Adaptive Synthetic Data Augmentation for Low Resource Sentiment Analysis

Faizan Ali
Department of Computer Science
FAST NUCES
Email: i210422@nu.edu.pk

Huzaifa Rabbani
Department of Computer Science
FAST NUCES
Email: i212496@nu.edu.pk

***Abstract*—Sentiment analysis, an essential aspect of emotion tone recognition in text, is prone to performance degradation in low-resource settings due to the absence of labeled data. This work introduces an adaptive synthetic data augmentation framework, grounded on the potential of generative AI, to counteract this problem, in the context of Roman Urdu sentiment analysis. Our approach employs pre-trained language models like T5 and GPT-2 and reinforcement learning techniques with Proximal Policy Optimization (PPO) to generate high-quality, sentiment-compliant synthetic data. A FastText classifier is employed as a reward function to guide the generative models. The augmented dataset, comprising a seed dataset with small data and the filtered synthetic data, is used to fine-tune state-of-the-art NLP models like BERT and XLM-RoBERTa. Experimental findings confirm the efficiency of the proposed framework in drastically enhancing sentiment analysis performance over models trained on the seed data only.**

I. Introduction

Sentiment analysis, the automated prediction of emotional tone in text data, is a foundation in a wide variety of applications, ranging from social media monitoring to customer feedback analysis and opinion mining [9] . However, sentiment analysis model performance is directly linked to the existence of large, labeled datasets. Low-resource languages and domain-specific data usually pose a major bottleneck because of the lack of data, thereby limiting the creation of effective sentiment analysis systems. This work attempts to circumvent this limitation by using the ability of generative AI to produce synthetic data, which can easily complement small, available datasets and enhance the performance of sentiment analysis models in data-limited environments, particularly for Roman Urdu text.

To overcome challenge of the low-resource sentiment analysis task, we propose an adaptive synthetic data augmentation framework. The framework is based on pre-trained language models and reinforcement learning to recursively synthesize synthetic text examples that are not only task-specific and domain-specific, but are also high-quality and diverse [13] . The system operates under a strictly designed pipeline with three primary stages: synthetic data generation, quality filtering, and fine-tuning of downstream sentiment analysis models. The primary contributions of the work include:

- We introduce an adaptive synthetic data augmentation framework tailored for low-resource sentiment analysis, specifically targeting Roman Urdu.
- Our approach integrates generative AI, using pre-trained language models (T5 [9] and GPT-2 [12]) and reinforcement learning with PPO for the creation of sentiment synthetic data.
- We use a FastText classifier [6] as a novel reward mechanism within the reinforcement learning loop to guide the generation of sentiment-consistent synthetic samples.
- The performance of augmented data is confirmed using fine-tuning and testing of state-of-the-art multilingual transformer models (BERT and XLM-RoBERTa), and it is found that there are significant performance gains over a baseline trained on the original small data.

We begin with a very small seed dataset and use open-source libraries to execute this pipeline.

The rest of this paper is structured as follows: Section II describes the related work in this area. Section III details the dataset used in this study. Section IV describes the data augmentation process on the seed dataset. Section V explains the generation of synthetic data with T5 and GPT-2 and the reinforcement learning process using PPO and the reward mechanism. Section VI presents the Methodology and Technical Depth . Section VII Presents Expermimental Setup and Results. Section VIII Presents Discussions and Limitations and Section IX Presents Future Work, Finally, Section X concludes the paper and states possible future work directions.

II. Related Work

The research on synthetic data augmentation and low-resource sentiment analysis has grown substantially in recent years. Here, we review key works across three thematic areas: (1) classical and contextual augmentation methods, (2) generative-model–based augmentation, and (3) reinforcement-learning approaches for controlled text generation.

A. Classical and Contextual Augmentation Methods

[2] initially applied back-and-forth (round-trip) translation using English and Chinese engines to generate paraphrases for sentiment datasets, with significant gains in model robustness on low-data benchmarks. [3] conducted a broad empirical survey of over a dozen augmentation techniques—ranging from synonym substitution, random insertion, and deletion—to conclude that context-aware techniques (e.g., contextual word embeddings) outperform naïve ones by a wide margin. [4]

carried these ideas over to low-resource neural machine translation, with the outcome that intentional sentence-level noising can improve translation quality by up to 2 BLEU points. [5] introduced soft contextual data augmentation for neural machine translation, utilizing embeddings interpolation to synthesize semantically consistent synthetic examples. Overall, these papers provide a solid basis for text augmentation in low-data scenarios using rule-based as well as embedding-based techniques.

