Power doesn't reside in size: A Low Parameter Hybrid Language Model (HLM) for Sentiment Analysis in Code-mixed data

Pavan Sai Balaga, Nagasamudram Karthik, Challa Vishwanath, Raksha Sharma,
Indian Institute of Technology Roorkee
{b_psai,n_karthik,c_vishwanath,raksha.sharma}@cs.iitr.ac.in

Rudra Murthy, **Ashish Mittal**
IBM Research, India
{rmurthyv,arakeshk}@in.ibm.com

Abstract

Code-mixed text—where multiple languages are used within the same utterance—is increasingly common in both spoken and written communication. However, it presents significant challenges for machine learning models due to the interplay of distinct grammatical structures, effectively forming a hybrid language. While fine-tuning large language models (LLMs) such as GPT-3, or Llama-3 on code-mixed data has led to performance improvements, these models still lag behind their monolingual counterparts and incur high computational costs due to the large number of trainable parameters.

In this paper, we focus on the task of sentiment detection in code-mixed text and propose a *Hybrid Language Model (HLM)* that combines a multilingual encoder (e.g., mBERT) with a lightweight decoder (e.g., Sarvam-1) ($< 3B$ parameters). Despite having significantly fewer trainable parameters, HLM achieves sentiment classification performance comparable to that of fine-tuned Large Language Models (LLMs) ($> 7B$ parameters). Furthermore, our results demonstrate that HLM significantly outperforms models trained individually, underscoring its effectiveness for low-resource, code-mixed sentiment analysis.

1 Introduction

Code mixing—the blending of two or more languages within a sentence or conversation—is common in multilingual communities. For instance, a Hindi-English speaker might say, "Kal cricket match haar gaye, so full day mood off raha" (which translates to "We lost the cricket match yesterday, so I was in an off mood the whole day"). Natural Language Processing (NLP) systems trained on monolingual data often struggle with such inputs, particularly for tasks like Sentiment Analysis (SA), due to grammatical inconsistencies, transliteration variations, and misspellings (Dhingra et al., 2016).

Encoder-only models like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2019), pretrained on large-scale unlabeled corpora, have achieved strong results across a range of NLP tasks. Their multilingual variants, when fine-tuned on code-mixed datasets, outperform traditional models but still underperform compared to monolingual baselines (Sharma et al., 2023).

Meanwhile, Large Language Models (LLMs) such as GPT-3 (Brown et al., 2020), Llama (Touvron et al., 2023), and Mistral (Jiang et al., 2023) demonstrate impressive zero-shot and few-shot performance, but are primarily optimized for English. Multilingual LLMs like BLOOM (Le Scao et al., 2023), mGPT (Shliazhko et al., 2022), Llama-3 (AI@Meta, 2024), and Gemma-2 (Riviere et al., 2024) attempt to bridge this gap. However, the high inference cost and large memory footprint of these models limit their applicability in real-time, resource-constrained settings.

Encoder-based models are well-suited for classification tasks due to their bidirectional context modeling via masked language objectives, whereas decoder-only models prioritize sequential coherence through autoregressive generation. Smaller decoder models improve deployability but often lack the discriminative power required for sentiment classification.

Motivated by recent advances in modular LLM design (Wan et al., 2024b), we introduce a low-parameter *Hybrid Language Model* (*HLM*) tailored for sentiment detection in code-mixed text. HLM integrates a multilingual encoder (e.g., mBERT or XLM-RoBERTa) with a compact decoder (e.g., Sarvam-1 or Llama-3.2-1B), merging their representations via a lightweight neural layer. This fusion leverages the encoder's deep contextual understanding and the decoder's generative priors, offering a balanced architecture for sentiment classification in code-mixed data.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 14820–14828

