在线健康问答助手

第一部分：课题背景与目标

1.1 课题背景

随着互联网信息的爆炸式增长，公众获取健康信息的渠道日益多样化。然而，海量的健康信息质量良莠不齐，充斥着大量未经证实、甚至是错误的观点，对公众健康构成了潜在威胁。近年来，以大型语言模型（LLM）为代表的生成式人工智能技术在自然语言处理领域取得了突破性进展，催生了众多智能问答助手。这些助手能够以流畅、自然的方式回答用户提问，极大地提升了信息获取的效率。

然而，在严肃的健康医疗领域，通用大模型的应用面临着严峻的挑战。其中最突出的问题是“模型幻觉”（Hallucination），即模型可能生成看似合理但实际上是捏造的、不准确的医疗信息。对于健康问题，一个错误的回答可能导致用户延误就医、误用药物，甚至造成严重的健康损害。因此，确保健康问答助手所提供信息的**准确性、可靠性和透明度**，是该领域技术应用的核心瓶颈。

为了解决这一问题，学术界和工业界正在探索将**循证医学（Evidence-based Medicine, EBM）的理念融入人工智能系统。循证医学的核心是强调任何医疗决策都应基于当前可获得的、最可靠的科学证据。其中，证据分级是对不同来源的医学证据进行质量和可信度评估的关键方法，例如国际广泛认可的GRADE（Grading of Recommendations Assessment, Development and Evaluation）系统**[1]。通过为AI的回答附上证据来源及其等级，用户可以清晰地了解信息的可靠性，从而做出更明智的判断。

此外，用户在咨询健康问题时，其提问往往是模糊的、口语化的，或者缺少关键的上下文信息（例如：“我最近肚子不舒服，怎么办？”）。传统的单轮问答系统难以在这种情况下提供精准的回答。因此，引入**多轮澄清机制（Multi-round Clarification Mechanism）**至关重要。该机制能够模拟医生问诊的过程，通过主动提问、追问和确认，逐步引导用户明确自己的意图和具体情况从而使系统能够基于更完整的信息，检索到更相关的证据，并生成更具针对性的回答。

本项目旨在结合以上两大核心理念，设计并开发一个基于证据分级与多轮澄清机制的在线健康问答助手。该助手不仅追求回答的智能化和自然度，更致力于成为一个负责任、可信赖、可追溯的健康信息

咨询工具。

1.2 课题目标

本课程设计旨在引导综合运用人工智能、自然语言处理、软件工程等领域的知识，完成一个具有现实意义和技术深度的项目。通过本课题的实践，应达成以下目标：

1. **核心目标：**
    - **构建一个健康问答助手原型系统。** 该系统能够理解用户的健康相关问题，并通过多轮交互进行澄清，最终提供附有明确证据来源和证据等级的回答。
    - **设计并实现证据分级机制。** 需要研究并实现一套可行的证据分级标准（可参考GRADE等成熟框架 [2]，对从公开数据源中检索到的健康信息进行自动化或半自动化的质量评估和等级划分。
    - **设计并实现多轮澄清对话策略。** 系统需能识别用户提问中的模糊性或信息缺失，并主动生成澄清式问题，通过2-3轮对话有效引导用户明确意图，提升回答的精准度[3]。

2. **技术能力目标：**
    - 熟练掌握至少一种主流的**大型语言模型（LLM）API**或**开源模型**的使用方法。
    - 掌握并实践**检索增强生成（Retrieval-Augmented Generation, RAG）**技术架构，理解其在解决模型幻觉和知识更新问题上的优势 [4]。
    - 学习和应用主流RAG框架技术，实现高效的语义检索。
    - 掌握Web应用开发的基本技术栈，能够独立开发和部署一个包含前端用户界面和后端逻辑的完整应用。
    - 培养系统设计、模块化编程和工程实践能力。

3. **研究与创新能力目标：**
    - 培养对前沿技术（如生成式AI）在垂直领域（如医疗健康）应用中的挑战与解决方案进行深入思考的能力。
    - 鼓励在证据分级算法的自动化、澄清问题的智能生成等方面进行探索和创新。
    - 通过撰写可行性分析报告，锻炼技术论证、文档撰写等能力。

