IE 7374

Machine Learning Operations

Group 18

Project Scoping – "AskNEU"

Ajin Frank Justin, Khushboo Galrani, Naga Kushal Ageeru, Pooja Kannan, Sarvesh Selvam & Yogita Bisht

1. Introduction

AskNEU is a conversational Retrieval-Augmented Generation (RAG) system designed to transform how users interact with Northeastern University's vast repository of information. AskNEU delivers accurate, context-aware answers to user queries in real time by integrating advanced AI language models with targeted data retrieval techniques. Whether you're a prospective student exploring academic programs, a current student navigating campus resources, or a faculty member seeking policy details, AskNEU serves as your intelligent assistant.

Traditional information retrieval systems often require users to sift through multiple web pages or documents to find specific answers. AskNEU eliminates this by combining the precision of search engines with the natural language understanding of GPT-based models. When a user asks a question, the system first retrieves the most relevant data from Northeastern's official website and then generates a concise, conversational response tailored to the command. This approach ensures that users receive accurate real-time information.

The project is driven by the goal of providing accurate information for the Northeastern community. AskNEU bridges the gap between complex institutional data and user-friendly communication. Our vision is to create a system that not only answers questions but also anticipates user needs, providing proactive suggestions and insights. This project represents a significant step forward in making institutional knowledge more accessible, intuitive, and engaging for everyone connected to Northeastern University.

2. Dataset Information

Dataset Introduction

The dataset being used for this project is publicly accessible information sourced from Northeastern University's websites.

Data Card

Model Card for ChatGPT Turbo

- Model: GPT-3.5 Turbo
- Type: A text-based AI model specializing in natural language understanding and generation.
- Base: Built on OpenAI's advanced language model architecture.
- Size: Designed for high efficiency, offering reduced latency compared to GPT-4, while maintaining robust performance across various tasks.
- Specialization: Optimized for conversational AI, general-purpose tasks, programming, content creation, and extended context comprehension.
- Performance: Faster and more cost-effective than GPT-4, with strong capabilities in generating concise responses and handling tasks of moderate complexity.
- License: Accessible through OpenAI's API, ChatGPT Pro subscription.
- URL: https://openai.com/chatgpt

Model Card for DeepSeek-V3

1. Model: DeepSeek-V3
2. Type: A multimodal AI model specializing in natural language understanding, generation, and advanced reasoning across text, code, and visual data.
3. Base: Built on DeepSeek's cutting-edge proprietary architecture.
4. Size: Optimized for a balance of power and efficiency, delivering reduced latency while handling high-complexity tasks.
5. Specialization: Tailored for advanced reasoning, complex problem-solving, programming, content creation, scientific research, and multimodal integration (text, code, and visuals).
6. Performance: Faster and more cost-effective than many larger models.
7. License: Accessible through DeepSeek's API or enterprise licensing agreements.
8. URL: https://www.deepseek.com

Data Sources

The dataset was compiled exclusively from publicly available Northeastern University resources, ensuring its relevance and accuracy for the intended applications. Key sources include:

1. Office of Global Services (OGS): Information related to international student support and visa processing.
2. Student Financial Services (SFS): Details about tuition, fees, scholarships, and financial aid options.
3. NEU Marino: Data about campus facilities and services.
4. Student Catalogue: Academic program information, course descriptions, and degree requirements.
5. Admissions: Comprehensive details for undergraduate, graduate, and PhD admissions processes.
6. University Health and Counselling Services (UHCS): Information about health services, mental health support, and wellness resources.
7. University Website Content: Pages from the official Northeastern University website, including academic catalogues, student handbooks, and faculty directories.
8. Frequently Asked Questions (FAQs): Curated from various official pages to provide direct answers to common inquiries.

This dataset ensures coverage of multiple domains critical for the university's community and stakeholders, without any inclusion of proprietary or restricted data.

Data Rights and Privacy

The dataset adheres to strict compliance with data protection regulations, including the General Data Protection Regulation (GDPR) and other applicable privacy laws. Only publicly accessible information was collected and utilized, ensuring no personal, confidential, or restricted data is included. Data handling was conducted responsibly to maintain transparency and respect the rights of individuals and organizations. This project complies with ethical data usage guidelines to ensure both legal and moral integrity in all applications derived from the dataset.