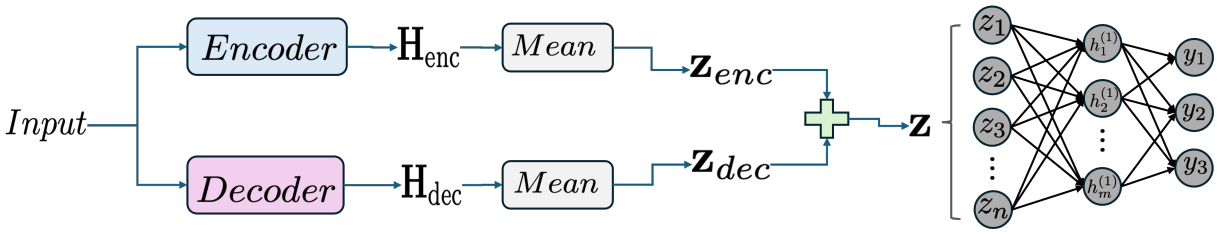

Figure 1: Architecture of HLM

The paper presents a novel architecture, HLM, which achieves performance comparable to Gamma-9B, the best-performing LLM on the task—while using only 1.5 to 3 billion parameters. The Results and Analysis section elaborates on and compares our approach with the Late-fusion model (Sharma et al., 2023), the state-of-the-art encoder-based strategy for this task, as well as with various LLMs. Notably, HLM models with as few as 1.5 to 3 billion parameters deliver sentiment classification results comparable to, or better than, those of LLMs with 7 to 9 billion parameters and the Late-fusion model (see Table 2 and Table 3). Despite having significantly fewer trainable parameters than conventional LLMs, HLM achieves competitive performance on code-mixed sentiment detection benchmarks. More importantly, its modularity and efficiency make it ideal for real-world applications where compute resources are limited—such as on-device inference or deployment in multilingual regions with low-resource infrastructure. Our experiments demonstrate that HLM not only narrows the gap with large-scale models but also provides a practical path forward for scalable and accessible code-mixed NLP.

2 Related Work

Early research on code-mixed text focused on dataset creation and encoder-based modeling. Benchmarks like GLUECoS (Khanuja et al., 2020) and LINCE (Aguilar et al., 2020) established standardized evaluation tasks, while datasets from Patwa et al. (2020) and Chakravarthi et al. (2020b) supported sentiment analysis in English-Hindi, English-Spanish, and Tamil-English.

Multilingual encoders such as mBERT showed improvements over monolingual models on code-mixed data (Fazili and Jyothi, 2022). To address data scarcity, several works explored synthetic generation using prompting strategies with multilingual LLMs (Yong et al., 2023; Kartik et al., 2024).

Architectural innovations have also emerged. Sharma et al. (2023) proposed late fusion of model predictions, while Das et al. (2023) introduced modified MLM objectives and structural changes to better handle code-switching. More recently, Huzaifah et al. (2024) provided a thorough evaluation of LLMs on code-switched translation. Wan et al. (2024a) and Zhang et al. (2023) investigated combining and analyzing multilingual LLMs across tasks. Our work contributes to architectural innovation while achieving improved performance.

3 Methodology

We propose a Hybrid Language Model (HLM) for sentiment detection in code-mixed text by combining a multilingual encoder with a lightweight decoder. Figure 1 illustrates the architecture of our approach. Given an input sentence $x = \{w_1, w_2, \ldots, w_T\}$, we tokenize it separately for the encoder and decoder models, producing x_{enc} and x_{dec}.

