Contents

1 Introduction

Intelligent Q&A systems have seen growing use in education, customer service, and online consultation. With advances in NLP, machine learning, and large language models (LLMs), chatbots are now more capable of understanding complex queries and managing multi-turn conversations.

In university settings, students often face practical questions related to course schedules, administrative procedures, and campus facilities. Traditional methods of information access—such as browsing websites or asking peers—can be time-consuming and inefficient.

This project aims to design and implement a chatbot tailored to student inquiry scenarios. It accurately detects user intent, recognizes key entities, and generates context-aware responses. By integrating multiple NLU components, including intent classification, entity recognition, and sentiment analysis, the system enhances user experience, streamlines information access, and improves overall information availability on campus.

2 Project Methodology

2.1 Research Design and Paradigm

This project adopts the Constructive Research / Design Science (DSR) research paradigm. The core objective is to design, build, and evaluate a novel AI system—a hybrid chatbot that integrates both retrieval-based and template-based features—to address real-world problems such as low information retrieval efficiency and lack of emotional interaction in campus settings(Yang et al., 2019).

The overall research process follows an iterative development cycle:

1. Requirement Analysis and Data Collection: Identify key information needs in campus scenarios and gather relevant raw data.
2. System Design and Implementation: Design and build a system architecture comprising a Natural Language Understanding (NLU) module, a response generation module, and an interactive front-end interface.
3. Model Training and Development: Train the core NLU model using synthetic data generated through LLM-based strategies.
4. System Integration and Evaluation: Integrate all modules into a working prototype and evaluate the system using both quantitative and qualitative methods.

5. Iterative Optimization: Refine the system by enriching training data and optimizing components based on evaluation feedback.

This design-oriented research approach ensures that the outcomes are not only theoretically grounded but also practically valuable. The iterative methodology further enables rapid feedback incorporation and continuous system improvement in real-world usage scenarios.

2.2 System Architecture

The chatbot system consists of a front-end user interface and a back-end processing core. The back-end adopts a modular design, primarily comprising a Natural Language Understanding (NLU) module and a response generation module. The functions of each component will be described in detail in the following subsection(Mohammed & Aref, 2022).

2.3 Data Collection and Preparation

High-quality data is fundamental to the performance of natural language processing systems. In this study, data collection was conducted on two levels: (1) organizing raw information for knowledge base construction, and (2) generating training corpora for the NLU model. Both types of data underwent structured processing and manual verification to ensure their usability and reliability in system development.

2.3.1 Knowledge Base Data

We manually collected information on campus facilities, dining options, academic procedures, and more by consulting the student handbook, official public accounts, and other campus information sources, and structured it into a searchable `JSON` file.

2.3.2 Model Training Data

To train the NLU model, we adopted a synthetic data strategy. Specifically, we first designed a data generator (`data_generator.ipynb`) that fills predefined intents and entities into prompt templates. Then, a locally deployed Ollama LLM is used to generate many feature-label pairs. Finally, all generated data undergo manual sampling, review, and correction to ensure quality and accuracy before being used for model training(Li, Bonatti, Abdali, Wagle, & Koishida, 2024).

2.4 Evaluation Methods

To comprehensively evaluate the effectiveness of this project, we adopted a combination of quantitative and qualitative assessment methods(Alfrink, 2024).

2.4.1 Quantitative Evaluation

The evaluation primarily focuses on the performance of the NLU model. We used an independent test set and employed scikit-learn's `classification_report` tool to automatically calculate a range of model evaluation metrics, including precision, recall, and F1-score—derived from the confusion matrix—to assess the model's performance across each class.

2.4.2 Qualitative Evaluation

The evaluation primarily targets the overall user experience and practicality of the chatbot system. We conducted multi-turn, open-ended conversations with the chatbot, with a particular focus on its ability to handle edge cases beyond the training data. This allowed us to assess the fluency of the dialogue, the accuracy of the responses, and the effectiveness of emotional interaction.

3 Design of the Intelligent System

3.1 Core Modules and Their Functions

3.1.1 User Input Interface

This module is responsible for capturing user input and sending it to the back-end interpretation module.

3.1.2 Natural Language Understanding (NLU) Module

The Natural Language Understanding (NLU) module serves as the system's initial processing stage, responsible for transforming users' free-form text input into structured semantic information(Smutek, Golec, Rymarczyk, Jarmuł, & Hernas, 2024). It consists of the following three subsystems:

- **Intent Classification**
  This model aims to identify the user's query intent within a specific context. The task is formulated as a multi-class text classification problem, implemented using a Scikit-learn `pipeline` combining a *TF-IDF* feature extractor and a *logistic regression* classifier.

- **Entity Recognition**
  To further extract key information from user input (such as location, time, or objects), this task is modeled as a *sequence labeling problem.* A custom Named Entity Recognition (NER) model is built using the `spaCy` framework, applying supervised learning to identify various entities in the text, thereby supporting knowledge retrieval.

- **Sentiment Analysis**
  To enhance the system's human-centered and emotional awareness capabilities, the *VADER sentiment analysis* tool is integrated to classify the emotional tone and intensity of user input. The output includes positive, negative, and neutral sentiment categories, along with a compound score, which informs template selection and interaction tone adjustment.

3.1.3 Response Generation Module

The response generation module is responsible for producing user-understandable and naturally styled text replies based on the intent, entities, and sentiment data output by the NLU module. It comprises the following two core components:

- **Knowledge Base Retriever**
  The system's knowledge is organized in a structured JSON format, covering common campus inquiry scenarios. The retriever receives keywords and labels from the NLU module and performs matching queries within the knowledge base to locate the most relevant information entries.

