

## 10 技术创新点

### 10.1 流式响应与动态渲染

- **实时流式对话:** 采用 Server-Sent Events (SSE) 技术实现流式响应, 使 AI 回答能够实时显示, 提升用户体验。以下代码展示了如何使用 `StreamingResponse` 实现流式数据传输:

```
from fastapi.responses import StreamingResponse

@app.post("/chat")
async def ask(question: Question):
    async def generate():
        async for chunk in qa_system.get_answer_stream(question.question,
            question.session_id):
            yield chunk

    return StreamingResponse(generate(), media_type='text/event-stream')
```

- **Markdown 动态渲染:** 使用 `marked.js` 实现 Markdown 实时解析, 支持富文本格式化显示。以下代码展示了如何在前端使用 `marked.js` 进行 Markdown 解析:

```
<script src="https://cdn.jsdelivr.net/npm/marked/marked.min.js"></script>
<div class="message-bubble" style="background: white;">
  <script>
    document.querySelector(".message-bubble").innerHTML = marked.parse(
      "**Markdown** content here"
    );
  </script>
</div>
```

### 10.2 智能图片处理机制

- **上下文感知图片选择:** LLM 根据对话内容智能筛选相关图片。以下 JSON 数据展示了如何存储图片路径:

```
{
  "问题类型": "食堂攻略",
  "具体问题": "益新食堂一楼环境怎么样",
  "标准回答": "益新食堂一楼为四人桌",
  "图片路径": "企创图片资料/益新食堂一楼环境.jpg"
}
```

- **图片标记智能处理:** 将图片信息嵌入到对话上下文中。以下代码展示了如何在 Markdown 中标记图片:

```
![Image Description](path/to/image.jpg)
```

### 10.3 交互体验优化

- **预览与全屏查看:** 实现图片预览缩略图和全屏查看功能。以下 CSS 代码展示了如何实现图片预览和全屏查看:

```
.image-modal {
  display: none;
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: rgba(0, 0, 0, 0.8);
  z-index: 1000;
  justify-content: center;
  align-items: center;
}

.modal-content img {
  max-width: 100%;
  max-height: 90vh;
  object-fit: contain;
}
```

### 10.4 响应式布局

- **响应式布局:** 采用 Tailwind CSS 实现自适应布局。以下代码展示了如何在 HTML 中使用 Tailwind CSS:

```
<link rel="stylesheet" href="https://cdn.tailwindcss.com" />
<div class="flex flex-col items-center p-4">
  <h1 class="text-xl text-teal-600 font-bold">SHU</h1>
</div>
```

### 10.5 知识检索优化

- **向量化检索:** 使用 FAISS 进行高效的向量相似度检索。以下代码展示了如何使用 SentenceTransformer 进行文本编码:

```
from transformers import AutoTokenizer, AutoModel

model_name = "sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModel.from_pretrained(model_name)
```

### 10.9 数据管理与维护优化

- **灵活的数据管理:** 数据采用 JSON 格式存储，并支持 CSV 转换和更新。以下 Python 脚本展示了如何将 Excel 数据转换为 JSON：

```
import pandas as pd

def excel_to_json(excel_file, json_file):
    df = pd.read_excel(excel_file)
    df.to_json(json_file, force_ascii=False, orient='records', indent=4)
```

## 10.10 部署与访问优化

- **云服务支持:** 项目部署在云服务器上，便于远程访问与维护。以下代码展示了 FastAPI 的基本设置：

```
from fastapi import FastAPI

app = FastAPI()

@app.get("/")
async def root():
    return {"message": "上海大学智能问答系统API"}
```

- **API 开放性:** 提供开放 API，便于未来与第三方应用集成。以下代码展示了如何定义一个简单的 API 端点：

```
@app.post("/chat")
async def ask(question: Question):
    # API logic here
    return {"response": "This is a response"}
```