The encoder (e.g., mBERT) produces contextual embeddings $\mathbf{H}_{\text{enc}} \in \mathbb{R}^{T \times h_{\text{enc}}}$, while the decoder (e.g., Sarvam-1) outputs $\mathbf{H}_{\text{dec}} \in \mathbb{R}^{T' \times h_{\text{dec}}}$. We apply mean pooling over each sequence to obtain sentence-level representations:

$$z_{\text{enc}} = \texttt{MeanPool}(\mathbf{H}_{\text{enc}}), z_{\text{dec}} = \texttt{MeanPool}(\mathbf{H}_{\text{dec}})$$

The pooled embeddings are concatenated to form a joint representation $\boldsymbol{z} = [\boldsymbol{z}_{\text{enc}}; \boldsymbol{z}_{\text{dec}}]$, which is passed through a feedforward classifier (f_{cls}) with a softmax output layer to predict sentiment probabilities:

$$\hat{\mathbf{y}} = \texttt{Softmax}(f_{\text{cls}}(\boldsymbol{z}))$$

We train the model using the standard cross-entropy loss between $\hat{\mathbf{y}}$ and the gold label $\mathbf{y}$. This hybrid formulation captures complementary strengths—contextual understanding from the encoder and generative priors from the decoder—resulting in efficient and effective sentiment classification for code-mixed text.

4 Experiment Setup

We evaluated our hypothesis using four code-mixed sentiment datasets: the Romanized English-Hindi (ENG-HIN) dataset (Patwa et al., 2020), which is part of the LINCE benchmark (Aguilar et al., 2020) (20K tweets scraped via Twitter API using a curated list of 10786 Hindi word tokens, split into 14 K train, 3K dev, 3K test with Positive/Negative/Neutral classes); the English-Spanish (ENG-SPA) dataset (Aguilar et al., 2020) (18.8K tweets from Twitter and CALCS workshop corpora (Pratapa et al., 2018), split into 12194 train, 1859 dev, 4736 test with 31.5% Positive, 25.7% Negative, 42.8% Neutral); the English-Telugu (ENG-TEL) dataset (Kusampudi et al., 2021) (19.9K instances collected via Twitter API and YouTube Comments API on movie-related topics, split into 80–10–10 train/validation/test ratio with 39.9% Positive, 38.8% Negative, 21.3% Neutral); and the English-Malayalam (ENG-MAL) dataset (Chakravarthi et al., 2020a) (5.9K YouTube comments on Malayalam movie trailers, annotated by volunteers with Krippendorff's alpha > 0.8, split into 4204 train, 480 dev, 1171 test with Positive/Negative/Neutral classes). Table 1 presents the dataset statistics. The preprocessing details for ENG-MAL are included in the Appendix A.1

Dataset	Train	Validation	Test
ENG-HIN	14,000	3,000	3,000
ENG-SPA	12,194	1,859	4,736
ENG-MAL	4,204	480	1,171
ENG-TEL	15,893	1,987	1,986

Table 1: Data split statistics for different datasets

4.1 Baselines

Encoder-only Models. Following the fine-tuning setup of Devlin et al. (2019), we add a linear classification head with softmax activation on top of each encoder model. All parameters are updated during training using the cross-entropy loss. We evaluate two widely used encoder-only models: mBERT and XLMR (Conneau et al., 2020). Table 3 reports the performance results across all datasets.

Large Language Models (LLMs). We also evaluate decoder-only LLMs under both zero/few-shot and fine-tuning settings. For fine-tuning, we adopt the Q-LoRA framework (Dettmers et al., 2023), which updates a small number of quantized low-rank matrices, significantly reducing computational cost. These matrices are integrated into the base model using low-rank parameterization of the original weight matrices. The LLMs are fine-tuned to generate the sentiment label corresponding to a given input sentence. Prompts used during training, validation, and testing are listed in Appendix A.2. Our experiments include Q-LoRA fine-tuning on Llama-3 (8B), and Gemma-2 (9B). We also fine-tune their instruction-tuned variants.

Late Fusion. We also compare with the Late Fusion technique (Sharma et al., 2023; Colnerič and Demšar, 2020), which combines predictions from two encoder-only models via a neural layer to produce a unified output. We follow the experimental configuration described in the original work to re-implement this baseline for sentiment classification and to facilitate a fair comparison with our proposed HLM model.

