# Ziyang Lin

📱 **Phone**: 13537685397 | ✉️ **Email**: ziyanglin1997@163.com | 💼 **LinkedIn**: www.linkedin.com/in/ziyang-lin | 💬 **WeChat**: zi24yang10la8 | 🌐 **Website**: https://ziyanglin.netlify.app | 📍 **Location**: Shenzhen
AI Software Development Engineer

## Professional Summary

- Bachelor's graduate in Computer Science and Artificial Intelligence from the University of Sheffield (graduated July 2022)
- Completed second-year coursework with First Class Honors in the 2019-2020 academic year
- Took a gap year in 2020-2021 to further study natural language processing technologies and related mathematical foundations
- Completed final year dissertation project on "Deep Learning-Based Automatic Speech Recognition" under the supervision of Dr. Anton Ragni
- Worked as a Software Development Engineer at Shenzhen Institute of Metrology and Quality Inspection's Technology R&D Center from November 2022 to March 2024
- Currently serving as an AI Software Development Engineer at Shenzhen Zhiruantong Technology Co., Ltd. since March 2024, focusing on RAG technology applications and real-time multimodal model application development

## Education

### University of Sheffield, United Kingdom

**BSc Computer Science and Artificial Intelligence, Department of Computer Science**
September 2018 - July 2022
Sheffield, United Kingdom

**Research-Based Dissertation:**
GLOM for Automatic Speech Recognition - Supervisor: Dr. Anton Ragni (Designed and developed a representation learning model with improved interpretability for automatic speech recognition, referencing Hinton's GLOM model, and trained it on academic high-performance computing clusters; supervisor's final feedback can be viewed here)

# Professional Experience

## Shenzhen Zhiruantong Technology Co., Ltd.

**AI Software Development Engineer**

March 18, 2024 - Present

Shenzhen

- Responsible for designing and developing intelligent systems based on Retrieval-Augmented Generation (RAG) technology, primarily utilizing Milvus and Weaviate vector databases to build efficient knowledge retrieval systems
  - Designed and implemented hybrid retrieval strategies, combining sparse retrieval (BM25) and dense retrieval (vector similarity) to improve retrieval accuracy
  - Optimized document chunking strategies, implementing semantic-based adaptive chunking to enhance the relevance of retrieval results
  - Implemented multi-level caching mechanisms to reduce computational overhead for repeated queries and improve system response time
- Participated in advanced Prompt Engineering practices to enhance LLM application effectiveness
  - Designed and implemented complex reasoning prompt strategies based on Chain-of-Thought and ReAct frameworks
  - Developed an automated prompt optimization system to evaluate different prompt templates through A/B testing
  - Implemented a dynamic prompt template library that adaptively selects the optimal prompt strategy based on user input and interaction history
- Successfully developed and deployed a financial data analysis agent that enables automated analysis and insight extraction from complex financial data
  - Designed a multi-stage analysis pipeline including data cleaning, anomaly detection, trend analysis, and predictive modeling
  - Implemented a custom DSL (Domain Specific Language) allowing financial analysts to query complex data using natural language
- Designed and implemented a government affairs Q&A agent to optimize government information retrieval efficiency and enhance user query experience
  - Built a knowledge graph for the government affairs domain, enhancing entity relationship understanding and complex query processing capabilities
  - Implemented a multi-turn dialogue management system supporting contextual understanding and interactive dialogue for query clarification

- Participated in the company's proprietary **multimodal low-latency real-time voice conversation agent**, integrating speech recognition, natural language processing, and speech synthesis technologies
  - Optimized end-to-end speech generation models to achieve 300ms latency
  - Implemented real-time streaming architecture supporting simultaneous speaking and understanding interaction modes
  - Designed an emotion recognition module to adjust response strategies based on user voice emotions
  - **Utilized llama.cpp efficient inference engine to deploy a 14B parameter multimodal voice model in GGUF format, achieving low-resource high-performance inference on a single RTX 3090 GPU, optimizing memory usage and computational efficiency**
  - Achieved **full-process real-time latency of 500ms**, providing tool calling and complex prompting capabilities; optimized deployment to run non-quantized models on just 2xA100 GPUs and enabled arbitrary base model replacement
- Participated in deployment optimization and system architecture design based on vLLM
  - Implemented model quantization techniques (A8W8, A4W4) to significantly reduce model size while maintaining performance
  - Implemented efficient low-latency inference service architecture supporting model parallelism and tensor parallelism
  - Implemented dynamic batching and request queuing mechanisms to improve system throughput
  - Implemented model hot-loading and caching strategies to optimize resource utilization in multi-model switching scenarios
  - Leveraged llama.cpp to optimize local inference performance, implementing efficient model quantization and memory management solutions
  - Built lightweight model services based on ollama, supporting rapid deployment and API calling for various open-source models
- Participated in the development of multiple generations of CAAS platforms, completing the optimization of the 'Mxgent' SOP Agent
  - Participated in the development of the company's dialogue management system, supporting complex multi-turn dialogue processes
  - Participated in the core implementation of the company's intent recognition Agent, improving the robustness and flexibility of the dialogue system
  - Participated in the design and implementation of the self-developed tool RPA module, enabling correct tool calling with low-parameter models and ensuring compatibility with function calling and MCP protocols

