Priyanshi Shah, Harveen Singh Chadha, and Vivek Raghavan. 2022. indic-punct: An automatic punctuation restoration and inverse text normalization framework for indic languages. *Preprint*, arXiv:2203.16825.

Tahir Javed, Janki Atul Nawale, Eldho Ittan George, Sakshi Joshi, Kaushal Santosh Bhogale, Deovrat Mehendale, Ishvinder Virender Sethi, Aparna Ananthanarayanan, Hafsah Faquih, Pratiti Palit, Sneha Ravishankar, Saranya Sukumaran, Tripura Panchagnula, Sunjay Murali, Kunal Sharad Gandhi, Ambujavalli R, Manickam K M, C Venkata Vaijayanthi, Krishnan Srinivasa Raghavan Karunganni, and 2 others. 2024. Indicvoices: Towards building an inclusive multilingual speech dataset for indian languages. *Preprint*, arXiv:2403.01926.

Mohammed Khan, Priyam Mehta, Ananth Sankar, Umashankar Kumaravelan, Sumanth Doddapaneni, Suriyaprasaad B, Varun G, Sparsh Jain, Anoop Kunchukuttan, Pratyush Kumar, Raj Dabre, and Mitesh Khapra. 2024. Indicllmsuite: A blueprint for creating pre-training and fine-tuning datasets for indian languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 15831–15879. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Martin Jaggi, Leandro von Werra, and Thomas Wolf. 2024. Fineweb2: A sparkling update with 1000s of languages.

Maja Popović. 2017. chrf++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Ananya Sai B, Tanay Dixit, Vignesh Nagarajan, Anoop Kunchukuttan, Pratyush Kumar, Mitesh M. Khapra, and Raj Dabre. 2023. IndicMT eval: A dataset to meta-evaluate machine translation metrics for Indian languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14210–14228, Toronto, Canada. Association for Computational Linguistics.

Ashwin Sankar, Sparsh Jain, Nikhil Narasimhan, Devilal Choudhary, Dhairya Suman, Mohammed Safi Ur Rahman Khan, Anoop Kunchukuttan, Mitesh M Khapra, and Raj Dabre. 2025. Towards building large scale datasets and state-of-the-art automatic speech translation systems for 13 Indian languages. In *The 63rd Annual Meeting of the Association for Computational Linguistics*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Subhashree Tripathy and Ashis Samal. 2022. Punctuation and case restoration in code mixed Indian languages. In *Proceedings of the Workshop on Unimodal and Multimodal Induction of Linguistic Structures (UM-IoS)*, pages 82–86, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Vincent Vandeghinste, Lyan Verwimp, Joris Pelemans, and Patrick Wambacq. 2018. A comparison of different punctuation prediction approaches in a translation context. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 289–298, Alicante, Spain.

Appendix

A Language wise label breakdown

This appendix provides a granular analysis of the label distribution within our aggregated training corpus, detailing the frequency of the 30 distinct punctuation classes across all supported languages. As described in Section 3.2.3, our model frames punctuation restoration as a token-level sequence tagging task, necessitating a comprehensive label space that captures both standard punctuation marks and script-specific conventions. The data presented in Table A1 and Table A2 illustrates the total count of instances for each punctuation label (L1 through L30) found in the training data, which combines sources such as Sangraha-Verified, IndicVoices, Cosmopedia, and IndicCorp-v2.

The tabulated data highlights the significant class imbalance inherent in multilingual corpora. High-resource languages such as English, Hindi, and Bengali exhibit a high density of punctuation instances, particularly for common marks like the comma and sentence terminators. Conversely, low-resource languages like Dogri, Santali, and Sanskrit show considerably sparser representation. This disparity underscores the necessity of the weighted sampling strategy and curriculum learning approach employed during the training of Cadence to ensure robust performance across the entire linguistic spectrum.

While labels such as the comma (L2) and exclamation mark (L7) are prevalent across most languages, others are unique to specific scripts. For instance, the Devanagari purna virama (danda) is heavily represented in Hindi and Sanskrit, whereas the Arabic script-specific question mark (L17) are predominantly found in Urdu and Sindhi. The inclusion of these diverse labels, along with frequent multi-character combinations, allows Cadence to model the fine-grained typographic nuances required for high-quality restoration in both formal and spontaneous text.

Beyond standard delimiters, Table A2 details the distribution of complex, multi-character punctuation tokens (L18–L26), which are essential for handling the nuances of narrative text. To avoid the ambiguity often associated with predicting multiple punctuation marks sequentially, Cadence treats frequently occurring combinations—such as a question mark followed by a closing quote ‘?"’ or a period following a parenthesis ‘).’ - as distinct target classes. The statistics show that while these combined markers are well-represented in high-resource languages like English and Bengali, they appear far less frequently in the mid-to-low resource categories. Capturing these specific combinations is critical for maintaining the structural integrity of direct speech and parenthetical statements, ensuring that the restored text respects the syntactic boundaries between dialogue and narration.