Dataset	LLM Models	0-shot	5-shot	FT F1
ENG-HIN	Llama-3 8B	43.47	64.16	75.01
	Llama-3.1 8B Instruct	37.93	66.40	74.03
	Gemma-2 9B	40.41	39.97	**75.49**
	Gemma-2 9B Instruct	43.96	68.68	74.77
ENG-SPA	Llama-3 8B	16.24	54.12	**59.25**
	Llama-3.1 8B Instruct	41.96	51.69	56.34
	Gemma-2 9B	11.77	41.05	57.03
	Gemma-2 9B Instruct	21.51	47.58	59.03
ENG-MAL	Llama-3 8B	43.37	59.09	75.89
	Llama-3.1 8B Instruct	41.76	58.81	75.65
	Gemma-2 9B	22.78	63.65	**76.21**
	Gemma-2 9B Instruct	27.858	59.93	75.41
ENG-TEL	Llama-3 8B	26.94	61.41	84.41
	Llama-3.1 8B Instruct	23.12	56.86	84.57
	Gemma-2 9B	7.52	56.09	85.34
	Gemma-2 9B Instruct	29.81	59.73	**85.63**

Table 2: Performance comparison of LLMS for ENG-HIN, ENG-SPA, ENG-MAL and ENG-TEL SA datasets showing 0-shot, 5-shot and FT(Finetuned) $F1$ Scores.

4.2 Hybrid Language Model (HLM)

In our HLM setup, sentence representations are extracted from the encoder and decoder models. These representations are fused and fed into a neural network, where the initial layer consists of $z_{enc}+z_{dec}$ neurons the final output layer consists of three neurons, each representing a sentiment class.

Training is performed using the Adam optimizer (Kingma and Ba, 2017). We also sweep over learning rate and number of training epochs, selecting the best configuration using validation loss.The

Dataset	Encoder	Decoder	Total Params	Encoder F_1	Decoder F_1	LateFusion F_1	HLM F_1
ENG-HIN	mBERT	Gemma-2 2B	2.72B	68.42	73.69	68.96	**74.54**
		Llama-3.2 3B	3.32B	68.42	69.82	68.45	73.36
		Sarvam-1	2.64B	68.42	67.86	68.59	72.79
	XLMR	Gemma-2 2B	2.88B	70.04	73.69	69.10	74.27
		Llama-3.2 3B	3.49B	70.04	69.82	69.65	73.21
		Sarvam-1	2.81B	70.04	67.86	68.73	72.44
ENG-SPA	mBERT	Llama-3.2 3B	3.32B	56.89	58.05	54.50	58.17
		Llama-3.2-1B	1.35B	56.89	57.89	53.93	58.16
		Gemma-2 2B	2.72B	56.89	54.91	54.56	57.01
	XLMR	Llama-3.2-1B	1.51B	57.21	57.89	52.91	**58.69**
		Llama-3.2 3B	3.49B	57.21	58.05	57.22	58.52
		Gemma-2 2B	2.88B	57.21	54.91	54.81	58.04
ENG-MAL	mBERT	Llama-3.2 3B	3.32B	73.85	73.19	73.78	**76.35**
		Sarvam-1	2.64B	73.85	73.28	74.46	74.78
		Gemma-2 2B	2.72B	73.85	72.82	73.15	74.77
	XLMR	Llama-3.2 3B	3.49B	70.38	73.19	72.58	75.18
		Gemma-2 2B	2.88B	70.38	72.82	69.78	74.01
		Sarvam-1	2.81B	70.38	73.28	70.41	73.86
ENG-TEL	mBERT	Llama-3.2 3B	3.32B	82.24	82.94	83.62	84.40
		Gemma-2 2B	2.72B	82.24	83.86	83.01	83.97
		Sarvam-1	2.64B	82.24	82.16	82.99	84.19
	XLMR	Sarvam-1	2.81B	83.33	82.16	83.83	**84.69**
		Llama-3.2 3B	3.49B	83.33	82.94	84.04	84.49
		Gemma-2 2B	2.88B	83.33	83.86	84.13	83.90

