南开大学

Nankai University

南　　开　　大　　学

计　算　机　学　院

中软国际实训项目

基于 Neo4j 的医疗知识图谱的智能问答系统

冯思程 杜旖芃 陈冠修 郭昱杰

2112213 2113958 2113068 2111066

年级：2021 级

专业：计算机科学与技术

指导教师：李伟

2023 年 7 月 21 日

目录

一、 项目背景

当今，医疗信息的特点受到科技发展和医疗设备进步的影响，呈现出以下特点：

- **爆炸性增长**：随着科技的不断进步，医疗信息的产生呈现出爆炸性增长的趋势。医学文献、临床数据、医疗诊断和治疗方案等不断涌现，导致医疗信息量急剧增加；
- **信息过载的挑战**：医生和医疗机构面临着信息过载的挑战。处理如此大量的信息，确保其准确性和及时性，成为一个巨大的挑战；
- **庞杂且不断更新的医学知识**：现代医学的知识非常庞杂，涵盖多个专业领域，并且不断更新。医生需要不断学习新知识，以保持专业水平，并将最新的医学进展应用于临床实践；
- **高效获取和整合医疗知识的需求**：医生在做出诊断和治疗决策时，需要考虑大量的患者信息和最新的医学知识。因此，他们迫切需要一种高效的方法来获取、整合和利用医疗知识，以便做出准确的决策。

为了应对这些挑战，医疗界正在积极采用信息技术和人工智能等先进技术。例如，智能医疗信息系统可以帮助医生快速获取和整理患者数据，辅助做出准确的诊断和治疗决策。同时，医学教育也越来越重视持续专业发展，使医生能够不断学习和更新医学知识，以适应不断变化的医疗环境。通过综合运用科技手段和知识管理，医疗信息的处理和应用将变得更加高效和精确。

智慧医疗知识图谱问答系统是一种基于知识图谱的医疗信息管理和智能决策系统。它融合了人工智能、自然语言处理、数据挖掘和图数据库等先进技术，旨在构建一个丰富的、结构化的医疗知识库，并通过对医疗数据的整合、关联和分析，为医疗决策提供支持和指导。

这种系统的核心是智慧医疗关系图谱，它是一个包含丰富医疗知识的图结构数据库。医疗知识图谱以实体（如疾病、药物、治疗方法、症状等）和实体之间的关系为基础，将医学知识进行结构化表示和存储。通过图谱的方式，不仅能够清晰展示各种医疗实体之间的关系，还可以更高效地进行知识的搜索和推理。

智慧医疗知识图谱问答系统的功能主要包括：

- **知识库构建**：通过从医学文献、专业数据库、临床实践等多渠道收集医学知识，并将其整合到医疗知识图谱中，形成一个庞大的知识库；
- **自然语言处理**：系统具备强大的自然语言处理技术，能够理解用户输入的自然语言问题，并将其转化成结构化的查询；
- **问答与推理**：根据用户的查询，系统可以在医疗知识图谱中进行检索，找到相关的医学知识并提供准确的答案。此外，系统还可以进行知识推理，给出一些潜在的、用户未直接询问的相关信息；
- **智能决策支持**：在临床决策过程中，医生可以通过该系统获取有关病情、诊断和治疗方案等方面的信息，帮助他们做出更明智的医疗决策；
- **持续更新**：医学知识不断更新，系统需要持续更新医疗知识图谱，以保持知识库的时效性和准确性。

智慧医疗知识图谱问答系统在医疗领域具有重要的应用价值，它为医生和医疗机构提供了强大的辅助工具，帮助他们更好地管理医疗信息，优化诊疗流程，并提供更加个性化和有效的医疗服务。同时，智慧医疗知识图谱问答系统也有助于促进医学研究和知识传播，推动医疗信息化的发展。

二、 需求分析

(一) 功能性需求

1. 为了实现智慧医疗服务,我们基于 寻医问药网 进行医学数据的爬取,然后将数据存入 Neo4j 数据库中。

2. 前端使用 HTML 搭建一个用户注册与登录的界面，用户可以在界面上注册自己的账号，账号会存入后端的数据库中，通过 Django 框架连接，从而用户能够在登录界面进行登录，进入网页主体。

