Hao-xuan (Horace) Wang

+86 18622468042

billweasley20092@gmail.com

Github: https://github.com/billweasley

Linkedin: https://www.linkedin.com/in/horace-haoxuan-wang

Personal Website: http://shellcottage.me

Work Experience

- 2022.03 - Now **Machine Learning Engineer, ASR and Language Tech** @ Zoom

  Tech stack: Torch/Pytorch, ONNX, Kubernates, Huggingface

  - Implement and train speech recognition and text punctuation models, worked on Danish language from 0 to 1. Initial WER on test dataset was about **8%**, and phonic data argument reduces the WER further, which **outperforms the MS Team's result**. Initial casing and punctuation overall F1 is around **70%**.
  - Implement, test, maintain, and evaluate key features of the inference system, including **decoding**, **multi-head attention (MHA) timestamp generation**, and **post-processing** code.
  - Write a meta-prompt, and then use a open-source Large Language Model (LLM) such as Mistral MoE 8x7B to generate hundreds of dialogue scenario prompts, and combine these prompts with different reading formats of digital sequences generated by the LLM model. By combining these two texts, create text with numbers in dialogue scenarios, and pass the text to colleagues to generate audio using internal/Microsoft TTS services, thereby obtaining a test set and some training data. After fine-tuning the production model with the training data, there is a slight improvement on another internal digital test dataset (Absolute digital WER is reduced by about 0.4%).
  - (Ongoing) Conduct internal evaluation for the ASR capabilities of large audio models, such as Qwen Audio, Whisper.
  - Independently completed the implementation, optimization, and performance evaluation (WER, RTF/latency/throughput) of Whisper inference support, using in-house VAD (Voice Activity Detection) model and the open-source WhisperX. Achieved higher throughput compared to the official implementation and lower WER on most test sets.
  - Implement the multi-head attention (MHA) based time alignment over LAS/seq2seq model so providing a good word level timestamp for multilingual transcription for fulfilling business requirements. [A U.S. patent has been filed for this]
  - Cooperate with the downstream web and infrastructure team to simplify the offline transcription system architecture and deployment to support **35+ single-language** models for **5+** different downstream services in **10+ different regions around the world**.
  - Build a systmatic model management practice from scratch to support complex combination in response to the business and languages requirements.
  - Evaluate the performance of the speech recognition system, and find the most suitable parameters to reduce real-time factor, or improve the throughput of the system.
  - Watch closely on online bugs reported from customers, and fix them either from code, or from data quality side.
  - Cooperate with hardware manufacturers (such as Nvidia) to evaluate the inference performance of proprietary inference systems in order to reduce the resource costs of inference.
  - Some cutting-edge explorations that are being done and planned to be done: e.g. LLM re-score/generative error correction etc.

- 2020.09 - 2021.12 **Data Scientist** @ Barclays

  Tech stack: Spark / PySpark, Amazon Deep Java Library (DJL), Tensorflow / Keras, Pandas, Jupyter, Pretrained Transformers / Likelihood Ratio

  - Company address matching and entity matching without internal GPU and labeled data available. Solve using an active learning method. Start from constructing some small datasets only with external data and training an XGBoost tree, then label samples in the boundary and fine-turn BERT models in an iterative way. Finish the inference on 6 million internal pair-wised samples with this model on a CPU cluster, using a DJL based pipeline built from scratch on my own. It achieved a very satisfying result of **94% F1 score on a noisy testing dataset from 89% where we started**. The model does inference offline on our Spark cluster in a distributed way. For 6 million pair-wised samples, the running time is under 1 hour (on a cluster with 80 CPUs).
  - Predict the aggregated user's transaction activity (volume and value) using the historical mean and Informer model, a variant of Transformer for time-series modeling. Following that, a counterfactual was constructed to provide an evaluation of how much finance loss that the bank suffers from system downtime and to find out the critical period for the system reliability.
  - Maintain the Spark cluster for the team, and build up pipelines for distributed inference by combining DJL / PySpark UDF with models.Collaborated with one of my colleagues, we created a team-wised package to start a Spark session within 4 lines of codes, which significantly reduces the overhead of using Spark for colleagues who are not with a distributed computing background.

- 2019.08 - 2020.09 **Java Backend Developer** @ Barclays

  Tech stack: Openshift (Kubernetes), GridGain, Maven, Gradle, Wiremock, Mockito, Spring Boot

  - End-to-end function development, testing (unit, functional, performance), deployment (CD)
  - Add cache layer to the existing APIs to reduce the latency for repetitive data access
  - Migrate legacy codes to internal Spring Boot templates, with refactors to enhance code readability and performance
  - Build up handy internal tools (e.g. git hooks) and scripts (python / bash) from scratch to automate software development processes

Education

2018 - 2019	MSc Web Science and Big Data Analytics @ **University College London,** Distinction
2016 - 2018	BSc Internet Computing @ **University of Liverpool *,** First class
2014 - 2016	BSc Information and Computing Science @ **Xi'an Jiaotong-Liverpool University ***

***Note:** 2+2 pathway routine (first 2 years in Suzhou, China and final 2 years in Liverpool, UK), dual degree.

Early Stage Projects

- 2019.06 - 2019.09 **Project Internship (Master Degree Thesis)** @ Astroscreen

  Social media posting language source identification (tweets and gabs) project. Finished a crawler for collecting language (posts) data from Gab.com, pre-processed data using Regular Expression, built models for classifying the source of these data by fine-turning BERT and XLNet, visualized results using t-SNE, did "leave-one-hashtag-out" cross-validation, and evaluated models using some common matrics (Accuracy, F1 score, Confusion Matrix, Matthews Correlation Coefficient). After fine-turning, XLNet shows a 86% F1 score on hashtag-balanced test dataset, reducing from 97% on random-balanced test dataset. The results show the protential for doing source checking using a model and also indicate the importance for avoiding data leakage.

- 2019.02 - 2019.03 **Integrating BERT and Embeddings into CommonsenseQA Chanllenge**

  We fine-turned Google BERT to CommonsenseQA challenge 1.0 (with 3 options for each question) and then integrated Conceptnet Numberbatch and ELMo embeddings attempting to improve the model performance. The challenge involves a set of MCQ questions requiring human commonsense knowledge. We achieved 68.79% of accuracy on validation set using BERT + ELMo (soly BERT : 67.47%; BERT + Numberbatch: 67.68%).

Technical Article

- **"Accelerating Deep Learning on the JVM with Apache Spark and NVIDIA GPUs"**

  Author: Haoxuan Wang, Qin Lan [AWS], Carol McDonald [Nvidia]; Link: https://www.infoq.com/articles/deep-learning-apache-spark-nvidia-gpu/?itm_source=articles_about_ai-ml-data-eng&itm_medium=link&itm_campaign=ai-ml-data-eng