Internship Project Report

on

Generative AI

Submitted in partial fulfillment of the requirement for the Degree

Master of Science (Computer Science)

Submitted by

Satyam Sahu
(23234747056)

Under the guidance of

Dr. Ankit Rajpal
(Assistant Professor)
Department of Computer
Science

Mr. Neeraj Shrivastava
(Senior Manager,
Software Engineering Systems)
Ciena India Pvt. Ltd.

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF DELHI
Academic Year 2024-2025

Declaration

I hereby declare that the work which is being presented in this project entitled **“Generative AI”** is an authentic record of my work is free of plagiarism and has sufficient grammatical clarity to be examined, carried out at Ciena India Pvt. Ltd., Gurugram, India, as requirements of six months of Industrial Training for the award of the degree of M.SC.(CS) under the supervision and guidance of Mr Neeraj Shrivastava, Senior Manager/Software Engineering Systems at Ciena India Pvt. Ltd. and Dr. Ankit Rajpal, Assistant Professor, Department of Computer Science, University of Delhi.

Date : **Satyam Sahu (Roll No: 23234747056)**

Place : New Delhi

Certificate

This is to certify that this work has been carried out by Satyam Sahu (Roll No: 23234747056) at Ciena India Pvt. Ltd. under the supervision of Mr. Neeraj Shrivastava, Senior Manager, Software Engineering Systems at Ciena India Pvt. Ltd. and Dr. Ankit Rajpal, Assistant Professor, Department of Computer Science, University of Delhi. This work has been carried out for the partial fulfillment of the requirements for the degree of M.SC.(CS) in the Department of Computer Science, University of Delhi. This project has not been submitted anywhere for the purpose of any other degree or diploma.

Dr. Ankit Rajpal
(Assistant Professor)
Department of Computer Science
University of Delhi

Prof. Neelima Gupta
(Head)
Department of Computer Science
University of Delhi

Mr. Neeraj Shrivastava
(Senior Manager, Software Engineering Systems)
Ciena India Pvt. Ltd.

Neeraj Shrivastava

Ciena India Private Limited
Plot No. 13, JSK Towers,
Echelon Institutional Area, Sector 32,
Gurgaon, Haryana-122001

+91 124 6254000- Phone
+91 124 6254444- Fax
www.ciena.com

09/06/2025 (D/M/Y)

Employment Verification: Satyam Sahu

To Whom It May Concern,

This letter is to confirm that Satyam Sahu (Employee ID: 33672) has been employed with Ciena India Pvt. Ltd. ("CIENA"), located in Gurgaon, India since Jan 06, 2025. His current designation is Intern/Co-Op.

If you have any further questions, please do not hesitate to contact the People Connect team at Peopleconnect_India@ciena.com
.

Kind Regards,

Brian Ivay
Vice President, People & Culture

Acknowledgement

I express my profound gratitude and deep regard for my supervisor, **Dr. Ankit Rajpal**, for his exemplary guidance, monitoring and constant encouragement throughout this project. The blessing, help, and guidance given by him from time to time shall carry me a long way in the journey of life on which I am about to embark.

I am thankful to **Mr. Neeraj Shrivastava** for allowing me to work on this challenging project. I am also grateful to Ciena India Pvt. Ltd for allowing me to work with them and undertake a project of such importance.

I also take this opportunity to express a deep sense of gratitude to the members of Team Generative AI for their cordial support, valuable information, and guidance, which helped me complete this task through various stages. The present work would not have taken this shape without their suggestions and valuable input.

My stay in the organization has been a great learning experience and a curtain-raiser to an interesting and rewarding career. This exposure has enriched me with technical knowledge and introduced me to the attributes of a successful professional.

Finally, I would like to thank Prof. Neelima Gupta, Head, Department of Computer Science, the University of Delhi, for giving me the opportunity to carry out this project at Ciena India Pvt. Ltd.

Satyam Sahu
(23234747056)

Abstract

Ciena is a global telecommunications networking equipment and software services provider. The company specializes in optical transport and packet networking solutions, enabling high-speed data transfer and connectivity for various industries worldwide. The report describes the internship journey and the project “Gen AI” I undertook in detail.

The report introduces team operations and their role, then discusses the technology and concepts utilized in the project, and discusses the tasks and project information in detail. The internship project introduced me to different tasks, the details of which are as follows.

The initial hands-on training activity mainly consisted of tutorials and small tasks to get familiarity with the workflow of the tasks and the organization’s product and technology. I was introduced to Navigator Network Control Suite and the various services it offers which also includes MCP-Chatbot to get an overall understanding of the project.

Navigator Network Control Suite is a multi-layer, multi-vendor network optimization tool that provides a centralized view of optical, Ethernet, and IP infrastructure. As part of Ciena’s Navigator NCS tool, MCP-Chatbot supports the gen ai conversational help in Navigator NCS. The main task of the internship project consists of “Developing a Conversational Chatbot for Navigator NCS”. It ranges from fixes issues, R&D for product optimization and accuracy, script writing for automate working and automate the debugging process through script writing.

The report mainly aims to convey my contribution, my learnings, and the technical stack I used during the internship. The report finally concludes with the skills acquired, key learnings, and experience of the internship.