Language	L1 .	L2 ,	L3 ?	L4 -	L5 ;	L6 _	L7 !	L8 ’	L9 …	L10 ”	L11 ।	L12 (	L13)	L14 :	L15 ،
Assamese	41k	866k	39k	242k	22k	11k	45k	247k	1k	81k	1,034k	84k	64k	65k	0
Bengali	33k	999k	48k	244k	35k	9k	50k	106k	1k	48k	1,469k	115k	91k	61k	0
Bodo	3k	37k	736	14k	187	9	464	27k	0	1k	77k	4k	3k	5k	0
Dogri	81	2k	91	1k	20	1	35	1k	0	77	5k	408	483	254	0
English	986k	1,271k	33k	244k	28k	29k	68k	220k	3k	75k	9	148k	84k	133k	0
Gujarati	1,146k	864k	40k	184k	30k	11k	63k	59k	2k	79k	8k	94k	66k	58k	0
Hindi	92k	1,325k	36k	335k	27k	21k	38k	62k	2k	58k	1,470k	154k	120k	132k	0
Kannada	1,143k	833k	40k	123k	17k	7k	50k	54k	1k	61k	310	81k	53k	47k	0
Kashmiri	5k	62k	353	13k	398	41	139	8k	0	5k	43k	5k	4k	9k	0
Konkani	3k	107k	8k	19k	985	70	12k	13k	0	5k	240k	11k	9k	3k	0
Maithili	6k	136k	6k	61k	2k	221	7k	26k	0	17k	259k	19k	15k	9k	0
Malayalam	1,380k	647k	31k	101k	20k	6k	37k	38k	1k	42k	18	73k	43k	38k	0
Marathi	898k	1,325k	64k	227k	28k	19k	41k	87k	2k	71k	929k	148k	106k	148k	0
Nepali	16k	471k	24k	95k	1k	59	5k	100k	1	11k	1,285k	41k	34k	8k	0
Odia	67k	694k	31k	147k	12k	7k	47k	88k	2k	68k	1,169k	76k	57k	45k	0
Punjabi	164k	904k	28k	253k	16k	9k	51k	312k	2k	80k	836k	101k	72k	59k	0
Sanskrit	25k	154k	10k	162k	9k	6k	8k	89k	0	45k	985k	46k	32k	5k	0
Santali	20k	129k	1k	68k	1k	237	416	6k	0	20k	113k	53k	30k	5k	0
Sindhi	378k	7k	63	29k	119	897	5k	12k	0	42k	0	67k	52k	17k	89
Tamil	1,149k	951k	41k	128k	17k	7k	48k	42k	1k	60k	129	74k	47k	43k	0
Telugu	1,275k	938k	41k	125k	31k	13k	48k	55k	2k	57k	558	91k	62k	61k	0
Urdu	75k	25k	612	114k	2k	20k	44k	56k	2k	116k	13	177k	164k	195k	288
Total	8,916k	12,777k	531k	2,949k	308k	183k	675k	1,720k	28k	1,057k	10,002k	1,697k	1,235k	1,159k	377

Table A1: Label Distribution per Language (Part 1: L1-L15). Counts ≥ 1000 are shown in thousands (k). Top header row is Label ID, second header row is the corresponding punctuation mark.