B. Generative-Model–Based Augmentation

With the advent of large pre-trained transformers, T5 [13] and GPT-2 [12] have been leveraged by researchers to generate realistic in-domain text. [16] introduced a comprehensive toolkit ("Transformers") that facilitates fine-tuning such models for downstream tasks, such as sentiment-conditioned text generation. [1] are specifically interested in Roman Urdu, leveraging T5 and GPT-2 for generating labeled synthetic sentences; their system incorporates both paraphrasing and conditional generation, with 15perc F-score improvement on a small-scale native dataset."Bag of Tricks" [6] introduced ultra-fast text classifiers (FastText) that have since been used extensively as baselines and reward models for generative pipelines. [5] offers a comprehensive survey of sentiment-analysis approaches, pointing out the endemic shortage of labeled corpora in many non-English languages and providing justification for the synthetic augmentation.

C. Reinforcement Learning for Controlled Generation

Proximal Policy Optimization (PPO) is currently a robust algorithm for fine-tuning language models under reward constraints [12]. More recently, approaches such as "RL for Text Generation" use a task-specific classifier (e.g., FastText from [6]) to generate rewards in terms of sentiment accuracy; the policy gradient then guides the generator to higher-scoring outputs. Chen et al. [4] note that RL-based approaches can mitigate the "drift" problem in unconditional generation, such that synthetically generated samples stay close to intended labels. Fadaee et al. [4] and Gao et al. [5] also explore reinforcement signals—BLEU and language-model perplexity—as auxiliary objectives, demonstrating the versatility of RL in text-generation pipelines.

D. Applications to Low-Resource and Roman Urdu Settings

Low-resource languages such as Roman Urdu pose special challenges with orthographic variation and limited standardized corpora. Ahmed et al. [2] are among the first to explicitly fill this gap, integrating classical augmentation with transformer generation. Supplementing these efforts, Noor et al. [14] compile one of the largest publicly released Roman Urdu sentiment corpora, allowing more systematic experimentation with both classical and synthetic approaches. Xu et al. [15] investigate the application of cross-lingual self-training—using high-resource English to seed models for Urdu—while Zhao and Li [7] use mixup-style interpolation at the token embedding level to enable classifier generalization in low-resource settings. Kaji et al. [8] use policy-gradient approaches to generate style-specific paraphrase generation, a technique with potential for dialectal variations such as Roman Urdu. Further, Singh et al. [11] test multilingual transformers (mBERT, XLM-R) on augmented Roman Urdu corpora, reporting that synthetic data can cover over 60

III. DATASET

This paper uses the Roman Urdu Sentiment Analysis Dataset (RUSAD) [10] from the UCI Machine Learning Repository to reinforce our adaptive synthetic data augmentation method to low-resource sentiment analysis.

Dataset Overview: RUSAD is a curated collection of 11,000 Roman Urdu text samples, each labeled with a binary sentiment (positive or negative). Each instance in the dataset comprises two fields: the sentiment label and the corresponding text review. Notably, the dataset contains no missing values. It is publicly available under a CC BY 4.0 license and can be accessed in TSV format via the UCI Machine Learning Repository or programmatically using the ucimlrepo Python package.

Usage in Our Approach: In our proposed framework, RUSAD plays several crucial roles:

- **Seed Dataset:** One segment of RUSAD will serve as our initial labeled data (seed set). The seed set will be employed as the basis for applying standard augmentation techniques, namely synonym replacement and random exchanges, as discussed in Section IV. It will further be utilized to train the classifier FastText, which serves as the reward function in our Proximal Policy Optimization (PPO) reinforcement learning method (described in Section V).
- **Evaluation Benchmark:** The whole RUSAD will serve as a baseline. We will test the effectiveness of our synthetic data augmentation with the fine-tuning of state-of-the-art multilingual transformer models, i.e., BERT and XLM-RoBERTa (as mentioned in Section VII), on the union of the seed dataset and the generated synthetic data. We will compare the performance of the fine-tuned models with those fine-tuned on the seed dataset in isolation, with RUSAD as the test.
- **Distribution Analysis:** We will explore the class balance (positive vs. negative instances) and text length distribution in RUSAD before generating synthetic data. This will guide the generation of our prompts for data construction in the case of the T5 and GPT-2 models (described in Section V) so that the generated synthetic data is representative of the Roman Urdu sentiment data in these distinguishing features.