Table 3: presents a comparative analysis of different fusion strategies across model combinations for the ENG-HIN, ENG-SPA, ENG-MAL, and ENG-TEL datasets. For each dataset, the highest F1 score achieved by the HLM is highlighted in bold.

weight parameters of both the models and feedforward classifier (f_{cls}) are iteratively updated during the optimization process. Full hyperparameter details are provided in Appendix A.3. Performance is reported using the weighted F_1 score.

5 Results and Analysis

Table 2 and Table 3 present sentiment analysis results using LLMs and fusion-based techniques, respectively.

Table 2 shows that fine-tuning LLMs on task-specific data consistently outperforms zero-shot and few-shot settings. For instance, on the ENG-HIN dataset, the Gemma-2 9B model achieves the highest F1 score of 75.49 when fine-tuned. Similarly, Llama-3 8B performs best on ENG-SPA with an F1 score of 59.25, and Gemma-2 9B Instruct obtains the best result on ENG-TEL with an F1 score of 85.63. For ENG-MAL, Gemma-2 9B obtains an F1-Score of 76.21.

Encoder-based models like mBERT typically contain between 100 and 200 million parameters, whereas decoder-only LLMs generally range from 7 to 9 billion parameters. To strike a balance between performance and efficiency, we present results using a low-parameter Hybrid Language Model (HLM), which combines an encoder with a lightweight decoder LLM. This design provides a more parameter-efficient alternative to full LLM fine-tuning while maintaining strong performance on sentiment analysis tasks.

Table 3 reports results for our Hybrid Language Model (HLM) across various encoder–decoder model combinations, alongside the Late Fusion approach (Sharma et al., 2023), which combines predictions from two encoder-only models. The Encoder and Decoder columns list the specific models used in each HLM configuration and the individual fine-tuning results in the columns Encoder F1 and Decoder F1. Total Params indicates the combined parameter count of the encoder and decoder. The final column, HLM F1, reports the performance after fusing their representations in our hybrid architecture.

Table 3 shows that HLM consistently outperforms individual encoder and decoder models, as well as the Late Fusion baseline, across most configurations. On ENG-HIN, HLM achieves strong gains with mBERT and Gemma-2 2B, and performs competitively with the much larger Gemma-2 9B, achieving an F1 score of 74.54 versus 75.49. On ENG-SPA, HLM outperforms Late Fusion in all cases; the XLMR + Llama-3.2 1B setup reaches 58.69 F1, slightly poorer compared to Llama-3 8B

model's 59.25 despite having less than a quarter of the parameters. Sarvam-1 is omitted from this setting due to its Indian language focus.

For ENG-MAL, HLM outperforms all individual models, with the mBERT + Llama-3.2 3B pairing surpassing even Gemma-2 9B. On ENG-TEL, HLM achieves 84.69 F1 with XLMR + Sarvam-1, compared to Gemma-2 9B Instruct's 85.34 while using nearly 70% fewer parameters. HLM yields statistically significant results compared to individual models using a paired t-test.

The HLM architecture demonstrates a significant advancement over prior fusion techniques such as Late Fusion (Sharma et al., 2023), which are typically constrained to combining homogeneous models—most commonly, dual encoder-based systems. In contrast, HLM enables the integration of heterogeneous components, such as pairing a encoder with a lightweight decoder, thereby offering greater architectural flexibility. As evidenced by the results in Tables 2 and 3, large language models (LLMs) already demonstrate superior performance compared to Late Fusion. however, HLM further elevates this performance by effectively fusing sentence-level representations across diverse architectures. This not only mitigates the parameter overhead associated with scaling up LLMs but also achieves results on par with, or superior to, significantly larger models.