Contents

1 Introduction 1

1.1 Company Introduction . . . 1

1.2 Products and Services . . . 2

1.2.1 Converged Packet Optical . . . 2

1.2.2 Routing and Switching . . . 2

1.2.3 Network Control and Management Software . . . 3

1.2.4 Intelligent Automation Solutions . . . 4

2 About Team 5

2.1 Objective . . . 5

2.2 Team Introduction and Composition . . . 5

3 About Project and Activities Performed 6

3.1 Objective . . . 6

3.2 My Learnings and Contribution to the Project . . . 7

3.2.1 Initial Hands-on Training Activity . . . 7

4 Software Development Methodology 9

4.1 Agile Software Development Methodology . . . 9

4.1.1 Agile Project Life Cycle . . . 10

5 Internship Report 11

5.1 Requirement Analysis . . . 11

5.2 Timeline . . . 11

5.3 Activity 1: Research and Development Using Open-Source Tools such as DSPy and TextGrad . . . 13

5.4 Activity 2: Data Parsing Script Development . . . 13

5.4.1 Task 1: Python scripts to parse HTML documentation files 13
5.4.2 Task 2: Developed Python scripts to convert PDF files into structured JSON data 13
5.5 Activity 3: Dataset Refinement for Ingestion 14
5.5.1 Task 1: Improved titles within our dataset for better ingestion 14
5.5.2 Task 2: Fixed Bugs Related to Special Characters in the Data 14
5.6 Activity 4: RAG Implementation and Chatbot Testing 15
5.6.1 Task 1: Generated embeddings and initialized a Retrieval Augmented Generation (RAG) search technique 15
5.6.2 Task 2: Tested Various Ideas/Use Cases on Our Chatbot 16
5.7 Activity 5: Chatbot Manual Testing UI Design 17
5.7.1 Task 1: Design a user interface for manual chatbot testing 17
5.8 Tools and Technologies: 17
5.8.1 GIT 17
5.8.2 Docker 18
5.8.3 Microsoft Azure 19
5.8.4 Atlassian Jira 20
5.8.5 Atlassian Bitbucket 21
5.8.6 Atlassian Confluence 21
5.8.7 Tiger VNC 21
5.9 Interfacess 22
5.9.1 CLI 22
5.9.2 REST 22

6 Conclusion **23**

7 References **24**

Chapter 1

Introduction

This chapter introduces the company and its operations, products, and services.

1.1 Company Introduction

Ciena Corporation[1] is an American telecommunications company recognized as a leading global provider of networking equipment, software, and services. Renowned for its expertise in optical connectivity, Ciena dominates the global market as the largest provider of network elements (NE), software, and services, primarily catering to the telecommunications industry.

The core activities of Ciena involve the development and marketing of equipment, software, and services, primarily targeted towards the telecommunications industry and major cloud service providers. Their comprehensive range of products and services supports the efficient transportation and management of voice. [2].

Ciena servers a diverse customer base that includes telecommunications services providers, cable operators, government agencies, enterprises, and data center operators. Ciena's Blue Planet software platform is the preferred choice for telecom companies like Bharti Airtel and Vodafone, enabling network programming and automation.

Ciena's research and innovation centers are focused on advancing optical networking, network automation, artificial intelligence, and machine learning capabilities to enable self-driving network and deliver a superior user experience.

1.2 Products and Services

This section provides a concise overview of the key products and services offered by **Ciena** [3].

1.2.1 Converged Packet Optical

The list of products of Ciena in the Converged Packet Optical[4] domain is as follows:

Figure 1.1: Converged Packet Optical

- **Waveserver**: A compact, stackable interconnect platform designed for high-efficiency Data Center Interconnect (DCI). It supports cloud-scale applications with flexible bandwidth scalability.
- **6500 Platform**: Integrates packet, Optical Transport Network (OTN), and photonics technologies into a single solution. It optimizes space, power, and network performance, and supports automated service provisioning via external software tools.
- **5400 Series**: A family of multi-terabit packet-optical switching systems that provide a cost-effective, scalable infrastructure capable of handling increasing traffic volumes and diverse service delivery requirements.

1.2.2 Routing and Switching

The list of the products of Ciena in the Routing and Switching[5] domain is as follows:

Figure 1.2: Routing and Switching platform

- **3000 Series**: Compact systems that support Passive Optical Networks (PONs), Ethernet, TDM, MPLS, IP, and virtualization, with interface speeds up to 10GbE.
- **5000 Series**: Designed for efficient traffic aggregation in metro and access layers. They support protocols such as Ethernet, IP, MPLS, PON, and Segment Routing.
- **6500 Packet Transport System**: Provides a seamless bridge between legacy TDM services and modern IP/Ethernet-based transport, allowing a gradual transition toward next-generation networks.

1.2.3 Network Control and Management Software

Ciena provides a suite of tools to manage and optimize network operations:

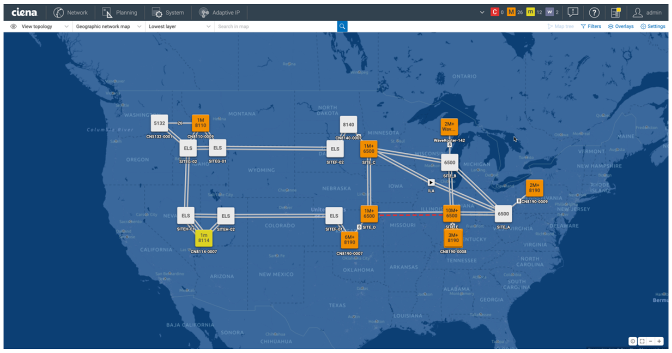


Figure 1.3: Navigator Network Control Suite

- **Service-Aware Operating System (SAOS)**: Embedded in both routing and optical platforms, offering robust Ethernet features and improved operational consistency.

- **Navigator NCS**: A domain controller that supports software-defined networking (SDN), streamlining network management and reducing manual intervention.
- **PlannerPlus**: A design and planning application for network build-outs and upgrades.
- **Emulation Cloud**: A virtual environment for developing, testing, and validating network functions and automation scripts.
- **Intelligent Control Plane**: Enables enhanced automation and network resilience by managing control protocols across the entire infrastructure.

1.2.4 Intelligent Automation Solutions

Ciena's intelligent automation[6] products include:

Figure 1.4: Blue Planet Intelligent Automation Suite

- **Blue Planet Route Optimization and Assurance**
- **Blue Planet vWAN**
- **Blue Planet Services**
- **Blue Planet Open Network Operating System**
- **Blue Planet NFV Orchestration**
- **Blue Planet Analytics**
- **Blue Planet Multi-domain Service Orchestration**

This ends the list of services and products offered by Ciena.

Chapter 2

About Team

2.1 Objective

The **Core AI Dev** team is responsible for the development of Ciena's upcoming product, MCP-Chatbot. This conversational chatbot uses artificial intelligence to assist users with queries about the Navigator Network Control Suite and to troubleshoot warnings related to the network.

The team's primary focus is on enhancing the artificial intelligence capabilities within the Navigator NCS software.

