Siddhant Singh Karki

AI ENGINEER

siddhantskarki.com
https://github.com/sidskarkii/
https://huggingface.co/sidskarki/

+61 406832800
siddhantskarki@gmail.com
St Peters, NSW 2044

AI Engineer who builds production systems end to end, from data pipelines and model training to APIs, evals, and deployment. Works below the abstraction layer when needed, with hands-on depth in tokenizer internals, model adaptation, distributed processing, and retrieval augmented generation (RAG). Turns ambiguous AI problems into shipped systems, from voice pipelines and analytics to knowledge infrastructure and language model research. Ships code that runs in production, trains models that beat baselines, and writes evals that prove it.

PROFESSIONAL EXPERIENCE

CallD.AI | AI Engineer https://www.calld.ai/ **March 2025 - Current**

Sole engineer owning two major production systems across live-call infrastructure and AI post-call analysis, from ingestion through storage, processing, APIs, and dashboards.

- Owned two generations of the AI analysis pipeline across **Twilio, LiveKit, and Asterisk** call sources, delivering end-to-end processing through **S3, Deepgram STT, LLM analysis**, and **PostgreSQL persistence**.
- Cut LLM token usage by **75%** and improved processing speed **3x** by redesigning speaker re-label system with turn-index maps, then further reduced cost through **OpenAI-to-Bedrock migration** and **Nova Micro prompt caching**.
- Implemented per-utterance LLM-based speaker **diarization** and enforced schema-validated **structured outputs** with **Pydantic**, eliminating malformed LLM responses across the analysis pipeline.
- Improved distributed processing reliability with **PostgreSQL FOR UPDATE SKIP LOCKED** row locking, **10 concurrent workers**, jittered retries, thundering herd prevention, and graceful shutdown handling.
- Built the analytics platform from data model to API to frontend, including **materialized-view-backed** reporting, **8+ dashboard endpoints**, and a configurable **widget system.**
- Reduced dashboard load times by **75%** through COALESCE removal, targeted PostgreSQL indexes, and SQL restructuring for high-traffic analytics queries.
- Delivered production infrastructure and security improvements with **Dockerized services**, **GitHub Actions CI/CD to ECR**, security hardening, **CVE** remediation, and multilingual call-analysis support.
- Expanded the auth, data model, and API surface from single-owner access to project-scoped M2M sharing, covering conversation APIs, transcript APIs, and **hard delete** across PostgreSQL and S3.

PROJECT EXPERIENCE

MCP Knowledge Base with Hybrid Retrieval and Wiki Synthesis

https://github.com/sidskarkii/Lore

Python, MCP, FastAPI, LanceDB, ONNX Runtime, FlashRank, networkx, spaCy

- Enabled AI agents to research across PDFs, videos, code, and web pages through a **23-tool MCP server** that returns metadata-only results first (**50 tokens** each), letting agents evaluate relevance before committing context window to full-text retrieval.
- Improved retrieval **MRR@5 by 68%** over vector-only search (0.256 → 0.431) with 6-signal hybrid search (vector, BM25, **fuzzy entity boost**, **RRF fusion**, **cross-encoder reranking**, query intent routing), benchmarked on a **211K-chunk corpus**, selected **MiniLM-L12** reranker at **34MB** over **Qwen3** at **1.19GB** for equivalent quality at **35x smaller size.**
- Automated cross-source **wiki generation** where every claim links back to its source chunks with a **verification status** (supported, conflicted, needs review), with **contradiction detection** across sources and recursive gap-filling ranked by link pressure, evidence count, and source diversity.
- Runs fully local with **CPU-first ONNX** inference, zero-config LLM routing that auto-detects the host agent's subscription, **7 specialized extractors** and **auto-compounding** that regenerates wiki pages on every new ingest.

Devnagari Tokenizer Efficiency: Cross-Model Benchmark and Remediation Pipeline

https://github.com/sidskarkii/nepali-tokenizer

Python, SentencePiece, PyTorch, HuggingFace, Qwen3, LoRA

- Proved a **2-6x tokenizer cost penalty** for Devnagari text by **benchmarking 17 tokenizers** across **9 LLM families** on 2,054 Nepali documents and 2,000 English documents with corpus-weighted metrics that directly map to API cost and context window impact.
- Reduced tokenizer-remediation training cost **3.5x** by writing custom **TrainableTokenEmbedding** wrappers that froze **151K** base embedding rows and trained only **15K** appended rows, cutting trainable params from **1.4B to 38M** with a split optimizer using separate learning rates for **LoRA** and new embeddings.
- Extended **4 production tokenizers** (Phi-4, Qwen 3.5, DeepSeek V4, Kimi K2.6) with delta vocabulary selection, then ran **LoRA CPT and SFT** on **Qwen3-4B** with English replay, cutting Phi-4 tokens/word by **52%,** improving Nepali BPC from **1.96 to 1.07,** and downstream script ratio from **0.57 to 0.92** with only **7.4%** English regression.

Nepali ASR: Fine-tuned Qwen3-ASR with Cross-Dataset Evaluation

https://github.com/sidskarkii/nepali-asr

Qwen3-ASR-1.7B, PyTorch, HuggingFace, A100

- Beat **Meta MMS-1B** on spontaneous speech (**55.8% vs 62.4%** WER on IndicVoices-R, 2060 speakers) and TTS speech (**31.4% vs 40.5%** on OpenSLR-43) using only **157 hours** of focused single-language fine-tuning against Meta's **1,100+** language multilingual model.
- Benchmarked **7 ASR** models across **3 domains** (read, spontaneous, TTS) to validate cross-dataset generalization, with end-to-end pipeline ownership from audiobook curation and **semi-Markov alignment** through Qwen3-ASR-1.7B SFT to published model on HuggingFace.

SKILLS

- **AI/ML:** PyTorch, LoRA/QLoRA, HuggingFace, Transformers, Agents SDK, ONNX Runtime, Deepgram, Whisper, Structured Outputs, Prompt Engineering, Harness Engineering, Context Engineering, Embeddings, RAG, MCP, LangGraph
- **Infrastructure:** Python, Django, FastAPI, PostgreSQL, Docker, CI/CD, GitHub Actions, AWS (S3, ECR, EC2, Bedrock), GCP, Azure, Terraform, Runpod, Vast.ai
- **Tools:** Git, Claude Code, Codex, Cursor, HuggingFace, LanceDB, MongoDB, TypeScript, Node.js, React, Svelte

EDUCATION

Victoria University

2023 - 2025

Bachelors in Information Technology

GPA: 6.46/7.0

Concentrations: Software Development & Networking Technologies