CMLFormer: A Dual Decoder Transformer with Switching Point based Positional Encoding for Code-Mixed Language Modeling

Aditeya Baral and **Allen George Ajith** and **Mrityunjay Abhijeet Bhanja** and **Roshan Nayak**
New York University
{ab12057, aa12938, mb9348, rn2588}@nyu.edu

1 Introduction

Globalization and cultural adaptation have fostered the growth of multilingualism in the modern world. This has lead to a widespread phenomenon of *code-mixing* (CM), where multiple languages are juxtaposed within a single utterance with the transition points being known as *switching points* (SP). In multilingual countries like India, the fusion of Hindi and English (popularly termed *Hinglish*) has become increasingly prevalent in conversational usage both online and offline. In code-mixed languages, the *base language* provides the grammatical structure, while the *mixing language* contributes loan words or phrases. For example, for the CM sentence $Mujhe_{\text{HI}}$ kal_{HI} ek_{HI} $important_{\text{EN}}$ $meeting_{\text{EN}}$ $attend_{\text{EN}}$ $karni_{\text{HI}}$ hai_{HI}, Hindi serves as the base language, while the words *important*, *meeting*, and *attend* from English are mixed in.

As Large Language Models (LLMs) become increasingly integrated into our daily lives, recent years have seen the development of multiple LLMs for a range of diverse languages, including high-resource ones such as English (Anil et al., 2024) and French (Faysse et al., 2024). Despite the steady increase in CM speakers today, most LLMs still fail to serve CM needs (Srivastava, 2025; Venkatesh et al., 2024). This is evidenced by the subpar performance on real-world tasks requiring encoder-only models, such as classification, semantic search and information retrieval. Their limitations can be traced back to their inadequate and poor neural representations of these languages (Mazumder et al., 2024; Jagdale et al., 2024). The unique linguistic structure of CM languages makes it a challenge to represent them effectively, limiting their efficacy in multilingual contexts.

Addressing these limitations requires a richer modeling of CM structure, particularly the role of switching points in transitional contexts. Since prior studies suggest that current pre-training approaches are insufficient to model CM languages, we propose **CMLFormer**, a novel dual-decoder Transformer architecture with a new switching point based positional encoding, specifically for code-mixed language modeling. Additionally, we introduce new supervised and unsupervised pre-training objectives such as *Switching Point Prediction*, *Bilingual Translated Sentence Prediction* and *Bilingual Language Translation* to learn richer representations of CM languages and encourage cross-lingual learning and alignment.

As we will be working with small to medium scale models, each with around 600M parameters, the project is computationally feasible requiring not more than two GPUs.

2 Related Work

Previous approaches have used multilingual models to bridge the gap between the languages in a CM language. However, studies have shown that CM languages being *inherently* multilingual, cannot be replaced by multilingual models since they are not natural code-mixers (Zhang et al., 2023) of monolingual languages. The lack of formal grammar, frequent occurrence of switching points, spelling variants and contextual nuances are some of the problems posed by CM languages. Additionally, models pre-trained on CM data have surprisingly shown inadequate improvements over multilingual models despite increasing the number of parameters (Patil et al., 2023; Santy et al., 2021). Since scaling up both data and parameters has only yielded marginal improvements, recent studies have moved away from the former and established the necessity of specialised pre-training techniques for CM languages.

(Li and Murray, 2022) introduce a language-agnostic approach that shows that masking the loan words from a CM sequence allows multilingual models to generalize better to downstream tasks. The effectiveness of cross-lingual representations

using approaches like Translation Language Modeling (Lample and Conneau, 2019) has also been extensively studied in MuRIL (Khanuja et al., 2021) and IndicBERT (Doddapaneni et al., 2023), both of which demonstrate significant improvements over multilingual BERT (Devlin et al., 2019) on Indian languages. However, these approaches still failed to sufficiently capture the linguistic nuances of CM text. Moving towards architectural modifications, (Sengupta et al., 2021) introduce a Hierarchical Transformer with a new outer-product attention mechanism to effectively capture the syntactic and structural characteristics of CM sentences. (Ali et al., 2021) was one of the first to introduce a new positional encoding approach that assigned weights to words near switching points, enabling the model to capture language transitions in a CM sentence better. Building upon this, (Ali et al., 2023) introduce an improved switching point based rotary matrix by combining it with rotatory positional encodings (Su et al., 2023), which rotate at switching points to denote language transitions.