2.2 Team Introduction and Composition

I worked with Team **Core AI Dev** for the duration of my internship. Mr. Neeraj Shrivastava, Senior Manager of Software Engineering Systems, assigned the tasks.

If I have any doubts or needs, I always reach out to my buddy, Varun Aggarwal, and other team members for help.

The Core AI Development team was recently formed to design AI-powered solutions for the Navigator Network Control Suite. We are exploring to develope a user-friendly conversational chatbot, built on an agentic framework, to effectively assist customers and address their queries.

Chapter 3

About Project and Activities Performed

3.1 Objective

The objective of Ciena's network management system, Navigator NCS (Navigator Network Control Suite), is to manage Ciena's network elements (NEs).

Key objectives include:

1. Performing network element (NE) management tasks such as monitoring alarm conditions, backing up and restoring NE software, and upgrading NE software.
2. Managing multi-layer packet and optical services with a real-time view of network and service topology.
3. Providing an integrated planning facility to assist network designers in planning and deploying network hardware and service configurations.
4. Utilizing the Blue Planet micro-services architecture through the Navigator NCS domain controller to automate lifecycle operations of Ciena's packet and optical networks.
5. Supporting tasks including Network Element Enrollment, Network Element Discovery, Service Intent Creation, and Deployment.
6. Offering a high-level understanding of advanced functions such as the Path Computation Engine, Service Stitching, and alarm management.

The following sections discuss the activities and tasks undergone during the whole internship period, with contributions and learnings.

3.2 My Learnings and Contribution to the Project

3.2.1 Initial Hands-on Training Activity

To train the interns in basic company and team operations, several activities are conducted, including providing information about the internal structure of Navigator NCS, an overview of the chatbot framework, building images, and deploying them.

1. **Training Sessions and Videos:** Interns were assigned communication improvement sessions and training videos to help them understand how the organization works
2. **Introduction to Core AI Dev Team:** To help interns understand the team's operations and the types of tasks it primarily handles, an introductory session on the Gen AI project was conducted. The session included a description of the Gen AI product, covering the general workings of the MCP Chatbot.
3. **Hands-on Session:** It included exercises on API calling, code review of the RAG (Retrieval-Augmented Generation) technique, and various methods for handling LLM outputs. Additionally, building and deploying Docker images and deploying.
4. **Get knowledge into practice:** After gaining an understanding of the project, I was assigned various tasks, including writing Python scripts to enhance our dataset, conducting research, and developing proofs of concept (PoCs) using open-source tools such as TextGrad and DSPy.
5. **Lab Visit:** To give us more understanding of the working of the hardware devices, interns were provided a lab visit in Ciena India R&D Lab.

As an intern, I contributed to the improvement and optimization of our chatbot's accuracy. My responsibilities included enhancing existing Python and shell scripts and developing new automation scripts for data processing, often in collaboration with my teammates, to ensure we met industry standards.

The following list describes the overall activities performed during this period :

1. **Activity 1: Researched and Developed with Open-Source Tools like DSPy and TextGrad**
2. **Activity 2: Data Parsing Script Development**
   - Wrote Python scripts to parse HTML documentation files.

- Developed Python scripts to convert PDF files into structured JSON data.

3. **Activity 3: Dataset Refinement for Ingestion**
   - Improved titles within our dataset for better ingestion.
   - Fixed bugs related to special characters in the data.
4. **Activity 4: RAG Implementation and Chatbot Testing**
   - Generated embeddings and initialized a Retrieval Augmented Generation (RAG) search technique.
   - Conducted research and tested various ideas/use cases on our chatbot.
5. **Activity 5: Chatbot Manual Testing UI Design**
   - Designed a user interface for manual chatbot testing using Streamlit.

The above activities are primarily focused on developing Proofs of Concept (PoCs) to achieve minimum processing and high accuracy. During these activities, I used my knowledge to learn about technologies like Python, shell scripting, Rust, etc. Tech stacks like Agile-jira, docker, confluence, BitBucket and some interfaces. Open source tools like Dspy, Textgrad, StreamlitUi, Flashtext, etc.

Chapter 4

Software Development Methodology

This chapter describes the details of the software development methodology utilized, scrums team and sprint planning.

4.1 Agile Software Development Methodology

Agile[7] is a software development methodology that emphasizes adaptability and takes a pragmatic approach to delivering the final product. It requires a cultural change in organizations as it prioritizes the incremental delivery of individual software components rather than the entire application.

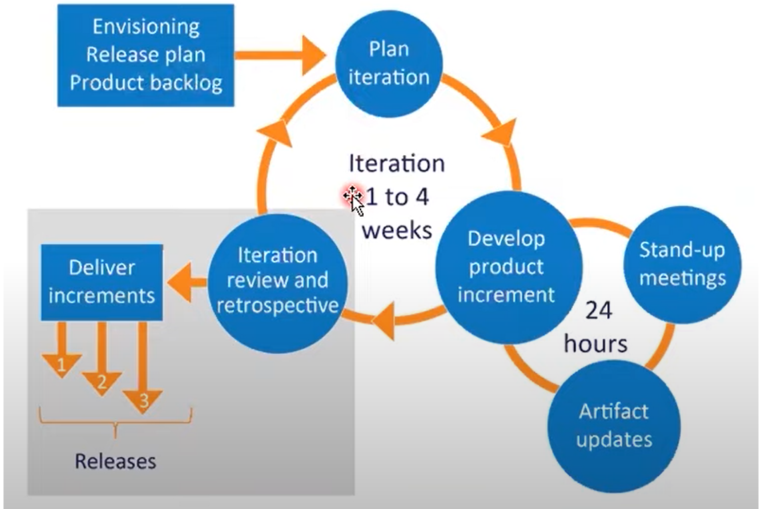


Figure 4.1: Agile Project Life Cycle

4.1.1 Agile Project Life Cycle

Figure 4.1 illustrates the project life cycle typically followed in Agile software development. Agile offers several benefits, including its ability to assist teams in navigating a changing landscape while focusing on efficiently delivering business value. The collaborative culture fostered by Agile also enhances overall organizational efficiency as teams work together and gain a clear understanding of their respective roles in the process.[8]