## Shenzhen Institute of Metrology and Quality Inspection

**Software Development Engineer, Technology R&D Center**

November 2022 - March 2024

Shenzhen

- Led the development of an electric vehicle charging station verification platform website based on the XAMPP (Apache + MySQL + PHP + PERL) technology stack, implementing test data collection, storage, and visual analysis functions
- Responsible for core development work on the CT online inspection of electronic components project:
  - Designed and implemented CT scan image data collection and preprocessing workflows
  - Conducted model selection and secondary development based on Baidu's PaddlePaddle framework
  - Optimized model training processes to improve defect detection accuracy
  - Completed model deployment and system integration
- Proficiently applied computer vision-related technologies, including OpenCV library for image processing and analysis
- Developed industrial camera applications, integrating Basler and Hikvision industrial camera SDKs
- Participated in cross-platform application development based on the Qt framework
- Technology stack: C++, Python, PHP, MySQL, PaddlePaddle, OpenCV, Qt

# Projects

## Imperial College London Summer Natural Language Processing Course + Project

**Summer Online Research Student**

June 2020 - August 2020

- Participated in the 2020 NLP Online Research Summer Project guided by Professor Lucia Specia from Imperial College London
- The project aimed to develop regression and classification models to evaluate the humor level of edited news headlines (in English)
- For specific details about my completed work for this project, please visit my personal website or my GitHub profile
- Link to Professor Lucia Specia's reference letter: Reference Letter

# Organizational Experience

## University of Sheffield Artificial Intelligence Society

**Event Officer**

September 2020 - July 2021

Sheffield, United Kingdom

- Selected as the Event Officer for the AI Society for the 2020-2021 academic year
- Responsible for hosting and organizing AI-related learning projects (planning workshops or alumni lectures) during the academic term

# Domain Knowledge

- **Large Language Models & Application Development**: In-depth understanding of Transformer architecture, attention mechanisms, self-supervised learning, and fine-tuning techniques
- **Prompt Engineering**: Mastery of advanced prompt engineering techniques including Few-shot Learning, Chain-of-Thought, ReAct framework, and prompt template design
- **LLM Deployment & Optimization**: Expertise in model quantization, Parameter-Efficient Fine-Tuning (PEFT), and inference acceleration techniques, proficient with llama.cpp efficient inference library's quantization techniques and memory optimization solutions, mastery of ollama local model management and deployment framework's containerized applications
- **Retrieval-Augmented Generation (RAG)**: Familiarity with vector retrieval, hybrid retrieval strategies, document chunking, and context optimization techniques
- **Multimodal Learning**: Understanding of cross-modal information fusion, multimodal representation learning, and alignment techniques
- **Machine Learning Fundamentals**: Deep learning, reinforcement learning, natural language processing, computer vision
- **Mathematical Foundations**: Linear algebra, multivariate calculus, probability theory, information theory

# Technical Skills

- **Programming Languages**: Python, C++, JavaScript
- **Deep Learning Frameworks**: PyTorch, TensorFlow, JAX, PaddlePaddle
- **LLM Toolchain**: LangChain, DSPy, LlamaIndex, PEFT, Transformers
- **Vector Databases**: Milvus, Weaviate, FAISS, Pinecone

- **Model Deployment**: ONNX Runtime, TensorRT, vLLM, DeepSpeed, Triton, llama.cpp, ollama
- **Speech Processing**: SpeechBrain, Whisper, ESPnet, Kaldi, FFmpeg, librosa, Wav2Vec2, HuBERT, pyAudioAnalysis
- **Audio Processing**: WebRTC, DTMF, VAD (Voice Activity Detection), Audio Enhancement
- **Development Tools**: Docker, Git, Linux, Ray
- **Computer Vision**: OpenCV, Pillow, torchvision
- **Web Development**: FastAPI, Flask, React

# Languages

Mandarin Chinese (Native), English (Fluent)