3 CMLFormer

This section introduces our proposed model, the CMLFormer, and includes details about the architecture, pre-training tasks, data, and other details.

3.1 Problem Definition

We formally represent a sequence of N_C tokens in the CM language as $C = \{c_1, c_2, ..., c_{N_C}\}$, a sequence of translated N_B tokens in the base language as $B = \{b_1, b_2, ..., b_{N_B}\}$ and a sequence of translated N_M tokens in the mixing language as $M = \{m_1, m_2, ..., m_{N_M}\}$. We also define the set of language labels with $L = \{l1, l2, ..., l_{N_C}\}$ and language transitions with a bit vector $T = \{t1, t2, ..., t_{N_C-1}, 0\}$ where t_i is a 0-1 bit that indicates a language transitions from c_i to c_{i+1} with a 0 bit added at the end to account for EOS. For example, for an input CM sentence C: *college*$_{\text{EN}}$ *mein*$_{\text{HI}}$ *aaj*$_{\text{HI}}$ *exam*$_{\text{EN}}$ *hain*$_{\text{HI}}$, we denote the language labels $L = \{\text{EN}, \text{HI}, \text{HI}, \text{EN}, \text{HI}\}$ and transitions $T = \{1, 0, 1, 1, 0\}$. The inputs to CMLFormer's encoder is the sequence C, with B and M being passed to the base and mixing language decoder respectively.

3.2 Architecture

We build the backbone of the CMLFormer (Fig. 1) using an enhanced multi-layer Transformer (Vaswani et al., 2017) with both the encoder and decoder layers. We propose two significant modifications to the vanilla Transformer architecture. First, we explore a novel multi-target pre-training setup where two fully synchronous decoders sharing the same encoder are coupled through an attention layer. Considering the usually significant syntactical and semantic differences between the base and mixing language, this allows the model to decouple the two languages and handle their specific nuances effectively while still learning cross-lingual representations. We further explore two variations of this enhancement through our ablations (Appendix A.1). Second, we introduce a new positional encoding that integrates Gaussian-scaled switching point information with token embeddings to capture language transitions between tokens.