第二部分：核心任务与技术要求

为实现上述目标，项目团队需要分阶段完成以下核心开发任务。每个任务模块都包含具体的功能描述和技术要求，旨在确保项目的深度和可评审性。

2.1 任务一：健康知识库的构建与预处理

这是整个系统的基石。你们需要从公开、权威的渠道获取数据，构建一个用于支撑问答的本地知识库。

- **功能描述：**
  a. **数据源选择与获取：** 识别并选择至少2-3个不同类型的公开健康信息源。数据必须能在外网公开访问，不涉及任何个人隐私和公司信息安全。
  b. **数据爬取与清洗：** 开发网络爬虫或利用API获取所选数据源的文本内容。对获取的数据进行清洗，去除HTML标签、广告、无关链接等噪声，提取核心的健康信息文本。
  c. **数据结构化处理：** 将清洗后的非结构化文本进行切分（如按段落、按主题），并为每个文本块（chunk）标注必要的元数据，例如：
    - `source_url`: 原始链接
    - `source_name`: 来源网站名称 (如: WHO, WebMD)
    - `publication_date`: 发布或更新日期
    - `title`: 文章标题
    - `document_type`: 文档类型（如：新闻、指南、研究论文摘要等，若可识别）
  d. **知识库索引：** 利用向量化将处理后的文本块转换为高维向量，并存入知识库中，建立高效的语义检索引擎。

- **技术要求：**
  - **数据源建议：**
    - **权威组织官网：** 世界卫生组织 (WHO)、美国疾病控制与预防中心 (CDC)、中国疾控中心等官方发布的健康指南和事实清单。
    - **医学百科/知识库：** 如WebMD、Mayo Clinic、丁香医生、有来医生等网站的公开文章。
    - **学术文献数据库：** 如PubMed/Medline的摘要部分。可通过API或爬虫获取。

- **公开数据集：** 可利用一些公开的医疗问答数据集作为补充，如Huatuo-26M、MedQuAD等。

2.2 任务二：证据分级机制的设计与实现

这是本项目的核心创新点之一。你们需要设计一套算法或规则，为知识库中的每一条信息（或在检索时动态地）评定一个证据等级。

- **功能描述：**
  a. **分级标准定义：** 参考**GRADE系统** [5]或**牛津证据分级标准** 的核心思想，定义一个简化的、可计算的证据分级模型。例如，可以定义4个等级：
    - **Level 4 (高证据强度):** 来自顶级权威机构（如WHO, CDC）发布的临床指南、系统综述。
    - **Level 3 (中等证据强度):** 来自知名医疗机构（如Mayo Clinic）、大学研究机构发布的专业文章或研究摘要。
    - **Level 2 (低证据强度):** 来自商业化但信誉良好的健康资讯网站、经过认证的医生撰写的科普文章。
    - **Level 1 (极低证据强度/仅供参考):** 来源不明、缺乏作者信息或带有明显商业推广性质的内容。
  b. **自动化分级算法：**
    - **基于来源权威性：** 维护一个权威来源网站列表（白名单），并为其分配基础分值。例如，`who.int` 域名的基础分值为100，`mayoclinic.org` 为80，普通资讯网站为50。
    - **基于内容时效性：** 根据元数据中的发布日期计算时效性得分。例如，近1年内的信息得分较高，超过5年的信息得分相应降低。
    - **基于文档类型（可选）：** 如果能通过关键词（如 "guideline", "systematic review", "meta-analysis"）或页面结构识别出文档类型，可以给予额外加分。
  c. **等级计算与标注：** 综合以上维度，设计一个加权评分公式，计算出每个文本块的最终“证据分数”，并根据分数区间映射到预定义的证据等级（Level 1-4）。此等级需要与文本块一并存储或在检索后动态计算。

- **技术要求：**
  - 需要以文档形式清晰地阐述你们定义的证据分级标准、评分规则和计算公式。
  - 实现一个函数或类，输入为一个文本块及其元数据，输出为其证据分数和等级。
  - 该模块应与知识库紧密集成，确保所有被检索出的信息都能附带其证据等级。

