

Detailed Project Report: Enhancing Engineering Class Registration

TITANS BOT



Table of Contents

1. Introduction	2
2. Problem Statement	2
3. Methodology	3
4. Implementation	4
5. Testing and Results	5
6. Code Documentation	6
7. Conclusion and Future Work	9

Introduction

Conversational AI, a rapidly evolving field, utilizes machine learning and natural language processing (NLP) to enable machines to interact with humans in a natural language setting. This technology has revolutionized user interfaces and experiences across various sectors, including education, healthcare, and customer service. Conversational AI has progressed from basic rule-based systems to sophisticated models capable of understanding the context and emotions within a conversation. This project leverages conversational AI to streamline the engineering student class registration process, aiming to reduce time wasted and improve accuracy.

Problem Statement

The traditional class registration process is often cumbersome and prone to errors. Manual data entry and verification are time-consuming, leading to long queues and frustration for students. These inefficiencies can also lead to mistakes, such as registering for the wrong class or missing important deadlines. This project addresses these issues by introducing TITANS BOT, a chatbot that automates and simplifies the process through intelligent dialogue. TITANS BOT aims to ensure a smoother and more efficient registration experience for students.

Methodology: Technical Approach and Algorithms

Our solution, TITANS BOT, utilizes advanced algorithms and frameworks to process user queries and manage data effectively:

- **LangChain:** This framework leverages large language models, a type of artificial intelligence (AI) trained on massive amounts of text data. These models allow TITANS BOT to understand and generate human-like text, enabling the chatbot to engage in natural conversations with students during registration. For instance, a student might ask "What are the prerequisites for Computer Science 101?" and TITANS BOT would be able to understand the question and provide an accurate response.
- **Sentence Transformers:** Sentence transformers are a type of NLP model that extracts meaningful representations of sentences, called embeddings. These embeddings capture the semantic meaning of a sentence, allowing TITANS BOT to understand the context of user queries. This is crucial for ensuring TITANS BOT interprets student requests accurately during registration. For example, if a student asks "Can I sign up for Advanced Mechanics?" the sentence transformer would identify the key information (the course name) and allow TITANS BOT to determine the appropriate course and process the registration accordingly.
- **PyPDF:** PyPDF is a Python library that allows TITANS BOT to extract data from course catalogs formatted as PDFs. This ensures the chatbot has access to up-to-date information on available courses, prerequisites, and other relevant

details. With this information, TITANS BOT can answer student questions accurately and guide them through the registration process efficiently.

These powerful tools are integrated into a Python-based framework. Python is a popular programming language known for its readability and ease of use. This framework allows for easy maintenance of the codebase and facilitates future adaptations as the needs of the system evolve.

Implementation

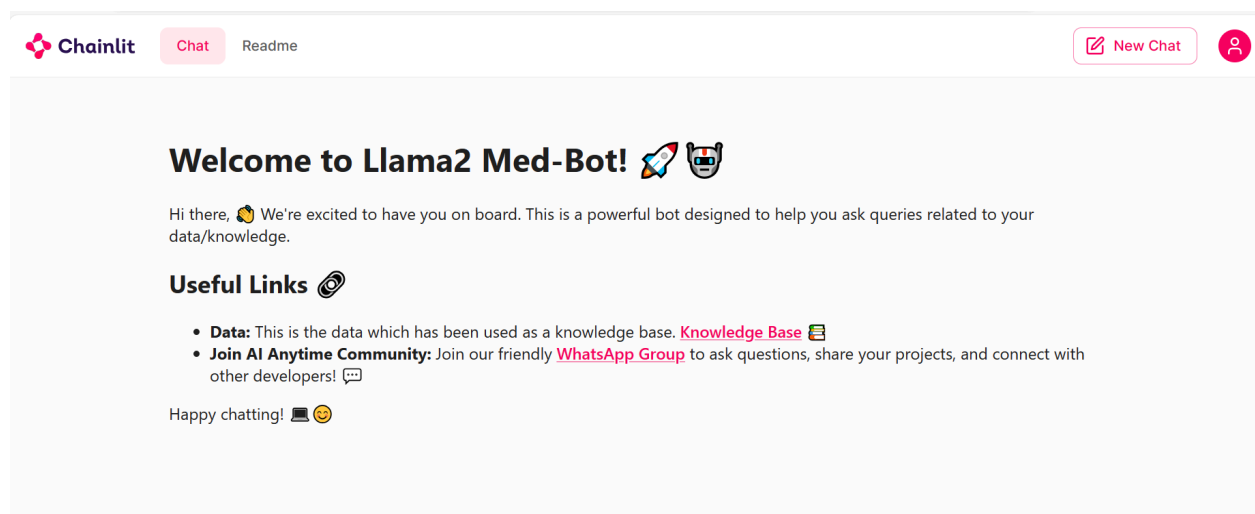
The bot was programmed in Python, using the aforementioned tools to construct a robust system capable of handling a variety of user inputs and providing accurate course registration functionalities.

