序号：14

评分标准	评分
（1）真实性和可用性：数据是真实可用的、具有可用性；	
（2）感知性和可行性：易于感知，方案可行；	
（3）艺术性和设计性：布局合理、数据展示美观；	
（4）原创性和有价值：实用性、内容新颖详实；	
（5）论文内容合理、书写工整。	
总分	

《鲁迅全集》导览可视化与人机交互（艺）

学号：23121677 姓名：范舒舰

摘要：本项目面向《鲁迅全集》构建了一体化“阅读—分析—可视化”的数字导览系统。前端基于 Vue 3 与组合式 API，提供三大模块：概览导航（鲁迅生平年谱检索与时间轴可视化、主要作品时间线、笔名汇编与版本背景）；书籍导览（epub.js 分页渲染，按“卷/册/章”分级目录，上一/下一章与分页跳转，字号与深浅色主题切换）；数据概览（词云、词/字频 Top-K 统计的 ECharts 交互柱状图，人物关系力导图并内置样式面板与原始 JSON 抽屉）。App 端支持可视主题编辑（渐变三色与角度、圆角模式）并持久化到 localStorage。后端采用 FastAPI，提供 /toc、/books、/wordcloud、/freq/word、/freq/char、/character/relationships、/chat/ask 等接口：利用 ebooklib+BeautifulSoup 解析 EPUB；以 HanziNLP 与 wordcloud 生成词云；人物关系抽取优先读取服务器缓存，必要时调用智谱 GLM 输出严格 JSON 并完整落盘日志；对话接口注入章节上下文以支持段落解读与问答。系统通过容器尺寸探测、渲染复用与自适应重绘保证性能与稳健性，结合开放 CORS 与静态资源挂载，便于部署与前后端联调。该平台在不改动底本风貌的前提下，实现对《鲁迅全集》的多视角浏览、统计与语义关联。

关键词：Vue 3，FastAPI，EPUB（epub.js），ECharts，词云，词/字频统计，人物关系抽取，LLM（智谱 GLM），HanziNLP，数据可视化。

1 引言

本系统是一个以 Vue3 与 ECharts 为核心构建的《鲁迅全集》深度数字导览与可视化平台。其整体架构包含：关系图页通过力导向图直观呈现人物关系，点击节点可弹出侧边栏查看详细出处并实现跳转；阅读器页采用 epub.js 渲染章节阅读，与统计图表深度联动，点击图表条目或人物节点即可定位至原文段落。所有分析均建立在真实、透明的数据基础上：原始文本直接来源于 luxun.epub 文件，经由 ebooklib 解包和 BeautifulSoup4 清洗，去除了目录、脚注等噪声。后端通过 FastAPI 提供了一系列固定且清晰的 RESTful 接口（如 /toc, /freq/word, /relationships 等），为前端提供了稳定可靠的数据服务。

在用户体验层面，系统通过柱状图、词云等图表清晰呈现字词频次，所有渲染均在章节/段落级数据量下稳定运行。界面布局主次层级分明：左侧为主工作区，右侧为信息与筛选侧栏，阅读、统计与关系图功能区保持了统一的工具栏与操作逻辑。整套界面采用统一的色板与字体，留白与间距经过调优，并支持浅色/深色主题自动适配。用户可在单一屏幕内完成“阅读文本→查看统计→分析关系”的完整分析闭环，覆盖了词/字频、词云、人物共现、时间-体裁分布等多维视角，且所有视图均可联动与章节定位。系统内置了章节级对话式检索原型，并利用轻量级 SQLite 数据库进行缓存，确保了所有交互操作都能迅速响应。该系统后端基于 Python 3.10+与 FastAPI，前端基于 Vue3、ECharts 与 epub.js，可在 Windows/macOS/Linux 等多种操作系统上通过主流的 Chrome、Edge、Firefox 浏览器流畅运行。本地开发可使用 uvicorn server.main:app --reload 与 npm run dev/vite 启动。源码与数据代码位于 server/ 与 web/ 两端；EPUB 放置在 server/books/luxun.epub。

