Zhiyuan Wang

zhiyuantom.wang@mail.utoronto.ca • 289-772-8339 • linkedin.com/in/zhiyuanwang552

EDUCATION

University of Toronto Scarborough | Scarborough, ON **2024 – Present**
Honours Bachelor of Science (Co-op), Computer Science Specialist & Statistics Major **cGPA: 3.67/4.0 Dean's List**
Relevant Courses: Software Design, Software Tools & Systems Programming, Theory of Computation, Linear Algebra, Probability, Discrete Mathematics, Linear Programming & Optimization, Calculus of Several Variables

EXPERIENCE

Dodonadata (Contracted to Telus) | Toronto, ON **Jan. 2026 – Present**
AI/ML Engineer

- Architected and deployed a production RAG system using **LangGraph**, **Pydantic AI**, and **Vertex AI Vector Search**, achieving **~30%** organic adoption (**~100** active users out of **300–500** team members) and reducing average query latency by **90%**—from **2–3 minutes** to a consistent **20–40 seconds**—while eliminating prior instability that caused response times up to **10 minutes**
- Selected **Gemini 3 Flash** as the primary LLM after benchmarking multiple models, leveraging its **1M-token** context window for large telecom document chunks and dynamically tuning thinking levels per pipeline step to minimize overthinking on planning stages while preserving deep reasoning for complex answer generation
- Engineered end-to-end automated data pipelines on GCP to migrate **20,000+** files from **Google Drive** to **GCS** via the Drive API, and built a multi-threaded web scraper to ingest **15,000+** vendor documents into the **Vertex AI Vector Search** index
- Implemented recursive chunking strategies tailored to telecom document structures and embedded chunks using Telus's proprietary **Fuelix** embedding model, then applied the **Vertex AI Ranking API** as a reranker to maximize retrieval relevance and minimize noise in retrieved context
- Deployed **Cloud Run** tasks to auto-extract compacted archives at scale and developed Python scripts to filter non-informative content (e.g., license keys, config stubs), improving overall embedding quality and downstream semantic retrieval accuracy
- Integrated **Langfuse** as the production observability platform, tracking per-node latency, token usage, cost per query, and Vertex AI reranker relevance scores across the agentic workflow—linked to a user feedback dashboard enabling rapid debugging and continuous iteration
- Built an **LLM-as-judge** evaluation pipeline comparing new RAG outputs against prior version baselines, computing delta quality scores across a standardized query set to quantify improvements and catch regressions pre-deployment
- Implemented a prompt guard security layer for the **MCP RAG** agent using input validation and content filtering techniques to detect and block prompt injection and jailbreak attacks, safeguarding system integrity across all user-facing interactions
- Designed a long-term memory system using **Firestore** and a dedicated vector store to persist user preferences and interaction history, retrieving relevant memory via semantic search on user queries to tailor responses to individual contexts
- Collaborated cross-functionally with Telus product managers and engineering leads on-site 5 days per week, translating stakeholder requirements into technical specifications and delivering iterative product demos to drive alignment

PROJECTS

Smart-Air Android App | Java, Android Studio, Firebase, JUnit, Git **November 2025**

- Developed a Java Android app in Agile to help children manage asthma symptoms, using Firebase for authentication and profile management, and authored **50+** JUnit tests achieving **100%** code coverage, reducing regression bugs by **60%**
- Managed **10+** feature branches via Git/GitHub, reviewed **10+** pull requests, and participated in daily standups analyzing **40+** user stories to ensure consistent on-time sprint delivery across the development lifecycle

Virtual Persona Chatbot | Python, LangChain, Ollama, PyTorch, RAG **October 2025**

- Developed a conversational AI chatbot using Retrieval-Augmented Generation, parsing and embedding **40+** pages of PDF data into a vector store to enable high-precision semantic search with context-aware response generation
- Designed **10+** structured prompt chains in LangChain and optimized Ollama model invocation flows, improving multi-turn conversational response accuracy by **~40%** through iterative prompt engineering

TECHNICAL SKILLS

Languages: Python, Java, C | **AI/ML:** LangChain, LangGraph, Pydantic AI, PyTorch, RAG, Gemini, Ollama, Langfuse, Vertex AI Vector Search
Tools: GCP (Cloud Run, GCS), Docker, Git, GitLab, Firestore, Firebase, Google Drive API, Jupyter, Jira, Linux