Project Report: Code-Mixed Urdu–English Sentiment Analysis Using Linguistically-Aware Transformer Models

Ibtesam Hussain, Ayan Khan, Safey Ahmed, Shaheer Uddin
Department of Artificial Intelligence
FAST National University of Computer and Emerging Sciences
Karachi, Pakistan
22K4125@nu.edu.pk, 22K8720@nu.edu.pk, 22K4039@nu.edu.pk, 22K8719@nu.edu.pk

***Abstract*—This report summarizes the project proposal and the current status of implementation for a linguistically-aware sentiment analysis system for Roman-script Urdu–English code-mixed text. The project integrates Roman Urdu normalization and token-level language identification (LID) with transformer-based models (XLM-R and byte-level transformers) and evaluates these methods on publicly available RU–EN datasets. This document highlights what was proposed, what has been implemented so far, obtained results, limitations, and planned next steps.**

***Index Terms*—Code-mixed NLP, Roman Urdu, Sentiment Analysis, XLM-R, Byte-level Models, Language Identification**

I. Introduction

This project aims to build a robust sentiment/emotion classification pipeline for Roman Urdu–English code-mixed social media text. The motivation stems from the high orthographic variability and token fragmentation typical of Roman Urdu which challenges multilingual transformer models. The proposed approach augments a transformer backbone with preprocessing (normalization) and a token-level LID auxiliary objective to improve performance and interpretability.

II. Problem Statement

Although multilingual transformers like mBERT, XLM-RoBERTa, and Indic-BERT have achieved promising results in multilingual NLP, they struggle with code-mixed Roman Urdu because:

- Roman Urdu is not part of the pre-training corpora of these models.
- High variation in spellings causes token fragmentation and reduces subword overlap.
- No integration of token-level language tags or normalized representations exists in current models.

This results in degraded performance for sentiment analysis on real-world Roman Urdu–English social media text.

Goal: To design and implement a linguistically aware sentiment analysis model that explicitly incorporates Roman Urdu normalization and token-level language identification within a transformer framework.

III. Objectives

The main objectives of the proposed research are:

1) To construct or adapt a dataset of Urdu–English code-mixed text with sentiment labels.
2) To develop preprocessing modules for Roman Urdu normalization and token-level language identification (LID).
3) To fine-tune a multilingual transformer model (XLM-RoBERTa / Indic-BERT) on this data.
4) To evaluate the model using standard NLP metrics and compare it against baseline models.
5) To analyze improvements introduced by LID and normalization components.

IV. What was proposed

The original proposal (see attached proposal document) outlined the following key components:

- Construct or adapt code-mixed datasets (MultiSenti, RU–EN-Emotion, LISACMT) with sentiment/emotion labels.
- Develop preprocessing modules for cleaning, Roman Urdu normalization, and token-level LID.
- Fine-tune multilingual transformer models (XLM-RoBERTa / Indic-BERT) with an auxiliary LID head, training jointly with sentiment labels.
- Evaluate using accuracy, precision, recall, F1, and confusion matrices; compare with baseline classifiers.
- Explore byte-level architectures (ByT5 / CANINE) to handle subword fragmentation in Romanized text.

V. Literature Review

Past research has addressed sentiment analysis in code-mixed settings using traditional ML and transformer models:

- Shakeel and Karim's MultiSenti corpus [10] introduced Roman Urdu–English tweets for sentiment classification.

- Younas et al. [1] and Hashmi et al. [2] applied multilingual transformers like mBERT and XLM-RoBERTa.
- Khan et al. [13] extended the task to emotion detection with RU–EN-Emotion dataset.
- Hussain et al. [14] proposed a token-level LID model but did not integrate it with sentiment classification.

Research Gap:

- No existing work integrates LID and normalization within a transformer framework.
- Roman Urdu normalization remains largely unexplored in deep models.
- Lack of joint training for LID and sentiment analysis tasks.