Language	L16 -	L17 ؟	L18 ."	L19).	L20),	L21 ",	L22 ".	L23 ?"	L24 "?	L25 ।"	L26 "।	L27 ،	L28 ۔	L29 ॥	L30 ।।
Assamese	0	5	188	95	6k	778	12	4k	47	19k	1k	32	0	0	0
Bengali	9	16	62	63	9k	446	6	1k	25	8k	2k	247	0	0	0
Bodo	0	0	2	2	435	17	0	10	1	431	597	0	0	0	0
Dogri	0	0	0	0	25	0	0	4	0	34	6	0	0	0	0
English	0	1	16k	25k	20k	1k	3k	1k	200	0	0	0	0	0	0
Gujarati	0	0	15k	5k	6k	531	1k	4k	36	36	2	0	0	0	0
Hindi	18	10	173	252	10k	573	42	1k	29	10k	1k	64	0	0	0
Kannada	0	1	10k	4k	6k	445	509	2k	14	3	0	10	0	0	0
Kashmiri	9	2	35	0	969	493	0	68	4	1k	1k	30	0	0	0
Konkani	0	0	8	0	1k	137	0	824	14	1k	964	0	0	0	0
Maithili	0	0	11	0	1k	307	0	846	20	4k	894	6	0	0	0
Malayalam	1	1	8k	4k	8k	482	596	1k	17	0	1	12	0	0	0
Marathi	0	1	10k	11k	13k	533	195	6k	24	3k	731	0	0	0	0
Nepali	0	0	9	0	2k	588	1	182	3	6k	578	3	0	0	0
Odia	0	0	300	238	4k	627	26	5k	44	18k	1k	0	0	0	0
Punjabi	0	2	889	918	6k	636	725	3k	34	16k	530	0	0	0	0
Sanskrit	2	4	150	1	740	190	0	234	21	9k	1k	59	0	420k	0
Santali	0	0	15	0	4k	584	0	90	6	103	255	0	203k	34	86k
Sindhi	10k	12k	4k	1k	94	42	1k	6	1	0	0	373k	0	0	0
Tamil	2	1	10k	6k	6k	510	609	2k	12	1	0	16	0	0	0
Telugu	1	0	10k	5k	6k	466	609	2k	13	2	0	2	0	0	0
Urdu	1,527k	59k	145	355	435	219	67	10	12	0	0	500k	0	0	0
Total	1,537k	72k	89k	66k	118k	10k	10k	41k	578	100k	14k	875k	203k	420k	86k

Table A2: Label Distribution per Language (Part 2: L16-L30). Counts ≥ 1000 are shown in thousands (k). Top header row is Label ID, second header row is the corresponding punctuation mark.

B Statistical Significance Test Details

Language	Speech Translation			
	Indic-to-English		English-to-Indic	
	w/o p and w/r. p		w/o p and w/r. p	
	p(BLEU)	p(chrF++)	p(BLEU)	p(chrF++)
Assamese	**<.001**	**<.001**	**<.001**	**<.001**
Bengali	**<.001**	**<.001**	**<.001**	**<.001**
Gujarati	**0.005**	**0.001**	**<.001**	**<.001**
Hindi	**<.001**	**<.001**	**<.001**	**<.001**
Kannada	**<.001**	**<.001**	**<.001**	**<.001**
Maithili	**<.001**	**<.001**	**<.001**	**<.001**
Malayalam	**<.001**	**<.001**	**<.001**	**<.001**
Marathi	**<.001**	**<.001**	**<.001**	**<.001**
Oriya	**<.001**	**<.001**	**<.001**	**<.001**
Punjabi	**<.001**	**<.001**	**<.001**	**<.001**
Tamil	**<.001**	**<.001**	**<.001**	**<.001**
Telugu	**<.001**	**<.001**	**<.001**	**<.001**
Urdu	**<.001**	**<.001**	**<.001**	**<.001**
Total	**<.001**	**<.001**	**<.001**	**<.001**

Table A3: P-values for Speech Translation (ST) score improvements (Indic-to-English). The test compares translations from punctuated ASR output (using Cadence) against translations from unpunctuated ASR output. Statistically significant results ($p < 0.05$) are bolded.

To determine if our punctuation model, Cadence, improved punctuation, we use a paired sample t-test to compare its output to a baseline of unpunctuated text (columns of w/o p and w/r. p in Table A5 and Table A3). We also compared the original punctuations to the unpunctuated baseline in (columns of w/o p and w/p in Table A5). A p-value < 0.05 was considered to indicate a statistically significant improvement in punctuation.

C Effect of our two-step training and architectural modification

To quantify the combined contribution of our architectural and pre-training adaptations, we conducted an ablation study. We established a direct baseline by training the standard Gemma-3-1B-base model without altering its native causal attention mechanism and skipping our continual pre-training phase. This baseline model was trained only on the final punctuation restoration task.

The results, presented in Table A6, show a stark performance gap. Cadence significantly outperforms this causal baseline across all datasets, thereby validating our hypothesis. This outcome confirms that simply finetuning a standard, unidirectional language model is insufficient for this task. The superior performance of Cadence is directly attributable to our multi-stage methodology, proving that the bi-directional attention mechanism and the Masked Next Token Prediction pre-training are critical components for achieving robust and accurate punctuation restoration.