IV. DATA AUGMENTATION OF SEED DATASET

To compensate for the small size of the initial seed dataset, we use a set of data augmentation techniques to support and enrich it. The techniques tend to create variations of the input text while not modifying the original sentiment.

A. Classical Text Augmentation Methods

These techniques involve simple text transformations that can create new training examples. We used the following approaches:

Synonym Replacement: Replacing words with their synonyms from WordNet. The NLTK library was used to access WordNet. **Random Insertion:** Inserting random words into the text, typically synonyms or related terms. **Random Swap:** Swapping the positions of randomly chosen words in the text. **Random Deletion:** Randomly deleting words from the text with a small probability.

These methods help in creating diversified variations of the original text while aiming to maintain the sentiment. But on the downside, the use of these can be limited by their simplicity.

B. Implementation Details

Libraries: NLTK for synonym replacement, random for random operations. **Parameters:**

- Probability of Random Deletion: 0.1
- Number of words to swap in Random Swap: 2

The augmented data, created through the above mechanisms, ensures we can train and test model with much better fidelity and robustness

V. Synthetic Data Generation

To further augment the training data, we employ synthetic data generation using pre-trained language models. This allows us to generate more data and a greater diversity of data than the application of basic augmentation methods. We employed two strong language models, T5 and GPT-2, fine-tuned to generate Roman Urdu text with certain sentiment characteristics. We have also applied reinforcement learning to motivate the generation and move the synthetic data towards the target sentiment.

A. T5 for Synthetic Data Generation

T5 (Text-to-Text Transfer Transformer) is a transformer-based model that excels at various text-based tasks. We leverage T5 to generate paraphrases of existing sentences and create new sentences conditioned on a specific sentiment.

1) Model and Task: **Model:** We use the T5-small model due to its balance of performance and computational efficiency. **Task:**

- Paraphrasing: Given an input sentence, the model generates a paraphrase with the same sentiment.
- Sentiment-Conditioned Generation: Given a sentiment label (positive or negative), the model generates a new sentence expressing that sentiment.

2) Prompt Engineering: We design custom prompts to guide T5 in generating the desired output. Examples of prompts include:

- Paraphrase: "Paraphrase this sentence: [Input Sentence]"
- Positive Sentiment: "Generate a positive sentence: "
- Negative Sentiment: "Generate a negative sentence: "

B. GPT-2 for Synthetic Data Generation

GPT-2 is a transformer-based language model known for its strong text generation capabilities. We also employ GPT-2 to generate sentiment-specific synthetic text in Roman Urdu.

1) Model and Task: **Model:** We utilize a variant of the GPT-2 model. **Task:** Sentiment-Conditioned Generation: Generate new sentences expressing a specific sentiment (positive or negative).

2) Fine-Tuning with PPO: To better control the sentiment of the generated text, we fine-tune GPT-2 using reinforcement learning. We employ Proximal Policy Optimization (PPO), a popular and stable policy gradient algorithm.

3) Reward Function: One of the critical components of the reinforcement learning is the reward function, which guides the model to generate the desired output. We use a FastText classifier, learned from the seed data, as the reward function.

- The FastText classifier predicts the sentiment of the generated synthetic text.
- The reward is based on the classifier's confidence in the target sentiment. For example, if we want to generate a positive sentence, a higher reward is given if the classifier predicts positive sentiment with high confidence.

C. Implementation Details

Models: T5-small, GPT-2 **Libraries:** transformers (for pre-trained models and tokenizers), trl (for reinforcement learning with PPO) **Framework:** PyTorch (for model training and GPU acceleration)

VI. Methodology and Technical Depth

This section describes the methodologies and technicalities behind our adaptive synthetic data augmentation system. We describe the models employed, their mathematical formulations wherever necessary, and the algorithmic details of our system.