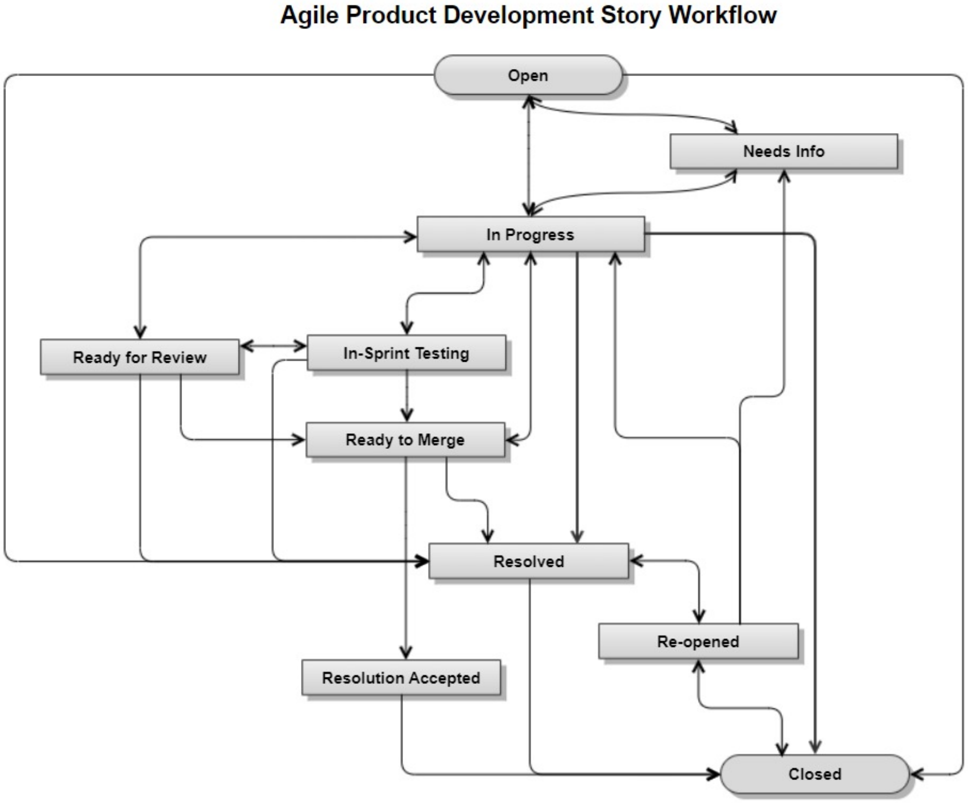


Figure 4.2: Agile Product Development Story Workflow

The different scrum teams have been allocated specific roles, and work is mostly done in compliance with Sprint planning, which is a timeboxed duration of 2 weeks to target specific tasks.

Chapter 5

Internship Report

This chapter briefly describes the requirement analysis, internship timeline, technologies, and tech stacks learned and utilized for the internship project. These concepts create a foundation for the tasks and activities done throughout the internship. The section discusses the approach used to tackle the described problem. This also discusses the requirement analysis of the situation, with my contribution to each task of the activity, learning, and tech stack used.

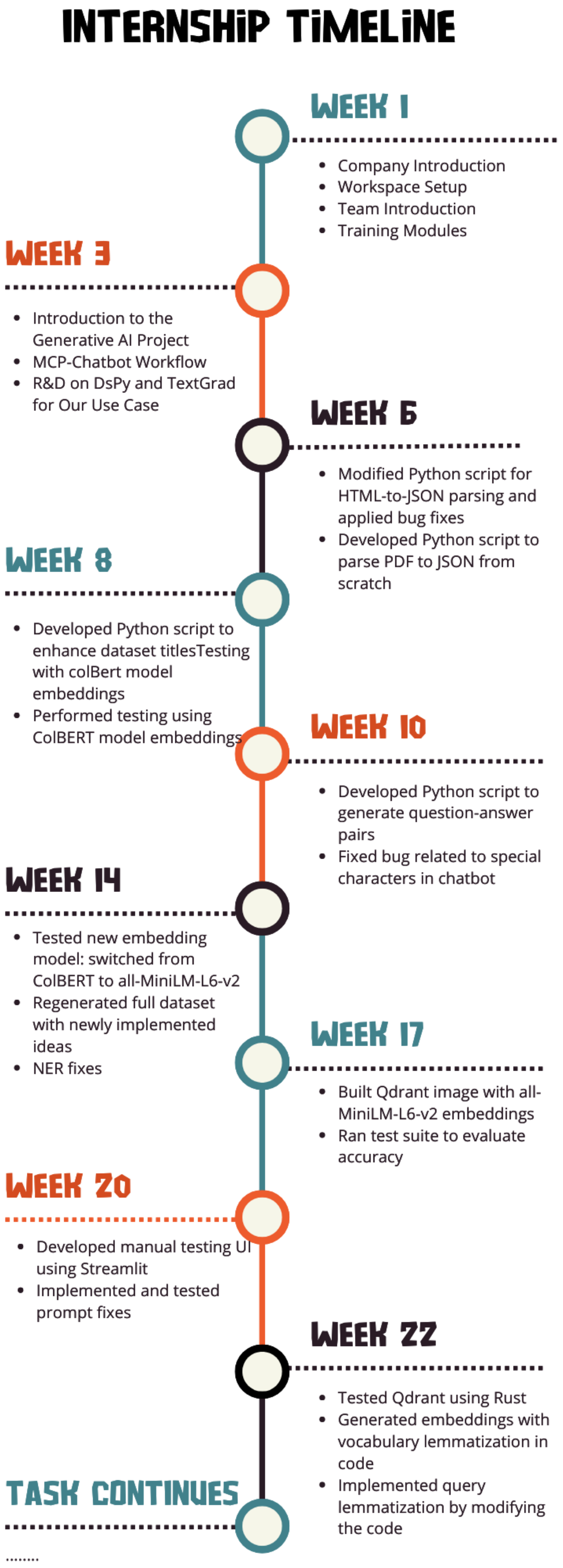
5.1 Requirement Analysis

During the internship, the tasks assigned required working knowledge of Python, Shell scripting, CLI interface, Rust, etc. Along with that, I was also able to learn about new technologies and tech stacks, which the task required.