3. Data Planning and Splits

Our plan is to systematically organize all Northeastern University (NEU) websites and scrape their webpages. The scraped pages will undergo thorough content extraction and preprocessing to clean and filter useful information. This processed data will then be stored securely in a centralized storage system.

Subsequently, the stored data will be segmented into smaller, meaningful chunks based on a suitable strategy. These chunks will be utilized to generate embeddings using a pre-trained language foundation model. The embeddings will be indexed and stored in a vector database, facilitating downstream tasks in the pipeline.

This process encompasses the critical stages of data collection, preprocessing, and storage, ensuring a robust foundation for subsequent analytical and operational workflows.

4. GitHub Repository

GitHub Repository Link: https://github.com/justin-aj/AskNEU

The repository will include all pipeline and code versions controlled via Git, with a README file detailing project installation, usage, and a structured folder organization reflecting different aspects of the project (data ingestion, model development, deployment).

5. Project Scope

Problems:

- Traditional search methods often return generic results or require users to navigate through multiple pages, leading to frustration and wasted time.
- Prospective students, international applicants, or individuals unfamiliar with the university's structure may struggle to locate relevant information due to complex website navigation.

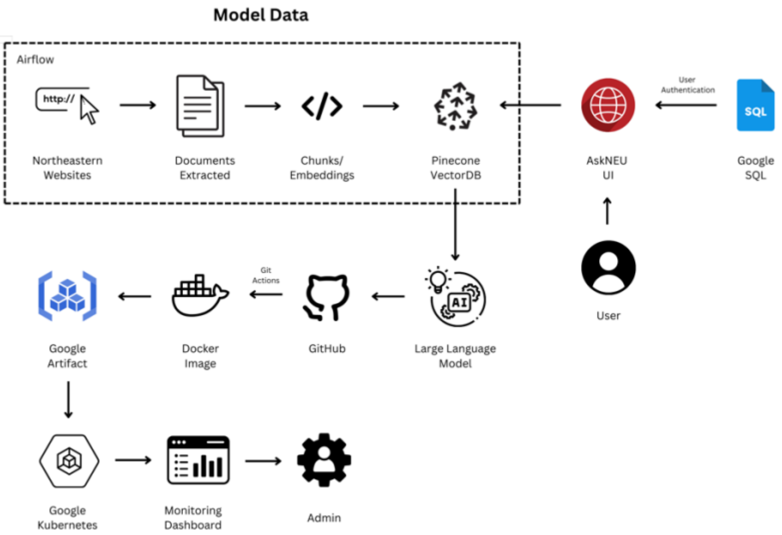
Current Solutions:

- Search functionality, but results are often overwhelming or irrelevant, requiring manual filtering.
- University staff and advisors are available to answer questions, but this approach is time intensive as it requires appointments to interact with them.

<u>Proposed Solutions</u>:

- By combining advanced data retrieval techniques with GPT-based language models, AskNEU can pull precise information from Northeastern's website and generate accurate, context-aware responses in real time.
- As an AI-powered tool, AskNEU can handle a high volume of queries simultaneously, making it a scalable solution for the entire Northeastern community.

6. Current Approach Flow Chart and Bottleneck Detection



7. Metrics, Objectives, and Business Goals

<u>Evaluation Metrics</u>:

- **Relevance**: Ensure responses directly address user queries and remain on-topic.
- **Accuracy**: Verify the correctness of the information, ensuring alignment with ground truth data.
- **Completeness**: Assess whether responses cover all critical aspects of the query without omissions.
- **Clarity**: Use clear and effective language for responses that are easy to understand.
- **Conciseness**: Avoid unnecessary details and overly verbose explanations in the generated output.

Objectives:

- Deliver responses that are accurate, relevant, and comprehensive to meet user needs effectively.
- Maintain clarity and conciseness to enhance the usability of the chatbot.
- Improve user engagement by ensuring a user-friendly and reliable conversational experience.

Business Goals:

- Enhance user satisfaction by providing accurate, on-topic, and easy-to-understand responses.
- Streamline operations by automating answers to frequent queries, reducing staff workload.
- Support scalability and adaptability, allowing the system to cater to growing user demands and evolving datasets.

8. Failure Analysis

Potential Risks:

- Data Scraping Limitations: Not all pages may be scrapped, leading to incomplete coverage of queries.
- Dependency on Third-Party APIs: Reliance on Pinecone and GPT API increases vulnerability to their failures.
- Concept/Data Drift: Changes in data patterns or distributions may degrade model performance over time.