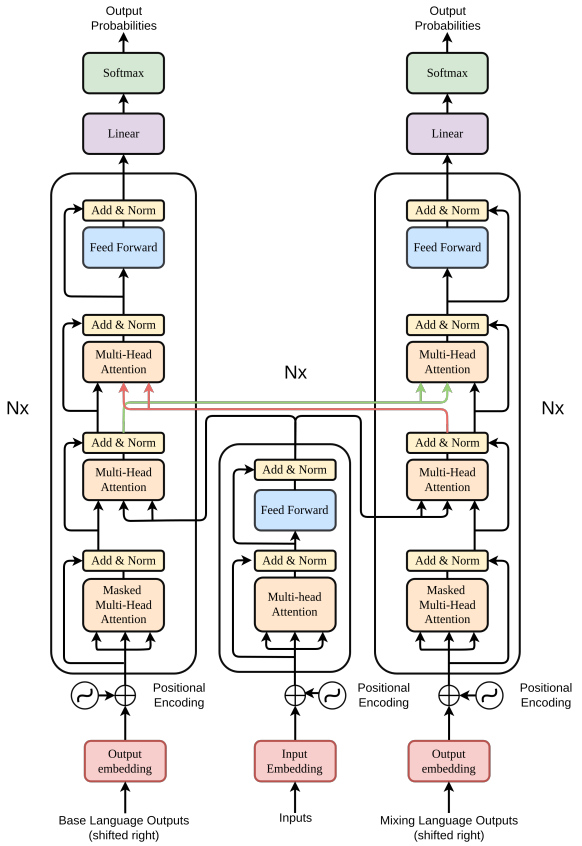

Figure 1: **The architecture of our proposed approach CMLFormer**: The outputs from each encoder-decoder attention sub-layer (arrows shown in green and red) are passed as input to the decoder-decoder cross-attention sub-layer. The decoders exhibit full synchronous coupling since each requires the hidden states from the other to compute its own hidden states. After pre-training, the encoder is extracted and fine-tuned on downstream tasks.

3.2.1 Dual Decoder with Decoder-level Cross Attention

We introduce a synchronous coupling of two decoders with the same shared encoder. In contrast to the vanilla Transformer architecture, we add a third attention layer to introduce inter-decoder cross-attention in each decoder block to mutually share the latent characteristics of each constituent language. Denoting the output hidden states from layer normalization after the encoder-decoder attention sub-layer of the base and mixing language decoder as a_l^b and a_l^m respectively, the decoder-decoder cross-attention is computed as,

$$\begin{aligned} o_l^b &= Attention(a_l^b, a_l^m, a_l^m) \\ o_l^m &= Attention(a_l^m, a_l^b, a_l^b) \end{aligned} \tag{1}$$

The corresponding output attention scores o_l^b and o_l^m are then passed as input to the feed-forward and layer normalization sub-layers as in the vanilla Transformer. This allows each decoder to attend to and "peek" at the other decoder's hidden states while decoding the input CM sentence into their corresponding language, helping it capture inter-block interactions and driving cross-lingual learning and alignment. The two decoders are thus fully synchronized as each requires the hidden states of the other in each block to compute its own.

3.2.2 Gaussian-scaled Switching Point Encoding

We introduce a novel switching point based positional encoding that aims to encode language transitions with token embeddings. We scale all the 1-bits in T with overlapping Gaussian distributions to emphasize the switching points. This results in a switching point activation vector with larger values concentrated around switching points (each switching point is assigned a value of 1.0) and lower values for contiguous sub-sequences of words in the same language.

3.3 Pre-Training Tasks

CMLFormer is expected to learn rich representations of the CM language, including cross-lingual representations for the base and mixing language, and the relationship between tokens and current context at switching points. To achieve this, we introduce multiple supervised and unsupervised pre-training tasks to learn rich contextual representations.

3.3.1 Masked Language Modeling

We incorporate the widely used Masked Language Modeling (Devlin et al., 2019) task as the base pre-training objective. More details can be found in Appendix A.2.1.

3.3.2 Switching Point Prediction

Switching points lead to frequent syntactic and semantic transitions within a CM sentence, which makes it a challenge to model these languages using standard approaches which fail to account for transitional contexts. To capture and model these linguistic nuances and learn richer contextual representations, we propose a new Switching Point Prediction (SPP) pre-training objective. Given a CM token sequence C and a sequence of transitions T, we are tasked with predicting the probability of switching points occurring between tokens in the code-mixed sequence, given by $P = \{p_1, p_2, ..., p_{N_C}\}$.

3.3.3 Bilingual Translated Sentence Prediction

The multilingual setting of CM languages enables concepts to be represented by words from the base or mixing languages. It is necessary to model cross-lingual relationships between these word variants by learning rich contextual representations of the CM, base and mixing languages. We propose a new Bilingual Translated Sentence Prediction (BTSP) objective to capture these cross-lingual relationships. Given a CM token sequence C, we randomly sample another token sequence S where 25% of the time S is B, 25% of the time it is M, and the remaining 50% of the time it is a randomly selected sequence C' from the corpus. The objective of BTSP is to predict whether sequence C is equivalent to the sampled sequence S.