2.3 任务三：多轮澄清对话机制的设计与实现

这是本项目的另一个核心创新点，旨在提升用户体验和回答的准确性。

- **功能描述：**
  a. **澄清触发条件识别：** 系统需要能够判断何时需要发起澄清。触发条件可以包括：
     - **意图模糊性检测：** 用户的问题过于宽泛（如“我该如何保持健康？”）。
     - **实体/术语歧义：** 用户提问中包含多义词（如“苹果”，是指水果还是公司？在健康领域此例较少，但类似歧义可能存在）。
     - **关键信息缺失：** 提问缺少必要的上下文，如症状描述不全、未提及患者年龄、性别等（如“发烧了吃什么药？”）。这可以通过基于规则或简单模型的方法来识别。例如，检测到“吃药”等关键词但未检测到具体药品名或患者群体。
  b. **澄清问题生成：** 当触发澄清条件时，系统应能生成引导性的澄清问题。
     - **基于模板生成：** 设计一系列澄清问题模板，如：“为了更好地回答您，您能具体描述一下[症状]吗？”、“您是指[选项A]还是[选项B]？”。
     - **（进阶）基于LLM生成：** 将用户原始问题和识别出的模糊点一同输入给LLM，让LLM生成一个自然的、引导性的澄清问题。
  c. **对话状态维护：** 系统需要维护一个简单的对话状态（session），记录用户的原始问题和在澄清对话中补充的信息。
  d. **查询重写（Query Rewriting）：** 在获取用户对澄清问题的回答后，系统需要将这些新信息与原始问题结合，形成一个更具体、更明确的新查询，然后用这个新查询去知识库中进行检索。

- **技术要求：**
  - 设计并实现一个对话管理模块（Dialogue Manager），负责判断是否需要澄清、生成澄清问题、维护对话状态以及重写查询。
  - 可以采用简单的状态机模型来管理对话流程。
  - 澄清问题的生成至少要实现基于模板的方式，鼓励尝试使用LLM生成。
  - 整个澄清过程应自然流畅，避免生硬的机器式问答。

2.4 任务四：集成与前端界面开发

将所有模块集成起来，并提供一个用户友好的交互界面。

- **功能描述：**
  a. **后端服务搭建：** 基于Web框架搭建后端服务，封装所有核心逻辑（对话管理、检索、分级、生成）为API接口。
  b. **核心问答流程（RAG流程）：**
     - 接收用户查询。
     - **（澄清循环）** 对话管理器判断是否需要澄清。若需要，则进入澄清流程，直至形成明确查询。
     - 使用明确查询的向量，在知识库中检索Top-K个最相关的文本块。
     - **（证据整合）** 对检索到的K个文本块，获取或计算其证据等级。
     - 将明确查询和带有证据等级的文本块（作为上下文）一同打包，构建一个精细的Prompt。
     - 将Prompt发送给大型语言模型（LLM）。
     - 接收LLM生成的最终回答。
  c. **前端界面开发：** 开发一个简洁的Web聊天界面。
     - 包含一个输入框供用户提问，一个聊天记录显示区域。
     - 当系统给出回答时，界面上必须**清晰地展示**以下信息：
       - LLM生成的自然语言回答。
       - **回答所依据的核心证据片段**（从检索到的文本块中摘录）。
       - **每条证据的来源链接和证据等级（Level 1-4）**。这需要进行可视化设计，比如用不同颜色或图标表示不同等级。
       - 一个明确的免责声明，告知用户本助手仅供参考，不能替代专业医疗建议。

- **技术要求：**
  - **后端：** Java（Spring AI Alibaba）/ Python（Flask/Django/FastAPI）。
  - **前端：** HTML, CSS, JavaScript。可以使用Vue.js, React等现代前端框架，或使用简单的原生JS + AJAX实现。
  - **LLM调用：** 熟练使用国内模型库与模型进行交互。
  - 需要特别注意Prompt Engineering的设计，引导LLM在生成回答时，必须忠实于所提供的、经过分级的上下文信息，并以指定格式输出。

第三部分：系统架构设计

为了帮助你们更好地理解系统各模块之间的关系，这里提供一个可参考的流程描述。

1. **简单查询流程（无需澄清）：**
    a. 用户通过**用户界面(UI)**输入问题（如“高血压患者的饮食建议”）。
    b. 请求发送至**API网关**，再由网关将问题交给**对话管理器(DM)**。
    c. DM分析后认为问题足够清晰，无需澄清。它将问题直接传递给**查询重写器(QR)**（此时QR可能仅做标准化处理）。
    d. **检索模块(RM)**接收到明确的查询，将其向量化后在**知识库**中进行相似度搜索，返回Top-K个相关的文本块。
    e. RM从**元数据库(KB)**中获取这些文本块的元数据（来源、日期等）。
    f. **证据分级器(EG)**接收文本块及其元数据，为每一块计算并标注证据等级。
    g. **回答生成器(GEN)**将用户的原始问题和所有经过分级的证据文本块组合成一个详细的Prompt。
    h. GEN调用**LLM**，LLM基于提供的上下文生成回答。
    i. GEN将LLM的回答、引用的证据来源及等级格式化后，通过API返回给UI进行展示。

