Bigyan Aryal

Tilottama-7, Rupandehi· 9847318266 · bigyan8aryal@gmail.com
www.linkedin.com/in/bigyan-aryal
www.bigyanaryal08.com.np
https://github.com/bigyan08

Education

Paschimanchal Campus, IOE, TU, Pokhara-16, Lamachaur **May 2023**
Currently Pursuing Bachelor In Computer Engineering,

Skills & Technology Stack

- **Technical Skills:** Backend Development, AI/ML Research, Generative AI, Agentic AI
- **Languages:** Python, PyTorch, Django REST, LangChain, LangGraph, FastAPI.
- **Tools:** Git, Github, Docker, LangChain, LangGraph
- **Cloud :** Mircrosoft Azure

Relevant Experience

Innovative Computer Engineering Student's Society, **Pokhara-16, Lamachaur** **July 2024-January 2025**
AI Bootcamp

- Presented a case study on AlexNet about the revolutionary CNN model, activation functions like ReLU and dropout regularization.
- Improved model understanding across the team, measured by higher training stability, by analyzing architecture choices and presenting findings.
- Enhanced project quality during hands-on labs by using AI tools for code review and documentation.

Projects

- RAG-Based AI Research Assistant (LangChain, FAISS, FastAPI)
  - Achieved efficient large-scale retrieval (handling 500k+ documents) as measured by sub-second vector search times, by designing a **FAISS** database and structuring ingestion workflows and wrapped the model into an endpoint using **FastAPI.**
- FormalNet – NLP Style Transfer Application (Django, Transformers)
  - Delivered a functional text-to-formal style **transformer**, measured by consistent inference quality, by integrating a transformer model into a **Django REST** backend.
  - Improved input reliability by building a preprocessing pipeline that reduced malformed inputs by ~40%.
- Containerized Image Captioning API (Docker + FastAPI)
  - Enabled scalable cloud deployment, measured by successful container uptime and API responsiveness, by packaging the captioning model into **Docker** and deploying it via **FastAPI**.
  - Used **Azure Container Registry** and **Web Services** for deploying the containerized model to production.
- Shakespeare GPT & Transformer Implementations
  - Improved understanding of **LLM** internals, measured by successful training convergence, by implementing attention, positional encoding, and transformer blocks from scratch in **PyTorch**.
  - Generated coherent text outputs by training a tiny GPT model on the Tiny Shakespeare dataset and iterating on training parameters.

Activities

- **Nepal Data Challenge 001**: Achieved 70% accuracy and secured 2nd place by fine-tuning nepBERTa and optimizing training cycles.
- **TechParva Datathon**: Increased loan prediction accuracy (A–G grading) through feature engineering and model comparisons.