

Automated Email Generation System for Job Seekers and Clients

Table of Contents

1. Introduction
 2. Objective
 3. Approach
 - 1. Creation of Case Studies and Data Storage in Neo4j
 - 2. Input Data Collection
 - 3. Intent Detection
 - 4. User Interest Identification
 - 5. Project Similarity Matching
 - 6. Personalized Email Generation
 - 7. Job Seeker Email Customization
 - 8. FastAPI Integration
 - 9. Dockerization
 4. Conclusion
-

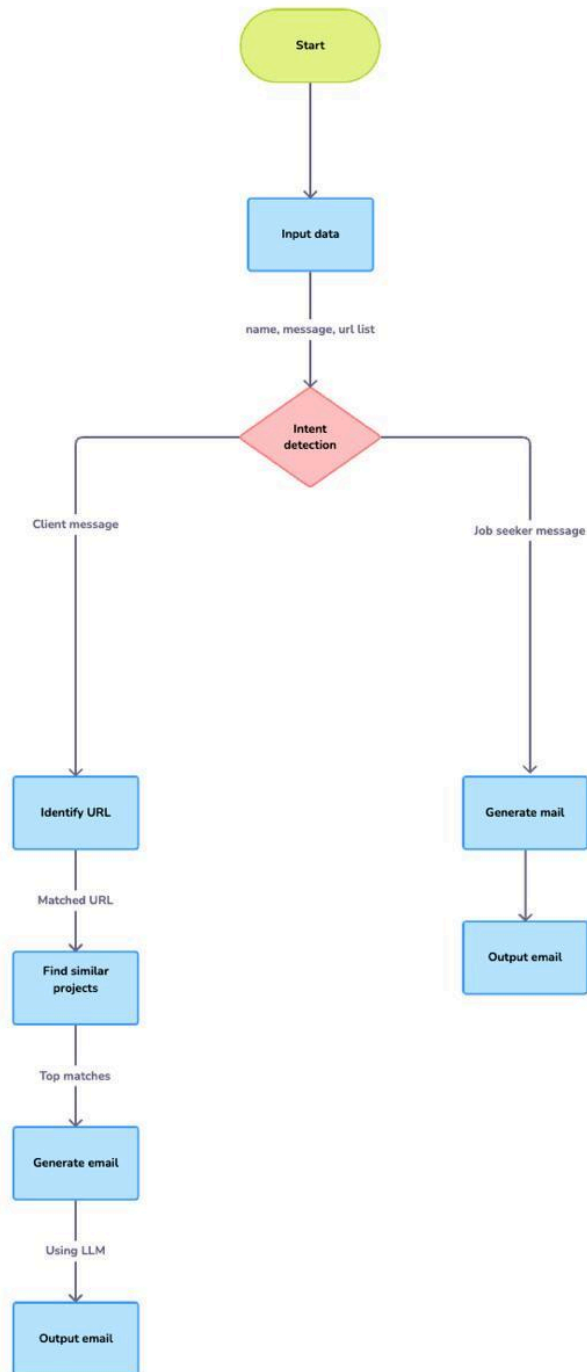
Introduction

In today's digital landscape, personalized communication is key to engaging potential clients and job seekers effectively. This project focuses on developing an automated email generation system that tailors responses based on user interactions, enhancing engagement and providing customized communication.

Objective

The primary goal is to create an automated system that generates personalized emails for job seekers and potential clients. By leveraging user data—such as message content, contact information, and browsing behavior—the system aims to assess user interests and deliver tailored email responses automatically.

Flow chart



Approach

To achieve the objective, the project follows a structured approach comprising several key steps:

1. Creation of Case Studies and Data Storage in Neo4j

Purpose: Optimize the identification of user interests and project relevance based on visited URLs.

Implementation:

- **Data Preparation:** Developed a set of case studies for different projects, each associated with specific URLs.
- **Database Storage:** Stored each URL-based project and its details—including client information, project title, challenges, solutions, results, tech stack, and URL sections—in a Neo4j graph database.
- **Embeddings:** Pre-generated embeddings for the projects and stored them in Neo4j to streamline the similarity matching process.

Advantage: Pre-storing embeddings reduces computational overhead by eliminating the need to generate embeddings for each project during user message processing.

2. Input Data Collection

Inputs:

- **User Name:** The name of the user.
- **User Message:** The content of the message received from the user.
- **List of URLs:** URLs visited by the user on the website.

Purpose: This minimal set of inputs ensures an efficient email generation process while allowing for a high degree of personalization.

3. Intent Detection

Objective: Differentiate between messages from potential clients and job seekers to customize the email format and approach accordingly.

Implementation:

- **Classification Model:** Employed a machine learning model to evaluate the user's message and determine the intent—whether it originates from a client or a job seeker.

Reasoning: Customizing the email content based on user intent ensures relevance and context-specific responses.

4. User Interest Identification

Process:

- **Interest Mapping:** Compared the user message with the list of visited URLs to identify topics or projects of interest.

Rationale: Understanding the user's interests helps tailor the email content to match their specific needs or inquiries, enhancing engagement and response quality.

5. Project Similarity Matching

Objective: Find the most relevant projects that align with the user's message and interests.

Implementation:

- **Embedding Retrieval:** Retrieved pre-stored embeddings from the Neo4j database for projects linked to the user's visited URLs.
- **Similarity Calculation:** Calculated the similarity between the user message and project embeddings.
- **Selection:** Identified the top three most similar projects for inclusion in the email.

6. Personalized Email Generation

Data Utilized:

- User Name
- User Message
- Top 3 Similar Projects
- Interest URLs
- User Intent

Method:

- **Language Model:** Used the open-source Llama 3 model to generate a customized email response based on the compiled data points.

Advantage: Ensures the generated email is contextually relevant and personalized according to the user's interests and message content.

7. Job Seeker Email Customization

Fixed Format for Job Seekers:

- **Template Usage:** Employed a predefined email template for job seekers to standardize the response format.
- **Customization:** Included placeholders for the user's name, visited URLs, and message content.

Reasoning:

- **Consistency:** A fixed format maintains a professional tone and ensures consistency in responses to job seekers.
- **Content Management:** Avoided scraping detailed information from visited URLs to prevent including excessive or irrelevant information.
- **Compliance:** Adhered to data access policies by not scraping content from URLs due to restrictions associated with open-source models.

8. FastAPI Integration

Implementation:

- **API Development:** Created a FastAPI endpoint to handle email generation requests.
- **Inputs:** Accepts User Name, User Message, and List of URLs.
- **Output:** Returns a personalized email response.

Purpose: The API serves as an interface for the automated email generation system, allowing seamless integration with web applications or Customer Relationship Management (CRM) systems.

9. Dockerization

Objective: Simplify the deployment process and ensure consistency across different environments.

Implementation:

- **Dockerfile Creation:** Developed a Dockerfile to automate the setup of the FastAPI application, including all dependencies and configurations.

Benefit: The containerized setup enables easy scalability and deployment across various platforms, ensuring the system operates smoothly in diverse environments.

Conclusion

The automated email generation system successfully addresses the need for personalized communication with potential clients and job seekers. By integrating machine learning techniques for intent detection and interest analysis, along with advanced language models for email generation, the system enhances user engagement and delivers tailored responses efficiently. The integration with FastAPI and Docker ensures a scalable and robust deployment, making it suitable for real-time applications.