Paul Miller-Schmidt

pmillerschmidt@gmail.com | paulmillerschmidt.com | github.com/paulmillerschmidt | linkedin.com/in/paul-miller-schmidt

EDUCATION

Yale University, New Haven, CT, GPA: 3.7 - Bachelor of Science, Computer Science, May 2023
Relevant Coursework: Software Engineering; Natural Language Processing; Systems Programming and Computer Organization; Data Analysis; Data Structures; Programming Techniques; Discrete Mathematics; Machine Vision

Languages: C, C++, Java, Python, JavaScript, TypeScript, NodeJS, HTML, CSS, SQL, Supercollider, Racket
Frameworks and Tools: React, Vue, Nuxt, Jira, Firebase, TensorFlow, PyTorch, Keras, Bitbucket, Git, TestRail, AWS

EXPERIENCE

Mathpix, Brooklyn, NY 2023-Present
Machine Learning Engineer

- Led end-to-end OCR and segmentation model development (data management, training, and performance optimization).
- Engineered custom loss functions, data augmentation algorithms, and model architectures, boosting OCR and segmentation model accuracy (TensorFlow, PyTorch). Fine-tuned hyper-parameters and implemented advanced optimization techniques.
- Spearheaded overhaul of the string/document processing pipeline, including tokenization, normalization, and entity extraction, resulting in 20% faster processing times and enhanced accuracy for complex text and specialized notation.
- Developed automatic annotation and verification tools, increasing annotator productivity by 30% (Python, JavaScript).
- Designed and implemented robust synthetic data generation systems, improving performance across multiple domains.
- Enhanced testability, visibility, and telemetry of OCR and segmentation model development pipelines.
- Designed and implemented new API features, improving system integration and expanding product capabilities.
- Optimized chart, table, and chemistry output parsers, elevating parse success rates by 50% and increasing user satisfaction.
- Conducted product demonstrations, managed customer-facing issues, and represented the technical team in client meetings.

Artiphon, Nashville, TN (Remote) 2022
Software Engineer Intern

- Developed cross-platform C++ code to facilitate MIDI communications between Mac/iOS and low-level embedded systems.
- Designed optimization algorithms to decrease time and space complexity of communication protocols between devices.
- Leveraged CI/CD pipeline and automated testing to deploy core product features using Bitbucket, Jira, and TestRail.

GravitateAI, Boston, MA 2021
Software Engineer Intern

- Built AI-based patient-provider matching algorithm in Python for Guidely.com (personal coach marketplace).
- Designed sentiment analysis model using Word2Vec and SpaCy libraries. Analyzed data from 500 patient intake surveys (client goals, demographics) and 150 provider profiles (education, methods, target base) to optimize provider-client pairings.
- Processed audio and video files with Google Cloud's speech-to-text API to feed as training data for analysis model.

Albo Climate, Tel Aviv, Israel (Remote) 2020
Head Technical Consultant

- Led team of five Yale students across all contracted Albo Climate objectives. Served as liaison to Albo's executive team.
- Built JavaScript interfaces to demo AI-powered carbon sequestration visualizer and improve client's UI/UX.

SOFTWARE PROJECTS

MelGen - Yale University Open Music Initiative 2023

- Developed LSTM-based melody generator with attention to underlying harmonic structures using Python and TensorFlow.
- Designed data pipelines to scrape and parse musical data into a functional MIDI format using Music21, JSON, and Pandas.
- Researched and applied various machine learning models for melodic generation, such as RNNs, GANs, and VQ-VAEs.

WalkHome 2022-2023

- Led a startup of six engineers to build an app for users to track their friends' commutes home with a variety of safety features.
- Developed Vue/Nuxt-driven PWA with Firebase backend. Integrated Google auth, Maps API, and push notifications.
- Automated unit testing (Jest) and implemented deployment processes. Iteratively improved features based on user feedback.

Markov Music Engine 2022

- Built context-sensitive, Markov-based melody generator in C++. Algorithm incorporated a Trigram Hidden Markov Model and user-specified hyper-parameters (repetition, variance, variability, chord tendencies) for precise melodic control.
- Program parsed artist-specific MIDI to emulate their style. Model was designed to work on low-memory, embedded systems.

GrooveBlocks 2020-2021

- Founded startup to create original, combinatory tools for composition-based music education. Led a team of five developers and two designers to build core hardware product and complementary Android application.
- Engineered modular MIDI devices using Teensy microcontrollers. Developed Java-based Android audio engine and implemented C++ drivers for seamless sensor and MIDI data communication between hardware and software components.