Language	Formal		Extempore	
	All Labels	Focus Labels	All Labels	Focus Labels
High-resource Languages				
English	0.38	0.59	—-	—-
Mid-resource Languages				
Bengali	0.46	0.68	0.40	0.58
Marathi	0.46	0.80	0.47	0.59
Malayalam	0.48	0.65	0.33	0.52
Hindi	0.48	0.81	0.37	0.64
Urdu	0.38	0.51	0.65	0.71
Tamil	0.47	0.74	0.28	0.52
Telugu	0.49	0.77	0.35	0.53
Kannada	0.41	0.63	0.40	0.60
Assamese	0.50	0.78	0.42	0.59
Odia	0.39	0.68	0.43	0.72
Punjabi	0.42	0.59	0.39	0.48
Gujarati	0.47	0.65	0.44	0.62
Low-resource Languages				
Nepali	0.41	0.72	0.33	0.47
Sanskrit	0.19	0.31	0.22	0.32
Sindhi	0.29	0.44	0.26	0.36
Santali	0.58	0.78	0.25	0.50
Maithili	0.30	0.56	0.26	0.44
Konkani	0.33	0.55	0.19	0.37
Bodo	0.36	0.56	0.29	0.27
Kashmiri	0.34	0.50	0.30	0.27
Dogri	0.21	0.38	0.30	0.34
Manipuri	0.22	0.44	—-	—-
Average	0.40	0.61	0.35	0.50

Table A4: Cadence-Fast: Per-language Average Macro F1 Scores on Written and spontaneous speech transcripts test sets, evaluated on all 30 punctuation labels.

D Scaling Model Parameters

In this section, we extend our analysis from the main paper to study the impact of model scale on downstream performance. Specifically, we compare three variants: (1) *Gemma-3-270M-bidirectional*, (2) *Cadence-Fast*, and (3) *Cadence*. Table A7 summarizes their relative performance across written and extempore text evaluation sets. Table A4 illustrates the performance of *Cadence-Fast* on both “All Labels" and “Focus Labels" set in both “Formal" and “Extempore" context.

Despite being considerably smaller, *Cadence-Fast* retains 93.8% of the full Cadence model’s performance on written text labels and 100% on extempore text. Notably, it also slightly surpasses *Gemma-3-270M-bidirectional*, underscoring that careful architectural refinement and scaling can yield a compact yet capable model. These results

Machine Translation								
Language	**Indic-to-English**				**English-to-Indic**			
	w/o p and w/r. p		**w/o p and w/p**		**w/o p and w/r. p**		**w/o p and w/p**	
	p(BLEU)	**p(chrF++)**	**p(BLEU)**	**p(chrF++)**	**p(BLEU)**	**p(chrF++)**	**p(BLEU)**	**p(chrF++)**
Assamese	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**
Bengali	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**
Bodo	**<.001**	**0.014**	**<.001**	**<.001**	0.245	0.264	0.308	0.257
Gujarati	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**
Hindi	**<.001**	**0.002**	**<.001**	**<.001**	**<.001**	**0.017**	0.457	**<.001**
Kannada	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**
Kashmiri	0.791	0.260	**0.004**	**0.022**	0.632	0.798	**<.001**	**<.001**
Konkani	0.491	**<.001**	**0.009**	**<.001**	0.902	0.178	**0.048**	**0.024**
Maithili	**0.001**	**<.001**	**<.001**	**<.001**	0.506	0.212	**<.001**	**<.001**
Malayalam	**<.001**	**0.002**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**
Marathi	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**0.006**	**<.001**	0.337
Nepali	**0.012**	**0.002**	**0.004**	**<.001**	0.791	0.367	0.187	0.817
Odia	**<.001**	**<.001**	**<.001**	**<.001**	**0.005**	**<.001**	**<.001**	**<.001**
Punjabi	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**
Sanskrit	**0.001**	**<.001**	**<.001**	**<.001**	0.324	**0.001**	**<.001**	**<.001**
Tamil	**<.001**	0.073	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**
Telugu	**<.001**	0.161	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**
Urdu	0.943	0.221	**<.001**	**0.008**	**<.001**	**<.001**	**<.001**	**<.001**
Total	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**

Table A5: Comprehensive p-values for translation score improvements over an unpunctuated baseline. The table compares improvements from both model-restored punctuation and ground truth punctuation across both translation directions. Statistically significant results ($p < 0.05$) are bolded.