6 Conclusion

We propose a parameter-efficient Hybrid Language Model (HLM) for sentiment analysis on code-mixed text, combining an encoder with a lightweight decoder. HLM consistently outperforms individual models and the Late Fusion baseline, while achieving performance comparable to large-scale LLMs. Despite using only 1.5–3 billion parameters, HLM matches or exceeds models with 7–9 billion parameters in weighted F1 score across multiple benchmarks. These results highlight HLM as a practical alternative to full LLM fine-tuning, offering competitive accuracy with significantly lower computational cost.

7 Limitations

In this work, we focus exclusively on sentiment analysis due to computational resource constraints. Exploring the effectiveness of HLMs on other related tasks such as emotion or sarcasm detection is left for future work. Additionally, extending HLMs to handle sequence generation tasks presents an interesting and non-trivial challenge for future exploration.

8 Ethical Considerations

Deployments of sentiment analysis systems to code-mixed data pose deep ethical questions that go beyond technical performance criteria. Code-mixed groups—typically comprising marginalized linguistic minorities—are at risk of having their cultural forms commodified without their agreement or equitable representation in training sets that overwhelmingly sample urban, digitally engaged populations. Our HLM design, though competitive in performance with reduced parameters, inherits possible biases from multilingual encoders trained on unbalanced corpora and may encode stereotypes regarding code-switching linguistic ability. The interaction of two or more languages in individual utterances brings difficult privacy expectations, as users might expect varying degrees of confidentiality depending on their language options, while existing consent channels do not recognize these multilingual environments. As such systems gain a larger role in content moderation and social media analysis, we support community-based development practices that protect code-mixed sentiment analysis from being used to exploit the rich linguistic communities it analyzes, with the need for constant monitoring of bias, culturally-sensitive evaluation metrics, and meaningful interaction with impacted language communities throughout the system life cycle

References

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A Centralized Benchmark for Linguistic Code-switching Evaluation. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France. European Language Resources Association.

AI@Meta. 2024. Llama 3 model card.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020.

Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Bharathi Raja Chakravarthi, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John Philip McCrae. 2020a. A sentiment analysis dataset for code-mixed Malayalam-English. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 177–184, Marseille, France. European Language Resources association.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John Philip McCrae. 2020b. Corpus creation for sentiment analysis in code-mixed Tamil-English text. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 202–210, Marseille, France. European Language Resources association.

Niko Colnerič and Janez Demšar. 2020. Emotion recognition on twitter: Comparative study and training a unison model. *IEEE Transactions on Affective Computing*, 11(3):433–446.

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. *Preprint*, arXiv:1911.02116.

Richeek Das, Sahasra Ranjan, Shreya Pathak, and Preethi Jyothi. 2023. Improving pretraining techniques for code-switched NLP. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1176–1191, Toronto, Canada. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William W Cohen. 2016. Tweet2vec: Character-based distributed representations for social media. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 269–274.

Barah Fazili and Preethi Jyothi. 2022. Aligning multilingual embeddings for improved code-switched natural language understanding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4268–4273, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Muhammad Huzaifah, Weihua Zheng, Nattapol Chanpaisit, and Kui Wu. 2024. Evaluating code-switching translation with large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6381–6394, Torino, Italia. ELRA and ICCL.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Kartik Kartik, Sanjana Soni, Anoop Kunchukuttan, Tanmoy Chakraborty, and Md Shad Akhtar. 2024. Synthetic data generation and joint learning for robust code-mixed translation. *Preprint*, arXiv:2403.16771.

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.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Siva Subrahamanyam Varma Kusampudi, Preetham Sathineni, and Radhika Mamidi. 2021. Sentiment analysis in code-mixed Telugu-English text with unsupervised data normalization. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 753–760, Held Online. INCOMA Ltd.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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.

Aditya Pratapa et al. 2018. Title of the paper. *Journal Name*, 1(1).

