Student: Joey Lim Jun Feng
Academic Supervisor: Tan See Teck

Leveraging Retrieval-Augmented Generation (RAG) and Large Language Models (LLMs) for Chatbot Development

Introduction

Problem

Universities face challenges in providing instant responses to students' administrative and academic queries. Traditional search methods are inefficient, requiring multiple clicks and manual navigation.

Solution

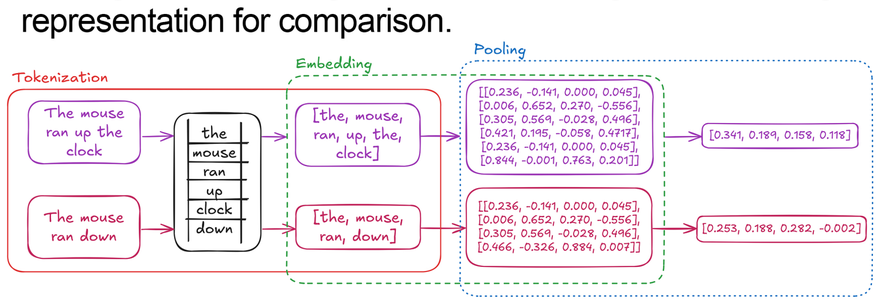
This thesis aims to develop an intelligent chatbot using Retrieval-Augmented Generation (RAG) and Large Language Models (LLMs) to enhance information retrieval efficiency by providing accurate, context-aware responses.

Methodology

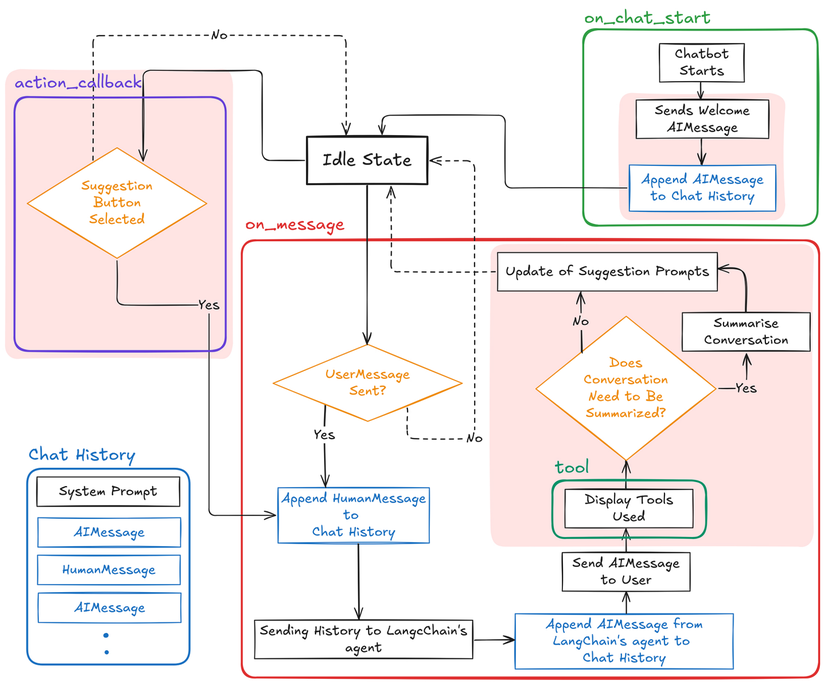
1. Sentence Embeddings

Sentences are converted into numerical representations called embeddings. This process involves three steps:

- Tokenization – Breaking the sentence into smaller units (tokens).
- Embedding – Converting each token into a numerical vector.
- Pooling – Combining token embeddings into a single representation for comparison.



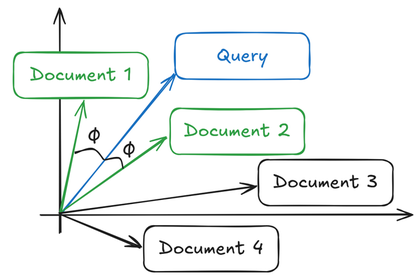
Text Processing in RAG: Tokenization, Embedding, and Pooling.



