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

Anonymous authors
Paper under double-blind review

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 Indian languages 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 impairs readability and degrades the performance of downstream NLP tasks like Machine Translation (MT) and Text Summarization.

While punctuation restoration has progressed for high-resource languages like English (Devlin et al., 2019), 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 danda. Prior efforts were often limited in language or punctuation scope, or used non-scalable, language-specific models, hindering cross-lingual generalization, particularly for under-represented languages (Gupta et al., 2022).

To address this gap for Indic languages, we introduce Cadence, a robust multilingual punctuation restoration model. First, we construct a large and diverse fine-tuning corpus from 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), to cover both formal written text and ASR transcripts while balancing linguistic representation. Second, we adapt a Gemma-1B model into a bidirectional transformer using a Masked Next Token Prediction (MNTP) objective (BehnamGhader et al., 2024). This allows for efficient, non-autoregressive sequence tagging over a fine-grained set of 30 punctuation classes, including Indic-specific symbols. Cadence supports English and all 22 scheduled languages of India, achieving new state-of-the-art performance that surpasses existing baselines. We release our model to empower downstream NLP tasks like machine translation and speech processing, particularly for under-resourced Indic languages. Our contributions include this carefully curated multilingual corpus and the adapted Gemma-

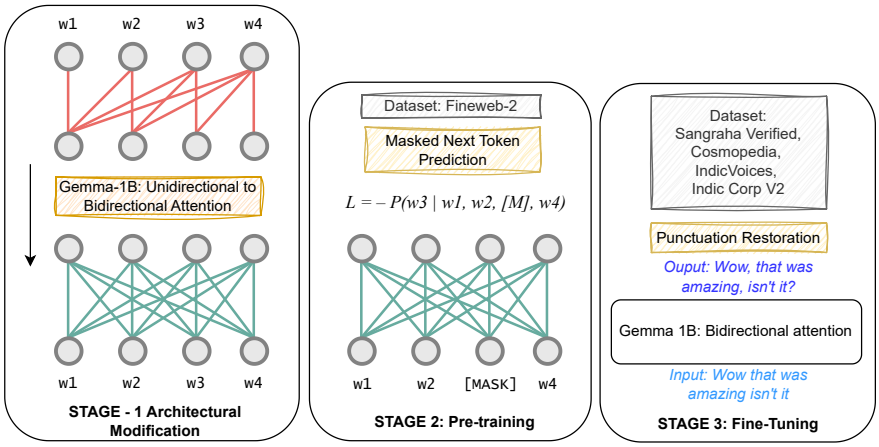

Figure 1: Overview of our training methodology. *Stage-1:* Modify causal attention to bidirectional attention. *Stage-2:* Pre-train with Masked Next Token Prediction Objective. *Stage-3:* Train for punctuation restoration, as a token-level classification task. Figure inspired from BehnamGhader et al.,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

based model, which offers a powerful and scalable solution for comprehensive punctuation restoration.

Cadence supports English and all 22 scheduled languages of India: Assamese, Bengali, Bodo, Dogri, Gujarati, Hindi, Kannada, Kashmiri, Konkani, Maithili, Malayalam, Manipuri, Marathi, Nepali, Odia, Punjabi, Sanskrit, Santali, Sindhi, Tamil, Telugu, and Urdu.

In summary, our key contributions are: **(i)** the creation of an extensive multilingual Indic punctuation corpus, carefully curated from diverse sources to address data scarcity and improve linguistic representation for low-resource languages and **(ii)** we present an adapted Gemma3-based model, transformed into an efficient bidirectional sequence tagger via an MNTP objective, capable of restoring a comprehensive set of 30 punctuation classes, including Indic-specific symbols and common combinations.