3. 实现问答系统相关功能模块的设计，根据问题智能查找结果，为项目提供一个接口。采用模糊匹配、子串计数、错字拼音识别、无监督学习（TFIDF、PCA）等算法实现基于关键词、长文本的单类实体检测、三类实体联合检测等功能，根据问题自动智能查找并预测用户可能想查询的结果。

4. 进入网页后，用户在主页能够看见关于网页的简介，下方有一个知识图谱的展示。
   在网页中，为了实现智慧医疗问答服务，有以下功能需求：

   (a) 猜你所想：针对用户输入的模糊病名进行智能匹配，可以选择返回疾病的个数，会按照相似程度的优先级返回给用户准确疾病名字的列表，然后用户可以通过点击链接跳转到病痛显微镜进行更详细信息的查询。

   (b) 病痛显微镜：针对输入的准确疾病名称进行 Neo4j 智能查询，通过查询跳转到该疾病的详细信息页。

   (c) 龙灵问疾：用户可以分别输入，共有 3 个输入块（病名、药品、症状）限制：用空格逗号顿号分开不同实体。通过对信息的智能匹配，按优先级返回用户可能患有的疾病。这里有两个可选项分别可以选择两种不同的输入格式：关键字和文本。

   (d) 龙灵问疾（Plus Beta）：用户可以任意输入一段自己症状的文本，智能的生成出一个疾病列表,也是按照优先级排列输出，然后可以跳转到对应的疾病详情页。这里的可选项是算法，一个是计数算法一个是向量算法（这里可以自己尝试一下验证适合的算法哦）（类似 ChatGPT）。

5. 编写项目演示 PPT、项目需求说明书等文档。

(二) 非功能性需求

1. 要求系统具有较高的性能，由于爬虫速度较慢且网页较多（长达 10000 多个网页），我们需要使用 Python 中的线程池进行并行加速爬虫。

2. 要求系统界面简洁美观，用户操作简单易懂，对于不同的用户群体，界面应尽量符合他们的使用习惯和体验要求。这一部分主要通过自己设置 CSS 进行设计。

3. 要求数据库的读写性能较好，能够支持大量数据的高效查询和存储。我们使用的 Neo4j 数据库的性能相对于关系型数据库有所改进，但我们仍然需要在查询时注意这一点，因为我们的数据库量级极大，非常容易由于查询语句写得不合适，Cypher 语句难以在短时间内完成。因此，我们需要在写查询语句时注意性能的优化，尽量不通过遍历关系（边）来查询。

三、 技术环境

项目实现具体需要的技术环境配置项见表1。

表 1: 技术环境

平台及版本	主要为 Windows 平台 Windows 10 Windows 11 等
硬件环境	可适用于多种硬件环境 AMD Ryzen 7 7840HS 等
适用分辨率	可适用多种分辨率 2560×1600 2560×1440 等
开发 IDE	VS Code PyCharm
使用语言	Python Html/CSS Cypher
类库支持	Python 中的多个第三方库 py2neo、os、jieba、numpy、pandas 等
数据库	Neo4j 图数据库 MySQL 关系型数据库
中间件服务器	Django Neo4j
浏览器 (版本)	下述浏览器版本均已通过测试: Microsoft Edge 版本 114.0.1823.82（正式版本）(64 位) 版本 114.0.1823.86（正式版本）(64 位) 联想浏览器 版本 8.0.1.5162（正式版本）(64 位)
三方插件	Echarts

四、 实现规划

(一) 项目分工

项目实现具体任务分工见表2。

(二) 进度总览

项目实现具体如下:

- **2023-7-12**: 最终项目开始启动, 从下午开始。这一天的工作目标主要是大家各自查阅相关资料。

表 2: 项目分工

功能名称	实现效果	负责人
智能问答系统（含模糊匹配和无监督学习扩展功能）	实现基于关键词、长文本的单输入、三输入等多功能切换，根据问题自动智能查找并预测用户可能想查询的结果，帮助用户了解疾病的详细信息或根据目前症状、服药情况、既往病史推断用户目前可能得患病等	郭昱杰、陈冠修
高量级的数据爬取与预处理以及并行加速模块（并行加速为扩展功能）	在目标网站上爬取到高量级数据并进行分词等预处理实现命名实体识别提取，解决多元混乱的网页结构爬取困难，利用并行思想对上万级别的页面爬取进程进行加速，提高数据爬取效率	冯思程、杜旖芃
构建 Neo4j 图数据库	设计爬取数据 csv 存储模式和 Neo4j 图数据库结构，根据爬取的大量级数据提取关系和节点构建出 Neo4j 数据库	郭昱杰、陈冠修
Django 框架开发	优美流畅的前端 web 界面，使用了 ECharts 来实现 Neo4j 图的可视化	冯思程、杜旖芃
基于 EChars 的数据可视化	利用学习的 ECharts 工具进行数据的动态可视化展示，提升用户使用体验感。	冯思程、杜旖芃
Neo4j 图数据库的操作实现	实现根据用户的提问智能进行图数据库的相关操作	郭昱杰、陈冠修

