Development of a Roman Urdu Medical Chatbot using Efficient Fine-Tuning and Hybrid RAG

Muhammad Faraz Ahmad
Dept. of Computer Science
LUMS
Lahore, Pakistan
26100250@lums.edu.pk

Muhammad Maaz Barki
Dept. of Computer Science
LUMS
Lahore, Pakistan
26100414@lums.edu.pk

Ameer Hamza
Dept. of Computer Science
LUMS
Lahore, Pakistan
24280033@lums.edu.pk

***Abstract*—Digital access to accurate medical information, especially in regional languages such as Urdu/Roman Urdu, is scarce. While modern-day LLMs excel at general tasks, they often underperform in domain-specific tasks, such as medical diagnosis. They're also very compute-intensive and may not work well in non-standard languages like Roman Urdu. Given the excessive use of Roman Urdu in the digital space over the standard Urdu script, we fine-tune three different open-source LLMs on a custom Roman Urdu dataset for the medical domain and complement it with a hybrid RAG system for accurate medical advice. We evaluate our models using automatic evaluation metrics, such as the BERT-F1 score and chrF++, as well as human evaluation. We found that the fine-tuned `qwen-2.5-7B` gave the best results with reasonable inference time.**

***Index Terms*—RAG, LoRA, Roman Urdu, parameter-efficient fine-tuning, LLMs, Unsloth**

I. Introduction

The advent of large language models (LLMs) has significantly impacted the daily lives of people worldwide. LLMs excel at a variety of tasks, such as coding, math, etc., but still underperform in tasks that require deep domain-specific knowledge, such as medical and legal domains. Also, since they are trained on the whole internet text, which mostly contains English, their performance in regional languages or non-standard languages like Roman Urdu declines, as shown by Pava et al. [1]. They are also extremely compute-intensive. Keeping these limitations in mind, we fine-tune three small-sized LLMs (1B-7B parameters) on a Roman Urdu medical dataset and evaluate them on metrics like Bert Score and chrF++. We analyze the quality vs inference speed tradeoff for these LLMs and conclude that 7B provides the best quality responses, with manageable inference time (using one GPU). We chose Roman Urdu, as opposed to standard Urdu, for our chatbot because of its greater prevalence in digital spaces, such as social media, compared to the standard Nastaliq script.

The detailed methodology of our project is in section II, while the detailed evaluation can be found in section III. Project's GitHub repository can be found at: `Link`

II. Methodology

A. Dataset Creation:

The most significant barrier in our project was the lack of a diverse medical-domain dataset in Roman Urdu. All the existing datasets were in either English or Chinese. To tackle this issue, we had to translate an existing dataset into Roman Urdu. We translated a dataset from Kaggle by `saifulislamsarfaraz`, which can be found `here`. The translation process itself was troubling, as no translation APIs currently translate into Roman Urdu, as it is a non-standard language. We also tried open-source translation models like `indictrans2` and `Helsinki-NLP/opus-mt-en-ur`, but their translation quality was really bad. They often mistranslated medical terms, and the original user intent was lost in the translated versions. The final option was to use LLMs with paid APIs. We used `gpt-4o-mini` for translation in batches. Our final dataset had $\approx 21,000$ training examples and ≈ 5200 validation examples.

B. Fine-tuning LLMs:

After dataset creation, we fine-tuned three open source LLMs, specifically:

1) `Llama-3.2-1B-Instruct`
2) `Llama-3.2-3B-Instruct`
3) `Qwen2.5-7B-Instruct`

The reason for choosing three different-sized models was to analyze the tradeoff between response quality and inference speed. For fine-tuning, we utilized the `unsloth` library with Low-rank Adaptation (LoRA) for faster and memory-efficient training. This allowed the fine-tuning to be completed on Colab and Kaggle's T4 GPU. We fine-tuned for 2600 steps with a batch size of 8 and a learning rate of `2e-4`, which constituted around 1 epoch of fine-tuning. We avoided tuning for more epochs to prevent catastrophic forgetting, where an LLM completely forgets its pretraining and memorizes the training data.

The fine-tuning times on T4 GPU were $\approx$ 1h 15m for 1B model, $\approx$ 3h 40m for 3B model, and $\approx$ 8h 10m for the 7B model.

After fine-tuning, we did a quick inference test to confirm whether the model outputs correctly or not.

C. Hybrid RAG:

After fine-tuning, we checked the quality of the model's outputs and consistently found this: the model talks fluently in Roman Urdu, often using commonly used Roman Urdu phrases, but struggled a lot with hallucinations. It would give terrible advice like *sabun ka paani piyen* (drink soapy water) and would start storytelling. Sometimes, it would start considering itself as the patient. To prevent such issues, it was essential to use RAG and instruct it strictly to avoid storytelling and be precise. So, we implemented RAG with a structured medical knowledge base covering the top 50 most common diseases, including each disease's symptoms, short-term relief, treatment, and precautionary measures. We initially tried `all-MiniLM-L6-v2` for vector embeddings, but it failed because it is primarily trained on English, with limited exposure to Roman Urdu. Then, we adopted `intfloat/multilingual-e5-large`, which was much better than `all-MiniLM-L6-v2`, but still imperfect. It treated all kinds of *dard* (pain) the same, considering *joron ka dard* (joint pain) and *pait ka dard* (stomach pain) identical. In order to prevent such issues, we had to make a hybrid RAG system: incorporating `BM25` for exact word matching in addition to the normal RAG pipeline. Our final retrieval pipeline ensembled the results of `intfloat/multilingual-e5-large` and `BM25`. We had to give huge weight (0.9) to `BM25` and only 0.1 to the normal RAG output, to avoid picking the wrong context. This was a fundamental limitation in the available resources, over which we had no control.