The following sections describe the timeline along with the activities performed with the workaround and my learning.

5.2 Timeline

The diagram below shows the timeline of activities performed during the internship period, along with information about each week.



INTERNSHIP TIMELINE

WEEK 1

- Company Introduction
- Workspace Setup
- Team Introduction
- Training Modules

WEEK 3

- Introduction to the Generative AI Project
- MCP-Chatbot Workflow
- R&D on DsPy and TextGrad for Our Use Case

WEEK 6

- Modified Python script for HTML-to-JSON parsing and applied bug fixes
- Developed Python script to parse PDF to JSON from scratch

WEEK 8

- Developed Python script to enhance dataset titlesTesting with colBert model embeddings
- Performed testing using ColBERT model embeddings

WEEK 10

- Developed Python script to generate question-answer pairs
- Fixed bug related to special characters in chatbot

WEEK 14

- Tested new embedding model: switched from ColBERT to all-MiniLM-L6-v2
- Regenerated full dataset with newly implemented ideas
- NER fixes

WEEK 17

- Built Qdrant image with all-MiniLM-L6-v2 embeddings
- Ran test suite to evaluate accuracy

WEEK 20

- Developed manual testing UI using Streamlit
- Implemented and tested prompt fixes

WEEK 22

- Tested Qdrant using Rust
- Generated embeddings with vocabulary lemmatization in code
- Implemented query lemmatization by modifying the code

TASK CONTINUES

....

Figure 5.1: Internship-Timeline

5.3 Activity 1: Research and Development Using Open-Source Tools such as DSPy and TextGrad

Problem Description: This activity involved conducting a PoC using open-source frameworks like DSPy and TextGrad. The focus was on understanding the types of loss functions these libraries utilize.

Workaround: The task required a comprehensive study of the internal architecture and functionalities of DSPy and TextGrad. It included in-depth exploration of how these tools can be effectively integrated into chatbot development.

Key Learnings: I got to learn about the DSPy and TextGrad frameworks, their operational mechanisms, and their potential for automating prompt engineering in conversational AI systems.

5.4 Activity 2: Data Parsing Script Development

5.4.1 Task 1: Python scripts to parse HTML documentation files

1. **Problem Description:** The task involved parsing HTML documentation files of Navigator NCS to remove unwanted content while preserving the hierarchical structure and additional contextual information.
2. **Workaround:** I was provided with a Python script previously developed by a teammate. I modified and extended its functionality to implement a solution that effectively filtered irrelevant content while maintaining the hierarchy during parsing.
3. **My Learning:** Through this task, I gained hands-on experience in parsing HTML files and handling nested HTML tags. I learned how to convert structured HTML content into Markdown format and subsequently transform it into a structured JSON file, which is used to generate the final dataset.

5.4.2 Task 2: Developed Python scripts to convert PDF files into structured JSON data

1. **Problem Description:** The task was to convert documentation files in PDF format into a structured JSON format suitable for ingestion into our chatbot system.
2. **Workaround:** I researched multiple models and tools capable of converting PDF files into Markdown. After conducting a proof of concept (PoC), I selected the open-source tool `pymupdf4llm` for its accuracy and ease of use. I then developed a custom Python script from

scratch to parse the generated Markdown files and transform them into the desired JSON format.

3. **My Learning:** I got exposure to various tools for processing PDF documents and learned how to effectively parse and structure Markdown content.

5.5 Activity 3: Dataset Refinement for Ingestion

5.5.1 Task 1: Improved titles within our dataset for better ingestion

1. **Problem Description:** Since some of our content is too large to generate embeddings directly, the task was to summarize the content by generating concise titles.
2. **Workaround:** I developed a Python script that takes the main content along with the preserved additional context and makes LLM API calls (using either OpenAI or Gemini) with a carefully instructed prompt. This served as a preprocessing step in our data pipeline.
3. **My Learning:** During this task, I learned about `Dask`, a Python library for parallel computing, which helped optimize the processing pipeline. I also learned about making API calls to different LLM endpoints and handling their responses effectively.

5.5.2 Task 2: Fixed Bugs Related to Special Characters in the Data

1. **Problem Description:** As part of our data preprocessing, we were removing special characters from the content. However, this introduced a bug in the chatbot UI, leading to the display of misleading or incomplete information.
2. **Workaround:** To address this issue, I modified the ingestion pipeline to map the ingested (cleaned) content back to the corresponding raw content, where no special character removal was applied. This ensured accurate context display in the chatbot.
3. **My Learning:** I gained hands-on experience with **QdrantDB**, specifically in maintaining a mapping between raw and processed data within our vector database for better traceability and debugging.

5.6 Activity 4: RAG Implementation and Chatbot Testing

5.6.1 Task 1: Generated embeddings and initialized a Retrieval Augmented Generation (RAG) search technique

1. **Problem Description:** The task was to generate embeddings from documentation content and store them in a vector database, enabling efficient data retrieval through similarity search algorithms.
2. **Workaround:** Initially, we used the ColBERT model for generating embeddings. However, since it required GPU resources, it was not cost-effective for production use. As a result, we switched to the lighter and more efficient `all-MiniLM-L6-v2` model for embedding generation, and used **Qdrant** as our vector database for storage and retrieval.