2 《鲁迅全集》可视化的模块、设计和实现

本系统设计紧密结合课程内容：遵循第 4 章的数据可视化流程、第 9 章的文本可视化方法，以及第 13 章的评估思路。在设计上，从数据获取（EPUB 文本、统计数据）到可视化呈现，再到用户交互反馈，完整地体现了数据可视化的一般流程。首先加载《鲁迅全集》数据（epub 格式电子书内容），随后进行数据预处理（如词频统计、人物关系计算等，依靠前端交互实现接口的调用），然后由前端组件渲染词云、柱状图、力导图和时间轴等可视化图形，最终通过交互功能（筛选、缩放、点击等）完成用户体验，并在交互过程中不断更新和评估可视化效果。

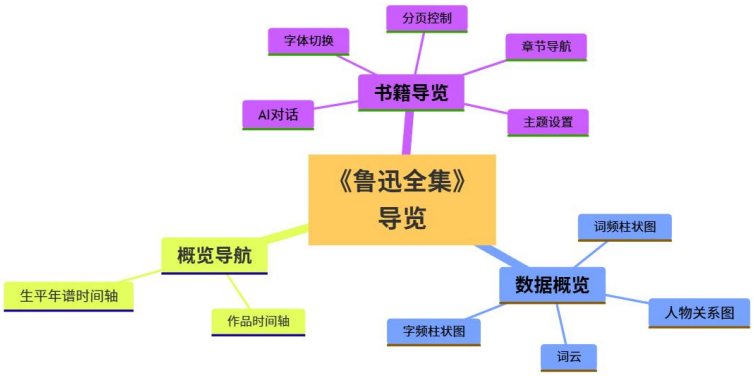


图 1. 系统架构思维导图

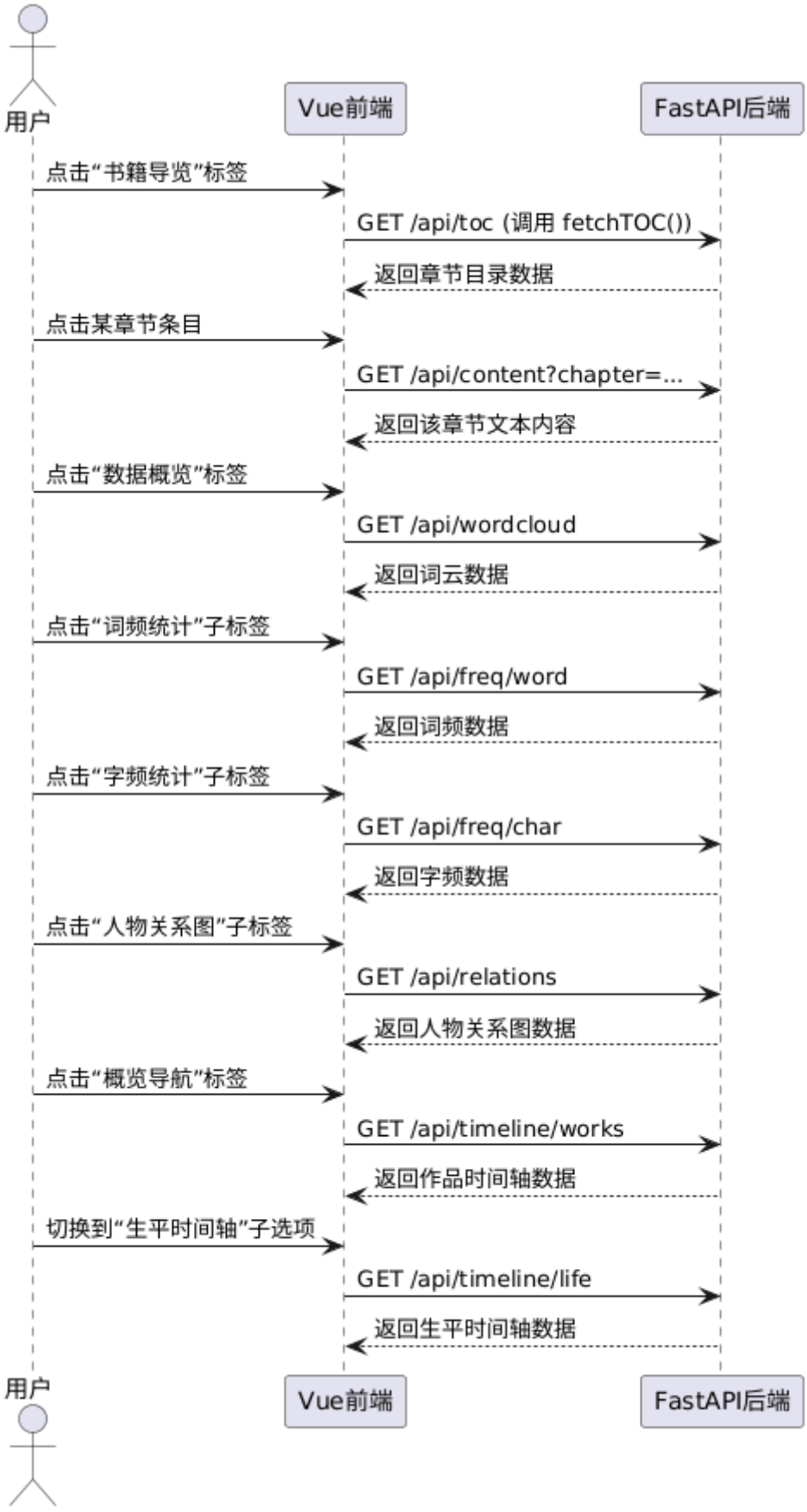


图 2. 整体系统请求流程（基于 Vue+FastAPI）

2.1 可视化一般流程

本项目的可视化流程涵盖数据加载、预处理、渲染以及交互反馈等阶段：首先加载系统界面并初始化各组件；然后从后端接口获取 EPUB 文本及各类统计数据；经过必要的预处理（统计词频、构建关系网络、整理时间轴事件等），调用前端可视化库（如 ECharts 或 D3）渲染词云、柱状图、力导图和时间轴等；在渲染完成后，系统持续监听用户交互事件（如点击、筛选、缩放等），并根据需要触发后续的数据更新和重新渲染过程，实现动态交互。

2.2 系统界面设计