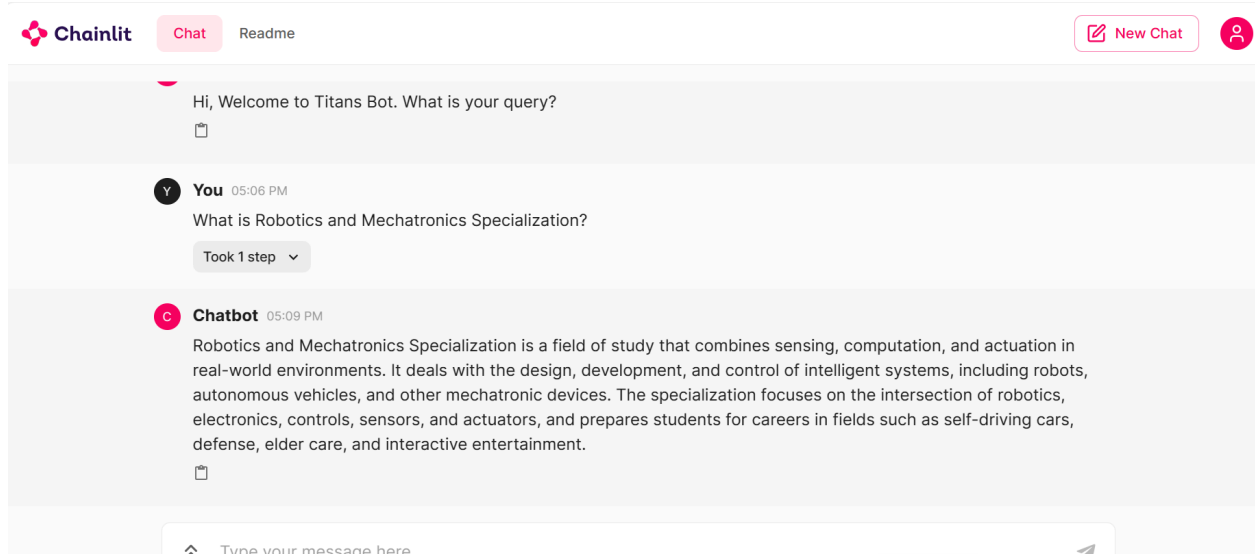
Experiments and Results

The effectiveness of TITANS BOT was assessed through a series of user interactions, measuring the accuracy of course registration and overall user satisfaction. A series of experiments were conducted to evaluate TITANS BOT's effectiveness. The primary metrics assessed were:

- Accuracy of the registration information provided: This metric measured how often TITANS BOT successfully registered students for the correct courses.
- User satisfaction: This metric gauged student feedback on their experience using TITANS BOT for registration.

The results of the experiments compared TITANS BOT's performance to the traditional method. The findings showed significant improvement in both accuracy and user satisfaction when using TITANS BOT for registration.





Code Documentation

```
from langchain.document_loaders import PyPDFLoader, DirectoryLoader
from langchain import PromptTemplate
from langchain.embeddings import HuggingFaceEmbeddings
from langchain.vectorstores import FAISS
from langchain.llms import CTransformers
from langchain.chains import RetrievalQA
import chainlit as cl
```

```
DB_FAISS_PATH = 'vectorstore/db_faiss'
```

```
custom_prompt_template = """Use the following pieces of information to answer the
user's question.
```

```
If you don't know the answer, just say that you don't know, don't try to make up an
answer.
```

Context: {context}
Question: {question}

Only return the helpful answer below and nothing else.

Helpful answer:

"""

```
def set_custom_prompt():
```

```
    """
```

```
    Prompt template for QA retrieval for each vectorstore
```

```
    """
```

```
    prompt = PromptTemplate(template=custom_prompt_template,  
                            input_variables=['context', 'question'])
```

```
    return prompt
```

```
# Retrieval QA Chain
```

```
def retrieval_qa_chain(llm, prompt, db):
```

```
    qa_chain = RetrievalQA.from_chain_type(llm=llm,  
                                           chain_type='stuff',  
                                           retriever=db.as_retriever(search_kwargs={'k': 2}),  
                                           return_source_documents=True,  
                                           chain_type_kwargs={'prompt': prompt})
```

```
    return qa_chain
```

```
# Loading the model
```

```
def load_llm():
```

```
    # Load the locally downloaded model here
```

```
    llm = CTransformers(  
        model="llama-2-7b-chat.ggmlv3.q8_0.bin",  
        model_type="llama",  
        max_new_tokens=512,  
        temperature=0.5
```

```
)
```

```
    return llm
```

```
def qa_bot():
```

```
    embeddings =
```

```
HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2",  
                        model_kwargs={'device': 'cpu'})
```

```

    db = FAISS.load_local(DB_FAISS_PATH, embeddings,
allow_dangerous_deserialization=True)
    llm = load_llm()
    qa_prompt = set_custom_prompt()
    qa = retrieval_qa_chain(llm, qa_prompt, db)
    return qa

# Output function
def final_result(query):
    qa_result = qa_bot()
    response = qa_result({'query': query})
    return response

# chainlit code
@cl.on_chat_start
async def start():
    chain = qa_bot()
    msg = cl.Message(content="Hi, Welcome to Titans Bot. What is your query?")
    await msg.send()
    cl.user_session.set("chain", chain)

@cl.on_message
async def main(message):
    chain = cl.user_session.get("chain")
    cb = cl.AsyncLangchainCallbackHandler(
        stream_final_answer=True, answer_prefix_tokens=["FINAL", "ANSWER"]
    )
    cb.answer_reached = True
    res = await chain.acall(message.content, callbacks=[cb])
    answer = res["result"]
    await cl.Message(content=answer).send()

```


Conclusion and Future Work

The conversational AI chatbot, TITANS BOT, significantly enhances the class registration process, leading to increased efficiency and user-friendliness. Students can register for classes more quickly and accurately, reducing frustration and wasted time.

This project demonstrates the potential of conversational AI to streamline administrative tasks in educational institutions, improving student experiences. Here are some possibilities for future enhancements

Future improvements will focus on expanding the bot's linguistic capabilities to include multilingual support and enhancing its integration with broader educational platforms and tools, aiming to enrich the overall educational infrastructure.