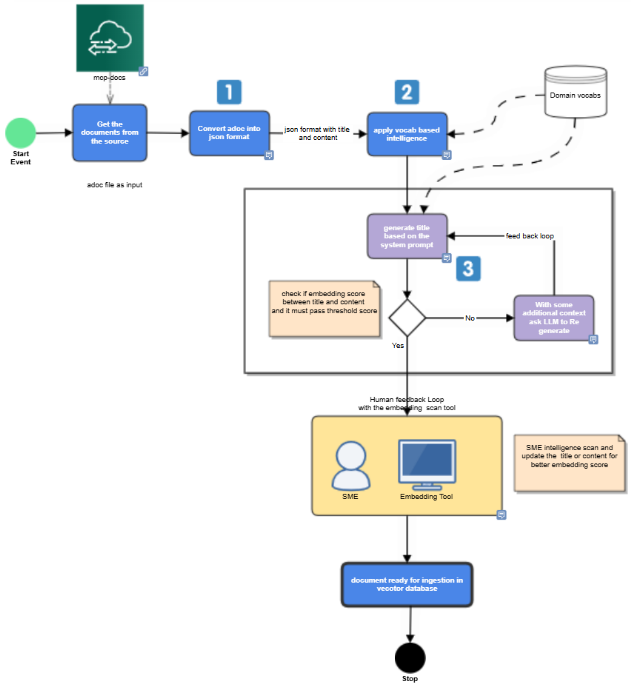

mcp-docs
1
2
Domain vocabs
Start Event
Get the documents from the source
Convert adoc into json format
json format with title and content
apply vocab based intelligence
adoc file as input
generate title based on the system prompt
feed back loop
3
check if embedding score between title and content and it must pass threshold score
No
With some additional context ask LLM to Re generate
Yes
Human feedback Loop with the embedding scan tool
SME
Embedding Tool
SME intelligence scan and update the title or content for better embedding score
document ready for ingestion in vecotor database
Stop

Figure 5.2: Document Ingestion Process

For similarity search, we adopted **cosine similarity** as our core algorithm to find the most relevant results based on vector distance.

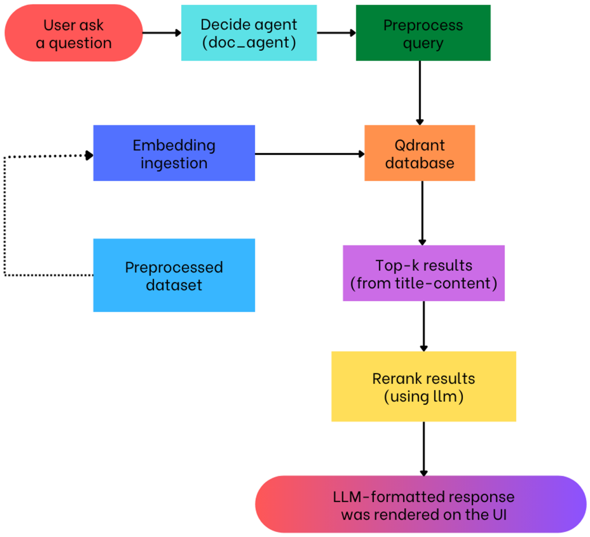


Figure 5.3: RAG Workflow

To further optimize performance and reduce latency, we transitioned the implementation from Python to Rust, a systems programming language known for speed and memory safety. Research and development on this task are ongoing.

3. **My Learning:** Through this task, I learned about Retrieval-Augmented Generation (RAG), the `all-MiniLM-L6-v2` embedding model, cosine similarity for vector search, the use of **Qdrant** for vector storage, and gained practical experience with the Rust programming language.

5.6.2 Task 2: Tested Various Ideas/Use Cases on Our Chatbot

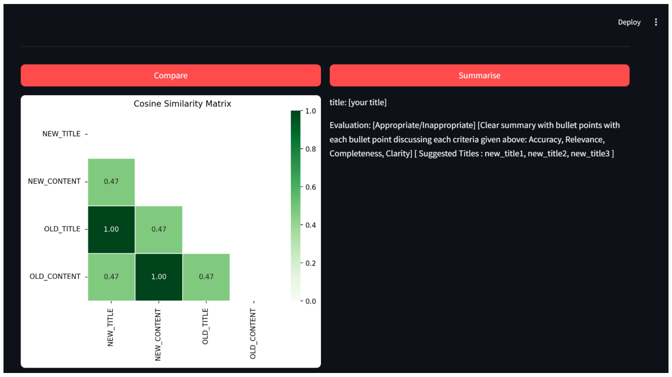
1. **Problem Description:** The task was to generate test bed to evaluate the accuracy and performance of our chatbot across various use cases.
2. **Workaround:** To generate the test bed, I developed a Python script that takes input content along with a well-instructed prompt to generate question–answer pairs. We then integrated this with a **TeamCity Automation** to automate the testing process. As of now, our chatbot achieves approximately **80% accuracy** on these test beds.
3. **My Learning:** Through this task, I learned how to implement automated testing pipelines using **TeamCity**.

5.7 Activity 5: Chatbot Manual Testing UI Design

5.7.1 Task 1: Design a user interface for manual chatbot testing

1. **Problem Description:** In this task, we wanted to analyze failed test cases where the chatbot's response did not match the expected answer. The objective was to understand why relevant content was not being retrieved during vector search.
2. **Workaround:** Since we were using **cosine similarity** as our search algorithm, I proposed the idea of analyzing embedding similarity in failed cases. By inspecting why certain content had low cosine similarity scores and was not retrieved, we could identify missing context and improve data coverage.

   To implement this solution, I designed a small application that visually displays similarity scores between content and the query. At runtime, the LLM also suggests additional context that could improve retrieval results.

	NEW_TITLE	NEW_CONTENT	OLD_TITLE	OLD_CONTENT
NEW_TITLE				
NEW_CONTENT	0.47			
OLD_TITLE	1.00	0.47		
OLD_CONTENT	0.47	1.00	0.47	

   title: [your title]

   Evaluation: [Appropriate/Inappropriate] [Clear summary with bullet points with each bullet point discussing each criteria given above: Accuracy, Relevance, Completeness, Clarity] [Suggested Titles : new_title1, new_title2, new_title3]

   Figure 5.4: Testing UI

   The application was built using **Streamlit**, a Python library that allows quick development of interactive web UIs for data applications.
3. **My Learning:** I explored the **Streamlit** library and learned how to build simple interactive UIs.