系统界面主要由顶部导航栏、左侧功能菜单、右侧内容显示区顶部导航栏和右侧内容显示区构成。顶部导航栏存在主题色选择、菜单分为“书籍阅读”、“数据概览”和“概览导航”三大模块，用户点击对应模块后，右侧显示相应的可视化或阅读组件。下图为系统界面示意图：

图 3. 页面示意图

2.3 各可视化模块详述

书籍导览

该模块基于 EPUB 阅读器组件，实现《鲁迅全集》文本的章节导航、分页显示、字体样式切换和主题设置等功能。用户可以在侧边目录中点击章节标题进行跳转，或者使用“上一页/下一页”按钮翻页；同时支持调整字体大小和字体类型，以及切换亮/暗色主题以提升阅读体验。

其中常用方法包括 loadChapterContent(chapterIndex)（加载指定章节内容）、nextPage() 和 prevPage()（分页翻页）、changeFontSize(size)（调整字体大小）、toggleTheme(isDark)（切换主题）。组件中可能使用 refs 访问底层阅读器实例（如 this.$refs.reader）调用对应方法。

书籍内容通常打包在前端应用中或通过静态资源提供，如已实现则无动态接口；若使用后端获取章节内容，可设计接口如 /api/epub/chapter/{id}。

页面加载时，组件调用阅读器 API 初始化并加载第一章内容。用户点击目录中的章节后，goToChapter(index) 被触发，依次调用 loadChapterContent() 更新内容渲染。用户点击翻页按钮时，分别调用 nextPage()/prevPage()，内部驱动阅读器组件展示相应页码。调整字体或主题时，触发对应方法修改样式并重新渲染页面。各功能调用逻辑清晰，均由组件方法串联调度图文内容和样式变化。