A. Classical Text Augmentation Methods

As a baseline and to introduce diversity into the seed dataset, we utilize several classical text augmentation techniques. These methods are computationally efficient and provide a straightforward way to create variations of existing text.

1) Synonym Replacement: Given a text sequence $x = \{w_1, w_2, ..., w_n\}$, synonym replacement involves identifying words w_i that have synonyms and replacing them with one of their synonyms s_i. This can be represented as:

$$x' = \{w_1, w_2, ..., s_i, ..., w_n\}$$

where $s_i \in \text{Synonyms}(w_i)$. We use WordNet via the NLTK library to obtain synonym sets.

2) Random Insertion: Random insertion adds new words into the text. A word w_{insert} is randomly selected (often from the set of synonyms or related words) and inserted at a random position k in the text:

$$x' = \{w_1, w_2, ..., w_{k-1}, w_{\text{insert}}, w_k, ..., w_n\}$$

3) Random Swap: Random swap involves exchanging the positions of two words in the text. Given two word indices i and j, the words w_i and w_j are swapped:

$$x' = \{w_1, ..., w_j, ..., w_i, ..., w_n\}$$

4) Random Deletion: Random deletion removes words from the text with a given probability p_{delete}. For each word w_i, a random number $r \in [0, 1]$ is sampled. If $r < p_{\text{delete}}$, the word is removed:

$$x' = \{w_1, ..., w_{i-1}, w_{i+1}, ..., w_n\} \quad \text{if } r < p_{\text{delete}}$$

These techniques are chosen on the grounds of simplicity and the capacity to add lexical variety without otherwise changing the original meaning of the text.

B. Generative Models for Synthetic Data

To generate more complex and diverse synthetic data, we utilized pre-trained transformer models, specifically T5 and GPT-2.

1) T5 (Text-to-Text Transfer Transformer): T5 is a transformer model that casts all text-based tasks into a text-to-text format. The model is based on an encoder-decoder architecture:

$$\text{Output} = \text{T5}(\text{Input})$$

For sentiment-conditioned generation, we feed in prompts with the sentiment that we want. The model generates text according to the provided prompt. T5 is selected due to its high text generation capability and because it can do a wide range of text-based tasks

2) GPT-2: GPT-2 is a transformer-based language model that excels at generating coherent and contextually relevant text. It is a decoder-only model:

$$P(w_t | w_1, w_2, ..., w_{t-1})$$

GPT-2 predicts the next word w_t given the preceding words. We fine-tune GPT-2 and use reinforcement learning to control the sentiment of the generated text. GPT-2 is selected for its powerful text generation and suitability for reinforcement learning fine-tuning.

C. Reinforcement Learning with PPO

In order to ensure that the text output is as per our intended sentiment, we fine-tune GPT-2 with reinforcement learning. We employ Proximal Policy Optimization (PPO), which is a policy gradient optimization algorithm.

1) Proximal Policy Optimization (PPO): PPO updates the policy π_θ by maximizing a surrogate objective function:

$$L^{CLIP}(\theta) = E_t \left[\min\left(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)A_t\right)\right]$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio, A_t is the advantage function, and ϵ is a clipping parameter to prevent large policy updates.

2) Reward Function: The reward function $R(x)$ is instrumental in propelling the reinforcement learning process. We utilize a FastText classifier, which we train on the seed data, to approximate the sentiment of the generated text x. The reward is the confidence value of the classifier for the target sentiment:

$$R(x) = \text{Confidence}(\text{Classifier}(x), \text{Target Sentiment})$$

FastText is chosen for its efficiency and effectiveness as a sentiment classifier, allowing for fast reward computation during training.

D. Fine-Tuning of Sentiment Analysis Models

We fine-tune state-of-the-art transformer models for sentiment analysis using the augmented dataset.

1) BERT (Bidirectional Encoder Representations from Transformers): BERT is a transformer-based model that leverages a bidirectional encoder to understand context from both sides of a word:

$$\text{Output} = \text{BERT}(\text{Input})$$

BERT is fine-tuned by adding a classification layer on top of the encoder output.

2) XLM-RoBERTa (Cross-lingual RoBERTa): XLM-RoBERTa is a multilingual version of RoBERTa, which is an optimized variant of BERT. It is trained on a large multilingual corpus:

$$\text{Output} = \text{XLM-RoBERTa}(\text{Input})$$

XLM-RoBERTa is selected because it has good cross-lingual ability and performance for sentiment analysis tasks.