3.3.4 Bilingual Language Translation Modeling

To capture the inherent multilingual characteristics of CM languages, it is crucial to not only learn rich contextual representations for the CM, base and mixing languages but also model a three-way alignment among them. Given a CM token sequence C, and its corresponding translations in the base and mixing language translations B and M respectively, we introduce a multi-target Bilingual Language Translation (BiLTM) objective. The base and mixed language decoders are tasked to simultaneously generate translations of the CM

sequence into its equivalent base language and mixing language forms L_B and L_M respectively. More details about computing $\mathcal{L}_{BiLTM}$ can be found in Appendix A.2.4.

CMLFormer is jointly trained on all the above pre-training objectives to minimise the overall training loss $\mathcal{L}_{total}$. This can be formulated as,

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{MLM} + \beta\mathcal{L}_{SPP} + \gamma\mathcal{L}_{BTSP} + \eta\mathcal{L}_{BiLTM} \quad (2)$$

where α, β, γ and η are scaling constants.

3.4 Datasets

3.4.1 Pre-training

CMLFormer will be pre-trained on the L3Cube-HingCorpus (Nayak and Joshi, 2022), a large-scale Hindi-English code-mixed corpus in Roman script. It comprises 52.93M sentences and 1.04B tokens from Twitter and reflects real-world code-mixing patterns.

3.4.2 Data Augmentation

To support the BiLTM task, we augment the dataset by generating parallel translations of code-mixed Hinglish sentences into pure Hindi and English in Roman script. Further details on the augmentation process and example transformations are provided in Appendix A.3.

3.4.3 Fine-tuning and Evaluation

We will discard CMLFormer's decoders and fine-tune just the encoder on three established Hinglish CM language classification benchmarks. The **HASOC 2021** (Mandl et al., 2021) dataset for hate speech detection comprises 7,000 code-mixed Hinglish tweets distributed across 5,740 train and 1,348 test examples. The **SentiMix 2020** (Patwa et al., 2020) dataset comprises 20,000 Hinglish sentences with sentence-level sentiment labels (positive, negative, neutral) for sentiment analysis. **GLUECoS** (Khanuja et al., 2020) is a Hinglish CM benchmark that includes the following tasks: token language identification, part of speech tagging, named entity recognition, and sentiment analysis. We will be evaluating CMLFormer on the above benchmarks and comparing its performance with current approaches and baselines.

4 Experiments

4.1 Comparison with Baselines

CMLFormer will be compared with existing models such as XLM-RoBERTa (Conneau et al., 2020), HingBERT (Nayak and Joshi, 2022), IndicBERT and MuRIL on the fine-tuning benchmarks. This will help us analyze the effectiveness of our pre-training strategies with the standard MLM approach of the aforementioned models. Additionally, we will also evaluate the zero-shot performance of these models on the benchmarks to investigate how well our pre-training can generalize to unseen tasks.

4.2 Ablation Studies

To identify the individual contributions of each of our pre-training strategies, we will be isolating each component and evaluating CMLFormer on the same benchmarks. Specifically, we are interested in investigating the improvement gained using (1) new pre-training objectives and (2) architectural and encoding modifications. This would help us evaluate the robustness of our approach and validate the insufficiency of using MLM for pre-training on CM languages. A few more ablation experiments have been detailed in Appendix A.

4.3 Intrinsic Evaluation of Representations

We will perform an intrinsic evaluation of the learnt representations by analyzing attention maps and identifying switching points in the attention scores. We expect CMLFormer to show higher attention around switching points, especially around frequent language transitions. This analysis will provide insights into the model's focus and decision-making process, allowing us to assess the quality and interpretability of the representations.

5 Division of Labour

Aditeya is working on the design and implementation of the CMLFormer architecture and will also be implementing the BTSP and BiLTM objectives and pre-training the model. **Roshan** is working on the Gaussian-scaled switching point based positional encoding architecture and will also implement the MLM and SPP objectives. **Allen** will be implementing and training the custom tokenizer for Hinglish CM text, setting up distributed training infrastructure on NYU HPC clusters, and documenting pretraining and fine-tuning results. **Mrityunjay** will be working on data augmentation to generate parallel English and Hindi translations using Gemini 2.0 Flash. He will be curating and cleaning the L3Cube-HingCorpus dataset and validate translation quality through manual sampling.