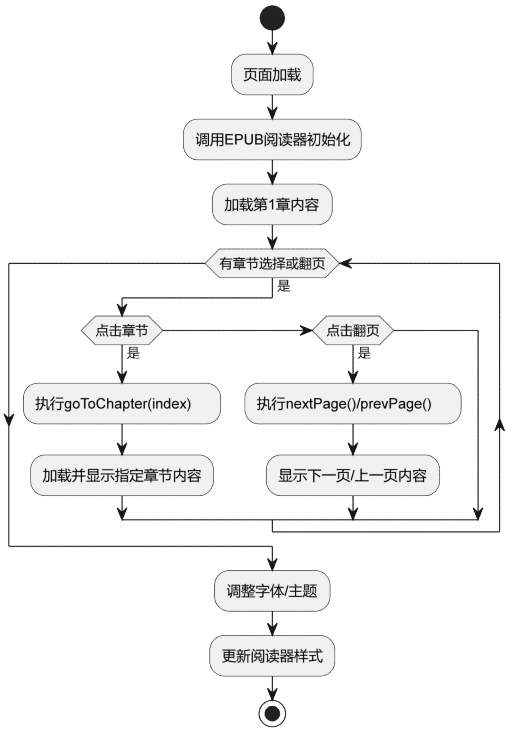


图 4. 书籍导览模块流程图

AI 对话

该模块嵌入于 RealEpubViewer.vue，为当前阅读的《鲁迅全集》章节提供基于章节上下文的问答能力。读者在页面右侧/下方的对话输入区键入问题后，前端会将当前章节的 href_key 与用户问题一并发送至后端，返回的答案与会话标识用于连续对话与消息列表渲染。

sendChat()：向后端发起对话请求（POST /chat/ask，携带 href_key、question、session_id），接收答案并追加到消息区。scrollChatToBottom()：在 nextTick 后将消息列表滚动至底部，保证新回复即时可见。响应包含答案文本与（可能更新的）session_id，前端据此维护同一会话的多轮问答。

页面加载时，RealEpubViewer.vue 完成 fetchTOC() → ensureContainerSized() → initEpubReader()，此后对话区即可使用。当用户在对话输入框提交问题后，触发 sendChat()：根据当前阅读位置携带 href_key，调用 /chat/ask 获取答案，将问答对追加到消息列表。若用户更换章节（如通过 prevChapter() / nextChapter() 或目录跳转），后续 sendChat() 会自动携带新的 href_key，使回答围绕新章节展开。

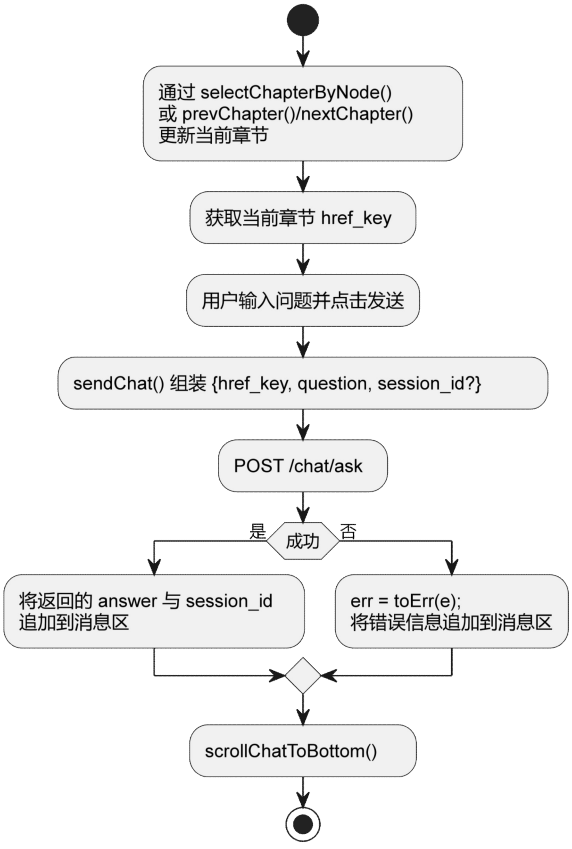


图 5. AI 对话模块流程图

词云模块

词云位于右侧 Tab（wordcloud）。目录数据通过 fetchTOC() 建立“卷/册/章”层级；当用户在左侧树选择章节时，selectChapterByNode() 会设置当前章节的 href_key 与标题，并触发右侧 Tab 的加载流程。

当用户切换到词云页签时，switchRightTab('wordcloud') 调用 loadWordCloud()。该方法仅基于当前章节的 href_key 拼出图片地址 /wordcloud?href_key=...，驱动 <img> 加载并展示词云。前端不做本地词频计算，不依赖图表实例，无需销毁/重建图表。