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 & Samal, 2022; Gupta et al., 2022). Gupta et al. (2022) introduced IndicPunct, a multilingual transformer model 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: State-of-the-art PR systems often use BERT-style token classifiers (Gupta et al., 2022; Guhr et al., 2021), with Large Language Models (LLMs) recently gaining traction (Sankar et al., 2025). Earlier models, trained mainly on clean written text, struggle with disfluent spontaneous speech, impairing real-

world ASR and ST applications. A key bottleneck is the scarcity of large, high-quality, punctuation-annotated corpora reflecting speech characteristics. While large multilingual text corpora like those by (Penedo et al., 2024; Doddapaneni et al., 2023) support training for diverse Indic languages, they mostly contain formal or web text. Resources such as IndicVoices (Javed et al., 2024), though unpunctuated, reveal stylistic phenomena PR models must address. However, non-autoregressive, generalist PR models for Indic languages that support large label sets and are robust across written and spoken styles remain rare. Cadence addresses this gap with a scalable, multilingual, LLM-based approach for varied domains and languages.

3 Methodology

3.1 Data Strategy for Multilingual Punctuation Restoration

Our methodology is founded on a two-pronged data strategy, employing distinct, large-scale corpora for the continual pre-training and task-specific fine-tuning phases.

Pre-training Data Corpus: For continual pre-training, we use large multilingual web corpora. This provides the model with broad exposure to general-domain text, helping it to build foundational representations that are adaptable across diverse linguistic contexts and writing styles.

Fine-tuning Data Amalgamation: For task-specific fine-tuning, we constructed a substantial and heterogeneous dataset by amalgamating text from numerous sources with varied domains and styles. This corpus intentionally includes both formal written text and less structured transcripts of spontaneous speech. The inclusion of spoken language data, with its characteristic disfluencies and fragmented syntax, is crucial for ensuring the model is robust and performs well on a wide spectrum of 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 is highly beneficial. Therefore, we adapt the model's attention mechanism to be fully bidirectional 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 Objective: We employ a *Masked Next Token Prediction* (MNTP) objective (BehnamGhader et al., 2024). In this setup, the model is trained to predict a masked token at position $i + 1$ using the contextual representation of the token at position i.

Crucially, the model employs bi-directional attention. This means the representation of token i (which serves as the basis for predicting token $i + 1$) is itself informed by the entire unmasked sequence, including tokens both preceding and succeeding token i. Despite

access to this broader context, the objective's design hones its ability to learn strong local dependencies between adjacent tokens; a skill highly relevant for punctuation prediction.

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:

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

2. **Expansion Phase 1 (Mid-to-High Resource):** The model is then exposed to a group of mid-to-high-resource languages. This phase allows the model to begin generalizing across related linguistic structures and benefit from these larger datasets.

3. **Expansion Phase 2 (Low Resource):** Subsequently, lower-resource languages are introduced. 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 the considered languages. This phase aims to consolidate learning across the entire linguistic spectrum and mitigate potential catastrophic forgetting of earlier-learned languages or features.

3.2.3 *Task-Specific Fine-tuning for Punctuation Restoration*

The final stage fine-tunes the model specifically for punctuation restoration. We frame this as a token-level sequence classification task, where for each token in an unpunctuated sequence, the model predicts the punctuation mark that should follow it (or a special "O" label for no punctuation). For this, the original language modeling head is replaced with a linear classification layer. To address data imbalance in the fine-tuning corpus, we employ a weighted sampling strategy that oversamples data from low-resource languages, promoting more equitable learning and robust performance across all target languages.

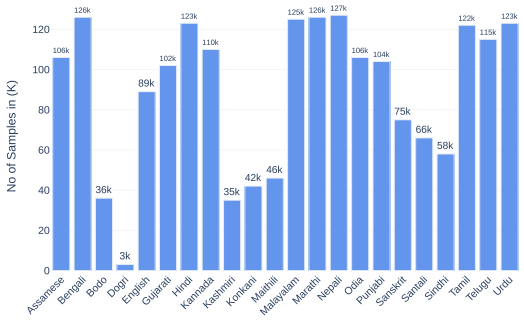

Figure 2: Statistics of our training corpus, showing the number of entries available for each supported language, represented in thousands.

4 Experimental Setup

This section outlines the datasets, training procedures, and evaluation metrics used to develop and assess 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.