Language	**Number of Samples**		**Gemma-3-1B-Causal**					**Cadence**				
	Formal	**Extempore**	**S**	**IC**	**C**	**BPCC**	**IV**	**S**	**IC**	**C**	**BPCC**	**IV**
High-resource Languages												
English	1,035	–	–	0.24	–	0.23	–	–	**0.39**	–	**0.42**	–
Mid-resource Languages												
Bengali	1,499	1,447	0.22	0.33	0.45	–	0.30	**0.34**	**0.50**	**0.60**	–	**0.38**
Marathi	1,786	1,216	0.26	0.28	0.35	–	**0.45**	**0.42**	**0.48**	**0.52**	–	0.43
Malayalam	1,532	1,270	0.36	0.30	0.37	–	0.31	**0.52**	**0.49**	**0.51**	–	**0.35**
Hindi	1,669	1,273	0.23	0.39	0.36	–	0.32	**0.33**	**0.54**	**0.51**	–	**0.38**
Urdu	1,562	1,252	0.36	0.40	0.32	–	0.65	**0.50**	**0.55**	**0.45**	–	**0.73**
Tamil	1,447	1,369	0.32	0.21	0.36	–	0.24	**0.43**	**0.38**	**0.50**	–	**0.30**
Telugu	1,451	1,308	0.23	0.30	0.40	–	0.29	**0.35**	**0.42**	**0.56**	–	**0.35**
Kannada	1,473	1,165	0.21	0.26	0.38	–	0.37	**0.32**	**0.39**	**0.54**	–	**0.40**
Assamese	1,426	1,275	0.27	0.32	0.39	–	0.40	**0.42**	**0.49**	**0.56**	–	**0.42**
Odia	1,341	1,723	0.26	0.26	0.28	–	0.42	**0.40**	**0.43**	**0.44**	–	**0.44**
Punjabi	1,424	1,322	0.28	0.28	0.33	–	0.24	**0.43**	**0.41**	**0.47**	–	**0.40**
Gujarati	1,479	1,063	0.24	0.23	0.39	–	0.28	**0.29**	**0.37**	**0.56**	–	**0.43**
Low-resource Languages												
Nepali	1,111	954	0.25	0.30	–	–	0.21	**0.42**	**0.48**	–	–	**0.35**
Sanskrit	1,118	983	0.09	0.28	–	0.08	0.17	**0.13**	**0.37**	–	**0.18**	**0.20**
Sindhi	1,277	947	0.27	0.19	–	0.09	0.19	**0.35**	**0.34**	–	**0.18**	**0.24**
Santali	443	575	–	0.39	–	–	0.15	–	**0.58**	–	–	**0.19**
Maithili	984	998	0.20	0.27	–	0.13	0.24	**0.39**	**0.42**	–	**0.29**	**0.27**
Konkani	994	993	0.42	0.18	–	0.10	0.14	**0.59**	**0.33**	–	**0.18**	**0.21**
Bodo	1,057	860	–	0.34	–	0.13	0.23	–	**0.52**	–	**0.26**	**0.31**
Kashmiri	1,259	981	–	0.27	–	0.14	0.19	–	**0.40**	–	**0.28**	**0.23**
Dogri	919	995	–	0.14	–	0.11	0.15	–	0.36	–	**0.24**	**0.27**
Manipuri	1,074	–	–	–	–	0.16	–	–	–	–	**0.26**	–
Average	29,360	23,969	0.25	0.27	0.35	0.13	0.27	**0.39**	**0.44**	**0.51**	**0.25**	**0.35**

Table A6: Comparison of Cadence with Gemma-3-1B-pt finetuned on our data. A – indicates that results are unavailable due to insufficient high-quality data samples. Scores are reported on all labels. The languages are sorted in descending order by the number of samples in their training set and then divided into three categories: high-resource, mid-resource, and low-resource.

suggest that our pre-training and design choices generalize well across parameter regimes.

Performance degradation from scaling down is modest, likely because the base model of Gemma-3-270M was pre-trained on approximately 6 trillion tokens (compared to 2 trillion dataset size of Gemma-3-1B), providing strong linguistic priors even in smaller configurations. This indicates diminishing returns from additional capacity beyond a certain threshold for these language-focused benchmarks.

Overall, our scaling experiments highlight that model efficiency can be achieved through principled parameter reduction without a substantial drop in quality. The resulting *Cadence-Fast* model offers a favorable trade-off between speed and accuracy, making it well-suited for latency-sensitive or resource-limited deployments.