表 3: 神龙无敌队实现项目功能及分工

- **2023-7-13**：继续收集资料，针对要实现的功能展开探讨，组长与组员在腾讯会议上召开会议进行讨论。并对项目进程时间控制进行商讨初步指定计划。
- **2023-7-14**：查找相关资料，确定小组分工，具体分工为郭昱杰、陈冠修负责设计导出数据的 csv 格式，将数据导入 Neo4j；杜旖芃，冯思程实现数据爬取以及前端页面实现。**里程碑 1**：确定出我们选择的领域在智能医疗领域，确定我们主要要实现的功能是智能问答系统。
- **2023-7-15**：设计好爬取数据的 csv 格式，并设计 Neo4j 图数据库的结构。
- **2023-7-16**：针对最后的项目流程做出规划，根据每个人的意见进行调整。之后每个小组进行自己负责部分的讨论，确定具体讨论结果。
- **2023-7-17**：杜旖芃、冯思程完成数据的爬取，封装到字典中；郭昱杰、陈冠修完成输出 csv 代码，以及完成数据的清晰和 Neoj 构建。**里程碑 2**：实现数据爬取以及数据预处理，并根据爬取的数据构建出 Neo4j 数据库（爬取与处理复杂，数据项量级在二十万左右）。
- **2023-7-18**：郭昱杰、陈冠修完成了后端基础功能的实现。冯思程、杜旖芃网页界面。**里程碑 3**：基于预定义问答模板实现智能问答系统（适当扩展）

- **2023-7-19**：郭昱杰、陈冠修完成了后端扩展功能的算法和实现。冯思程、杜旖芃实现了前端网页界面的实现。**里程碑 4**：实现前端（JS 以及 Echarts），实现前端美化；后端扩展功能实现。
- **2023-7-20**：小组一起完成前后端工作的交接；之后一起合作完成提交项目所需文档。
- **2023-7-21**：项目答辩与技术能力测评

五、 实现方法

(一) 数据获取

1. 来源选定

首先我们利用多个搜索引擎进行相关资料的搜索，寻找我们目标的爬取网站，再经过数据对比后，为了能够获得性能更好的问答模型，我们选定了寻医问药网站进行数据爬取，因为这个网站的数据量级十分庞大，而且数据具有一定的格式，另外经过检测发现，爬取机制并不强，于是我们最终选定寻医问药网站进行数据爬取。

2. 爬取加速

由于爬取的数据量级非常大，而且我们的数据爬取方式多界面深层爬取，所以速度方面是非常慢的，估算约需要 10 余小时，于是我们开发了并行加速模块，我们尝试了多种并行加速的方法，例如像线程池、异步并行加速、分布式爬取等等方法。经过测试后，综合考虑速度、安全性，我们最后选定了利用线程池方法进行加速，并经过对比测试，选定调用线程数为 16 线程。经过并行加速后，爬取过程大概加速了 10 倍，只需要 1 个小时多既可以完成全部的数据爬取。

(二) 数据规范与导入 Neo4j

- 设计 csv 的结构格式，将信息分为疾病，节点，关系三个 csv 文件。建立从爬取数据到输出 csv 的代码部分。在这一部分中，也是先对一些不合格的节点和关系数据的去除。
- 通过 py2neo 实现 Python 与 Neo4j 的连接，之后根据信息拓扑依赖关系排序先创建节点，后创建关系。
- 在导入 Neo4j 的过程中进行数据的筛选: 判断是否已经存在要创建的节点实现去重，在这里要注意是否有多出的属性，如果有多出的属性，进行添加。

(三) 后端功能

我们采用模糊匹配、子串计数、错字拼音识别、无监督学习（TFIDF、PCA）等算法实现基于关键词、长文本的单类实体检测、三类实体联合检测等功能，根据问题自动智能查找并预测用户可能想查询的结果，帮助用户了解疾病的详细信息或根据目前症状、服药情况、既往病史推断用户目前可能得患病等，用户自主切换能多种功能选择最适合自己的智能解决方案。

