Raghav Patidar

Indore, India

+91 9926514128 | r.patidar181001@gmail.com | linkedin.com/in/raghav-patidar-997842210/

Education

Indian Institute of Technology (IIT), Ropar *Ropar, India*

Bachelor of Technology, Computer Science and Engineering *Aug 2020 – May 2024*

- Courses: Data Structure and Algorithms, Software Engineering, Operating Systems, Computer Networks, Database and Management System, Blockchain Technology

Skills

Programming C, C++,Java, Python, SQL, Javascript, GraphQL

Framework/Tools LangChain, Hugging Face, LLM , GraphDB, LangGraph, Scikit-learn, TensorFlow, FastAPI, PyTorch

Deployment AWS, Docker, kubernetes, Streamlit, Chainlit

Work Experience

SCA Technologies *Gurgaon, India*

Software Engineer *June 2024 – Feb 2025*

- **Generic Log:** Implemented custom logging feature according to service and functionality on the top of tomcat catalina using Spring Boot Slf4. Which reduces the production debug time by 50%.
- **Chatting Feature:** Extended database procedures to implement role-based chatting functionality within the product, ensuring secure and efficient communication at role level.
- **Production Release and Hotfixes:** Resolved production-level issues and implemented critical hotfixes to ensure system stability. Additionally, supported the deployment and demonstration of key features during the client presentation.

Projects

QueryMind *Github Link*

Multi-Agent Stateful RAG Chatbot

- **Multi-source retrieval:** Developed an advanced Retrieval-Augmented Generation (RAG) pipeline for intelligent question answering, integrating multi-source retrieval and LLM reasoning.
- **RAG-Based Retrieval System:** Architected a RAG pipeline to efficiently fetch information from multiple sources – a VectorStore (AstraDB) and Wikipedia API.
- **Query Router:** Implemented dynamic query routing using LLaMA-3.3-70B (Open Source) to intelligently decide whether to fetch answers from the VectorStore or Wikipedia.
- **Graph-Based Retrieval:** Designed a state-based retrieval workflow using LangGraph, ensuring efficient query processing.Constructed a conditional graph state machine that dynamically selects retrieval paths based on query intent.

Quora Duplicate *Github Link*

NLP Application | Data preprocessing + Feature Engineering + Model evaluation + scikit-learn | **Website**

- **Duplicate Question Detection using NLP:** Built an ML pipeline to identify semantic similarity between Quora question pairs using advanced text preprocessing, feature engineering, and machine learning models.
- **Custom Text Preprocessing:** Applied advanced NLP techniques such as regex-based normalization, HTML tag stripping (BeautifulSoup), contraction expansion, stopword filtering (NLTK), and word count analysis to generate clean, tokenized input data.
- **Feature Engineering and Vectorization:** Engineered over 30 semantic, syntactic, and fuzzy matching features such as word overlap and token similarity, and integrated them with a 3000-dimensional Bag-of-Words vector using CountVectorizer for enriched text representation.
- **Modeling and Evaluation:** Trained and evaluated Random Forest and XGBoost classifiers using scikit-learn and XGBoost, achieving strong predictive accuracy with accuracy score and confusion matrix on validation data.

Gemma Fine-Tuning *Github Link*

LoRA-Based Fine-Tuned LLM for Text Generation

- **Text Generation Model:** Developed an instruction-tuned text generation model by fine-tuning GemmaCausalLM (2.2B parameters) using Low-Rank Adaptation (LoRA), optimizing compute efficiency for deployment on resource-constrained environments.
- **Data Preperation:** Integrated KerasNLP, Hugging Face Datasets, and Google Colab for dataset preprocessing, tokenization using GemmaTokenizer, and optimized training using mixed-precision computation (bfloat16/float32) to reduce memory footprint.
- **Model efficiency:** Reducing trainable parameters from 2.61B to 2.93M and memory footprint from 9.75GB to 11.17MB using LoRA (Low-Rank Adaptation) with rank=4, significantly lowering computational costs while preserving model performance.

Open Source Contribution

. **5937/5864**, InfiniFLow : Contributed to open source project working on RAG flows *India*