- **Response Templating Engine**
  A hierarchical response templating strategy is employed, with standardized language templates designed for different intent types. Based on the user's identified intent, extracted entities, and sentiment scores, the engine dynamically selects the appropriate template and fills in placeholders with information retrieved from the knowledge base, ultimately generating coherent and complete natural language responses.

3.1.4 Interactive Front-end Interface

The front-end interface is built using a web framework and provides a chat-like, turn-based interaction experience.

3.2 Data Flow and Workflow

The system operates based on a single-turn interaction model following the sequence: *Input* → *Interpretation* → *Generation* → *Output*. For each user query, the system sequentially invokes its modules and returns a structured, readable response. The following describes a typical data flow during system operation.

3.2.1 Input Stage

System operation begins when the user submits a natural language query through the input interface, such as ”What time does the library close?”. Before entering the main processing pipeline, the input is first passed through a sentence segmentation step to enable sentence-by-sentence analysis for multi-sentence queries.

3.2.2 Natural Language Understanding Stage

Once the input text enters the Natural Language Understanding (NLU) module, it undergoes the following processing steps in sequence:

- Intent Classification Each sentence is classified using an intent classifier built on TF-IDF features and a logistic regression model to identify its semantic purpose, such as ”location inquiry,” ”time request,” or ”contact information request.”
- Named Entity Recognition (NER) A custom NER model trained with the spaCy framework extracts key entities from the input—such as ”library,” ”academic office,” or ”17:30”—to support content generation in the response module.
- Sentiment Analysis The VADER tool is used to perform basic sentiment polarity analysis, producing a positive, neutral, or negative tendency. This result can inform future system versions to support tone-adaptive responses.

After each clause is processed, the system organizes the output into a unified intermediate format. For example, given the input ”What time does the library close?”, the structured output is:

{} JSON

```json
{
        "query": "What time does the library close?",
        "intent": "ask_facility_time",
        "entities": {"library": "LOCATION"},
        "sentiment": {"compound": 0.0}
}
```

3.2.3 Response Generation Stage

Based on the structured output, the system invokes the response generation module. The current version primarily relies on a rule-based matching mechanism: the system locates the appropriate response template according to the identified intent category and fills the predefined slots with the recognized entities to generate a semantically appropriate reply.

3.2.4 Output Stage

The final response text is processed by the output formatting module and then returned to the user interface for display, completing a full question-and-answer interaction cycle.

4 Implementation

4.1 System Architecture and Modules

4.2 Prototype Screenshots

4.3 System Execution and Testing

5 Discussion

5.1 Summary of Overall Effectiveness

This system is designed to improve information access for university students and enhance the accessibility of campus-related content. During development, a modular architecture was adopted to facilitate system construction and deployment.

Evaluation results indicate that the system performs reliably in intent classification, effectively covering most common campus-related queries. The sentiment analysis module also contributed to tone adjustment in responses, enhancing the naturalness and user-friendliness of interactions. While the current version does not yet support multi-turn dialogue or context modeling, it has demonstrated strong task completion and user experience in single-turn scenarios, meeting the intended design goals.

5.2 Advantages of Models and Systems

The system demonstrates several key strengths in both core modeling and overall architectural design. First, the intent and entity recognition models performed well on the test set, showing high accuracy and stability, which provides a reliable foundation for information retrieval. Additionally, the integration of the sentiment analysis module significantly enhanced the user interaction experience, making responses more human-like and context-sensitive.

5.3 Issues and challenges

During the development and deployment of the system, we also encountered several practical challenges and limitations. One of the major issues was the limited timeliness of the raw data, which made it difficult to keep up with real-time changes in campus operations and nearby businesses. In certain query scenarios, this could lead to outdated information, affecting the accuracy and credibility of the responses.

5.4 Directions for Improvement

Although the current system has demonstrated stable performance in single-turn question-answering scenarios, there are several areas for improvement in future iterations. First, incorporating a context-tracking mechanism would enable the system to understand multi-turn conversations, enhancing its ability to handle complex queries with greater coherence and continuity. Second, on the front-end side, further optimization of the interface design and response feedback mechanisms could improve usability and provide a smoother user experience.

References

Alfrink, K. (2024). *Contestable artificial intelligence: Constructive design research for public artificial intelligence systems that are open and responsive to dispute* (Dissertation (TU Delft), Delft University of Technology). doi: 10.4233/uuid:0b014ee6-67a8-4c59-8598-4bfd771595a3

Li, Y., Bonatti, R., Abdali, S., Wagle, J., & Koishida, K. (2024). *Data generation using large language models for text classification: An empirical case study.* Retrieved from https://arxiv.org/abs/2407.12813

Mohammed, M., & Aref, M. M. (2022). *Chatbot system architecture.* Retrieved from https://arxiv.org/abs/2201.06348

Smutek, T., Golec, E., Rymarczyk, P., Jarmuł, M., & Hernas, A. (2024, 08). Enhancing conversational ai with the rasa framework: intent understanding and nlu pipeline optimization. *Journal of Modern Science*, *57*, 531-548. doi: 10.13166/jms/191223

Yang, L., Hu, J., Qiu, M., Qu, C., Gao, J., Croft, W. B., … Liu, J. (2019). *A hybrid retrieval-generation neural conversation model.* Retrieved from https://arxiv.org/abs/1904.09068