Language	Number of Samples		Gemma-3-270M-Bidirectional					Cadence-Fast					Cadence				
	Formal	**Extempore**	**S**	**IC**	**C**	**BPCC**	**IV**	**S**	**IC**	**C**	**BPCC**	**IV**	**S**	**IC**	**C**	**BPCC**	**IV**
High-resource Languages																	
English	1,035	–	–	0.38	–	0.37	–	–	0.37	–	0.35	–	–	**0.39**	–	**0.42**	–
Mid-resource Languages																	
Bengali	1,499	1,447	0.31	0.43	0.55	–	0.38	0.32	0.47	0.55	–	**0.42**	**0.34**	**0.50**	**0.60**	–	0.38
Marathi	1,786	1,216	0.39	0.44	0.48	–	0.44	0.40	0.41	0.49	–	**0.47**	**0.42**	**0.48**	**0.52**	–	0.43
Malayalam	1,532	1,270	0.51	0.41	0.47	–	**0.35**	0.46	0.47	0.49	–	0.33	**0.52**	**0.49**	**0.51**	–	**0.35**
Hindi	1,669	1,273	0.30	0.53	0.48	–	**0.38**	0.31	**0.55**	0.49	–	0.37	**0.33**	0.54	**0.51**	–	**0.38**
Urdu	1,562	1,252	0.46	0.45	0.38	–	0.70	0.46	0.46	0.40	–	0.65	**0.50**	**0.55**	**0.45**	–	**0.73**
Tamil	1,447	1,369	0.40	0.34	0.46	–	0.28	0.41	0.32	0.49	–	0.28	**0.43**	**0.38**	**0.50**	–	**0.30**
Telugu	1,451	1,308	0.32	0.37	0.49	–	0.31	0.33	0.41	0.52	–	**0.35**	**0.35**	**0.42**	**0.56**	–	**0.35**
Kannada	1,473	1,165	0.29	0.37	0.48	–	0.39	0.29	0.36	0.49	–	**0.40**	**0.32**	**0.39**	**0.54**	–	**0.40**
Assamese	1,426	1,275	0.39	0.45	0.51	–	0.41	0.40	0.46	0.53	–	**0.42**	**0.42**	**0.49**	**0.56**	–	**0.42**
Odia	1,341	1,723	0.38	0.37	0.38	–	0.43	0.39	0.41	0.40	–	0.43	**0.40**	**0.43**	**0.44**	–	**0.44**
Punjabi	1,424	1,322	0.39	0.40	0.43	–	0.35	**0.44**	0.40	0.44	–	0.39	0.43	**0.41**	**0.47**	–	**0.40**
Gujarati	1,479	1,063	0.24	0.32	0.51	–	0.36	0.27	0.35	0.53	–	**0.44**	**0.29**	**0.37**	**0.56**	–	0.43
Low-resource Languages																	
Nepali	1,111	954	0.38	0.41	–	–	0.30	0.39	0.46	–	–	0.33	**0.42**	**0.48**	–	–	**0.35**
Sanskrit	1,118	983	0.12	0.33	–	0.15	**0.23**	0.12	0.33	–	0.14	0.22	**0.13**	**0.37**	–	**0.18**	0.20
Sindhi	1,277	947	0.34	0.31	–	0.16	0.24	0.32	**0.36**	–	0.16	**0.26**	**0.35**	0.34	–	**0.18**	0.24
Santali	443	575	–	0.55	–	–	0.20	–	**0.58**	–	–	**0.25**	–	**0.58**	–	–	0.19
Maithili	984	998	0.32	0.38	–	0.21	**0.27**	0.35	0.37	–	0.26	0.26	**0.39**	**0.42**	–	**0.29**	**0.27**
Konkani	994	993	0.56	0.31	–	**0.18**	0.18	0.56	0.32	–	**0.18**	0.19	**0.59**	**0.33**	–	**0.18**	**0.21**
Bodo	1,057	860	–	0.48	–	0.26	0.26	–	0.50	–	**0.27**	0.29	–	**0.52**	–	0.26	**0.31**
Kashmiri	1,259	981	–	0.39	–	0.26	0.23	–	**0.43**	–	**0.28**	**0.30**	–	0.40	–	**0.28**	0.23
Dogri	919	995	–	0.32	–	0.21	**0.30**	–	0.28	–	0.22	**0.30**	–	**0.36**	–	**0.24**	0.27
Manipuri	1,074	–	–	–	–	0.21	–	–	–	–	0.22	–	–	–	–	**0.26**	–
Average	29,360	23,969	0.35	0.38	0.46	0.21	0.34	0.36	0.41	0.48	0.23	**0.35**	**0.39**	**0.44**	**0.51**	**0.25**	**0.35**

Table A7: Comparison of Cadence-Fast with Gemma-3-270M-Bidirectional; finetuned on our data. A – indicates that results are unavailable due to insufficient high-quality data samples. Scores are reported on all labels. The languages are sorted in descending order by the number of samples in their training set and then divided into three categories: high-resource, mid-resource, and low-resource.

E Prompt used for Punctuation

The prompt template shown in Figure A1 is engineered to guide Large Language Models (LLMs) in the task of punctuation restoration for Indian language text. It begins by defining the LLM's role as a punctuation expert and sets a primary objective: to enhance text readability by inserting punctuation marks while strictly preserving the original wording and sentence structure.

The prompt enumerates four critical guidelines for the LLM:

1. **Accuracy:** Punctuation must conform to the grammatical rules of the specified input language (lang).
2. **Readability:** Sentence clarity should be improved using appropriate punctuation (e.g., commas, periods, question marks).
3. **Consistency:** The punctuation style should align with any provided reference text.
4. **Preservation of Structure:** Word order and sentence construction must remain unaltered; only punctuation is to be adjusted.