Chainlit Logic Flow.

2. Retrieval Augmented Generation (RAG)

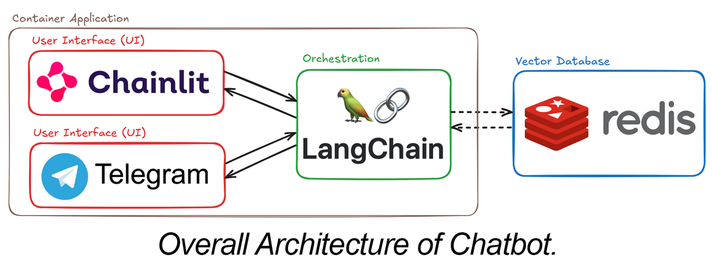
Each document is represented as an embedding vector, and when a user submits a query, it undergoes the same embedding process. The chatbot retrieves the most relevant documents using cosine similarity, enabling the Large Language Model (LLM) to generate a precise response.



Vector Space Representation of Query and Documents.

$$\cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Cosine Similarity Formula.



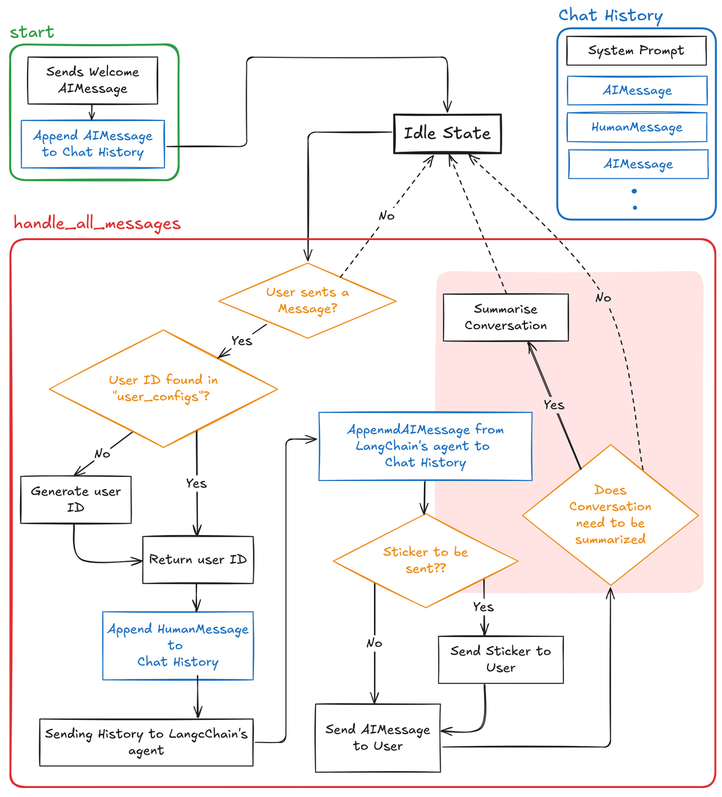
Overall Architecture of Chatbot.

3. Architecture of the Chatbot

The chatbot integrates Chainlit and Telegram as the front-end interfaces, LangChain as the orchestration tool for processing queries, and Redis as the database.

- Telegram – Mobile-friendly interface for chatbot interactions with sticker support.
- Chainlit – Web-based interface with suggestion prompt buttons and tool visibility.

The red-marked areas in the logic flow diagrams highlight newly implemented functions based on the User Study feedback.

