# Introduction

近年来，智能问答系统在教育、客服、在线咨询等领域得到了广泛应用。这类系统通过自然语言处理（NLP）与机器学习技术，能够理解用户提问并及时提供准确的回复。随着大语言模型（LLM）和深度学习算法的快速发展，聊天机器人在处理多轮对话和复杂意图识别方面表现出越来越强的能力。

在高校校园中，学生在日常学习和生活中经常会遇到各类实用性问题，如课程安排、行政流程、校园设施使用等。传统的信息获取方式（例如浏览网页、查阅论坛、询问工作人员或同学）往往效率不高，耗时耗力。

本项目旨在设计并实现一个面向学生提问场景的聊天机器人系统，能够准确理解用户意图，识别相关实体，并提供符合语境的响应。该系统集成了多种自然语言理解（NLU）模块，包括意图分类、实体识别与情绪分析，力求提升用户体验，优化信息获取途径，增强学生在校园中的信息可达性。

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.

# Project Methodology

## Research Design and Paradigm

本项目采用构建性研究/设计科学(Constructive Research / Design Science)的研究范式。其核心目标是设计、构建并评估一个新的人工智能系统——一个结合了检索式(Retrieval-based)与模板式(Template-based)特点的混合式聊天机器人——以解决校园场景下信息获取效率低下和情感交互缺失的实际问题。

本研究的整体流程遵循一个迭代的开发周期：

1. 需求分析与数据采集: 识别校园场景下的核心信息需求，并收集原始数据。

2. 系统设计与构建: 设计并实现一个包含NLU模块，响应生成模块和交互式前端的系统架构。

3. 模型训练与实现: 利用LLM生成模拟数据的策略训练核心的NLU模型。

4. 系统集成与评估: 将所有模块集成为一个可工作的原型，并通过定量和定性方法进行评估。

5. 迭代优化: 根据评估结果，补充训练数据并对系统进行优化。

这种设计导向的研究路径确保了研究成果不仅具备学术理论意义，同时具备高度的现实适用价值。迭代式方法论的引入也有助于在真实使用场景中快速响应反馈，持续提升系统表现。

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.

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.

## System Architecture

本聊天机器人系统由前端用户界面和后端处理核心组成。后端核心采用模块化设计，主要由自然语言理解（NLU）单元和响应生成单元构成。各个模块的功能将在下一小结详细描述。

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.

# Data Collection and Preparation

高质量的数据是自然语言处理系统性能的基础保障。本研究的数据采集工作从两个层面展开：一是面向**知识库构建**的原始信息整理，二是面向**NLU模型训练**的语料数据生成。两类数据均经过结构化处理与人工校验，确保其在系统开发中的可用性与可靠性。

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.

### 知识库原始数据Knowledge Base Data

我们通过查阅学生手册、官方公众号以及其他校园信息渠道，手动整理了关于校园设施、餐厅、学术流程等信息，并将其结构化为一个易于检索的JSON文件。

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.

### 模型训练数据Model Training Data

为训练NLU模型，我们采用了合成数据(Synthetic Data)策略。具体流程是：首先设计一个数据生成器 (data\_generator.ipynb)，它将预设的意图和实体填入Prompt模板；然后调用本地运行的Ollama LLM生成大量的特征-标签对；最后，所有生成的数据都经过人工抽样检查与修正，以确保其质量和准确性，最终用于模型训练。

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.

## 评估方法Evaluation Methods

为了全面衡量本项目的成效，我们采用了定量和定性相结合的评估方法。

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

### 定量评估Quantitative Evaluation

主要针对NLU模型的性能。我们使用独立的测试集，并通过scikit-learn的classification\_report工具自动计算一系列模型评估指标(Model Evaluation Metrics)，包括基于混淆矩阵(Confusion Matrix)计算出的精确率(Precision)、召回率(Recall)和F1分数(F1-Score)，以评估模型在每个类别上的性能。

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.

### 定性评估Qualitative Evaluation

主要针对整个聊天机器人的用户体验和实用性。我们通过与机器人进行多轮、开放式的对话来执行评估，尤其关注其处理训练数据之外的边缘问题(edge cases)的能力，以检验对话的流畅性、答案的准确性和情感交互的有效性。

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.