Gemma Team Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, L'eonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ram'e, Johan Ferret, Peter Liu, Pouya Dehghani Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stańczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Boxi Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Christoper A. Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozi'nska, D. Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Pluci'nska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost R. van Amersfoort, Josh Gordon, Josh Lipschultz, Joshua Newlan, Junsong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, L. Sifre, Lena Heuermann, Leti cia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Gorner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Peng chong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Perrin, S'ebastien M. R. Arnold, Se bastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomás Kociský, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeffrey Dean, Demis Hassabis, Koray Kavukcuoglu, Clément Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving open language models at a practical size. *ArXiv*, abs/2408.00118.

Gagan Sharma, R Chinmay, and Raksha Sharma. 2023. Late fusion of transformers for sentiment analysis of code-switched data. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6485–6490, Singapore. Association for Computational Linguistics.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mgpt: Few-shot learners go multilingual. *arXiv preprint arXiv:2204.07580*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024a. Knowledge fusion of large language models. *Preprint*, arXiv:2401.10491.

Fanqi Wan, Haoyu Zheng, Yaqing Zhu, Jianguo Liu, Hao Peng, and Philip S Yu. 2024b. Knowledge fusion of large language models. *arXiv preprint arXiv:2401.10491*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Zheng Xin Yong, Ruochen Zhang, Jessica Forde, Skyler Wang, Arjun Subramonian, Holy Lovenia, Samuel Cahyawijaya, Genta Winata, Lintang Sutawika, Jan Christian Blaise Cruz, Yin Lin Tan, Long Phan, Long Phan, Rowena Garcia, Thamar Solorio, and Alham Fikri Aji. 2023. Prompting multilingual large language models to generate code-mixed texts: The case of south East Asian languages. In *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, pages 43–63, Singapore. Association for Computational Linguistics.

Ruochen Zhang, Samuel Cahyawijaya, Jan Christian Blaise Cruz, Genta Winata, and Alham Fikri Aji. 2023. Multilingual large language models are not (yet) code-switchers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12567–12582, Singapore. Association for Computational Linguistics.

A Appendix

A.1 Data Preprocessing

In the ENG-MAL dataset described in (Chakravarthi et al., 2020a), five sentiment

classes are defined. However, for our work, We preprocessed the ENG-MAL dataset by filtering out sentences labeled as not-Malayalam. The resulting data set contains instances classified into positive, negative and neutral sentiment classes.

A.2 Prompts for LLMs

the prompt utilized for SA training and validation data is shown in Figure 2 and for test data is shown in Figure 3.

> Analyze the sentiment of the news headline enclosed in square brackets. Determine if it is positive, neutral, or negative, and return the answer as the corresponding sentiment label: "positive", "neutral", or "negative".
>
> $$[\{text\}] = \{label\}$$

Figure 2: Prompt utilized for fine-tuning the LLMs on the train and validaion data for SA task

> Analyze the sentiment of the news headline enclosed in square brackets. Determine if it is positive, neutral, or negative, and return the answer as the corresponding sentiment label: "positive", "neutral", or "negative".
>
> $$[\{text\}] = \{\}$$

Figure 3: Prompt utilized for fine-tuning the LLMs on the test data for SA task

A.3 Hyper Parameters

We have performed hyper-parameter tuning for the learning rate (lr) and number of epochs, selecting the optimal values based on the validation set. We achieved lr: $1e^{-5}$ for encoder models, $5e^{-4}$ for LLMs and HLM. The number of epochs varied depending on the specific model used, with each model requiring a different amount of training time to converge effectively. We used Q-LoRA fine-tuning for the LLMs with the alpha, rank, and dropout parameters. We experimented with two different ranks, 16 and 64, to adjust the number of low-rank adapters used for fine-tuning. We tested alpha values of 16 and 32, which control the scaling factor for the low-rank adapters. We applied a dropout rate of 0.05 across all LLMs to help prevent overfitting during training. For late fusion, we have used the default values mentioned in (Sharma et al., 2023)