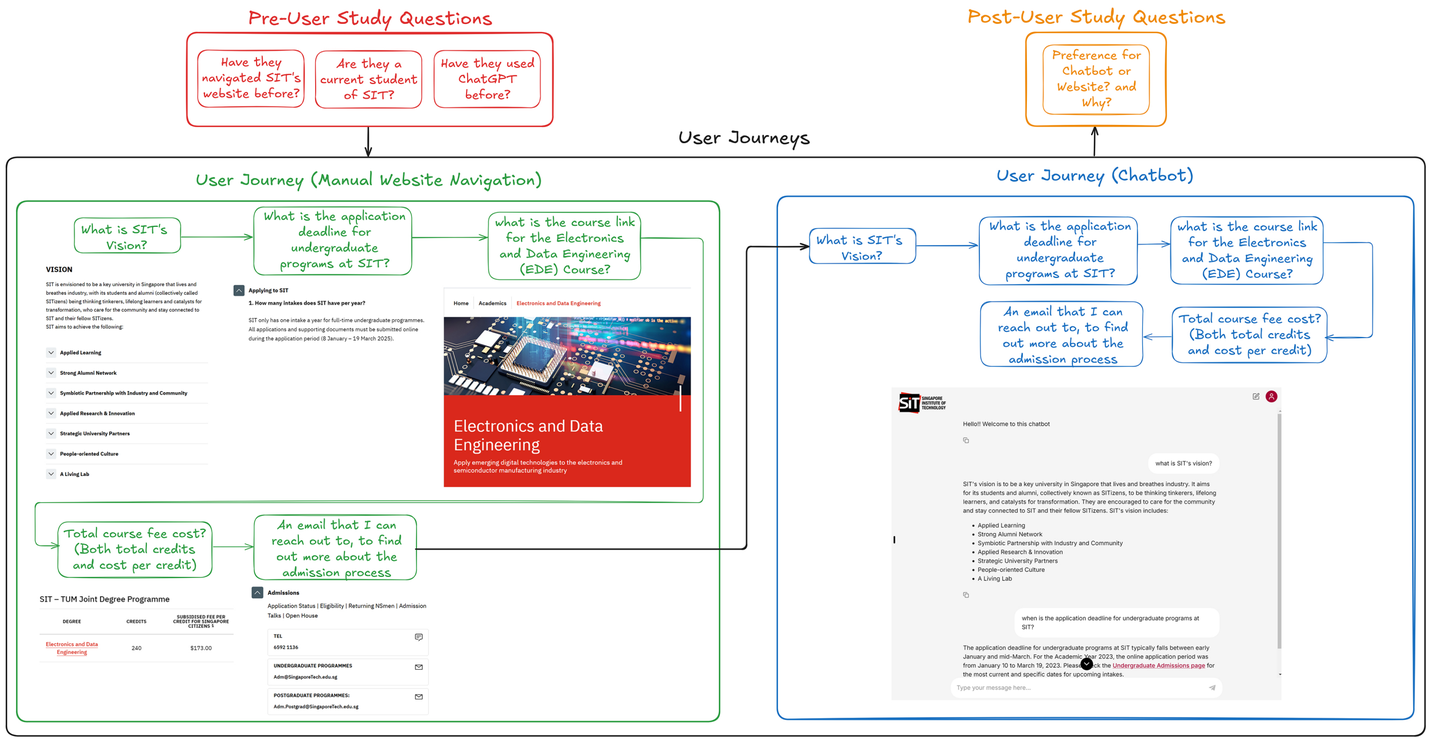


Telegram Logic Flow.

User Study & Results

User Study

The User Study involved 23 participants comparing chatbot interactions with manual website navigation. Each participant attempted to find answers to five questions, simulating a student's typical user journey. This process was conducted twice – once using the chatbot and once through manual website navigation. Response times, user feedback, number of prompts sent to the chatbot and the number of pages navigated were evaluated to assess efficiency and usability.



Overall Structure of User Study.

Results

Results showed that 82.6% (19 participants) preferred the chatbot, which demonstrated up to a 300% increase in speed compared to manual website navigation. Qualitative analysis was conducted by visualizing user feedback through word clouds and categorizing responses to understand the reasoning behind their preferences, leading to key improvements in the chatbot's functionality.

	Website Avg. Time(s)	Chatbot Avg. Time(s)
Question 1	14.65	19.57
Question 2	112.83	31.70
Question 3	31.22	22.78
Question 3	98.00	25.90
Question 4	93.26	28.82
Overall Average	70.00	25.75

Comparison of Average Response Times.

Word Cloud of Website Preference.

Word Cloud of Chatbot Preference.