2. **复杂查询流程（需要澄清）：**
    a. 用户输入模糊问题（如“我头疼”）。
    b. 请求到达**对话管理器(DM)**，DM通过内置规则或模型判断出该问题缺乏关键信息，需要澄清。
    c. DM触发**回答生成器(GEN)**（或使用模板）生成一个澄清问题（如“为了更好地帮助您，您能描述一下是哪种类型的头痛吗？比如是刺痛、胀痛还是跳痛？”）。
    d. 该问题通过API返回给**UI**。
    e. 用户在UI上回答（如“是跳痛，主要在太阳穴附近”）。
    f. 用户的回答再次通过API发送给DM。DM记录此信息。
    g. DM判断信息是否足够。如果不够，可以继续追问（例如“这种情况持续多久了？”）。
    h. 当信息足够时，DM将原始问题和所有澄清轮次中收集到的信息交给**查询重写器(QR)**。
    i. QR将这些信息整合成一个新的、详细的查询（如“太阳穴附近跳痛的可能原因及缓解方法”）。
    j. 后续流程同“简单查询流程”的步骤4-9。

第四部分：扩展挑战

为了鼓励学有余力的同学进行更深入的探索，本课题设置了以下扩展挑战任务。完成任何一项都将在最终评分中获得显著加分。

1. **高级证据冲突检测与处理：**
   - **挑战描述：** 在检索到的Top-K证据中，可能会出现相互矛盾的信息（例如，两种不同的治疗建议）。设计一种机制来检测这种冲突，尤其是在高证据等级的来源之间。
   - **实现思路：** 利用LLM的语义理解能力，设计一个特定的Prompt，要求其判断给定的若干证据片段之间是否存在事实性冲突。如果检测到冲突，系统可以在最终回答中将冲突点明确展示给用户（如：“关于[某问题]，来源A（证据等级4）认为...，而来源B（证据等级4）则指出...，目前存在不同观点。”），而不是简单地选择其一或模糊处理。这在MEGA-RAG等前沿研究中有所体现。

2. **个性化与用户画像：**
   - **挑战描述：** 在遵循隐私保护原则的前提下，系统可以根据用户的对话历史（在同一会话中）建立一个临时的用户画像（如：提到的症状、年龄段、关注点等），从而使后续的澄清问题和最终回答更具个性化。
   - **实现思路：** 扩展对话管理器，使其能够提取和存储会话中的关键实体和信息。在生成澄清问题或最终答案时，将这些画像信息作为额外的上下文提供给LLM。

3. **可解释性证据分级：**
   - **挑战描述：** 当前的证据分级只给出了一个结果（Level 4），但没有解释原因。实现一个“可解释性”模块，当用户将鼠标悬停在证据等级标签上时，系统能给出一个简短的解释，说明为什么该证据被评定为这个等级。
   - **实现思路：** 在证据分级器计算分数时，记录下各个维度的得分（如“来源权威性得分：95/100，时效性得分：80/100”）。然后根据这些记录，动态生成解释性文本，如：“此条目被评为‘高证据强度’，因为它源自世界卫生组织官方指南，且发布于近期。”

4. **开源模型本地化部署与微调：**

- **挑战描述：** 摆脱对商业LLM API的依赖，选择一个合适的开源大模型（如ChatGLM, Llama系列, Qwen等），在本地进行部署。进一步地，可以尝试使用公开的中文医疗问答数据集对模型进行微调（Fine-tuning），以期提升其在健康领域的表现。
- **实现思路：** 学习模型量化、本地部署（如使用 `vLLM`, `Ollama`）等技术。学习使用 `Hugging Face` 的微调脚本，在特定数据集上对模型进行参数高效微调（如LoRA）。