VI. What has been implemented so far

The following tasks have been implemented in the provided Jupyter notebook (`NLP Project.ipynb`):

A. Data acquisition and preprocessing

- Downloaded and loaded the RU–EN dataset using `kagglehub` and `pandas`.
- Implemented cleaning steps: removal of URLs, mentions, hashtags, emojis; lowercasing; removal of non-letter characters; reduction of elongated characters; whitespace normalization.
- Implemented a comprehensive Roman Urdu *normalization dictionary* (many common variants, slang, intensifiers, negations, pronouns, and common tokens) and a normalization function to map noisy tokens to canonical forms.
- Created label mapping and stratified train/test split.

B. Modeling experiments

Three modeling lines are present in the notebook:

1) Baseline: XLM-R fine-tuning:

- Tokenization with `XLMRobertaTokenizer`, dataset prepared with the HuggingFace `datasets` API.
- Model: `XLMRobertaForSequenceClassification` fine-tuned for 3 epochs (learning rate 2e-5, batch size 16 as implemented).
- Sample evaluation metrics printed in the notebook: **accuracy ≈ 0.703, F1 ≈ 0.685**. The notebook includes code to save the trained model and tokenizer.

2) LID-aware XLM-R (multi-task):

- Built a Roman Urdu lexicon set and implemented a heuristic `get-lid-tags` function to label tokens as Roman Urdu vs English.
- Tokenization routine aligns word-level LID tags to subword tokens (a simple and brittle alignment that was attempted via decoding tokens).
- Implemented `LIDAwareXLMR` model with two heads: sentiment (sequence-level) and LID (token-level), with combined loss: $L_{total} = L_{SA} + 0.5L_{LID}$.
- Custom trainer `LIDTrainer` for evaluation using sentiment logits only.
- Sample evaluation metrics printed in the notebook: **accuracy ≈ 0.696, F1 ≈ 0.472**. The notebook notes LID supervision was noisy and the LID approach did not improve F1 in the current implementation.

3) Byte-level Model (ByT5 / CANINE):

- Prepared an experiment pipeline using a byte-level model (e.g., `google/byt5-base`) for tokenization at byte level and classification.
- Converted data to HuggingFace Datasets and tokenized for ByT5. The training cell is present and marked to run on a GPU ("NEEDS HEAVY COMPUTATION GPU TO RUN...WILL IMPLEMENT THIS IN NEAR FUTURE"). No final evaluation metrics are present in the notebook for this line since the training was not executed there.

VII. Obtained results and observations

- The baseline XLM-R model achieves reasonable performance for the task (sample printed $eval_{f1}$ 0.685).
- The LID-aware multi-task model, as implemented, produced lower F1 (sample 0.472), indicating that the current LID supervision (lexicon + naive alignment) is noisy and can hurt joint training unless improved.
- Byte-level experiments are prepared; their performance is pending GPU execution.

VIII. Limitations and issues found

- LID labeling is simplistic and sometimes misaligns with subword tokenization, producing noisy auxiliary labels.
- No per-class metrics or confusion matrices were produced yet; class-wise weaknesses are unexamined.
- No hyperparameter search, CV, or repeated seed experiments have been executed yet to quantify variance.
- Byte-level model training requires GPU resources and has not been evaluated yet.

IX. Planned next steps (ongoing work)

1) Improve token-level LID labels: use $tokenizer.word_{ids}()$ for robust alignment or a small annotated seed to train a proper LID tagger and then use it as supervision.
2) Re-run LID-aware multi-task training with improved LID supervision and experiment with task weighting (tune λ_2).
3) Execute byte-level model training on an NVIDIA GPU (ByT5 / CANINE) and compare against XLM-R baseline.
4) Add per-class metrics, confusion matrices, and training curves; log experiments with WB or MLflow.
5) Explore class balancing (loss weighting, oversampling, and data augmentation) if needed.

X. Conclusion

The project’s proposal and implementation so far align well: the core preprocessing pipeline and multiple modeling experiments are implemented and documented in the notebook. Preliminary results show a working baseline and indicate that LID auxiliary supervision requires better label quality to be beneficial. Continuing work will focus on improving LID labeling, running byte-level experiments on GPU, and expanding the evaluation and reporting.