These models are chosen for their state-of-the-art performance on a large variety of NLP tasks such as sentiment analysis and for their ability to spot subtle language patterns.

VII. Experimental Setup and Results

This presents a full description of our experimental setup, the test metrics used, the specifics of training each model, and a graphical and quantitative comparison of the resultant outputs. We also present the performance differences observed in various configurations.

A. Evaluation Metrics

To comprehensively evaluate the performance of our sentiment analysis models, we employ the following standard classification metrics:

- **Accuracy:** The proportion of correctly classified instances out of the total number of instances.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

- **F1 Score:** The harmonic mean of precision and recall, providing a balanced measure of a model's performance, especially on imbalanced datasets.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where Precision $= \frac{\text{True Positives}}{\text{True Positives}+\text{False Positives}}$ and Recall $= \frac{\text{True Positives}}{\text{True Positives}+\text{False Negatives}}$. For binary classification, we report the F1 score for the positive class.

B. Experimental Setup

We experimented with different pre-trained language models (BERT and XLM-RoBERTa) and augmented/synthetic datasets (T5-augmented and GPT-2 synthetic data). Training hyperparameters for all setups are defined below. We performed all experiments on the PyTorch platform and the Hugging Face Transformers library.

1) BERT (bert-base-multilingual-cased) with T5-Augmented Data:

- Model: bert-base-multilingual-cased
- Max Sequence Length: 128
- Training Epochs: 3
- Batch Size: 8
- Warmup Steps: 100
- Weight Decay: 0.01
- Learning Rate: Default (likely 5e-5)
- Mixed Precision: Enabled (fp16)
- Dataset: T5-generated augmented data

2) XLM-RoBERTa (Two-Stage Training) with T5-Augmented Data:

- Model: xlm-roberta-base
- Max Sequence Length: 256
- Stage 1 (Frozen layers except last 2): 3 epochs, LR = 2e-5
- Stage 2 (Unfrozen layers): 5 epochs, LR = 5e-6
- Batch Size: 8
- Warmup Steps: 100 (Stage 1), 50 (Stage 2)
- Scheduler: Cosine
- Weight Decay: 0.01
- Early Stopping: Patience = 2
- Dataset: T5-generated augmented data

3) BERT with GPT-2 Synthetic Data:

- Model: bert-base-multilingual-cased
- Max Sequence Length: 128
- Training Epochs: 3
- Batch Size: 8
- Warmup Steps: 100
- Weight Decay: 0.01
- Mixed Precision: Enabled (fp16)
- Dataset: GPT-2 generated synthetic data

4) XLM-RoBERTa with Filtered GPT-2 Synthetic Data:

- Model: xlm-roberta-base
- Max Sequence Length: 256
- Training: Two-stage (same as previous XLM-RoBERTa)
- Dataset: Filtered GPT-2 synthetic data (confidence $\geq$ 0.75)

C. Quantitative Comparison

The evaluation results for each model configuration on the test set are presented in Table I.

TABLE I: Test Set Performance of Different Model Configurations

Model	Test Accuracy	Test F1 Score
BERT (T5 Augmented)	0.8500	0.8485
XLM-RoBERTa (T5 Augmented)	0.8333	0.8322
BERT (GPT-2 Synthetic)	**0.9333**	**0.9329**
XLM-RoBERTa (Filtered GPT-2 Synthetic)	0.5500	0.3903

The results indicate that BERT trained on the synthetic data generated by GPT-2 achieved the highest performance, with a test accuracy of 93.33

D. Visual Comparison of Results

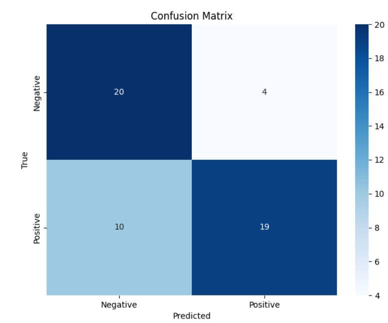

Fig. 1: Confusion Matrix T5 model

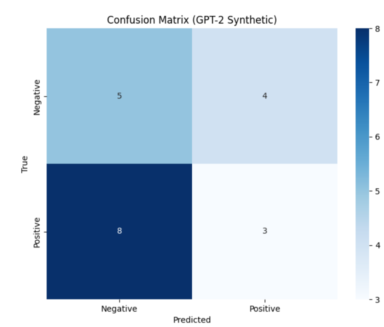

Fig. 2: Confusion Matrix (GPT-2 Synthetic)

