

Kelei Gong

Tel: 919-986-1543/ Email: kgong@ncsu.edu
Add: 4210 Sterlingworth Dr, Raleigh, NC 27606

SUMMARY

- Backgrounds in Computer Science and Mathematics. Experience in Back-End development, DevOps, Database Design, Data mining. I work as a Research Assistant in the Department of Management, focusing on developing an auto-rating system including Web crawling and Data mining.
- Skills: Python, Java, R, C, SQL, Git, Ruby, Mac OS, Linux, HTML/CSS.

EDUCATION BACKGROUND

North Carolina State University / MS in Computer Science / Current GPA: 4.0/4.0 **2014-2016**
Shenzhen University / Master in Finance **2011-2013**
Nanjing University of Science and Technology / BS in Applied Mathematics **2007-2011**

EXPERIENCE

Research Assistant at North Carolina State University **2014-Present**
Supply Chain Management Auto-Rating system

- Designed a whole Data Mining process involving crawling data from Internet, cleaning/processing data, storing data and applying Machine Learning algorithms to data to generate required results.
- Perform asynchronous crawling through proxies with different IPs (to avoid being blocked). Proxy servers are deployed on AWS EC2, achieved quick deployment by using AWS API and Ansible-playbook
- Process the collected text data, including parsing HTML, removing stop words, stemming, etc, generate formatted data.
- Apply Machine Learning algorithm on the collected data, assigning ratings(1 ~ 5) to companies' supply chain management performance in different aspects.
- Created a website (developed in Django Framework) for results presenting and data maintaining.

PROJECTS

Operating System Design **2015**

- Extended an OS called Xinu, worked on its source code directly. Including implementing a Linux-like scheduling policy based on process priority, and a Multiqueue scheduling policy
- Archived the Demand Paging feature, including the implementing of the whole paging system(like page directory and page table), page fault handler, page replacement policy, and other needed system calls.
- Implemented Readers/Writer locks, used to synchronize access to a shared data structure where a lock acquired for writing must exclusive.

Continuous Integration and Deployment Pipeline Design **2015**

- Integrated a set of Git Hooks, shell scripts, Redis, Docker and CI tools(Jenkins) to create a pipeline to build, test, analysis and deploy newly committed code.
- Started from a pre-commit hook, perform a series of test and analysis to avoid commit on broken code. Commit will be rejected once certain errors/warnings are detected.
- Upon a successful commit, use a post-commit to send notification to Jenkins to build and test the code, then deploy the new code to servers if nothing goes wrong. Used Ansible and Docker for the production environment configuration.
- Implemented a Canary release strategy for deploying new version, routing most traffics to the stable version and the rest to the new version. Once the new version works well, routing more traffic to it.
- Created a load balancer and monitoring app to ensure all the servers have similar workload.

Spam filter based on Semi-Supervised Learning **2015**

- Expended Self-training algorithm as a wrapper that turns a Supervised learning algorithm into a Semi-Supervised version, thus can make use of unlabeled data and improve accuracy
- Implemented Bagging method to further improve the performance, finally got a 95% accuracy compare to 91% in Supervised Learning.