Acknowledgment

The authors acknowledge the dataset creators and HuggingFace tools used in experimentation.

References

[1] A. Younas, R. Nasim, S. Ali, G. Wang, and F. Qi, “Sentiment Analysis of Code-Mixed Roman Urdu-English Social Media Text using Deep Learning Approaches,” 2020.

[2] E. Hashmi, S. Y. Yildirim, and S. Shaikh, “Augmenting sentiment prediction capabilities for code-mixed tweets with multilingual transformers,” Social Network Analysis and Mining, vol. 14, pp. 1-15, 2024. doi: 10.1007/s13278-024-01245-6.

[3] M. K. Nazir, C. N. Faisal, M. A. Habib, and H. Ahmad, “Leveraging Multilingual Transformer for Multiclass Sentiment Analysis in Code-Mixed Data of Low-Resource Languages,” IEEE Access, vol. 13, pp. 7538-7554, 2025. doi: 10.1109/ACCESS.2025.3527710.

[4] H. Z. Ali and A. R. Chaudhry, “Anti-social Behavior Detection using Multi-lingual Model,” 2023 4th International Conference on Advancements in Computational Sciences (ICACS), pp. 1-9, 2023. doi: 10.1109/ICACS55311.2023.10089659.

[5] A. Ilyas, K. Shahzad, and M. K. Malik, “Emotion Detection in Code-Mixed Roman Urdu - English Text,” ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 22, pp. 1-28, 2022. doi: 10.1145/3552515.

[6] G. I. Ahmad and J. Singla, “(LISACMT) Language Identification and Sentiment analysis of English-Urdu ‘code-mixed’ text using LSTM,” 2022 International Conference on Inventive Computation Technologies (ICICT), pp. 430-435, 2022. doi: 10.1109/ICICT54344.2022.9850505.

[7] N. A. Found, “‘Adapting Machine Learning And Deep Learning Approach Towards Language Identification And Sentiment Analysis Of Code-Mixed Urdu-English And Hindi-English Social Media Text ‘,” Unknown Journal.

[8] I. Ameer, G. Sidorov, H. Gómez-Adorno, and R. M. A. Nawab, “Multi-Label Emotion Classification on Code-Mixed Text: Data and Methods,” IEEE Access, vol. 10, pp. 8779-8789, 2022. doi: 10.1109/ACCESS.2022.3143819.

[9] B. H. Vedula, P. Kodali, M. Shrivastava, and P. Kumaraguru, “PrecogIIITH@WASSA2023: Emotion Detection for Urdu-English Code-mixed Text,” Unknown Journal, pp. 601-605, 2023. doi: 10.18653/v1/2023.wassa-1.58.

[10] M. Shakeel and A. Karim, “Adapting deep learning for sentiment classification of code-switched informal short text,” Proceedings of the 35th Annual ACM Symposium on Applied Computing, 2020. doi: 10.1145/3341105.3374091.

[11] M. Bilal, A. Khan, S. Jan, S. Musa, and S. Ali, “Roman Urdu Hate Speech Detection Using Transformer-Based Model for Cyber Security Applications,” Sensors, vol. 23, 2023. doi: 10.3390/s23083909.

[12] M. Zain, N. Hussain, A. Qasim, G. Mehak, F. Ahmad, G. Sidorov, and A. Gelbukh, “RU-OLD: A Comprehensive Analysis of Offensive Language Detection in Roman Urdu Using Hybrid Machine Learning, Deep Learning, and Transformer Models,” Algorithms, vol. 18, 396, 2025. doi: 10.3390/a18070396.

[13] Khan et al., “RU–EN-Emotion: A Dataset for Emotion Detection in Roman Urdu–English Code-Mixed Text,” 2022.

[14] Hussain et al., “LISACMT: Token-Level Language Identification for English–Urdu Code-Mixed Text,” 2020.