# Design of the Intelligent System/Application

## Core Modules & Responsibilities

### 用户输入接口模块（User Input Interface）

该模块负责捕获用户输入并发送至后端解释模块。

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

### 自然语言理解模块（NLU）Natural Language Understanding (NLU) Unit

自然语言理解模块作为系统的第一处理环节，负责将用户输入的自由文本转化为结构化语义信息。该模块内部由以下三个子系统组成：

* **意图识别模型（Intent Classification）**  
  本模型旨在识别用户在特定语境下的查询意图。我们将其建模为一个**多类别文本分类问题**，训练方式采用**TF-IDF特征提取器**与**逻辑回归分类器**组成的Scikit-learn Pipeline。
* **实体识别模型（Entity Recognition）**  
  为进一步提取用户输入中的关键信息（如地名、时间、对象等），我们将该任务建模为**序列标注问题**。采用spaCy框架构建自定义命名实体识别（Named Entity Recognition, NER）模型，通过监督学习识别文本中的各类实体，进而辅助知识检索。
* **情感分析模块（Sentiment Analysis）**  
  为增强系统的人文关怀与情绪感知能力，我们引入了VADER情感分析工具，对用户文本进行情绪分类与强度评分。输出包括积极、消极、中性三类情感倾向以及复合情感得分（compound score），用于辅助模板选择与交互语气调节。

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. 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.

### 响应生成模块Response Generation Unit

响应生成模块负责根据NLU输出的意图、实体与情绪数据，生成用户可理解、风格自然的文本回复。该模块由以下两个核心单元构成：

* **知识库检索器（Knowledge Base Retriever）**  
  系统知识以结构化JSON格式组织，涵盖常见校园问询场景。检索器接收NLU模块输出的关键词与标签，在知识库中进行匹配查询，定位最相关的信息条目。
* **响应模板引擎（Response Templating Engine）**  
  本系统采用分层级响应模板策略，针对不同意图设计标准化语言模板。该引擎会根据用户意图类型、识别到的实体及情感得分，动态选择适配模板，并将从知识库提取的信息填入占位符，最终输出连贯完整的自然语言回答。

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.

### 交互式前端界面Interactive Front-end Interface

前端界面采用Web框架构建，提供类聊天窗口的一问一答的操作体验。

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

## Data Flow & Workflow

本系统的运行流程遵循“输入 → 解释 → 生成 → 输出”的单轮交互模式。每一次用户发起查询，系统都会依次调用各个模块，并输出一个结构化且可阅读的回答。以下为系统运行时的典型数据流程说明：

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:

### Input Stage

系统运行始于用户通过输入界面提交的自然语言查询，例如“图书馆几点关门？”。该输入在进入主处理流程前，首先经过句子分割处理，以支持多句输入的逐句分析。

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.

### Natural Language Understanding Stage

输入文本传入自然语言解释模块后，依次完成如下处理步骤：

* **意图识别（Intent Classification）**  
  利用基于 TF-IDF 特征和逻辑回归模型构建的意图分类器对每句输入进行分类，识别其语义目的，如“查询地点”、“咨询时间”、“请求联系方式”等。
* **实体识别（Named Entity Recognition）**  
  采用基于 SpaCy 框架训练的命名实体识别模型，抽取输入中的关键实体，例如“图书馆”“教务处”“17:30”等，以提供给响应模块进行内容填充。
* **情感分析（Sentiment Analysis）**  
  引入 VADER 工具进行基础情感极性分析，获得正向、中性或负向倾向。该分析结果可用于未来扩展版本中实现语气自适应回应。

每一子句处理完毕后，系统将其组织为统一格式的中间结果。以示例输入“图书馆几点关门？”为例，结构化输出如下所示：

{

"query": " What time does the library close? ",

"intent": "ask\_facility\_time",

"entities": { "library": "LOCATION" },

"sentiment": { "compound": 0.0 }

}

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:

{

"query": " What time does the library close? ",

"intent": "ask\_facility\_time",

"entities": { "library": "LOCATION" },

"sentiment": { "compound": 0.0 }

}

### Response Generation Stage

系统根据上述结构化输出，调用响应生成模块。当前版本主要基于规则匹配机制：系统根据意图类别查找对应模板，并将识别出的实体填入预设槽位中，生成符合语义的回复内容。

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.

### Output Stage

最终生成的回复文本经由输出格式化模块处理后，返回至用户界面进行展示，完成一次问答交互闭环。

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.

# Implementation

## System Architecture and Modules

## Prototype Screenshots

## System Execution and Testing

# Discussion