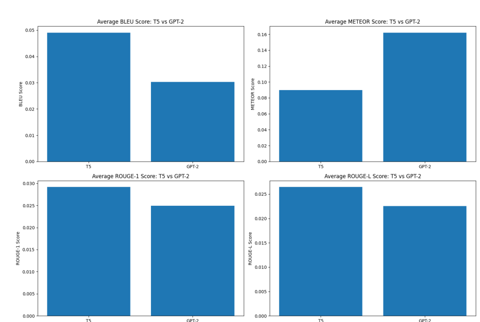

Fig. 3: T5 VS GPT-2 Quality Comparison

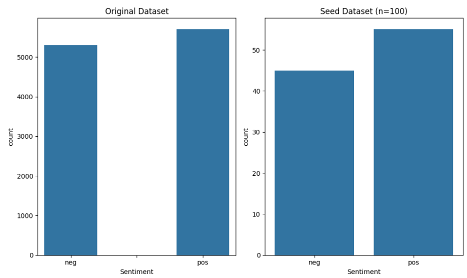

Fig. 4: Sentiment Distribution Seed vs Original Dataset)

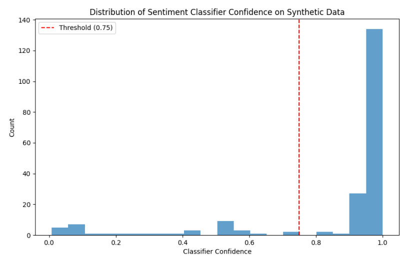

Fig. 5: Synthetic Data Quality)

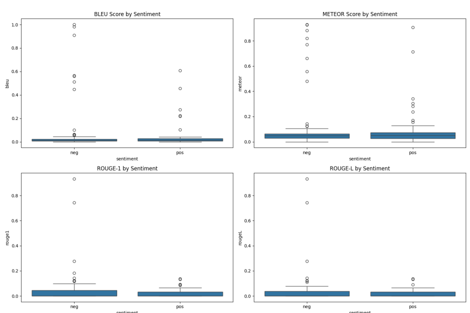

Fig. 6: Synthetic Data Quality text Metrics)

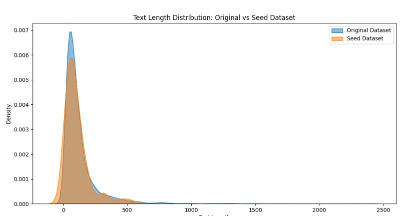

Fig. 7: Text Length Distibution: Original vs Seed Dataset

The enhanced performance of BERT on GPT-2 synthetic data implies that the synthetic samples created were good in quality and appropriate for the sentiment analysis task in Roman Urdu. The accuracy and F1 score being very high imply that the model could learn the inherent sentiment patterns from the synthetic data well.

Two-stage training of XLM-RoBERTa with T5-augmented data was no improvement over that of BERT with T5-augmented data. This might be explained by the dissimilarities

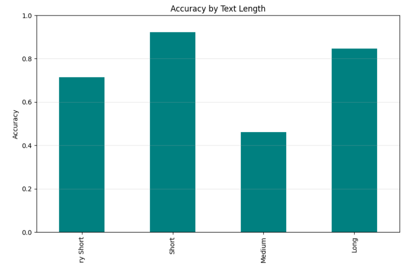

Fig. 8: model trained on T5-augmented data

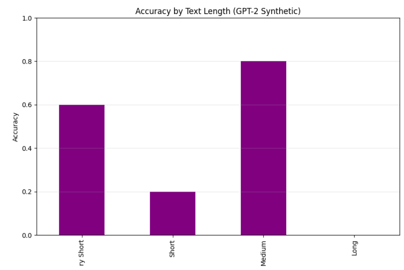

Fig. 9: model trained on GPT-2 synthetic data

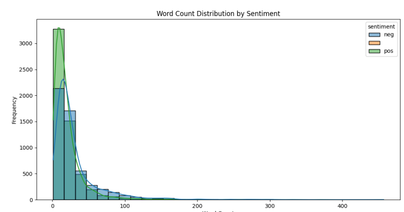

Fig. 10: Word Count Distribution by Sentiment Chart

between the architectures and pre-training tasks of the models, as well as the intrinsic characteristics of the T5-created augmented data.

