Hao-xuan (Horace) Wang

+44 (0)7774857427

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 **Software Engineer, ASR and Language Tech** @ Zoom
  - Develop, test and productise ASR offline transcription service, and turn the system parameters to minimise its inference duration (i.e. real time factor, RTF)
  - Develop computational operators of Pytorch for ASR (automatic speech recognition) models, using CUDA/OneDNN, for speeding up the inference and minimising the cost of the existing inference system
  - Trail and experiment those emerging end-to-end ASR + machine translation models, to explore the possibility of landing SLT (speak language translation) models in Zoom products
- 2020.09 - 2021.12 **Data Scientist** @ Barclays

  Tech stack: Spark / PySpark (on Elastic Data Platform), Amazon Deep Java Library (DJL), Tensorflow / Keras, BitBucket, Pandas, Jupyter, Pretrained Transformers / 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. 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.
  - Participated in a fully immersed 6-weeks cloud DevOps training, which involves the deployment of a working pipeline including GitHub, DockerHub, Jenkins, and AWS EKS (Kubernetes) cluster, using Terraform and Ansible.
- 2019.08 - 2020.09 **Java Backend Developer** @ Barclays

  Tech stack: Jenkins, Jira, Confluence, BitBucket, Openshift (Kubernetes), Docker, GridGain, Maven, Gradle, Wiremock, Mockito, Spring Boot, SonarQube, Karate, AppDynamics
  - 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 **University College London** , MSc Web Science and Big Data Analytics, Distinction

  Probability Graphical Models;Deep Learning; Complex Network; Affective Computing; Statistical NLP; Information Retrieval
- 2016 - 2018 **University of Liverpool** , BSc Internet Computing, First class
- 2014 - 2016 **Xi'an Jiaotong-Liverpool University** , BSc Information and Computing Science

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

Selected Projects

- 2021.09 - 2021.10 **Wechat chat history analysis**

  It was a gift for one of my important friends for a friendship anniversary. I collected all of our chat history (in Chinese). Take an analysis of the following aspects: the emotion appeared in our single chat sentences (80% accuracy for fine-turning a Chinese version Roberta model for a 6-classes dataset including happy, natural, angry, fearful, anxious, exciting), word cloud generation, Wechat emoji counting, and hourly chat statistics. The final delivery is a mobile html5 page constructed using a library wechat-h5-boilerplate. This was an end-to-end project from data collection (WeChat does not provide any public way of exporting chat history).
- 2021.08 - 2021.09 **Implementation of Conflict-Free Replicated Data Type (CRDT) - set and graph**

  A project from one of my previous take-home interview exercises finished in a week. It is a self-contained,fully-functional and properly-tested implementation of Last Write Win (LWW) Graph and Set in Python, which is one type of CRDT. CRDTs can be replicated across systems, they can be updated independently and concurrently without coordination between the replicas, and it is always mathematically possible to resolve inconsistencies that might result. Github: https://github.com/billweasley/Last-write-win-CRDT-graph
- 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).
- 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