Yijiashun(Elijah) Qi

Email: elijahqi@umich.edu
Mobile: +1-734-450-5998
Personal Website
Github: github.com/elijahqi

Education

- **University of Michigan** — Ann Arbor, MI
  Bachelor of Science - Computer Science and Data Science; GPA: 4.0/4.0 — *Sep 2021 - May 2024(expected)*
  Honors/Awards: Three terms University Honors, James B. Angell Scholar (Two consecutive A terms)
- **The Hong Kong Polytechnic University** — Hong Kong
  Bachelor of Business Administration - Financial Services; — *Sep 2019 - May 2021*
  Honors/Awards: Two years Dean's List (Awarded to top 10% of the whole Business School)

Technical Skills

- **Programming language**: C++, Python, HTML/CSS, JavaScript, SQLite, R
- **Frameworks/Technologies**: React, Bootstrap, jQuery, Flask, Jinja
- **Tools**: Git, Linux

Professional Experience

- **School of Information, University of Michigan** — Ann Arbor, MI
  Research Assistant with Professor Jinseok Kim — *May 2022 – Present*
  - Read all the json files including subfolders, parsed them according to the author's name and the name of the published paper, assigned each article and author a different id, and merged them by comparing the similarity of their last name+first name's first three letters+suffix strings to determine if authors need to merge their ids.
  - Used dictionary in python to save authors' name for every paper and used Union find to merge the authors whose name's similarity(provided by difflib.SequenceMatcher()) is greater than 0.85.
  - Improved efficiency of finding all the JSON files by finding one path of a JSON file and read one JSON file instead of saving all the paths and reading the JSON file from the beginning of the list, reducing the runtime from 43 minutes to about 2 minutes.
- **The Pacific Securities** — Shanghai, China
  Industry Analyst Intern — *July 2021 – Aug 2021*
  - Retrieved the latest data on the number of stock prices and research reports in the automotive industry and saved them as JSON files by using Pandas and NumPy.
  - Automated the data collection and collation including market value in circulation, latest trading day increase or decrease,reduced the data collecting time from 2 hours to 10 minutes.

Projects

- **Insta485 Web Project**
  - Developed an Instagram-like web application using React for front-end, REST APIs in Python using Flask as the back-end, and SQLite for database.
  - Implemented features, which includes creating, updating, and deleting users, posts, comments, likes and infinite scrolling, using click and trigger event listeners, and browser routing.
  - Deployed this website using EC2 instance on Amazon Web Service.
  - Encountered problems that the number of likes and new comments do not update automatically. Managed by lifting up state to allow like-component and comments-component to share the data and prevent complex state management.
- **Piazza Posts Classifier**
  - Designed and implemented a classifier to classify Piazza posts data according to the probability of the tag of the posts.
  - Used the training data to figure out the percentage of posts of each type to the overall posts, figured out which type the post has the highest probability of belonging to based on the read-in data from the test set, and assigned it to the current post.
  - Increased the efficiency of worst case from $O(n^2)$ to $O(nlog(n))$ by implementing self-balancing Binary Search Tree to create a container called map to save the probability for each tag and word.
- **Automation Web Screenshot**
  - Designed and implemented an automated web screenshot software that scrolls through the first two pages of a search engine and generates screenshots based on a given company name and corporate name.
  - Opened the search engine type in the search content using Selenium; found whether there is a "next page" in the CSS selector; saved at most two pages of results according to the CSS selector result. Used JavaScript to get the length and width of the page and then take a screenshot, looped every 1080PX to cut a picture, and then named it according to the order of precedence.
  - Reduced the preparation work for an enterprise risk assessment to 2 hours on average.