The much worse performance of XLM-RoBERTa on filtered GPT-2 synthetic data is a surprising result. Filtering the synthetic data by a confidence threshold (0.75 in this experiment) could have inadvertently removed good diverse samples or introduced a data bias, resulting in bad generalization.

The comparison of accuracy across text length implies that the model learned from GPT-2 synthetic data is very accurate for varying text lengths, which points to a strong learning process from the enriched data.

VIII. Limitations

Though our adaptive synthetic data augmentation model shows large improvements in low-resource Roman Urdu sentiment analysis, there are certain limitations to be noted:

- **Dependence on Seed Dataset Quality:** The success of synthetic data generation largely depends on the quality and representativeness of the seed dataset. A biased or unrepresentative seed dataset can theoretically generate

synthetic samples that still carry such biases, which can restrict model generalization.
- **Filtering Threshold Sensitivity:** Weak performance of XLM-RoBERTa on filtered GPT-2 synthetic data indicates that the confidence-based filtering mechanism (threshold of 0.75) might inadvertently exclude valuable samples or introduce unforeseen biases. The threshold needs to be well-tuned, and it was not deeply investigated in this work.
- **Computational Cost:** The procedure of applying reinforcement learning through PPO with large pre-training models such as T5 and GPT-2 is a computationally demanding process, a potential disadvantage in the case of researchers or working professionals with small budgets.
- **Language-Specific Challenges:** Although the framework is defined for Roman Urdu, no experiment is performed for evaluating its feasibility for other low-resource languages. Roman Urdu's variability and non-standardization of orthography present special challenges, and the performance of the framework can be varied for languages with different linguistic properties.
- **Evaluation Scope:** It was tested on one dataset (RUSAD). Testing on other Roman Urdu datasets or even on other low-resource languages will provide a better idea about its generalizability and strength.

IX. Future Work

Building on the findings and limitations of this study, several avenues for future research are proposed:

- **Adaptive Filtering Mechanisms:** Future work may explore dynamic or context-dependent filtering techniques for synthetic data, e.g., employing multiple classifiers or diversity metrics to achieve balance between quality and diversity in filtering, which can reduce issues with hard confidence thresholds.
- **Cross-Lingual Generalization:** The model can be applied to other low-resource languages like Pashto or Sindhi to prove its application beyond Roman Urdu. Cross-lingual transfer learning methods can also be used to tap into high-resource languages for better performance.
- **Optimization of Computational Efficiency:** Investigating lighter generative models or better reinforcement learning algorithms would reduce the computational burden, and the architecture would become more scalable on a larger scale.
- **Dataset Diversification:** Having more Roman Urdu datasets available or merging datasets from various domains (e.g., product reviews, social media, news) could help improve the stability of the trained models and more closely reflect real-world settings.
- **Human-in-the-Loop Validation:** Combining human judgment to assess synthetic sample quality and sentiment accuracy can improve reward functions from automated systems to offer improved data quality generation.
- **Advanced Reward Functions:** Exploring more sophisticated reward mechanisms, i.e., averaging sentiment accuracy and linguistic diversity or coherence scores, has the potential to improve the synthetic data produced in quality and continue improving the performance of the downstream model.

By combining these constraints and venturing into these areas of future research, the proposed framework can be enhanced to offer even greater impact in low-resource sentiment analysis and other NLP tasks.