1. 模糊疾病的查询

该部分实现猜你所想和病痛显微镜两个子功能。从猜你所想板块，输入一个模糊的病名，然后在界面上给出用户匹配的病名结果，之后用户可以点击某个病名，跳转到病痛显微镜来获取该疾病的具体信息。

病痛显微镜 功能：输入一个准确的病名，完成对病名信息（包括简介，患病、治愈比例，传染方式，常见药物等信息）的获取。

实现方式：通过 py2neo 连接到数据库中，对数据库的信息进行查询，获取所有的信息。

猜你所想（扩展板块） 功能：输入一个比较模糊的病名，可以给出匹配程度最高的几个病名。

实现方式: 最初设计是通过最小编辑代价来实现模糊病名和准确病名的匹配，但这种匹配方式对于错字和空字符匹配的代价需要不断调整，在最后实现出来的结果并不理想。

考虑到上述方法的缺点, 我们采用分词, 然后进行匹配。这里分词是对比 jieba 分词和 thulac 分词后，选取 jieba 作为分词工具。之后遍历分词后的结果，每个结果进行匹配，这里的匹配规则为：如果结果为某个疾病名字字串，则获取该结果长度的分数。

之后考虑到需要允许用户有错别字的输入，因此考虑的是音相同也进行匹配，但匹配分数会有送降低。这里采用 pypinyin 包来完成从汉字到拼音的转化。之后就是判断分词结果的拼音是否是要匹配病名的拼音的字串。

上述方法具有非常好的性能，同时也允许用户能够输入错别字（音相近的错别字）。经过测试，在输入错别字的情况下，依旧能匹配出理想的结果。之后优化的方向是增加处理形相近的错别字的处理，这样能够容忍用户输入音相近或者形相近的错别字。

2. 单类实体关联预测

龙灵问疾 plus 功能：通过输入一段症状描述的文本，分词后提取其中症状关键词，根据关键词利用不同算法预测相关的疾病。

算法 1：关联症状计数：对于提取的症状关键词，累计其关联的疾病的分值，选取分值最高的若干疾病作为预测结果。

算法 2：特征向量相似度（扩展）：预训练将数据库中的疾病基于其关联的症状形成对应的描述文本，将这些文本使用 TFIDF 向量化表征对应疾病，由于向量维度较大，使用 PCA 数据降维和特征浓缩；在用户查询时，将提取的症状关键词形成的文本利用已训练的 TFIDF 和 PCA 模型转化为特征向量，将该特征向量与预训练的特征向量矩阵求取余弦相似度和欧几里得相似度并将二者加权得到最终估计相似度，选取相似度最高的若干疾病作为预测结果。

3. 三类实体联合预测

龙灵问疾 功能：通过不同模式分别输入既往病史、近期症状、近期服药三类信息，模糊匹配对应实体形成提取列表，分别计算与相关疾病的关联度，将三类关联度根据信息量加权得到最终关联度，并筛选最终关联度最高的若干疾病作为预测。

模式 1：关键词：需要用户输入相关关键词，准确度相对较高。

模式 2：文本（扩展）：在关键词基础上可以输入文本，输入更加自由。这里采用 NLP 来进行处理。

（四） 前后端交互

前端用户数据输入获取： 在 html 中的 script 的标签进行前端数据的获取，这里利用的方法是首先对默认表单提交的行为进行阻止，然后利用 fetch 方法以 POST 的形式去获取到前端用户传送过来的数据。并暂存到一个 message 中。

数据发送： 当在利用 fetch 方法获取到用户输入的数据后，我对数据进行以 json 格式的封装，然后将 json 格式的数据发送到后端。

接受数据并返回结果： 在后端收到数据后，就利用已经接好的 API 调用功能模块，返回一个结果，这里调用功能模块是通过在视图文件中进行 import 引入定义好的实例和其中的方法实现的。

发送结果回到前端： 在获取到返回结果后，我们需要对返回结果进行一个处理，因为这里我是按照 json 响应格式去返回一个结果的，所以要将结果都通过代码转化成列表进行返回。