Mitigation Strategies:

- Fallback to Open-Source Platforms: Transition to cost-effective, open-source platforms if costs become prohibitive.
- Cost Optimization: Regularly review and optimize cloud resource usage, such as scaling down resources during off-peak times.
- Budget Monitoring and Alerts: Set up real-time budget tracking to avoid overspending.
- Addressing Concept/Data Drift: Implement continuous monitoring and retraining to adapt to new data patterns.

9. Deployment Infrastructure

Supported Platforms

- **Google Cloud Platform (GCP):**
    - GKE (Google Kubernetes Engine): To orchestrate containerized services.
    - Cloud Run: For lightweight, serverless API endpoints.
    - Cloud Storage: For data and artifact storage.
    - BigQuery: For analytics and reporting.
    - Vertex AI: For model training, tuning, and deployment.

Infrastructure Details

Core Components:

1. GKE Cluster:
    - Node pools: Separate for model training, data processing, and monitoring.
    - Autoscaling: To handle variable traffic loads.
2. Cloud Storage:
    - Separate buckets for raw data, processed data, and artifacts.
3. Networking:
    - VPC with firewalls and private subnets.
    - Load Balancer for external API traffic.

Monitoring and Logging:

- Prometheus and Grafana for application metrics.
- Google Operations Suite for centralized logging and error tracking.

Automation Tools:

- Terraform: To define infrastructure as code.
- GitHub Actions: To automate CI/CD workflows.

10. Monitoring Plan

What to Monitor

1. **Application Performance**:
    - API latency and throughput.
    - Error rates and request retries.
2. **Infrastructure Metrics**:
    - GKE node health (CPU, memory, and disk usage).
    - Pod and container statuses.
3. **Pipeline Metrics**:
    - Training time and resource utilization.

    - Data ingestion rates and job failures.
4. **Model Performance**:
    - Prediction latency.
    - Model accuracy (e.g., via periodic evaluation).
5. **Security Metrics**:
    - Unauthorized access attempts.
    - Network traffic anomalies.

Why Monitor These Metrics

- To ensure system reliability, scalability, and security.
- To quickly detect and resolve pipeline or deployment failures.
- To track model performance and trigger retraining as needed.

Tools

- **Prometheus**:
    - Collects and stores metrics from Kubernetes and application services.
- **Grafana**:
    - Visualizes metrics in dashboards and sets alerts for threshold breaches.
- **Google Operations Suite**:
    - Provides centralized logging and traces for debugging.

11. Success and Acceptance Criteria

- The chatbot must be accessible through a user-friendly front-end interface, enabling users to easily submit queries and receive responses in real time.
- Performance will be evaluated using metrics such as ROUGE scores (ROUGE-1, ROUGE-2, ROUGE-L), measuring the similarity between generated responses and ground truth answers.
- The system should adapt dynamically to new data inputs, improving its responses over time as more information becomes available.
- Mechanisms should be implemented to handle edge cases, ambiguous queries, and out-of-scope questions, ensuring the chatbot remains reliable and user-centric.
- Continuous performance monitoring must be in place to ensure the system consistently meets predefined metrics for accuracy, timeliness, and user satisfaction.

12. Timeline Planning

Week 1: Planning and Requirements

Define project scope and objectives
Create a simplified project brief
Outline the pipeline for Ask NEU
Assign roles to team members
Define datasets and ML models to be used

Week 2: Database and Backend Setup

Scrape data from NEU Websites
Implement data pipeline
Develop APIs for data retrieval and storage
Design a basic UI for the app

Week 3: Model Integration

Set up the RAG pipeline architecture
Integrate a pre-trained large language model

Week 4: Monitoring and Evaluation

Implement logging and monitoring systems for the AI pipeline
Develop evaluation metrics for model performance
Create dashboards for visualizing model accuracy and system health
Conduct basic unit testing

Week 5: Cloud Deployment

Start generating docker images and uploading on cloud
Start implementing the pipeline

Week 6: Frontend Development and Integration

Develop a basic user interface for user data entry
Integrate relational database to store username
Integrate the frontend with the backend APIs

Week 7: Testing and Refinement

Perform integration testing of all components
Conduct end-to-end testing of the entire system
User testing with sample queries
Bug fixes and minor improvements

Week 8: Documentation and Presentation Preparation

Prepare project documentation
Create a presentation showcasing the app's functionality
Conduct a final round of testing