X. Conclusion

This paper presents an adaptive synthetic data augmentation system to solve the problem of low-resource sentiment analysis, i.e., for Roman Urdu. By integrating pre-trained language models (T5 and GPT-2) with reinforcement learning using Proximal Policy Optimization (PPO) and a FastText-based reward function, we generated high-quality synthetic data to augment a small seed dataset from the Roman Urdu Sentiment Analysis Dataset (RUSAD). Fine-tuning state-of-the-art transformer models, BERT and XLM-RoBERTa, on the augmented dataset led to excellent performance improvement. Specifically, BERT trained on GPT-2 synthetic data achieved a test accuracy of 93.33

The contribution of this work is that it makes it possible to enable strong sentiment analysis for low-resource languages like Roman Urdu, where there is no labeled data. Employing PPO in the model with a sentiment classifier as a reward function ensures task-specific and high-quality synthetic data, offering a scalable solution for other low-resource languages and NLP tasks.

Nonetheless, the framework is not without vulnerabilities. Firstly, the representativeness of the seed dataset is instrumental in determining the quality of the synthetic data; a small or biased seed dataset could contaminate the synthetic samples with errors. Secondly, the confidence filtering of synthetic data (e.g., a 0.75 threshold for GPT-2 data) resulted in poor performance, as observed with XLM-RoBERTa, implying that useful samples might have been eliminated. Thirdly, the training costs of large language models and the use of PPO might be prohibitive in environments with constrained resources. Lastly, evaluation was limited to one dataset (RUSAD).

To overcome these limitations, future work will focus on several significant enhancements. We will develop adaptive filtering methods that trade off sentiment confidence against diversity metrics to preserve informative samples. Extending the framework to other low-resource languages, like Pashto or Sindhi, and evaluating it on other datasets will enhance its robustness and usability. To reduce computational costs, we will explore lighter generative models and more efficient reinforcement learning methods. We will also incorporate human-in-the-loop validation to improve synthetic data quality by augmenting automated reward functions. These enhancements will enhance the applicability and usability of the framework to more general NLP research.

In summary, this paper demonstrates the promise of synthetic data augmentation to transform low-resource sentiment analysis, offering a feasible solution to data scarcity. By alleviating the constraints set, we aim to further advance and improve this framework, resulting in better sentiment analysis and related tasks in underrepresented linguistic settings.

References

[1] Naeem Ahmed, Rashid Amin, Hamza Aldabbas, Muhammad Saeed, Muhammad Bilal, and Houbing Song. A novel approach for sentiment analysis of a low resource language using deep learning models. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, September 2024. Just Accepted.

[2] Thomas Body, Xiaohui Tao, Yuefeng Li, Lin Li, and Ning Zhong. Using back-and-forth translation to create artificial augmented textual data for sentiment analysis models. *Expert Systems with Applications*, 178:115033, 2021.

[3] Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. An empirical survey of data augmentation for limited data learning in nlp. *Transactions of the Association for Computational Linguistics*, 11:191–211, 03 2023.

[4] Marzieh Fadaee, Arianna Bisazza, and Christof Monz. Data augmentation for low-resource neural machine translation. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[5] Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. Soft contextual data augmentation for neural machine translation. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5539–5544, Florence, Italy, July 2019. Association for Computational Linguistics.

[6] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759, 2016.

[7] Bohan Li, Yutai Hou, and Wanxiang Che. Data augmentation approaches in natural language processing: A survey. *AI Open*, 3:71–90, 2022.

[8] Xun Liang, Hanyu Wang, Yezhaohui Wang, Shichao Song, Jiawei Yang, Simin Niu, Jie Hu, Dan Liu, Shunyu Yao, Feiyu Xiong, and Zhiyu Li. Controllable text generation for large language models: A survey, 08 2024.

[9] Bing Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 2022.

[10] Essam Daryl Mehmood, Khawar and Muhammad Kamran Malik. Roman Urdu Sentiment Analysis Dataset (RUSAD). UCI Machine Learning Repository, 2018. DOI: https://doi.org/10.24432/C59046.

[11] Muhammad Kashif Nazir, Cm Nadeem Faisal, Muhammad Asif Habib, and Haseeb Ahmad. Leveraging multilingual transformer for multiclass sentiment analysis in code-mixed data of low-resource languages. *IEEE Access*, 13:7538–7554, 2025.

[12] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[13] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), January 2020.

[14] Bareera Sadia, Farah Adeeba, Sana Shams, and Kashif Javed. Meeting the challenge: A benchmark corpus for automated urdu meeting summarization. *Inf. Process. Manage.*, 61(4), July 2024.

[15] Xiaojun Wan. Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, page 235–243, USA, 2009. Association for Computational Linguistics.

[16] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.