前端结果的可视化： 前端获取到返回结果后，会对 json 数据格式进行相应的解析，然后调用可视化函数进行前端界面的可视化展示。

六、 成果展示

猜你所想功能交互见图1，用户输入“干冒可搜”实际是“感冒咳嗽”的同音错字，但猜你所想功能也能智能识别并检索出高度相关的疾病，如若干不同类型的伴随咳嗽症状的感冒。

图 1: 猜你所想

病痛显微镜功能交互见图2，用户直接查询百日咳，页面上部显示疾病的简介、治疗期等信息，下部左侧显示成因、预防措施、详细信息等内容，可以选择展开收起，下部右侧显示 Neo4j 图数据库中与该疾病相关的所有信息，用户可以通过点击上侧的对应色块进行对应节点的隐藏或显示的切换。

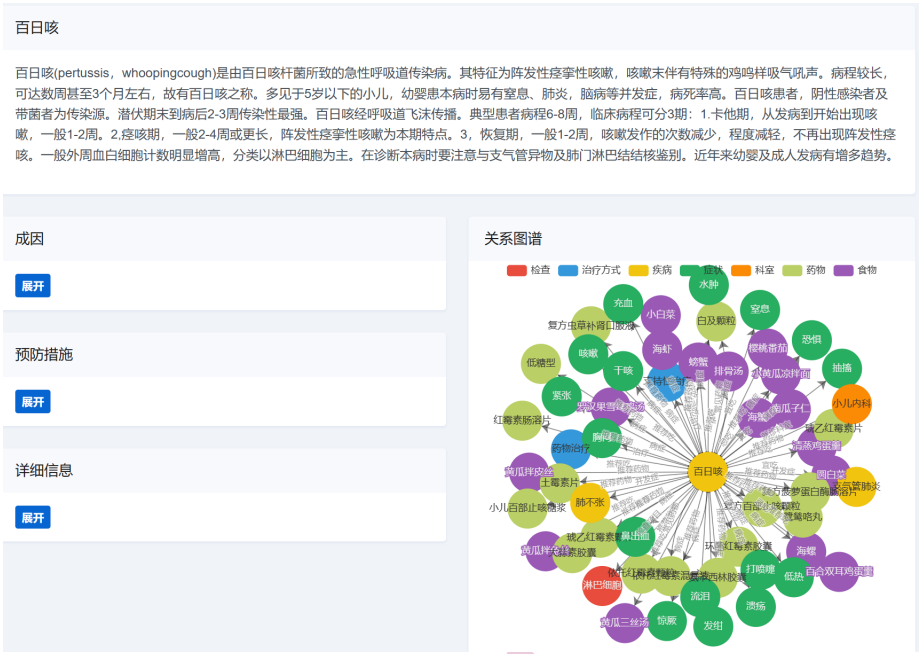

图 2: 病痛显微镜

龙灵问疾 Plus 功能交互见图3，用户分别输入既往病史、近期症状、近期服药的信息，可以在右下角发送键的左侧按钮选择“关键词”和“文本”模式，根据用户提供的信息提取相关要点，对库中疾病进行关联度分析，返回相关度最高的若干疾病，可以点击相关疾病名跳转至对应的病痛显微镜界面了解详细信息。

图 3: 龙灵问疾 Plus：三类实体关联预测

龙灵问疾功能交互见图4，用户输入对于当前症状的文本描述，可以在右下角发送键的左侧按钮选择“计数算法”和“向量算法”，与上文龙灵问疾类似分别对库中疾病进行关联度分析，返回相关度最高的若干疾病，可以点击相关疾病名跳转至对应的病痛显微镜界面了解详细信息。

图 4: 龙灵问疾：单类实体联检预测

七、 评价总结

优势创新

- **大量级数据**：由于问答系统的实现准确度和完善度严重依赖于数据的量级，ChatGPT 的成功也证明了大量级数据的重要性，于是我们的项目创新点之一就是爬取到了大量级的数据，去重处理后约为 30w 级别。
- **ECharts 实现的动态前端关系图**：这一个创新点大幅度的增强了对于用户的便利性和个性化，提升了用户与系统交互的友好性。
- **人性化的匹配方法**：利用拼音自动修正用户描述中可能得错别字，提升模糊匹配的鲁棒性和兼容性，为用户提供更自由的输入途径。
- **无监督学习**：利用 TFIDF 逆文本频率将用户描述向量化、数据化，再利用 PCA 进行数据降维和特征浓缩，使得疾病的匹配可以量化评估，提高程序可扩展性。

