Sangchun Ha

47, Misagangbyeondong-ro, Hanam-si, Gyeonggi-do, South Korea, (+82)10-4725-9896, seomk9896@gmail.com

LINKS

Github, Huggingface, Linked In, Tech blog

EMPLOYMENT HISTORY

Jan 2022 — Present

Research Engineer, ReturnZero

Gangnam, Seoul

Developed end-to-end speech recognition models and sLLM models for call summarization.

Implemented model optimization using Triton Inference Server, OpenVINO, and TensorRT.

Check more details about ReturnZero at [link].

Aug 2021 — Jan 2022

Internship, TUNiB

Gangnam, Seoul

Developed Korean-English bilingual language models. [link]

Mar 2021 — Aug 2021

Internship, NeuroAI Lab

Nowon, Seoul

Dept. of Electronic and Communication Engineering, Kwangwoon University

Advisor: Prof. Young-Seok Choi

EDUCATION

Mar 2016 — Feb 2022

Bachelor of Engineering, Kwangwoon University

Nowon, Seoul

Major in Electronic and Communication Engineering, Minor in Data Science.

Major GPA : 4.28 / 4.5, **Total GPA** : 4.07 / 4.5

Relevant Coursework :

- AI & Speech Signal Processing, Digital Signal Processing, Algorithm
- C language, Operating System, Computer Architecture, Computer Network
- Data Structure, Database, Data Science, Data Communication, Object-Oriented Programming

PROJECT EXPERIENCE

May 2024 — Sep 2024

Korean language model

Developed Korean-language conversational model based Gemma 2 model using Continuous Pre-Training (CPT), Supervised Fine-Tuning (SFT), and Direct Preference Optimization (DPO) techniques. [link]

Developed various Korean embedding and reranking models to enhance Retrieval-Augmented Generation (RAG) and improve retrieval performance. [link]

Applied vllm, sglang, and TensorRT-LLM optimization techniques to the service, enhancing performance and efficiency.

Aug 2023 — Aug 2023

Awesome-Korean-Speech-Recognition

Released an open-source collection of Korean speech recognition API, datasets, and error rates. [link]

May 2021 — Jul 2021

OpenSpeech

Implemented a framework to easily make a speech recognizer in various languages. [link]

EXTRA ACTIVITIES

CPython Study

Studied the low-level operations of CPython, including lexical analysis, parsing, and Abstract Syntax Tree (AST) generation. [link]

Focused on understanding how memory is allocated, optimized, and managed at the code level, with an emphasis on reference counting, circular references, and garbage collection (GC) mechanisms to handle memory efficiently and prevent leaks.