Label ID	Punctuation Mark	Instances	Label ID	Punctuation Mark	Instances
1	.	8,916k	16	۔	1,537k
2	,	12,777k	17	؟	72k
3	?	531k	18	."	89k
4	-	2,949k	19	).	66k
5	;	308k	20	),	118k
6	_	183k	21	",	10k
7	!	675k	22	".	10k
8	'	1,720k	23	?"	41k
9	…	28k	24	"?	578
10	"	1,057k	25	।"	100k
11	।	10,002k	26	"।	14k
12	(	1,697k	27	،	875k
13	)	1,235k	28	۔	203k
14	:	1,159k	29	॥	420k
15	،	377	30	।।	86k

Table 1: Breakdown of supported punctuation marks, their internal Label IDs, and the number of instances for each in our training corpus, represented in thousands. For language wise breakdown, refer to Appendix A.

4.1.1 *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.

4.1.2 *Fine-tuning Datasets for Punctuation Restoration*

To fine-tune our model effectively, we constructed a multilingual training corpus by aggregating data from four diverse sources, each contributing complementary strengths. **Sangraha-Verified** provides high-quality, accurately punctuated formal text (Khan et al., 2024); **IndicVoices-ST** offers punctuated transcripts of spontaneous speech, capturing spoken language patterns (Sankar et al., 2025); the **Translated Cosmopedia** dataset introduces syntactically varied, structured knowledge content (Ben Allal et al., 2024); and **IndicCorp-v2** contributes wide-domain natural language text with rich punctuation usage (Doddapaneni et al., 2023). This combination ensures broad linguistic coverage and stylistic diversity. Dataset composition and statistics are detailed in Figure 1.

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 was originally designed as a causal decoder for text generation, punctuation restoration benefits from access to bidirectional context. We modify the GEMMA-3-1B's attention mechanism to attend to both left and right contexts, thus making it bidirectional.

4.2.2 *Continual Pretraining*

Curriculum Learning Strategy: Given the wide variation in data availability and linguistic structure across Indic languages, we employ a four-phase curriculum learning strategy:

Phase 1: English only – Initializes the model with a high-resource language to establish stable representations. masking ratio was set to 0.30.

Language	Number of Samples		Cadence (Ours)					IndicPunct				DMP	
	Formal	Extempore	S	IC	C	BPCC	IV	S	IC	C	IV	IC	BPCC
Nepali	1,111	954	**0.69**	**0.78**	–	–	**0.51**	x	x	x	x	x	x
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**	–	**0.74**	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
English	1,035	–	–	0.54	–	**0.63**	–	x	x	x	x	0.54	0.50
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
Overall	29,360	23,969	**0.68**	**0.76**	**0.78**	**0.45**	**0.60**	0.31	0.54	0.26	0.54	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, when evaluated on the test set. 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.

Phase 2: High- and mid-resource Indic languages – Introduces 13 languages, which includes Hindi, Telugu, Tamil, Bengali, Malayalam, Marathi, Kannada, Gujarati, Assamese, Oriya, Punjabi, Sindhi, Urdu. 0.25 masking ratio was employed.

Phase 3: Low-resource Indic languages – Adds the languages: Bodo, Dogri, Konkani, Kashmiri, Maithili, Manipuri, Nepali, Sanskrit, Santali, encouraging generalization. Masking ratio was 0.15.

Phase 4: Mixed multilingual training – We train on all 23 languages (22 Indic languages + English) for the final 10% of steps to consolidate knowledge and mitigate catastrophic forgetting. 0.25 masking ratio was used.

This staged progression allows the model to incrementally adapt to increasing linguistic diversity while maintaining stability across training phases.

4.2.3 Finetuning

The label set, described in Table 1, covers standard English punctuation, Indic-specific and Urdu-script marks, and frequent multi-character combinations. This setup enables the model to capture stylistic and orthographic variation across languages and domains.

We trained Cadence using the AdamW optimizer (Loshchilov & Hutter, 2017) with a max learning rate of $2e-4$ with 10% of the training steps as warmup followed by a cosine decay to $1e-6$. The effective batch size was 64 and the model was trained on 8×H100 GPUs for 8 hours.

4.3 Evaluation

Test Set: We held out a test set from IndicCorp-v2, Sangraha-Verified, translated Cosmopedia, and IndicVoices (spontaneous speech), supplemented with the BPCC dataset (Gala et al., 2023). Since some portions of this data are machine-generated, we implemented a rigorous quality control process. Manual verification is prohibitively costly and time-intensive, especially for low-resource languages where expert evaluators are scarce. Therefore, we employed Google's Gemini 2.5 Flash model as an automated judge. Using a rubric-based prompt (Appendix C) to assess grammatical correctness and contextual appropriateness, each instance was assigned a quality score from 1 to 5. Only examples scoring 4.5 or higher were retained, ensuring our final benchmark is diverse and meets a high-quality standard.