不足

- **数据来源精确度不足**：由于网站较为不规范、且其数据零散，最后形成的医学知识图谱数据有一定的偏差。
- **语义解析还可以进一步提升**：我们目前使用了机器学习来优化模型的语义分析能力，但是我们没有应用到神经网络等大数据训练下的模型，所以效果是有限的。
- **病理分析的细粒度不足**：在医学上，知识图谱还包括基因与蛋白质相关的极为复杂的病理分析，我们的数据不是非常全面的，这部分我们也是有一定的缺陷性。

总结 在这次小组项目的开发过程中，我们不仅学到了技术知识，还深刻认识到团队合作的重要性。通过密切的协作和相互学习，我们解决了一个个技术难题，攻克了一个个困难，共同完成了这个具有挑战性的项目。

在前端开发中，我们掌握了 HTML、CSS 和 JavaScript 等技术，学会了设计精美的用户界面，并实现了丰富的用户交互功能。同时，我们还学习了前端框架和库，如 React 和 Vue.js，这为我们开发更加复杂和高效的应用打下了基础。

在后端开发方面，我们熟悉了 Django 框架，掌握了数据库设计和 API 接口的开发。我们学会了如何处理用户请求，进行数据处理和业务逻辑，最终返回用户所需的结果。同时，我们还学习了如何优化后端性能，提高系统的稳定性和响应速度。

在 NLP 自然语言处理方面，我们深入了解了文本处理、语义分析和情感识别等技术。我们探索了如何构建问答系统，通过处理用户的自然语言输入，快速找到匹配的答案，并向用户提供准确的回复。

除了技术上的成长，我们还提升了团队协作和沟通能力。我们学会了倾听他人意见，尊重不同观点，并通过有效的沟通和讨论达成共识。在解决问题的过程中，我们展现了团结合作的精神，相互支持和帮助，共同克服了困难。

这次项目开发让我们对计算机科学和人工智能充满了热情和兴趣。我们期待将这些宝贵的经验和知识继续发扬光大，在未来的学习和实践中，不断追求创新和进步。我们相信，通过不断学习和探索，我们能够为科技发展和社会进步贡献自己的力量。让我们一起走向更加美好的未来！

八、 未来展望

- 在我们项目的基础上，我们还有许多能够进一步改进的地方，它们包括：
  - 更精确化的数据来源：由于网站较为不规范、且其数据零散，最后形成的医学知识图谱数据有一定的偏差，我们今后可以通过寻找更为专业的网站，获取得到更精确的数据。此外，网站爬虫使用的 NLP 处理部分也可使用神经网络进行，牺牲一定的爬虫效率的同时，优化数据的精确程度。
  - 更智能化的语义解析：我们目前使用了机器学习来优化模型的语义分析能力，但是 NLP 处理依旧可利用深度学习和大规模语料库的训练，对症状匹配算法进行优化，从而更准确地理解用户的意图，提供更精准和个性化的回答。
  - 更细粒度的病理分析：在医学上，知识图谱还包括基因与蛋白质相关的极为复杂的病理分析，我们可以获取这些数据，利用病理分析帮助医生确定疾病的类型、分级、分期和预后等信息，更广泛地应用于临床诊断中。
- 此外，我们还可以使用科技前沿——液体神经网络，在网页运行的同时进行学习，美国麻省理工学院（MIT）的研究人员开发了一种能在工作状态下学习的神经网络（而非仅在训练过程中学习）。这些被称为“液体神经网络”的灵活算法改变了基本方程式，可以应对不断输入的新数据。[1]
- 对应在我们的项目中，我们可以在与用户的交互过程中进行神经网络的训练，这样，我们的智能问答系统的回答更符合用户的需求。
- 我们由于项目开发的时间有限，未能全部完成这些功能。在今后，我们还可以对我们当前的知识图谱进行进一步的扩充与完善，在学术领域中取得更好的成绩。
- 总之，这次小组项目的开发不仅为我们带来了技术上的成长，也加深了我们对前、后端知识和 NLP 自然语言处理技巧的理解与熟练程度。我们期待将这些宝贵的经验和兴趣在未来的学习与实践中继续发扬光大，为更多创新和进步贡献自己的力量。

[1] https://techxplore.com/news/2021-01-liquid-machine-learning-conditions.html