References

Mohsin Ali, Kandukuri Sai Teja, Neeharika Gupta, Parth Patwa, Anubhab Chatterjee, Vinija Jain, Aman Chadha, and Amitava Das. 2023. Conflator: Incorporating switching point based rotatory positional encodings for code-mixed language modeling.

Mohsin Ali, Kandukuri Sai Teja, Sumanth Manduru, Parth Patwa, and Amitava Das. 2021. Pesto: Switching point based dynamic and relative positional encoding for code-mixed languages.

Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Gemini Team, Jiahui Yu, and Radu Soricut et. al. 2024. Gemini: A family of highly capable multimodal models.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

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.

Manuel Faysse, Patrick Fernandes, Nuno M. Guerreiro, António Loison, Duarte M. Alves, Caio Corro, Nicolas Boizard, João Alves, Ricardo Rei, Pedro H. Martins, Antoni Bigata Casademunt, François Yvon, André F. T. Martins, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2024. Croissantllm: A truly bilingual french-english language model.

Shruti Jagdale, Omkar Khade, Gauri Takalikar, Mihir Inamdar, and Raviraj Joshi. 2024. On importance of code-mixed embeddings for hate speech identification.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. Muril: Multilingual representations for indian languages.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. GLUECoS: An evaluation benchmark for code-switched NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3575–3585, Online. Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining.

Shuyue Stella Li and Kenton Murray. 2022. Language agnostic code-mixing data augmentation by predicting linguistic patterns.

Thomas Mandl, Sandip Modha, Gautam Kishore Shahi, Hiren Madhu, Shrey Satapara, Prasenjit Majumder, Johannes Schaefer, Tharindu Ranasinghe, Marcos Zampieri, Durgesh Nandini, and Amit Kumar Jaiswal. 2021. Overview of the hasoc subtrack at fire 2021: Hate speech and offensive content identification in english and indo-aryan languages.

Debajyoti Mazumder, Aakash Kumar, and Jasabanta Patro. 2024. Revealing the impact of synthetic native samples and multi-tasking strategies in hindi-english code-mixed humour and sarcasm detection.

Ravindra Nayak and Raviraj Joshi. 2022. L3Cube-HingCorpus and HingBERT: A code mixed Hindi-English dataset and BERT language models. In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 7–12, Marseille, France. European Language Resources Association.

Aryan Patil, Varad Patwardhan, Abhishek Phaltankar, Gauri Takawane, and Raviraj Joshi. 2023. Comparative study of pre-trained bert models for code-mixed hindi-english data. In *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*, page 1–7. IEEE.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas PYKL, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. SemEval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 774–790, Barcelona (online). International Committee for Computational Linguistics.

Sebastin Santy, Anirudh Srinivasan, and Monojit Choudhury. 2021. BERTologiCoMix: How does code-mixing interact with multilingual BERT? In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 111–121, Kyiv, Ukraine. Association for Computational Linguistics.

Ayan Sengupta, Sourabh Kumar Bhattacharjee, Tanmoy Chakraborty, and Md. Shad Akhtar. 2021. HIT - a hierarchically fused deep attention network for robust code-mixed language representation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4625–4639, Online. Association for Computational Linguistics.

Varad Srivastava. 2025. DweshVaani: An LLM for detecting religious hate speech in code-mixed Hindi-English. In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025)*, pages 46–60, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Dilip Venkatesh, Pasunti Prasanjith, and Yashvardhan Sharma. 2024. BITS pilani at SemEval-2024 task 10: Fine-tuning BERT and llama 2 for emotion recognition in conversation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 811–815, Mexico City, Mexico. Association for Computational Linguistics.

