# （一）基于大语言模型的实时对话系统

## 系统背景:

大语言模型的发展显著提升了自然语言处理的能力，具备强大的文本生成、翻译和问答功能。其上下文感知能力使得模型能够生成连贯的文本，并基于广泛的知识基础快速提供信息和建议。此外，自动化处理极大提高了工作效率，并支持多种语言，促进跨语言交流。这些模型的灵活性使其可以根据特定需求进行微调，同时具备创作富有创造性的内容的能力，从而为用户提供数据驱动的决策支持。这些优点推动了人工智能在各行各业的广泛应用。然而，大语言模型在与用户的交互中仍然局限于逐句对话的形式，存在以下局限性：

(1) 逐句对话限制：用户需要等待对方的回复才能继续交流，这限制了互动的灵活性和流畅性。

(2) 信息传递延迟：这种逐步对话方式可能导致信息传递的延迟，尤其在处理复杂问题时，用户常常需要通过多轮问答才能达到预期效果。

(3) 实时交互不足：这一局限性使得大语言模型在某些实时交互场景中显得不够高效，亟需改进以提升其在动态对话中的适应能力和响应速度。

因此，开发大语言实时对话系统显得尤为重要，这类系统可以使大语言模型与用户的互动更接近现实中人与人之间的交流方式。通过实时对话，模型能够利用文本、语音等多种模态，及时理解用户在交流过程中可能出现的打断和新话题提出等情况，并迅速做出响应。这种灵活性不仅提升了交互的流畅度，还增强了用户体验，使交流更加自然高效，从而实现更深层次的理解与信息交换。

## 主要功能

实时对话系统的主要目的是提升大语言模型与用户之间的交互效率，主要功能模块包括大模型推理、多模态识别和模型实时对话三大模块。

（1）大模型推理  
系统能够基于用户输入，让大模型做出响应的推理。模型能够生成自然、连贯的回复，帮助用户解答问题或提供信息。

（2）多模态识别  
系统能够同时处理文本和语音输入，实时识别用户在对话中可能的打断或新话题提出情况，确保互动的灵活性和顺畅性。

（3）模型实时对话  
用户与大语言模型之间的对话是实时的，系统能够快速响应用户的提问或请求，从而提升交流的效率，增强用户体验。用户在互动过程中可以自由切换话题，系统会及时适应这些变化。

## 开发环境

操作系统及版本：Ubuntu 20.04

支撑软件及版本列表：Python 3.10, Flask 2.0, Node.js 14.0, Vue 3.0

数据库环境：MongoDB 4.4

版本控制系统：Git 2.x

## 基本的架构体系

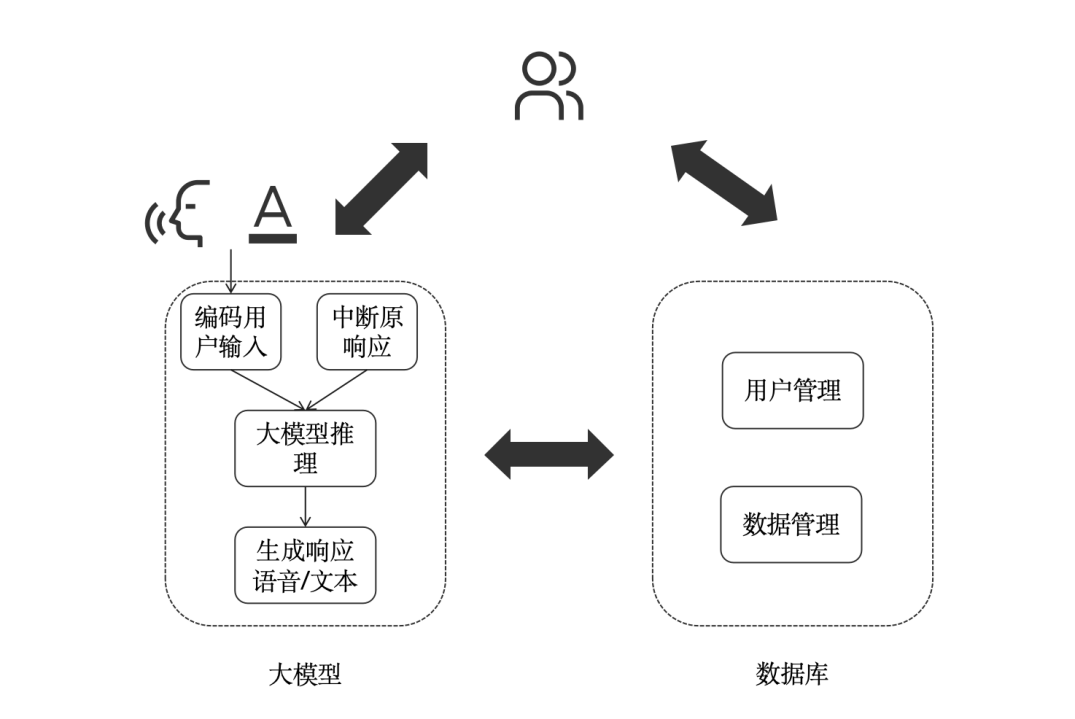


图 1 大语言实时对话系统功能结构图

用户管理：由用户和管理员共同操作，用户可以进行注册、登陆，管理员可以操作用户数据。

实时对话：由用户操作，用户可以选择开启新会话、继续旧会话或删除旧会话。用户可以和大语言模型进行实时对话，包括语音输入和文本输入等，系统会根据用户输入做出响应，如正常推理、中断、开启新话题等。

数据管理：由用户和管理员共同操作，用户可以访问或删除自己的历史对话记录。管理员可以管理所有用户的历史聊天记录。