第五部分：交付成果

1. **可行性分析报告：**
   - **引言：** 课题背景、目标和意义的简要重述。
   - **技术可行性分析：**
     - **数据源分析：** 详细列出计划使用的公开数据源，并评估其数据质量、可访问性和获取难度。
     - **算法与模型分析：** 阐述计划使用的核心技术（RAG, 向量检索, LLM）及其可行性。对证据分级和多轮澄清的初步实现思路进行描述。
     - **技术栈选型：** 确定前后端开发语言、框架、数据库等，并说明选型理由。
   - **进度可行性分析：** 根据项目周期，制定详细的、可行的周度计划（Gantt图）。
   - **团队协作计划：** 明确团队成员的分工。
   - **风险分析与应对策略：** 预测可能遇到的技术难题、数据问题等，并提出备用方案。

2. **项目演示Demo：**
   - **要求：** 必须是可运行的、交互式的Web应用。
   - **演示内容应覆盖：**
     - **场景一（简单查询）：** 输入一个明确的健康问题，展示系统如何给出带有证据来源和等级的回答。
     - **场景二（澄清查询）：** 输入一个模糊的健康问题，展示系统如何通过1-2轮澄清对话，最终给出精准的回答。
     - **证据展示：** 清晰地展示回答所引用的证据片段、来源链接和直观的证据等级标识。
     - **（可选）扩展功能演示。**
   - **代码注释：** 核心模块（特别是证据分级和多轮澄清）有清晰的代码注释。

3. **源代码与文档：**
    - 提交包含所有前后端代码的完整项目工程。
    - 代码应有必要的注释，结构清晰，易于理解。
    - 提供一份详细的 `README.md` 文件，说明项目的部署和运行方法。

第六部分：评价标准

最终成绩将由以下几个部分综合评定：

评价维度	权重	具体考察点
功能完整性 (Functionality)	45%	- 核心功能（RAG问答、证据分级、多轮澄清）是否全部实现。 - 系统是否稳定运行，无明显BUG。 - 用户界面是否友好，信息展示是否清晰。
技术深度与创新性 (Depth & Innovation)	30%	- 证据分级模型设计的合理性与有效性。 - 多轮澄清策略的智能化程度与流畅性。 - RAG流程中Prompt Engineering的精细程度。 - 是否成功挑战了扩展任务。
项目文档质量 (Documentation)	15%	- 可行性分析报告是否全面、深入。 - 系统设计文档、代码注释是否清晰、规范。 - 结题报告是否逻辑严谨，总结到位。
演示 (Demo)	10%	- Demo演示是否流畅，能否清晰展示项目亮点。

		- 团队协作和个人贡献的陈述。

第七部分：参考材料

7.1 核心概念与论文

- **证据分级：**
  - GRADE Working Group. "Grading of Recommendations Assessment, Development and Evaluation (GRADE)." (可搜索相关官方文档和教程，理解其核心思想)
- **检索增强生成 (RAG)：**
  - Lewis, P., et al. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." (RAG的开山之作)
- **多轮对话与澄清：**
  - Aliannejadi, M., et al. "Asking clarifying questions in open-domain information-seeking conversations." (了解澄清问题生成的经典研究)

7.2 推荐工具与框架

- **LLM平台:** Deepseek API，阿里百炼 API
- **工程框架:** Java（Spring AI Alibaba）
- **RAG框架:** RagFlow，LangChain
- **前端框架:** Vue.js, React, or vanilla JavaScript

7.3 推荐数据集

- **中文医疗问答:** Huatuo-26M, CMedQA v2
- **英文医疗问答:** MedQuAD, PubMedQA, BioASQ

第八部分：学术诚信与道德规范

1. **学术诚信：** 本项目所有代码、报告等产出物必须由团队成员独立完成。严禁任何形式的抄袭、剽窃行为。如引用他人成果或代码，必须在文档中明确注明出处。
2. **数据隐私：** 本项目只能使用公开数据，严禁爬取、使用任何涉及个人隐私的数据。
3. **道德规范：** 健康问答是严肃领域。必须在系统界面显著位置添加**免责声明**，明确指出“本系统为课程设计原型，其提供的信息仅供学术研究和参考，不能作为专业的医疗诊断和治疗建议。如有任何健康问题，请务必咨询执业医师。”

附录

1. https://ifis.libguides.com/systematic_reviews/evaluation-of-included-studies
2. https://pmc.ncbi.nlm.nih.gov/articles/PMC12442678/
3. https://openreview.net/pdf/39acbab0a4779243917c1921b60c3708ba5226d5.pdf
4. https://arxiv.org/pdf/2505.01146v3
5. https://www.gradeworkinggroup.org/