For HLM, we used a neural network comprising an input layer of size z_{enc} + z_{dec}, followed by three hidden layers with dimensions 768, 256, and 64 respectively, and an output layer of 3 neurons (one for each sentiment category). Dropout layers were also incorporated into the network to improve regularization In HLM For low-parameter LLM, we used Q-Lora with alpha: 32, rank: 64, dropout: 0.05. We have trained HLM using Adam Optimizer with lr: $1e^{-}5$ for ENG-SPA, ENG-HIN, ENGTEL and lr: $5e^{-}5$ for ENG-MAL. Table 4, 5, 6, 7 present the hyperparameters employed for various model combinations across all datasets. Table 8 represents number of trainable parameters for the HLM.

MODEL	batch	max len	epochs
(mBERT, Gemma-2-2b)	8	128	2
(XLMR, Gemma-2-2b)	4	256	2
(mBERT, Llama-3.2-3b)	4	256	3
(mBERT, Sarvam-1)	4	128	3
(mBERT, Sarvam-0.5)	8	128	3
(XLMR, Sarvam-1)	4	256	2
(XLMR, Llama-3.2-3b)	4	256	2

Table 4: Hyperparameters for ENG-HIN data

MODEL	batch	max len	epochs
(BERT, Gemma-2-2b)	8	128	5
(BERT, Llama-3.2-1b)	4	128	4
(BERT, Llama-3.2-3b)	4	256	5
(XLMR, Gemma-2-2b)	4	256	2
(XLMR, Llama-3.2-3b)	4	256	4
(XLMR, Llama-3.2-1b)	4	256	4

Table 5: Hyperparameters for ENG-SPA data

MODEL	batch	max len	epochs
(XLMR, Llama-3.2-3b)	4	128	5
(mBERT, Llama-3.2-3b)	4	128	5
(mBERT, Gemma-2-2b)	4	128	5
(XLMR, Gemma-2-2b)	4	128	4
(XLMR, Sarvam-1)	4	128	2
(mBERT, Sarvam-1)	4	128	3

Table 6: Hyperparameters for ENG-MAL data

MODEL	batch	max len	epochs
(XLMR, Llama-3.2-3b)	4	256	2
(mBERT, Llama-3.2-3b)	4	128	3
(mBERT, Gemma-2-2b)	4	128	3
(XLMR, Gemma-2-2b)	6	128	2
(XLMR, Sarvam-1)	4	128	2
(mBERT, Sarvam-1)	4	128	2

Table 7: Hyperparameters for ENG-TEL data

MODELS	Trainable Params
(XLMR, Llama-3.2-3b)	$376M$
(mBERT, Llama-3.2-3b)	$207M$
(mBERT, Gemma-2-2b)	$193M$
(XLMR, Gemma-2-2b)	$364M$
(XLMR, Sarvam-1)	$365M$
(mBERT, Sarvam-1)	$205M$
(XLMR, Llama-3.2-1b)	$324M$
(mBERT, Llama-3.2-1b)	$155M$

Table 8: Table represents number of trainable parameters for HLM

A.4 Computation and Memory Stats

For the ENG-HIN, ENG-TEL, and ENG-SPA datasets, the HLM model required an average training time of approximately 16.57 minutes per epoch, whereas the large LLMs took around 40 minutes per epoch. In the case of ENG-MAL, which has fewer training instances, HLM trained significantly faster, averaging 4.01 minutes per epoch compared to 8.24 minutes for the LLMs. Additionally, HLM was more resource-efficient, requiring 24 GB of GPU memory, while the large LLMs needed 40 GB.

A.5 Device Specifications

All experiments were conducted on a workstation equipped with an NVIDIA GeForce RTX A5000 GPU with 24GB of memory, utilizing CUDA for parallel processing .