Evaluation Metric: We evaluate performance using the Average Macro F1 score. This metric computes the F1 score for each punctuation class independently and then averages them, giving equal weight to each class. This is ideal for handling the imbalanced class distributions common in punctuation restoration.

Baselines: We compare Cadence with two key baselines: **(i)** *IndicPunct* (Gupta et al., 2022): A series of language-specific models based on IndicBert, supporting a limited set of languages and punctuation marks (sentence-end, question mark, comma). **(ii)** *Deepmultilingualpunctuation (DMP)* (Guhr et al., 2021): A model trained on European languages, which we use as a baseline for English. It also supports a limited label space (period, question mark, comma, hyphen, colon).

Since these baselines support different and more limited sets of punctuation, we established a common set of "focus labels" for fair comparison. This set includes the period (.), comma (,), colon (:), question mark (?), and script-specific marks such as the Devanagari danda, Urdu full stop, and Santali mucaad.

5 Results

This section evaluates Cadence's performance. We first compare its efficacy on a defined set of "focus labels" versus all supported punctuation labels, alongside a comparison with baseline models. We then analyze its performance across formal written text and spontaneous extempore transcripts. This is followed by an examination of the relationship between training data volume and performance, and finally, its generalization capabilities to unseen languages.

5.1 Performance on Focus vs. All Labels and Baseline Comparison

Cadence demonstrates strong performance on critical punctuation, achieving an F1 score of **0.79** on written text and **0.62** on spontaneous speech for "focus labels" (Table 3). As expected, performance on the full set of 30 supported labels is lower, reflecting the increased complexity of predicting rarer and more stylistic marks. Crucially, Cadence substantially outperforms existing baselines across all evaluated datasets (Table 2). On the large-scale IndicCorp-v2, it achieves an F1 of **0.76**, a significant leap from the **0.54** score of both IndicPunct and DeepMultilingualPunctuation. Similar gains are observed on Sangraha-Verified (**0.68** vs. **0.31**) and Translated Cosmopedia (**0.78** vs. **0.26**). It also surpasses baselines on noisier speech data from IndicVoices (**0.60** vs. **0.54**) and the English BPCC dataset (**0.63** vs. **0.50**). Unlike the baselines, Cadence's support for a wide range of Indic languages demonstrates its broader utility and scalability.

5.2 Performance on Formal Written Text vs. Extempore Transcripts

Cadence consistently performs better on formal written text than on spontaneous speech transcripts. For "focus labels," the overall F1 score is **0.79** for written text versus **0.62** for speech (Table 3). This gap is expected, as spontaneous speech is characterized by greater syntactic irregularity, fragmented constructions, and disfluencies, making punctuation

Language	Written		Extempore	
	All Labels	Focus Labels	All Labels	Focus Labels
Nepali	0.44	0.73	0.36	0.51
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
English	0.38	0.59	—	—
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	—	—
Total	**0.59**	**0.79**	**0.45**	**0.63**

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

prediction inherently more ambiguous. The challenge is compounded by the relative scarcity of accurately annotated spontaneous speech corpora for training.

5.3 Generalization to Unseen and Low-Resource Languages

We evaluated Cadence's ability to generalize to languages with little to no fine-tuning data. We tested on Bhojpuri text, a language absent from our fine-tuning pipeline, Cadence achieved a Macro F1 score of **0.46** on "focus labels." This demonstrates a promising capability for zero-shot adaptation to new languages, likely inherited from the base model's pre-training exposure. Manipuri also served as a low-resource test case, further complicated by the need to transliterate it into the Bengali script due to tokenizer limitations. Despite these constraints, Cadence achieved a respectable F1 score of **0.44** on "focus labels", underscoring its utility in challenging, data-scarce scenarios.

