Haoran Cheng

📞 971-900-6780 ✉ haorancheng@brandeis.edu 🔗 Haoran Cheng ⬡ Rsirp0c

Education

Brandeis University, Waltham, MA <div align="right">**Sep 2021 – June 2025**</div>

Bachelor of Science in Computer Science / Bachelor of Art in Anthropology <div align="right">*CS GPA: 3.8*</div>

Awards: Dean's List (Fall 2022 - present)

Coursework: Data Structure, Operating System, Natural Language Processing, Deep Learning, Software Engineering, Information Visualization, Object-oriented programming in Java, Database management, Web Application Development

Experience

Chewy | E-commence <div align="right">**June 2024 – present**</div>

Software Engineer Intern <div align="right">*Boston, MA*</div>

- Developed the software template feature for a corporate-level internal tool built on **Backstage**, actively used by 500+ developers. Automated the deployment of **Helm charts** and **Terraform** onto a shared **AWS EKS cluster**.
- Implemented **GitHub Actions** and **Jenkins** CI/CD pipelines to streamline development workflows and project config.
- Coded the backend **Cookiecutter** action plugin in **Typescript**, later simulated and verified with 10 **Jest** unit test.
- Utilized **Jenkins**, **Docker**, **Poetry**, and **Python** script to auto deploy customized Docker Image to **AWS ECR**.

Zbyte Technology | Data Warehouse Startup <div align="right">**May 2023 – Sep 2023**</div>

Technical Product Manager Intern <div align="right">*Hangzhou, China*</div>

- Involved in the design, development and product launch of a **RAG** (LLM-dataset) web application, improving domain-specific response accuracy; collaborated with a cross-functional team of 10+ members.
- Directed **design reviews** bwtween UI/UX and engineer teams, resolving over 150+ frontend component errors.
- Drafted **documentation** for the data warehouse, including, 50+ SQL Commands and 10+ Build-in Functions.; published three high-engagement **market analysis articles**, each attracked 500+ read.

Branda | Brandeis Campus App <div align="right">**Jan 2023 – Jan 2024**</div>

Software Engineer <div align="right">*Waltham, MA*</div>

- Responsible for developing mobile application using **React Native**. Deployed a interactive calendar daily used by 1.6K student to keep track of school events. Used **Redis** to cache data, reducing the workload on main database by 20%.
- Managed database migration from Heroku to **MongoDB**, implemented migration testing within the CI/CD pipeline.

Quant Club | Brandeis student club <div align="right">**Jan 2023 – Sep 2023**</div>

Data Analytic/Engineer <div align="right">*Waltham, MA*</div>

- Implemented a data warehousing solution using **Snowflake** to centralize financial data, facilitating big data anlysis.
- Developed the ETL pipline using **Airflow** to integrate 10 years historical data from pyfinance and earning calender.
- Built a interactive earning calendar in **React** that retrieve data from data warehouse using a **Django** backend.

Projects

RAG LLM-dataset chatbot | *Pinecone, Langchain, Cohere, OpenAI api, Streamlit* | *git*

- Developed a **retrieval-augmented generation** chatbot that support difference choices over Embedding model, chunking strategy, top k retreival, LLM model, prompt engineering, and meta-data retreival search.
- Implemented the file uploading workflow; Utilized **Langchain splitter module** and **Python** script to clean and chunk the uploaded file and use **Huggingface sentence transformers** and **Pinecone** to vectorize and store data.
- Used **SpaCy NER model** to retreive meta data from prompt and ran a **hybrid search** using meta-data and vector.

Desktop LLM App | *Qdrant, Tauri, React, Typescipt, OpenAI api* | *link*

- Designed and developed a cross-platform desktop LLM Chat app, enabling chat over user-upload dataset; providing a more accurate response to domain-specific inquiries than ChatGPT.
- Utilized Embedding and Searching from **OpenAI API** to optimize Chat app's response. Split the user-upload file into small chunks; used OpenAI Embedding model to vectorize each chunk and save them into **Qdrant** database. Transform user query to vector using OpenAI, and then inquire the top match text chunk from Qdrant database using top_k value.
- Developed the frontend of the desktop app in **Tauri** and implemented **React** components with **Typescript** and **MUI**.

Technical Skills

Programing Languages: Python, Java, C++, UNIX/Linux, JavaScript, Typescript, HTML/CSS
Frameworks: Pandas, PySpark, D3.JS, FastAPI, Flask, React, Node.JS, Express.JS, Docker, Kubernetes, Kafka
Database and DevOps: Jenkins, Terraform, PostgreSQL, DynamoDB, Pinecone, AWS, Github Actions
Other: Figma, OpenAI API, Excalidraw, Trello, REST