5.8 Tools and Technologies:

5.8.1 GIT

Git[9] is a free, open-source distributed version control system known for its ability to efficiently manage projects of all sizes. Its lightweight design and high-speed performance make it easy to

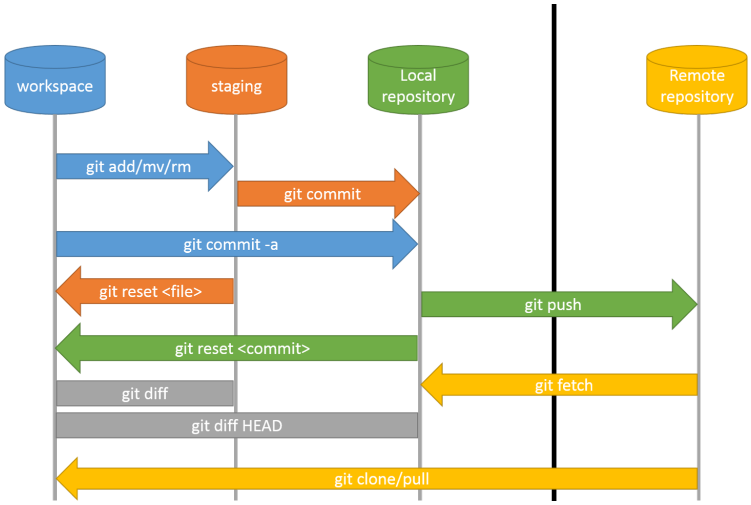


Figure 5.5: Git Workflow

learn and use.

Git surpasses traditional SCM tools like Subversion, CVS, Perforce, and ClearCase with features such as affordable local branching, convenient staging areas, and flexible workflows.

One of Git's standout features is its branching model, which supports multiple independent local branches. Creating, merging, and deleting branches are quick operations, taking only seconds.

Git's data model ensures the cryptographic integrity of your project by checksumming every file and commit, ensuring that only the exact data you put in can be retrieved.

Git's *staging area* or *index* allows commits to be formatted and reviewed before finalizing. This feature lets you stage and commit specific files without affecting other modified files in your working directory or having to list them all during the commit process.

5.8.2 Docker

Docker[10] is an open platform designed for developing, shipping, and running applications. It enables you to separate applications from your infrastructure, facilitating quicker software delivery. By leveraging Docker's methodologies, you can significantly reduce the time between writing code and running it in production.

Docker packages applications in *containers*, which are lightweight, isolated environments containing everything needed to run an application. This ensures consistent behavior across different environments, as containers do not depend on the host's current setup.

Docker's client-server architecture involves the Docker client communicating with the Docker daemon, which handles building, running, and distributing containers.

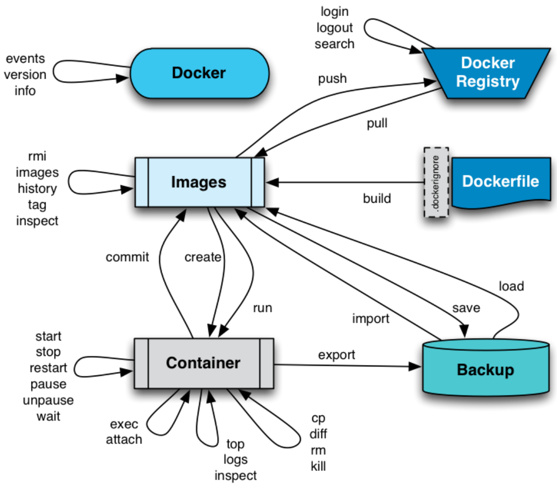


Figure 5.6: Docker Commands

Docker enhances the development lifecycle by enabling developers to work in standardized local container environments, making it ideal for continuous integration and continuous delivery (CI/CD) workflows. This standardization allows for easy sharing of containers, ensuring consistent application performance across different stages of development and deployment.

5.8.3 Microsoft Azure

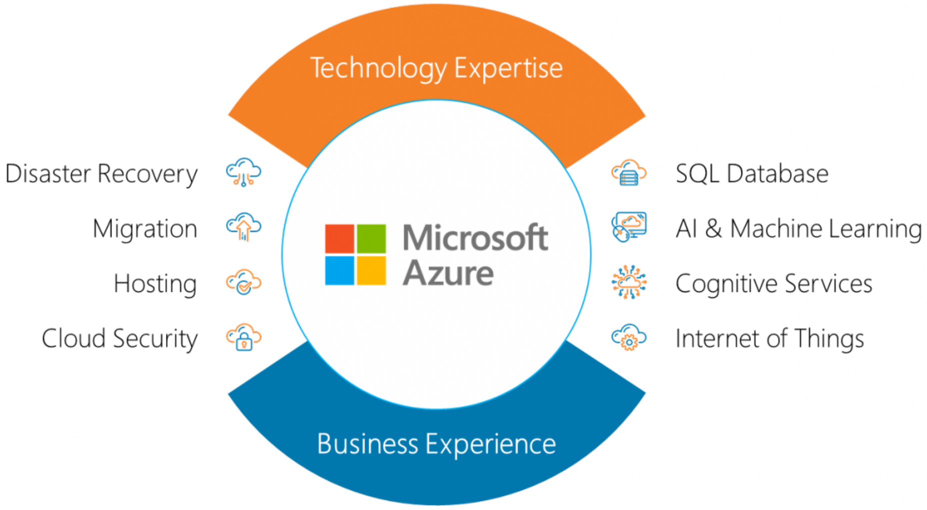


Figure 5.7: Microsoft Azure

Microsoft Azure[11] is a cloud computing platform and service developed by Microsoft, offering a wide array of cloud-based services, including computing, analytics, storage, networking, and more.

It allows businesses and developers to build, deploy, and manage applications through a global network of data centers.

Azure supports various programming languages, frameworks, and tools, which makes it versatile for different development and operational needs. Key features of Azure include virtual machines, databases, storage, artificial intelligence (AI) and machine learning (ML) services, and Internet of Things (IoT) solutions.

Azure ensures high availability, scalability, and security, making it a critical component for modern enterprises seeking to enhance their IT infrastructure.

