NLU - DL Midterm Proposal

魏知原　2300012875

一、项目题目：《“智览”——基于大模型的智能信息聚合与分析系统》

二、项目背景

当前互联网信息爆炸，各类媒体渠道信息质量参差不齐，人们每天面临海量的新闻、文章和资讯，但人工浏览、筛选和分析这些信息既耗时又低效，难以快速获取高质量、全面可靠的信息摘要。根据统计，普通用户每天需要花费1-2 小时浏览各类新闻和资讯，但其中真正有价值的信息占比不足20%。

传统的信息聚合方法主要依赖关键词匹配、简单的文本摘要算法（如TF-IDF、TextRank等），往往缺乏对信息质量的深度判断和跨源的综合分析能力。这些方法存在以下局限性:

- 无法理解深层语义，容易遗漏重要但表述方式特殊的信息;
- 缺乏对信息来源可靠性的判断能力;
- 生成的摘要缺乏逻辑性和可读性;
- 无法根据用户偏好定制个性化报告。

随着大语言模型（如Qwen、GPT 系列、Claude 等）的发展，其强大的语义理解、信息提取和生成能力为智能化信息处理提供了新的解决方案。大模型在信息处理领域已有多个成功应用案例，如微软的Copilot News、Google 的AI Overview 等。

本项目旨在构建一个自动化的信息分析系统”智览”，通过调用大模型API、网页搜索引擎、数据可视化等技术，自动从网络中收集指定主题和时间范围内的信息（新闻、文章、论文等），并智能生成高质量、信息全面且可靠的分析报告。相比传统方法，大模型能够:

- 理解复杂的语义关系，准确判断信息的相关性和重要性;
- 自动进行多源信息的交叉验证，识别虚假或低质量信息，提升信息可靠性;
- 生成结构化、逻辑清晰、易读的分析报告，辅以数据可视化;
- 支持多模态内容生成（文本+ 图表+ 配图），提升报告的专业性和可读性;
- 根据用户配置灵活调整报告风格和内容侧重点。

三、项目方案

本项目计划实现以下自动化流程（Pipeline）：

Step1: **配置输入与解析**:

- 用户通过YAML/JSON 配置文件设定关注的主题领域（如"人工智能"、"金融市场"、"国际政治"等）；
- 指定时间范围（今日、近三天、近一周、自定义日期区间）；
- 选择报告风格（简明新闻风格、深度分析风格、学术刊物风格等）；
- 设定信息源偏好（官方媒体、学术期刊、社交媒体等）。

Step2: **多源信息采集**:

- 调用搜索引擎API（如Bing Search API、SerpAPI）进行通用信息检索;
- 对特定领域调用专业数据源API（如arXiv API 用于学术论文、NewsAPI 用于新闻）；
- 获取信息的标题、来源、发布时间、摘要、URL 等元数据;
- 对采集到的原始数据进行去重和初步清洗。

Step3: **智能信息筛选与分析**:

- 设计专门的Prompt 模板，将采集到的内容分批输入大模型（如Qwen API）；
- 让模型根据相关性、重要性、时效性、可靠性等多个维度对信息进行评分;
- 过滤低质量、重复或不相关的信息;
- 提取每条信息的关键要点、核心观点和潜在影响;
- 识别信息间的关联关系（如因果关系、时间序列关系等）。

Step4: **数据统计与可视化**:

- 使用matplotlib/seaborn 对筛选后的信息进行统计分析;
- 生成热点话题词云图、时间趋势折线图、信息源分布饼图等可视化图表;
- 将图表保存至assets 文件夹，命名规范化以便后续引用;
- 记录统计数据（如总信息量、筛选率、热点关键词等）。

Step5: **智能报告生成**:

- 设计报告生成Prompt，将分析结果和统计信息输入大模型;
- 生成结构化文本报告，包含：摘要、重点新闻解读、趋势分析、相关建议等章节;
- 根据配置的报告风格调整语言风格和详略程度;
- 将报告初稿保存为report.txt 或Markdown 格式。

Step6: **多模态配图生成**:

- 调用多模态大模型API（如DALL-E、Midjourney API 或开源替代方案）；

- 根据报告的主题和核心内容生成相关配图（如主题插图、概念示意图等）；
- 对生成的图片进行质量检查和筛选；
- 将合格的配图保存至assets 文件夹。

Step7: **LaTeX 自动排版与编译**:

- 使用Jinja2 等模板引擎将报告内容、数据图表、生成配图整合到预设的LaTeX 模板中；
- 自动处理特殊字符转义、图片路径引用等细节；
- 调用pdflatex 或xelatex 编译生成格式美观的PDF 报告；
- 若编译失败，解析错误信息并尝试自动修复（如调整图片大小、修正格式错误）。

Step8: **错误处理与日志管理**:

- 在整个流程中使用try-except 机制捕获异常；
- 对API 调用失败实现自动重试机制（指数退避策略）；
- 详细记录每个步骤的执行状态、耗时、错误信息至日志文件；
- 若关键步骤失败超过阈值，发送告警通知；
- 支持断点续传，避免因单点失败导致整个流程重新执行。

四、计划时间点

- **2025.11.07** 提交proposal，完成项目初步方案设计和技术调研，确定使用的API和工具栈。
- **2025.11.14** 完成配置模块和信息采集模块，实现基本的搜索API 调用功能，能够从至少两个信息源获取数据。
- **2025.11.21** 实现信息筛选与分析模块，完成Prompt 设计和大模型API 集成，能够对信息进行智能评分和筛选。
- **2025.11.28** 实现数据可视化模块和报告生成模块，完成文本报告生成，能够输出基本的分析报告。
- **2025.12.05** 实现多模态配图生成和LaTeX 自动排版编译功能，能够生成完整的PDF 报告。
- **2025.12.12** 完善错误检测、日志记录功能，进行系统测试和优化，提升系统稳定性和报告质量。
- **2025.12.19** 完成项目报告和演示PPT，整理项目文档和代码仓库。