

Work Experience

• 2020.09 - Current **Data Scientist** @ Barclays

Tech stack: Spark / PySpark (on Elastic Data Platform), Amazon Deep Java Library (DJL), Tensorflow / Keras, BitBucket, Neo4J, Pandas, Jupyter Notebook, Pretrained Transformers / RNNs / 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-tuning BERT models in an iterative way. Finish the inference on 1.5 million internal samples with this model on a CPU cluster, using a DJL based pipeline that built from scratch on my own. It achieved a very satisfying result.
- Build up a transactional graph, with a billion level of dataset. Apply network analysis on targeted clients to find insights for them, and train the account embeddings using DGL (Deep Graph Library), Keras on Tensorflow, and graph attention network (GAT) for a small part of nodes in the large graph (This work in progression).
- Maintain the Spark cluster for the team, and build up pipelines for distributed inference by combining DJL / PySpark UDF with models.

• 2020.08 - 2020.09 **Natural Language Processing Engineer** @ Kwai Inc (Kuaishou)

Tech stack: Spring, Ceph, Dragonfly (A Opensource P2P file distributed system From Alibaba), Tensorflow, Faiss, Docker, Gitlab

- Model Migrations for an newly built internal model management system
- Maintain an internal inference system, build upon Spring
- Retaining a transformer-based user language model using newly incoming data (~ millions, fetched by Hive SQL) to mitigate data shift issue

With more than 300 million daily active users (DAU), Kwai is one of the largest short video sharing and live streaming social platforms in mainland China and also the world. It was a great experience. My leave was for a better work-life balance, and a plan to travel around Europe.

• 2019.08 - 2020.07 **Backend Developer** @ Barclays

Tech stack: Jenkins, Jira, Confluence, BitBucket, OpenShift (Kubernetes), Docker, GridGain, Maven, Gradle, Wiremock, Mockito, Spring Boot, AWS, 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
- Take some NLP side projects (e.g. Address NER using BERT), and trace the state-of-the-art progress in NLP area, especially in the topic of transfer learning, model compression (for low resource inference) and multi-modality fusion
- Keep a good communication and delivery efficiency during COVID-19 pandemic, when the team has to work from home for months

Education

• 2018 - 2019 **University College London** , MSc Web Science and Big Data Analytics, Distinction

Core subjects: Probability Graphical Models; Introduction to Deep Learning; Complex Network; Affective Computing; Statistical NLP; Information Retrieval; Multi-agent AI, Applied Machine Learning

• 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.

Projects

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

Tech stack: Python, Keras, Tensorflow, MulticoreTSNE, Matplotlib

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-tuning BERT and XLNet, visualised results using t-SNE, did "leave-one-hashtag-out" cross-validation and evaluated models using some common metrics (Accuracy, F1 score, Confusion Matrix, Matthews Correlation Coefficient).

• 2019.03 - 2019.04 **Information Retrieval Course Project**

Multiple practices using Fact Extraction and Verification (FEVER) dataset Including word counting and verification of zip's law; implementation of vector space information retrieval (TF-IDF); implementation of query likelihood document retrieve (applying Laplace Smoothing, Jelinek-Mercer Smoothing and Dirichlet Smoothing, respectively); implementation logistic regression to predict sentence relevance; implementation of Precision, Recall and F score function; using neural networks to predict document truthness.

• 2019.02 - 2019.03 **Integrating BERT and Embeddings into CommonsenseQA Challenge**

We fine-tuned Google BERT to CommonsenseQA challenge 1.0 (with 3 options of 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%).

• 2019.02 - 2019.03 **Maximise number of clicks through AD CTR prediction and bidding functions selection**

Tech stack: Python, Keras, XGBoost, Numpy, Pandas, Matplotlib

Predict whether a user would click the online AD (advertisement) on an AD real-time DSP bidding history dataset. The prediction results then were inputted to a bidding strategy function to predict a bid price. The total pay price is bounded by a constant total number. The dataset is unbalanced with only about 3000 positive samples (clicks) among more than 300000 bidding records. We tried many different models (XGBoosting, Shallow NN, Logistic Regression) and some bidding strategies. We also applied downsampling and re-calibration techniques in the project. We did a competition in a leaderboard with other students (30 groups) and ranked in the 3rd place (with 185 clicks and the first 2 are with 186 clicks).