SALT LAKE CITY, UT

RICHIE FROST

(385) 335-0115
RICHIE.FROST@UTAH.EDU

EXPERIENCE

RESEARCH ASSISTANT UNIVERSITY OF UTAH AUGUST 2016-PRESENT

- Produced and completed a thesis on a highly scalable machine learning algorithm for unstructured data. Reduced network load by 55% with 94% less memory consumption.
- Published a paper on spatiotemporal sentiment analysis at a top-tier data conference (KDD) in 2017.
- Led five other research assistants as team lead to optimize database latency and throughput.

SOFTWARE ENGINEER CONSULTING AUGUST 2014- AUGUST 2016

- Wrote large-scale applications with external API integrations for a freight shipping platform.
- Developed custom data-driven modules for the State of Nevada's digital library.

SOFTWARE ENGINEER ROCK SOLID INTERNET SYSTEMS MAY 2012 – AUGUST 2014

- Managed MySQL database; wrote and optimized queries for an enterprise reporting platform.
- Pair-programmed a secure API to streamline the customer ordering process.

EDUCATION

SALT LAKE CITY, UT UNIVERSITY OF UTAH JANUARY 2015 – MAY 2019

- M.S. in Computer Science, expected May 2019. GPA: 4.0
- B.S. in Computer Science, August 2017. GPA: 3.72
- Graduate Coursework: Information Extraction, Database Systems (in progress), Machine Learning (in progress).
- Undergraduate Coursework: Natural Language Processing, Artificial Intelligence, Math for Data, Computer Security, Databases, Algorithms, Linear Algebra, Calculus III, Operating Systems.

PROJECTS

- **Frost LDA** (2017). Built for massive scale, it uses a distributed, asynchronous machine learning algorithm to discover the latent topics in unstructured data with only 6% of the memory usage of the industry standard Python solution and 55% less network bandwidth usage. Written with Python, Numpy and Pyro4 (currently porting to C++ and Protobuf).
- **Event Extractor Plus** (2017). An app that uses machine learning to summarize bodies of text by learning which key words and phrases are the most important in a document. Takes a document's raw text as input and outputs a structured summarization of the document. Written in Python and NLTK, based on XML data.
- **Co-reference Resolver** (2017). As one of the top-performing co-reference resolver programs in a class project, this program uses a sequence of semi-deterministic heuristics to group nouns and pronouns that refer to each other in a body of text. Written in Python, based on XML data.

LANGUAGES AND TECHNOLOGIES

- **Data Science/Machine Learning**: Python (proficient), Numpy, NLTK, Scikit-Learn, Scrapy, Matlab (previous experience)
- **Back-End**: Python, Java, C/C++ (previous experience), Node JS, TypeScript
- **Front-End/Visualization**: JavaScript, CSS/HTML (proficient), React JS, D3
- **Databases**: MongoDB, MySQL/SQL, Cassandra (previous experience)
- **Other**: Git (proficient), Github, LaTeX, Agile Methodologies (Scrum, Kanban)