



Heman Gandhi

Pittsburgh, USA

Phone : +1 (609) 903-8902

Email: hemangandhi@gmail.com

linkedin: [linkedin.com/in/hemangandhi](https://www.linkedin.com/in/hemangandhi)

Web: hemangandhi.github.io

Date of Birth: 29th August 1997

Summary

I am passionate about learning, problem solving, and creating software. From working in Google's Shopping Data Org, I have not only learned the technical skills needed for Big Data Analytics and Machine Learning, but developed the soft-skills needed to focus my analysis on business needs. Prior to this, I completed a BS in Computer Science and a BS in Mathematics at Rutgers University while also completing various summer internships and volunteer experiences to broaden my understanding.

Career

Google Inc.

June 2019 - Present

Title: Software Engineer III/Software Engineer II

Project	Dates
Question Answering Model Inference Library	June 2022 – Present

Application: Our NLP question answering model was running offline, outputting daily results that we would upload into our serving system. However, this offline design blocked our pivot to rely on the model instead of older techniques. Hence, I was tasked with understanding the resource requirements to reach the required serving 1 million QPS within the latency bound of 100ms per request.

Software: C++, Tensorflow Serving*, Cloud monitoring*.

Role: After learning the serving systems we were targeting, I led the design for an inference library to productionize our offline code and maximize its online usability after understanding the model's codebase, requirements, and tuning parameters. A collaborator and I have began the refactor to allow the offline pipeline to use the library and began its implementation.

Structured Data Framework

January 2022
– December
2022

Application: Our text-based question answering systems could not leverage structured key-value data. As more structured sources were being provided inside the shopping data models, we saw an opportunity to increase our coverage. Furthermore, we saw that our understanding of key-value pairs could be used by other teams to help curate our semantic data model. I was one of the collaborators in integrating this new data source into our question answering system, launching a 4% coverage increase across our 36 billion items with no quality loss. After the launch, I also implemented a metrics pipeline to monitor the framework's understanding of the 200 billion key-value pairs it processes.

Software: C++, Flume*, Protocol Buffers, BigQuery*, Borg, gUnit, Continuous Integration Tool, ACL management tooling, Looker dashboards*, Colossus, Capacitor.

Role: Initially, I was tasked with running the experiments to prove the utility of the new structured data sources. After estimating a 3 to 4% coverage gain, I was tasked with designing and implementing the conversion from the myriad sources into a unified format. After the MVP launch, I took on some of the cross-team collaboration to enable other teams to use the values we could not interpret to augment our semantic model. I also implemented a weekly metrics pipeline to monitor the overall health of the framework.

Classifier Data Migration

March 2020 –
January 2022

Application: Google shopping's product identity organization began an initiative to internationalize their data by providing a new "global" data type and deprecating the old "local" type. As a critical component of the shopping data platform, the support of our text classifier was an early requirement for the initiative. I was tasked with ensuring that we provided accurate classifications on the new type, that our classifications on the global level became the source of truth for cross-entity smearing, and finally that the old type could be removed from our ML training and evaluation processes.

Software: C++, Tensorflow, Flume*, BigQuery*, TFX, Borg, Cloud monitoring*, gUnit, Colossus, gRPC, Capacitor.

Role: I implemented the required feature extraction and flume pipelines to evaluate our model's performance on the new type. With team approval, I implemented the serving functionality to launch our initial support of the new type, and after further evaluations, launched the use of the newer type as the source of truth. After this, I proposed a process to migrate our training and evaluation sets to use the newer type and remove the older type, and with buy-in from senior team members, performed the required classifier training experiments to work towards this.

Changelog Migration

May 2021 –
August 2021

Application: the underlying Google Shopping Graph semantic data models have curation changelogs, enabling the tracking of human operator and linguist efforts and tracking high-level statistics about the growth of the data models. The storage for the data models was being unified into a graph database representation to simplify its maintenance and use, but this required a migration so that the changelogs could be brought inline with the new representation.

Software: C++, gRPC, Spanner, Protocol Buffers, Integration Testing Framework (RPC replaying tool), Bazel*, Google Graph Storage.

Role: I learned about the new service mesh being implemented to enable the curation of the data models, implementing the changelog for the taxonomy to unblock an MVP launch. After learning the graph representation and implementing a traversal, I also launched the changelog for the semantic model.

Classifier Online Experimentation

August 2020
– January
2021

Application: When launching a new classifier, our infrastructure uploaded experimental classifications computed offline which would lead to noise in the launch experiment. To remove this noise and make our launches quicker and easier to justify, I was tasked with implementing an online experiment framework.

Software: C++, Cloud Monitoring*, A/B Testing Framework*.

Role: Designed, and with approval from senior team members, implemented and launched the online experimental framework, successfully accelerating a subsequent classifier launch.