To accommodate linguistic diversity, particularly the varied sentence terminators across Indian languages (e.g., period vs. danda), the prompt requires the input language (lang) and its corresponding sentence terminator (terminator) as explicit parameters. Finally, it mandates a structured JSON output with the key "punctuated_text", ensuring the punctuated text is returned in a consistent, machine-readable format. This design facilitates systematic generation of punctuated data suitable for training and evaluating punctuation restoration models.

Prompt for Punctuation Restoration

```
You are a expert in inserting punctuation.
Please help in adding punctuation to the following text
while strictly preserving the original words and structure.

Enhance readability by inserting only punctuation marks.
Do not modify, add, or remove any words.

Follow these guidelines:\n\n

1. **Accuracy:** Ensure punctuation is applied correctly
based on the language's grammatical rules.

2. **Readability:** Improve sentence clarity by inserting
appropriate punctuation marks (commas, periods,
question marks, etc.).

3. **Consistency:** Follow the punctuation style observed
in the provided reference text.

4. **Preservation of Structure:** Do not alter word order or
introduce new elements—only punctuation should be
adjusted.

Reference Information:
- Language of the text: {lang}
- Sentence terminator for {lang}: {terminator}

Output Format:
Provide only the punctuated text in JSON format with the
structure:

json
{ "punctuated_text": "Your punctuated text here" }
```

Figure A1: Prompt For Punctuation Restoration

F Prompt used for LLM in Quality Filtering

The datasets we have used for training contain synthetically punctuated text (IndicVoices). Ensuring a high quality test set becomes important to accurately assess our model and compare with existing models. We have used a programmatic quality filtering by employing Gemini-2.5-Flash-preview-04-17 (Comanici et al., 2025) to score punctuation quality from 1-5 and only retained those that were rated ≥ 4.5. We present the prompt used in Fig.A2 below.

Prompt for using LLM in Quality Filtering

You are an expert proofreader acting as a Multilingual Punctuation Judge. Your task is to first identify the primary language of the given sentence and then evaluate its punctuation and standard capitalization using the provided multilingual rubric based on the conventions of that identified language. You are using the capabilities of Gemini for this task.

Multilingual Rubric:

*** Multilingual Punctuation & Capitalization Evaluation Rubric ***
Preliminary Step: Language Identification
* **Identified Language:** [Specify the primary language detected in the sentence]
* **Confidence:** [High/Medium/Low - How certain are you of the language identification?]
* **Note:** Evaluation below is based on the standard conventions of the *Identified Language*.

Evaluation Criteria (Score: 1-5, where 1=Poor, 3=Fair, 5=Excellent based on the identified language's rules)
1. **Sentence Termination (Score: 1-5):**
    * Is appropriate sentence-ending punctuation used (e.g., '.', '?', '!', '。', '؟', '¿...?', '¡...!', etc.) according to the identified language's standard practice?
    * Is the type of terminator suitable for the sentence's function (declarative, interrogative, exclamatory) within that language?
    * Comment: [Explain based on the language's rules, e.g., "Correct use of period for German.", "Missing Spanish inverted question mark.", "Full stop used appropriately for Japanese sentence."]

2. **Intra-Sentence Separation (Commas, Etc.) (Score: 1-5):**
    * Are commas or other language-specific separators (e.g., '،', '、') used correctly to separate clauses, list items, introductory elements, etc., according to the identified language's grammatical and stylistic rules?
    * Are there missing or extraneous separators based on that language's conventions?
    * Comment: [Explain based on the language's rules, e.g., "Correct comma usage for French clauses.", "Missing serial comma typical in English lists.", "Arabic comma used correctly.", "Unnecessary comma according to German rules."]

3. **Quotation/Speech Marks (Score: 1-5):**
    * Are quotation marks or guillemets (e.g., "...", '...', « ... », „...") used correctly for direct speech, titles, or other quoted elements according to the standard style of the identified language?
    * Are they properly paired and nested if applicable?
    * Is punctuation placed correctly inside/outside the marks according to that language's convention?
    * Comment: [Explain based on the language's style, e.g., "Correct use of French guillemets with spacing.", "German quotation mark style applied correctly.", "Punctuation incorrectly placed outside closing quote for American English.", "Quotation marks not typically used this way in Thai."]