Ruochen Zhang, Samuel Cahyawijaya, Jan Christian Blaise Cruz, Genta Indra Winata, and Alham Fikri Aji. 2023. Multilingual large language models are not (yet) code-switchers.

A Appendix

A.1 Architecture Ablations

A.1.1 Dual Decoder without Decoder-level Cross Attention

To isolate and measure the impact of the proposed decoder cross-attention sub-layer in each layer, we will decouple the two decoders by removing cross-attention from both branches. This would degenerate both decoders to their vanilla architecture and remove the need for synchronous decoding since the overall architecture would function as two independent decoders with a shared encoder. While the model retains the multi-target training setup, this approach prevents each decoder from learning from the other, inhibiting cross-lingual alignment and learning inter-language relationships.

A.1.2 Ablating with Cross-Attention Inputs

We will investigate the effect of swapping the inputs to the cross-attention sub-layer with the final output hidden state from the previous layer of the other decoder after being passed through all its sub-layers. We hypothesize that this will ensure that each layer's cross-attention is based on a *stable* and *richer* representation from the other decoder branch. Denoting the output hidden states of the previous layer of the base and mixing language decoder as H^b_{l-1} and H^m_{l-1} respectively, this can be formulated as,

$$\begin{aligned} o^b_l &= Attention(a^b_l, H^m_{l-1}, H^m_{l-1}) \\ o^m_l &= Attention(a^m_l, H^b_{l-1}, H^b_{l-1}) \end{aligned} \tag{3}$$

Output attention scores o^b_l and o^m_l are then passed as input to the feed-forward and layer normalization sub-layers as done previously. After each layer l in the decoder, we store the final output hidden states H^b_l and H^m_l so they can be used in the cross-attention sub-layer of the decoder's $l+1$ layer.

A.1.3 Additional Switching Point based Positional Encoding Approaches

We are also exploring other encoding methods such as using a wavelet transform to account for local and global frequencies of switches, an exponential decay to factor in distance from the next switch and adding a new switching position embedding to the token embedding. These approaches will be used in conjunction with the standard token and positional embedding functions as in BERT.

A.2 Pre-Training Tasks and Training Losses

A.2.1 Masked Language Modeling

Masked Language Modelling (MLM) was first introduced in BERT (Devlin et al., 2019) to mitigate the problem of traditional bidirectional language modelling allowing the model to "see" the tokens being predicted. Given a CM token sequence $C = \{c_1, c_2, ..., c_{N_C}\}$, a subset of tokens are randomly sampled denoted by L_{mask}, typically 15% of which 80% are replaced by $[MASK]$ token, 10% are replaced by a random token, and 10% remain unchanged. The objective of MLM is to predict the masked tokens by using the surrounding unmasked tokens in the input sequence as context, which is formulated by,

$$\mathcal{L}_{MLM} = \sum_{c_i \in L_{mask}} -\log p(c_i|\theta) \tag{4}$$

A.2.2 Switching Point Prediction

The Switching Point Prediction loss $\mathcal{L}_{SPP}$ can be formulated with binary cross-entropy as,

$$\mathcal{L}_{SPP} = -\frac{1}{N_C}\sum_{i=1}^{N_C}[t_i \log(p_i) + (1-t_i)\log(1-p_i)] \tag{5}$$

where $p_i = p(t_i = 1|c_i, \theta)$ denotes the predicted probability of a switching point occurring after token c_i given model parameters θ.

A.2.3 Bilingual Translated Sentence Prediction

The Bilingual Translated Sentence Prediction loss $\mathcal{L}_{BTSP}$ can be formulated with binary cross-entropy as follows:

$$\begin{aligned} \mathcal{L}_{BTSP} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p(S_i|C_i, \theta)) \\ +(1-y_i)\log(1-p(S_i|C_i, \theta))] \end{aligned} \tag{6}$$

where N is the total number of training examples and $p(S_i|C_i, \theta)$ is the model's predicted probability that S_i is related to C_i given model parameters θ.