Classifier Tuning and Metrics

September
2019 – March
2020

After moving from legacy systems to Tensorflow, the ease of experimentation opened up the opportunity for various hyperparameter tuning experiments. To learn about machine learning experimentation, I was assigned a few experiments to run. I found a better learning rate, increasing our precision by 1%. After this, I also refactored our hyperparameter configuration and integrated Vizier to automate some of the tuning.

Software: Tensorflow, TFX, Python, Vizier*, Protocol Buffers.

Schema Curation Changelog

June 2019 –
September
2019

As my introductory project, I implemented a changelog for the semantic graph data model we use in question-answering about products. This curation changelog is used as the source of truth for tracking the productivity of our graph curation processes. With help from my mentor, I learned about best practices at Google and began understand software development.

Software: C++, Spanner, Protocol Buffers, gUnit, Borg.

HackRU RnD

January 2017 – May 2019

Title: Co-director

Application: As a volunteer, co-founded and led my university's hackathon's development team, building and managing various applications. With the new team, we expedited the event registration process, added an online judging website, and a mobile application, among other improvements to the hackathon experience. I owned the web development, performing full-stack tasks but eventually designing, reviewing, and project managing as the team grew.

Software: Python, PyTest, Python Bcrypt, ReactJS, Express JS, Nginx, AWS (EC2, Lambda, S3, Cloudfront, API Gateway), MongoDB, JSON.

Role: In the beginning, I performed full-stack web development tasks to understand and maintain the existing website. However, as we built more functionality, my co-director and I redesigned the system to implement SSO, allowing new applications to integrate with the website and mobile app seamlessly. I also built the core backend API to enable this new design, reducing costs and maintenance overheads by migrating from EC2s to AWS Lambdas. As we grew, my co-director and I also project-managed with other volunteers, coordinating tasks for various applications while mentoring newer members.

Skillshare Inc.

May 2017 – August 2017

Title: HackNY Fellow

Application: SkillShare is an educational marketplace offering a platform for experts to teach and learn, trading skills. I interned with them as a software engineer.

Software: PHP, Yii, Backbone JS, SCSS, MySQL, Git.

Role: I performed routine maintainance, learning to track tasks through JIRA, while also improving their test coverage, fixing their search result structured data, and building a splash page for their v5 launch. Learned Git best-practices, understanding the use of feature branches and rebasing. As a HackNY fellow, I also got to learn about the NYC startup scene through various sessions with entrepreneurs and other leaders in the NYC tech scene.

Optum Inc.

June 2016 – August 2016

Title: Talent Development Program Intern

Application: Optum owns a large COBOL mainframe codebase. Navigating this codebase to understand the relationships between form fields and the variables across the COBOL procedures was a challenging task that Optum sought to simplify. We implemented a website that rendered the form screens and used a SQL database to store and render the details about what copybook and variables each screen would use per field.

Software: C#, JSON, HTML, Bootstrap.

Role: As an individual contributor, I implemented the parsing logic, first understanding the specification of the input form screens and outputted an HTML rendering of the screen with hints for the variables. After this, I also implemented a parser to read copybooks and understand the variables used inside COBOL procedures, including understanding MOVE statements to track values.

Mphasis Inc.

June 2015 – August 2015

Title: Intern

Application: To test ideas for multi-user collaboration interfaces at large scales, where users would simultaneously edit elements on a common screen, we built a multi-screen game. In the game, each user could move a circle about and consume smaller circles to grow until only one user remained.

Software: WAMP, jQuery, JS, CSS.

Role: With a collaborator, implemented the multi-screen game. I owned the logic for the user's controller UI, that each user would interact with to manipulate the common game state on the larger display screen. We coordinated the message passing to be concise to minimize latency, powering an engaging experience.

Skills

- Programming Languages: C++ (17), Python 3, C#, Java, SQL, Rust, JS, C.
- Data science/ML: BigQuery*, Flume*, Tensorflow, Python, Google Colab*.
- Web development: Python (Flask), HTML, CSS, JS, jQuery, NodeJS, ExpressJS, ReactJS, MongoDB, MySQL, gRPC.
- Development Process: Git, Linux, JIRA.
- Cloud: AWS EC2, AWS Lambda, AWS API Gateway, AWS Cloudfront, AWS S3.
- Databases: Spanner, Capacitor, MongoDB, MySQL.
- Languages: English (native speaker), Japanese (intermediate), Spanish (intermediate).

* a Google-internal tool with equivalent functionality was used.

Education

Rutgers University

August 2015 – May 2019

Bachelor of Science in Computer Science

Bachelor of Science in Mathematics

GPA: 3.88

Summa Cum Laude

Honors College

High Honors in Mathematics
