BOYI LI

+1-512-207-0536 ◇ boyizjuer2017@gmail.com ◇ https://clamli.github.io ◇ ◇

EDUCATION

The University of Texas at Austin — Austin, TX
Master of Science in Computer Science, GPA: 3.89/4.00, Graduate Research Assistant — *Aug. 2018 – May. 2020*

Zhejiang University — Zhejiang, China
Bachelor of Engineering in Computer Science and Technology; GPA: 3.91/4.00 (10%) — *Aug. 2014 – June. 2018*

PROGRAMMING SKILLS

Skills: Database, Compiler System, Web Development, NLP, RecSys, Data Analytics, Wireless, Automation Testing
Languages: Java, JSP, JavaScript, HTML/CSS, Python, C/C++, Matlab, R, Verilog, SQL, Git

INTERNSHIP/EXPERIENCE

Automate Pricing of Amazon Seller Central — May 2019 - Aug 2019
Software Development Engineer Intern, Pricing Systems, Amazon.com, inc.

- Implemented feedback loop on automate pricing landing page of Amazon Seller Central (**Java**) to show seller's action status, designed and added additional columns in the table (**DynamoDB**) to save information related.
- Added additional API calls in the controller of automate pricing editing page, fetched data related with the impact of seller's editing action and displayed on the web page.
- Worked closely with UX team to figure out user-friendly interactions on both landing page and editing page, added and adjusted web elements (**JSP/HTML/JS**) for informing sellers the status of their action.
- Optimized front and back end code logic involved when seller's editing action takes place, alleviated the burden on the web server by decreasing requests sent to the back end API.
- Performed front end integration testing by **DiamondToolkit**, ran **A/B testing** with Amazon's Weblab and rolled out new features to sellers incrementally.

Collaborative Active Learning Tree for Cold-Start Item Recommendation — Oct 2017 - Feb 2018
Visiting student in Simon Fraser University, Canada, supervised by Prof. Martin Ester and Prof.Chung Fu-Lai

- Proposed a Collaborative Active Learning Tree model (**Matlab**) for cold-start movie recommendation.
- Empowered the tree-based learning algorithm by exploiting content information of movies, solved low responsiveness and unbalanced tree problems.
- Applied matrix factorization implemented by Spark to predict movie preferene of user clusters.
- Achieved low RMSE, outperforming baseline models with **5%** boost on accuracy for predicting ratings of cold-start movies.

Mobile App for Food Logging and Recommendation — Jul 2017 - Sep 2017
Research intern in Ubiquitous Computing Lab, National University of Singapore, Singapore

- Designed a RESTful API (**Python**) with Flask to interact with the app, processed the pictures posted by users and fetched the nutrient values of the food in the pictures.
- Implemented two-constraint sorting algorithm (**Python**), sorted foods according to their taste and health value.
- Reduced the app's built-in camera delay by using AsyncTask.

PROJECTS

Corruption-aware Tensor Robust PCA. An optimized **Tensor Robust PCA (TRPCA)** for image recovery. Incorporated corruption-aware mechanism based on Rank-Ordered Logarithmic Difference to locate corrupted pixels and constrain the entries of the noise matrix. Boosted the PSNR performance by 10%.

Topic-aware Seq2Seq Model for Text Summarization. An attention-based encoder-decoder model (**Python**). Implemented **attention-based encoder-decoder model**, optimized it by adjusting probability of repetitive word phrases in beam search and incorporating pretrained **LDA** topic model.

WatchOut!. A wireless intrusion detection system based on acoustic information (**C++/Python**). Achieved data recording and transmission on **Raspberry Pi**, implemented acoustic event detector by using sound energy, built a web monitor by **Dash** and displayed dynamic wave data in real time.

MiniSQL. A MySQL-like Database Managerment System (**Java**). Implemented basic SQL commands, built index manager on **B+ tree** to double query efficiency, achieved buffer manager based on **LRU algorithm** to minimized disk access time.

Room Break. A game implemented by **OpenGL**. Created game scene by **Maya**, built light controller and texture controller to optimize the scene, designed movement and collision detection modules to trigger events.