5.8.4 Atlassian Jira

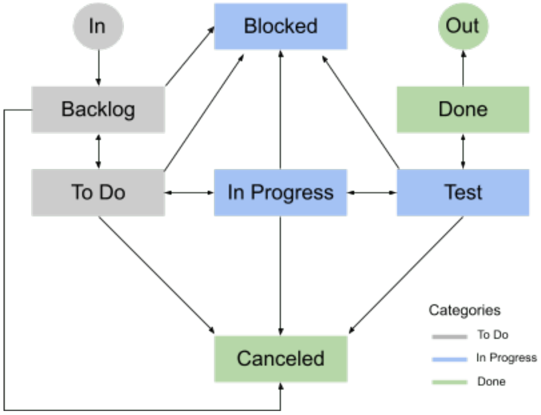


Figure 5.8: Jira Workflow

Jira Software[12], developed by Atlassian, is a project management tool tailored for software development teams. It is designed to enhance the planning, tracking, and management of projects. A standout feature of Jira Software is its issue tracking system, which enables teams to create, monitor, and manage tasks, bugs, user stories, and other project-related issues in a centralized and organized fashion. Each issue can be assigned to specific team members, prioritized, and tracked throughout its lifecycle, ensuring transparency and accountability.

Jira Software also supports agile methodologies like Scrum, offering customizable boards to help teams visualize and manage their workflow.

Additionally, it integrates with various development tools, version control systems, and CI/CD pipelines, facilitating a seamless connection between development and release processes.

Figure 5.9: Atlassian Bitbucket

5.8.5 Atlassian Bitbucket

Bitbucket[13] is an online platform designed for version control and collaboration in software development projects.
It mainly hosts and manages Git repositories, providing features like code versioning, branching, merging, and pull requests to support effective teamwork and code review.
Additionally, Bitbucket integrates with various development tools, such as issue tracking systems and CI/CD pipelines, to enhance the software development process. Developers can use Bitbucket to create both private and public repositories, set access permissions for team members, and manage multiple branches for concurrent development.

5.8.6 Atlassian Confluence

Figure 5.10: Atlassian Confluence

Confluence[14], developed by Atlassian, is widely used for collaboration and documentation. It is intended to enhance team collaboration, knowledge sharing, and project management within organizations.
Confluence offers a centralized platform where teams can create, organize, and work together on a variety of content types, such as documents, meeting notes, project plans, product requirements, and more.

5.8.7 Tiger VNC

TigerVNC[15] is an open-source software that allows users to remotely access and control graphical desktops over a network connection. It operates on the Virtual Network Computing (VNC)

protocol, enabling remote display and input functionalities.
TigerVNC utilizes a client-server architecture, with the VNC server running on the remote machine and the VNC client used to connect to and manage the remote desktop.

5.9 Interfacess

5.9.1 CLI

A command-line interface[16] (CLI) is a text-based user interface (UI) that allows users to execute programs, manipulate computer files, and interact with the computer using text commands. CLIs are also called command-line user interfaces, console user interfaces, or character user interfaces. Users enter commands via the keyboard, and the computer executes the commands entered at the command prompt.

CLI-based shells provide users with a streamlined and efficient way to interact with the operating system, eliminating the need for a graphical user interface and its associated overhead.

5.9.2 REST

A RESTful[17] API, also known as a REST API, is a web service that adheres to the principles of Representational State Transfer (REST) architecture. REST is a set of constraints applied to web services, enabling the creation of scalable and flexible services. One of the essential characteristics of a REST API is its statelessness. This means that the server does not retain any client state information and relies solely on the information provided within each request. This stateless nature enhances scalability and flexibility since the server is not burdened with managing client state information.

Chapter 6

Conclusion

This chapter concludes the report by highlighting the key learnings and experiences gained during my internship. It has the various tasks and activities I undertook throughout the internship.
The tasks assigned to me provided valuable exposure to a range of tools and technologies, including Docker, shell scripting, Git, and many more. The primary focus of my internship was to develop a processing-optimized and accurate chatbot.

Throughout the internship, I gained insights into writing professional code and documentation, understanding their significance in a collaborative environment.
One of the most important lessons I learned was the value of identifying the root cause of errors/problem rather than simply solving them.
I also explored several open-source tools such as Qdrant, ColBERT, and LangChain, and enhanced my skills in designing REST APIs and integrating services within existing infrastructures.

Overall, this experience significantly improved my technical expertise and deepened my understanding of building scalable, user-friendly solutions in complex network environments.
Additionally, I became proficient with several industry-standard tools like Agile-Jira, Docker, Confluence, Bitbucket, and OpenGrok. From a soft skills perspective, I gained experience in team collaboration and time management.

In conclusion, my internship experience at **Ciena** has been extremely rewarding and encouraging. It marked my first exposure to a professional environment, and the learning curve I experienced as an intern was fruitful.

Chapter 7

References

[1] Ciena. *Company*. 2025. URL: https://www.ciena.com.

[2] Ciena. *We are Ciena*. 2025. URL: https://www.ciena.com/about.

[3] Ciena. *Products*. 2025. URL: https://www.ciena.com/products.

[4] Ciena. *Converged Packet Opticals Products*. 2025. URL: https://www.ciena.com/products/converged-packet-optical.

[5] Ciena. *Routing and Switching Products*. 2025. URL: https://www.ciena.com/products/routing-switching.

[6] Ciena. *Intelligent Automation Products*. 2025. URL: https://www.blueplanet.com/.

[7] TechTarget. *Agile software development*. 2025. URL: https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development.

[8] Agileful. *Agile product development lifecycle*. 2025. URL: https://agileful.com/the-agile-product-development-lifecycle-steps-and-best-practices/.

[9] Wikipedia contributors. *Git — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Git. 2025.

[10] official-site. *Docker*. 2025. URL: https://www.docker.com/.

[11] Wikipedia contributors. *Microsoft Azure (software) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Microsoft_Azure. 2025.

[12] Wikipedia contributors. *Jira (software) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Jira_(software)&oldid=1156290651. 2025.

[13] Wikipedia contributors. *Bitbucket — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Bitbucket&oldid=1138440928. 2025.

[14] Wikipedia contributors. *Confluence (software) — Wikipedia, The Free Encyclopedia.* https://en.wikipedia.org/w/index.php?title=Confluence_(software)&oldid=1157376319. 2025.

[15] Tiger VNC. *About.* 2025. URL: https://tigervnc.org/.

[16] Wikipedia contributors. *Command-line interface — Wikipedia, The Free Encyclopedia.* https://en.wikipedia.org/wiki/Command-line_interface. 2025.

[17] IBM. *What is rest apis?* 2025. URL: https://www.ibm.com/think/topics/rest-apis.