Finally, to stop the model from storytelling and considering itself as the patient, we added strict instructions in the system prompt saying: ”Do not tell stories about yourself” and ”Act as a professional Doctor”. This reduced such cases.

III. Evaluation

For the evaluation of our fine-tuned LLMs, we did automatic evaluation of 100 randomly selected examples from the validation split using Bert-F1 score, chrF++ score, and ROUGE-L score:

- **Bert-F1:** Measures semantic similarity between the generated text and the referenced text
- **chrF++:** Measures character n-gram and word n-gram overlap between the generated text and the referenced text
- **ROUGE-L:** Measures similarity between the generated text and the referenced text by Longest Common Subsequence (LCS) overlap

For human evaluation, 10 query-response pairs were rated on a scale of 1 to 5, and their average was calculated.

IV. Results

Table I summarizes the automatic evaluation results using the metrics discussed above. We perform robust testing for each model by comparing the base pre-trained model, the fine-tuned (FT) model without RAG, and then with RAG. The fine-tuned `Qwen2.5-7B` model performs the best across all metrics. However, one observation, that may seem counterintuitive at first, is that fine-tuned versions without RAG outperform their counterparts with RAG for all three models, across all the metrics. This is explained by the following: All the evaluation metrics measure some sort of overlap between the generated and reference text. When using RAG, we essentially force the model to output medical terms like names of tablets, which are not found in the reference text; it usually provides general advice. Because of this, the scores for the FT model (without RAG) get higher as it doesn't mention such names, whereas the RAG one does mention them. However, in reality, having precise information about the treatment (if accurate) is more beneficial than simple general advice.

For human evaluation, we gave scores from 1-5 and took the average for each configuration. We can see the results in Table II. As expected, `Qwen2.5-7B` with RAG performed the best with the average human rating of 3.67. All the base models suffered terribly as they generated Urdu Nastaliq script even after being instructed to answer in Roman Urdu. Another interesting result is that `Llama-3.2-3B-Instruct` with RAG performed worse than without RAG. This is because sometimes the model got confused with the additional context, especially if it was for an irrelevant disease. Even `Qwen2.5-7B` suffered from this on some questions, where the model without RAG performed better while the one with RAG produced bad responses. But, overall, it provided more stable responses.

TABLE I
Automatic Evaluation Results

Model	Mode	BERT-F1	ROUGE-L	CHRF++
Llama 1B	Base	0.7350	0.0384	6.6217
	FT	**0.8413**	**0.1173**	**18.5206**
	RAG	0.8359	0.1095	16.5166
Llama 3B	Base	0.7586	0.0566	9.0776
	FT	**0.8411**	**0.1189**	**18.4851**
	RAG	0.8380	0.1131	16.8233
Qwen 7B	Base	0.7138	0.0191	3.6988
	FT	**0.8417**	**0.1229**	**18.9669**
	RAG	0.8394	0.1172	18.4333

TABLE II
Human Evaluation Results (1-5 Scale)

Model	Mode	Avg Human Rating
Llama 1B	Base	1.35
	FT	2.775
	RAG	2.87
Llama 3B	Base	2.5
	FT	3.2
	RAG	2.825
Qwen 7B	Base	1.5
	FT	3.56
	RAG	**3.67**

V. Challenges & Conclusion

The main challenges were the lack of resources for Roman Urdu, as it is not a standard language. At each step, we struggled with very limited options to choose from, and even those were not very accurate.

In conclusion, this project demonstrates that fine-tuning small LLMs combined with a decent RAG framework is a viable option for building medical domain chatbots in low-resource languages. Better resources, such as models made specifically for Roman Urdu vector embeddings, can greatly enhance the quality of the model's responses.

VI. Future Extensions

Possible future extensions of our work can be:

- Creating a more diverse Roman Urdu dataset for finetuning, or using an existing diverse & high-quality Roman Urdu dataset.
- Studying the impact of fine-tuning on much larger models
- Implementing RAG more robustly to further reduce hallucination rate, by perhaps creating new models for creating vector embeddings.

References

[1] J. N. Pava, C. Meinhardt, H. Badi Uz Zaman, T. Friedman, S. T. Truong, D. Zhang, E. Cryst, V. Marivate, and S. Koyejo, "Mind the (language) gap: Mapping the challenges of llm development in low-resource language contexts," Stanford Institute for Human-Centered Artificial Intelligence (HAI) and The Asia Foundation, Policy White Paper, 2025. [Online]. Available: https://hai.stanford.edu/policy/mind-the-language-gap-mapping-the-challenges-of-llm-development-in-low-resource-language-contexts