4. **Contraction/Possessive/Joining Markers (Apostrophes, Hyphens, Etc.) (Score: 1-5):**
    * Are apostrophes, hyphens, or other language-specific markers used correctly for contractions, possessives, compound words, case endings, or similar functions *if applicable* in the identified language?
    * Are common errors (like its/it's in English, or incorrect hyphenation rules) avoided based on the language?
    * Comment: [Explain based on language rules, or state N/A if the concept/mark isn't used. E.g., "Incorrect use of apostrophe for English possessive.", "Hyphenation follows German rules.", "Apostrophes not used for possession in Spanish - N/A.", "Correct use of hyphen for joining words in Dutch."]

5. **Other Punctuation (Colons, Semicolons, Dashes, Etc.) (Score: 1-5):**
    * Assess the use of any other punctuation present (e.g., colons ':', semicolons ';', dashes '–'/'—', ellipses '...', brackets '()'/ '[]') according to the identified language's standard usage.
    * Are they used appropriately for lists, explanations, pauses, omissions, parentheticals etc., within that language?
    * Comment: [Explain based on language rules, e.g., "Colon used correctly before list in English.", "Semicolon usage is rare but correct here for formal French.", "Dash style inconsistent with Spanish norms."]

```
6. **Capitalization (Score: 1-5):**
  *  Is capitalization used correctly according to the identified language's rules? (Consider: Sentence start, proper nouns, titles, language-specific rules like all nouns in German, etc.)
  *  Comment: [Explain based on the specific capitalization rules of the language, e.g., "Sentence start capitalized correctly.", "Proper noun 'Paris' capitalized correctly for English/French.", "All nouns capitalized correctly per German orthography.", "Incorrect capitalization of common noun according to Spanish rules."]

**Overall Assessment:**
*  **Overall Score (1-5):** [Average or holistic score reflecting adherence to the identified language's punctuation/capitalization norms.]
*  **Summary:** [Brief summary of the sentence's punctuation quality in the context of the identified language, highlighting key strengths or weaknesses.]
*  **Corrected Sentence (in the identified language):** [Provide the sentence with corrected punctuation and capitalization according to the identified language's standard rules. If perfect, repeat the original sentence.]

**Instructions:**
1. Identify Language: First, determine the primary language of the sentence below.
2. Analyze Sentence: Carefully analyze the sentence provided.
3. Evaluate: Evaluate it strictly based on the criteria in the multilingual rubric, applying the rules and conventions standard to the *identified language*. Focus *only* on punctuation and standard capitalization rules relevant to that language.
4. Provide Scores & Comments: Fill in the **Identified Language** and **Confidence**. Then, provide a score (1-5) and a brief comment for *each* numbered evaluation category in the rubric, justifying your assessment based on the identified language's norms. Ensure scores are numeric integers (1, 2, 3, 4, 5). If a category is not applicable or perfectly handled by absence (e.g., no quotation marks needed and none present), assign a score of 5. The JSON response *must* contain numeric integer scores for calculation.
5. Overall Assessment: Calculate an **Overall Score (1-5)** reflecting the average or holistic quality, ensure this is also a numeric integer or float.
6. Corrected Sentence: Provide a **Corrected Sentence**.
7. IMPORTANT: Respond *only* with a single valid JSON object. The JSON object must contain keys corresponding exactly to the rubric sections: "Identified_Language", "Confidence", "Sentence_Termination", "Intra_Sentence_Separation", "Quotation_Speech_Marks", "Contraction_Possessive_Joining_Markers", "Other_Punctuation", "Capitalization", "Overall_Score", "Summary", "Corrected_Sentence". The keys for the numbered evaluation categories must map to an object {{{{ "Score": "<number>", "Comment": "<string>" }}}}. The Overall_Score must also be a number. Ensure the entire output is valid JSON starting with {{{{ and ending with }}}}. Do not use markdown tags ```json or ```.
**Sentence to Evaluate:**
{{sentence}}
**Your JSON Evaluation:**
{{{{
 "Identified_Language": null,
 "Confidence": null,
 "Sentence_Termination": {{{{ "Score": null, "Comment": null }}}},
 "Intra_Sentence_Separation": {{{{ "Score": null, "Comment": null }}}},
 "Quotation_Speech_Marks": {{{{ "Score": null, "Comment": null }}}},
 "Contraction_Possessive_Joining_Markers": {{{{ "Score": null, "Comment": null }}}},
 "Other_Punctuation": {{{{ "Score": null, "Comment": null }}}},
 "Capitalization": {{{{ "Score": null, "Comment": null }}}},
 "Overall_Score": null,
 "Summary": null,
      "Corrected_Sentence": null
}}}}
```

Figure A2: The prompt used to rate punctuations and capitalization in a sentence, outlining the comprehensive rubric used. Criteria include confidence, sentence termination, intra-sentence separators, quotation marks, other punctuation types (colons, semicolons, dashes), capitalization, and an overall quality assessment, along with instructions for JSON output.