5.4 Impact of Punctuations on Downstream Tasks

To assess the downstream relevance of punctuation, we evaluated its impact on machine translation (MT) quality. Using parallel corpora where the target side remains punctuated and the source side is either punctuated or unpunctuated, we measured translation performance across BLEU, and chrF++ metrics. Results, summarized in Table 4, show that punctuating the source consistently improves translation quality across both translation directions.

Across both translation directions, the presence of punctuation leads to consistent improvements in translation quality for all languages and metrics. In the Indic-to-English direction, BLEU scores generally increase with punctuation, with larger gains for some languages such as Punjabi (14.47 $\rightarrow$ 26.37) and Bengali (11.60 $\rightarrow$ 16.91). Improvements in the English-to-Indic direction are smaller on average but still positive across most cases. For example, Telugu improves from 9.85 to 16.20, and Urdu from 17.72 to 20.83 BLEU. These results indicate that punctuation provides useful syntactic cues that MT models can leverage, particularly in morphologically rich or word-order flexible languages, reinforcing its utility as a preprocessing step for multilingual MT systems.

Language	Indic to English		English to Indic	
	w/o punct.	w/ punct.	w/o punct.	w/ punct.
Assamese	9.61 / 44.35	16.80 / 50.49	5.43 / 38.40	8.26 / 40.80
Bengali	11.60 / 45.80	16.91 / 48.73	10.07 / 46.52	13.52 / 48.53
Bodo	17.23 / 49.11	22.95 / 53.07	12.49 / 35.43	11.05 / 33.08
Gujarati	12.78 / 45.64	18.86 / 50.64	12.63 / 43.57	20.63 / 48.44
Hindi	15.74 / 49.53	19.37 / 51.95	19.76 / 50.89	19.47 / 49.21
Kannada	9.91 / 44.83	16.39 / 49.85	7.00 / 43.52	13.72 / 49.06
Kashmiri	4.80 / 32.16	7.20 / 33.98	3.22 / 21.40	5.19 / 24.20
Konkani	1.44 / 24.73	2.17 / 28.26	0.68 / 22.97	0.79 / 22.33
Maithili	2.81 / 27.13	4.25 / 30.89	0.85 / 20.25	1.15 / 20.92
Malayalam	8.64 / 44.14	15.53 / 50.02	5.57 / 41.98	13.32 / 48.97
Marathi	12.17 / 46.97	17.84 / 51.41	9.64 / 44.68	12.10 / 45.04
Nepali	8.21 / 34.68	13.52 / 41.71	3.33 / 32.09	4.53 / 32.49
Oriya	8.09 / 41.50	14.21 / 46.78	3.29 / 34.01	5.73 / 36.58
Punjabi	14.47 / 50.67	26.37 / 59.92	15.45 / 46.02	24.82 / 52.55
Sanskrit	1.66 / 26.68	2.37 / 29.11	0.51 / 22.23	0.80 / 23.59
Tamil	9.76 / 44.00	14.74 / 47.96	7.12 / 47.02	14.07 / 53.54
Telugu	9.77 / 43.83	13.98 / 46.02	9.85 / 44.27	16.20 / 49.48
Urdu	12.47 / 48.45	14.91 / 49.23	17.72 / 46.72	20.83 / 49.64
Total	**9.51 / 41.34**	**14.35 / 45.56**	**8.03 / 37.89**	**11.45 / 40.47**

Table 4: Translation quality for Indic–English directions. Each cell reports BLEU/chrF++ scores evaluated **without** and **with punctuation**. The scores with punctuations are statistically significant (with p-value < 0.05 for either chrF++ or BLEU) for all languages except in the case of Bodo, Nepali where there is no statistically significant difference between the scores.

6 Conclusion

We presented Cadence, a novel multilingual punctuation restoration model for English and 22 scheduled Indian languages. By adapting the GEMMA3-1B-PRETRAIN model with bidirectional attention and utilizing a curriculum-based continual pre-training strategy with MNTP on Indic web data, we successfully created a robust foundation model. Fine-tuning this model on a diverse aggregation of datasets with weighted sampling yielded a single model capable of handling 23 languages and 30 punctuation types, including Indic-specific marks. Our model significantly outperforms existing monolingual baselines across various languages, demonstrating the power of multilingual learning and our tailored pre-training approach.This achievement highlights the potential of unified multilingual models to address linguistic disparities in the realm.