A.2.4 Bilingual Language Translation Modeling

The objective of BiLTM can be formulated with an autoregressive modeling function for each decoder with an added alignment factor to enforce

alignment between the CM sequence C and its equivalent forms B and M. We compute the Bilingual Language Translation Modeling loss $\mathcal{L}_{BiLTM}$ as,

$$\mathcal{L}_{BiLTM_b} = -\frac{1}{N_B}\sum_{i=1}^{N_B}\log p(b_i|b_{<i}, C; \theta_B)$$

$$\mathcal{L}_{BiLTM_m} = -\frac{1}{N_M}\sum_{j=1}^{N_M}\log p(m_j|m_{<j}, C; \theta_M)$$

$$\mathcal{L}_{BiLTM_{align_b}} = -\frac{1}{N_C}\sum_{i=1}^{N_C}\left[\log\left(\sum_{j=1}^{N_B}\alpha_{B_{ij}}\right)\right]$$

$$\mathcal{L}_{BiLTM_{align_b}} = -\frac{1}{N_C}\sum_{i=1}^{N_C}\left[\log\left(\sum_{k=1}^{N_M}\alpha_{M_{ik}}\right)\right]$$

$$\mathcal{L}_{BiLTM_{align}} = \mathcal{L}_{BiLTM_{align_b}} + \mathcal{L}_{BiLTM_{align_b}}$$

$$\mathcal{L}_{BiLTM} = \mathcal{L}_{BiLTM_b} + \mathcal{L}_{BiLTM_m} + \lambda\mathcal{L}_{BiLTM_{align}} \quad (7)$$

where $\mathcal{L}_{BiLTM_b}$ and $\mathcal{L}_{BiLTM_m}$ are the translation losses for the base and mixing language, $\alpha_{B_{ij}}$ and $\alpha_{M_{ik}}$ are the attention weights between the i-th token in C and the j-th and k-th token in B and M respectively, $\mathcal{L}_{BiLTM_{align_b}}$ and $\mathcal{L}_{BiLTM_{align_m}}$ are the alignment losses between C and B and M respectively, $\mathcal{L}_{BiLTM_{align}}$ is the combined alignment loss, and $\mathcal{L}_{BiLTM}$ is the total BiLTM loss.

A.2.5 Other Pre-Training Tasks

We are currently also exploring other pre-training objectives such as a regression task to measure the CM Index and a supervised alignment task to learn specific alignments between C, B and M. However, these are subject to time constraints and might not be fully explored.

A.3 Dataset Augmentation and Curation

A.3.1 Augmentation Details and Example Transformations

We augment the corpus with parallel translations of Hinglish into English and Hindi in Roman script. We utilize a multilingual Large Language Model, Gemini 2.0 Flash, for translation. This augmentation strategy generates a trilingual parallel corpus in Hinglish-English-Hindi(Roman) that facilitates the BiLTM objective.

Below is an example from our augmented dataset showing Hinglish sentences alongside their English and Hindi (Roman) translations:

Example 1:

- **Hinglish:** *aapki logo ki help krne ki soch bhut acchi h ...manav seva bhut hi accha kary krte ho di.*
- **English:** *Your thought of helping people is very good... you do a very good deed of human service, sister.*
- **Hindi (Roman):** *Aapki logo ki madad karne ki soch bahut achchhi hai... manav seva bahut hi achchha karya karte ho di.*

Example 2:

- **Hinglish:** *office se ghar aate time traffic mein 2 ghante waste ho gaye. Metro service improve karo please*
- **English:** *Wasted 2 hours in traffic while coming home from office. Please improve the metro service*
- **Hindi (Roman):** *Karyalaya se ghar aate samay safar mein 2 ghante vyarth ho gaye. Metro seva mein sudhar kijiye*

A.3.2 Data Curation and Pre-processing

During augmentation, we conduct a manual review of several hundred samples to ensure the quality of translation. However, we observed examples where producing a valid translation could not be generated because of unintelligible text or highly ambiguous phrasing. Such training examples were omitted from the dataset.