References

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. LLM2Vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=IW1PR7vEBf.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. Cosmopedia, 2024. URL https://huggingface.co/datasets/HuggingFaceTB/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. 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)*, pp. 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10888018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. Towards leaving no Indic language behind: Building monolingual corpora, benchmark and models for Indic languages. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12402–12426, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.693. URL https://aclanthology.org/2023.acl-long.693.

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. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=vfT4YuzAYA.

Oliver Guhr, Anne-Kathrin Schumann, Frank Bahrmann, and Hans Joachim Böhme. Fullstop: Multilingual deep models for punctuation prediction. June 2021. URL http://ceur-ws.org/Vol-2957/sepp_paper4.pdf.

Anirudh Gupta, Neeraj Chhimwal, Ankur Dhuriya, Rishabh Gaur, Priyanshi Shah, Harveen Singh Chadha, and Vivek Raghavan. indic-punct: An automatic punctuation restoration and inverse text normalization framework for indic languages, 2022. URL https://arxiv.org/abs/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, Pratyush Kumar, and Mitesh M Khapra. Indicvoices: Towards building an inclusive multilingual speech dataset for indian languages, 2024. URL https://arxiv.org/abs/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. 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)*, pp. 15831–15879. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.843. URL http://dx.doi.org/10.18653/v1/2024.acl-long.843.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Martin Jaggi, Leandro von Werra, and Thomas Wolf. Fineweb2: A sparkling update with 1000s of languages, December 2024. URL https://huggingface.co/datasets/HuggingFaceFW/fineweb-2.

Ashwin Sankar, Sparsh Jain, Nikhil Narasimhan, Devilal Choudhary, Dhairya Suman, Mohammed Safi Ur Rahman Khan, Anoop Kunchukuttan, Mitesh M Khapra, and Raj Dabre. 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*, 2025. URL https://openreview.net/forum?id=QYAb7tbPKZ.

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, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, and 14 others. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Subhashree Tripathy and Ashis Samal. Punctuation and case restoration in code mixed Indian languages. In Wenjuan Han, Zilong Zheng, Zhouhan Lin, Lifeng Jin, Yikang Shen, Yoon Kim, and Kewei Tu (eds.), *Proceedings of the Workshop on Unimodal and Multimodal Induction of Linguistic Structures (UM-IoS)*, pp. 82–86, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.umios-1.9. URL https://aclanthology.org/2022.umios-1.9/.

Vincent Vandeghinste, Lyan Verwimp, Joris Pelemans, and Patrick Wambacq. A comparison of different punctuation prediction approaches in a translation context. In Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Miquel Esplà-Gomis, Maja Popović, Celia Rico, André Martins, Joachim Van den Bogaert, and Mikel L. Forcada (eds.), *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pp. 289–298, Alicante, Spain, May 2018. URL https://aclanthology.org/2018.eamt-main.27/.

Appendix

A Language wise label breakdown

The tables show language wise breakdown of label distribution.

Language	.	,	?	-	;	_	!	’	…	”	।	(	)	:	،
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 5: Label Distribution per Language (Part 1: first 15 labels). Counts $\geq$ 1000 are shown in thousands (k). Top header row is Label ID, second header row is the corresponding punctuation mark.

Language	ـ	؟	.”	).	),	”,	”.	?”	”?	।”	”।	،	۔	॥	।।
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 6: Label Distribution per Language (Part 2: second 15 labels). Counts $\geq$ 1000 are shown in thousands (k). Top header row is Label ID, second header row is the corresponding punctuation mark.

B Prompt used for Punctuation

The prompt template shown in Figure 3 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 3: Prompt For Punctuation Restoration

C Prompt used for LLM as a Judge

The datasets we have used for training contain web-scraped text (Sangraha-verified, IndicCorpV2) and also synthetically punctuated text (IndicVoices). As a result punctuations may not always be correct. Ensuring a high quality test set becomes important to accurately assess our model and compare with existing models. We have employed Gemini-2.5-Flash-preview-04-17 as a judge to validate our test set. We present the prompt used in Fig.4 below.

Prompt for using LLM as a Judge

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 4: The LLM-as-a-Judge prompt, outlining the comprehensive rubric used for evaluating punctuation and capitalization. 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.