词云图片由 GET /wordcloud 生成：服务端会依据 href_key 提取章节文本（extract_text_by_href_key()），统计词频（build_word_frequencies()，优先 HanziNLP.word_freq，失败回退 jieba+Counter），并返回 PNG。前端仅负责设置 <img src> 与切换章节/Tab 时的链接更新。

词/字频柱状图模块分别统计文本中词语和单字出现频率，并以柱状图形式展示。仅与词云模块的可视化流程不同，其余 api 调用流程基本一致。

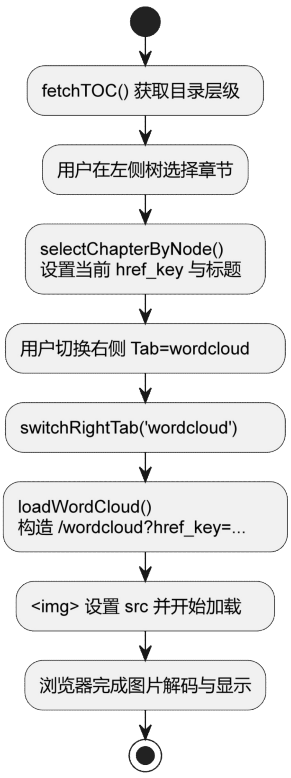


图 6. 词云模块流程图

人物关系模块

模块定位与入口。 人物关系可视化位于右侧 Tab 的 relations 分页，和“词云”“统计”并列。用户在左侧目录树选择章节后，selectChapterByNode(会设置当前 href_key 与标题；当用户切换到 relations 页签时，switchRightTab('relations') 调用 loadRelations() 进入关系数据的加载与渲染流程。

数据获取与规整。 loadRelations() 优先尝试缓存；必要时发起 GET /character/relationships（由 fetchRelationships(preferCache=true) 执行）。返回结果可能是严格 JSON、数组或 fenced JSON 变体，前端使用 normalizeRelationshipsPayload(raw) 规整为 {characters:[...]} 统一结构；随后用 toGraph(res) 将 characters[].relationships[] 转换为 ECharts 所需的 nodes/links。

图渲染与样式联动。 在容器具备有效尺寸后（ensureContainerSized(el)），renderRelGraph(nodes, links) 初始化并绘制力导向图，绑定节点点击等交互。后续仅做样式微调时，通过 applyRelStyle() 在不重布局的前提下动态更新 label/node/edge/edgeLabel 的颜色与字号；对应取色器与滑块变更由 watch([relStyle.*], ...) 触发。组件卸载时用 disposeRelChart() 释放实例；窗口尺寸变化由 handleResize() 统一自适应。

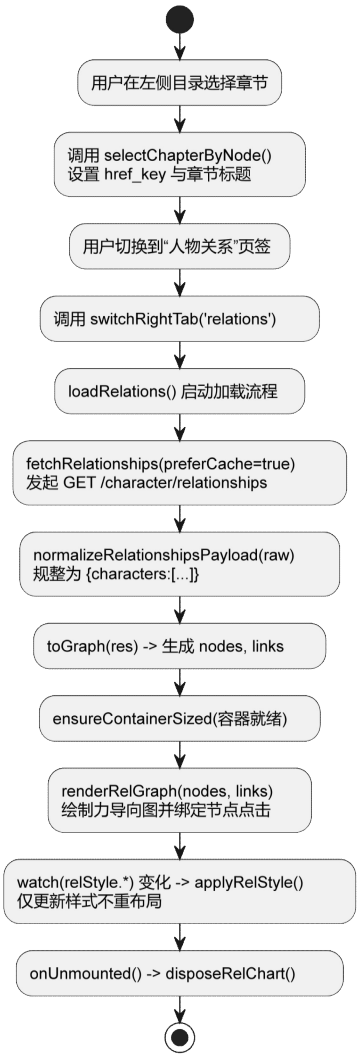


图 7. 人物关系图流程图

概览导航模块

该模块提供两个时间线视图：鲁迅作品发表时间轴和鲁迅生平年谱时间轴，帮助用户快速浏览时间相关信息。

作品时间轴模块展示鲁迅各主要作品的发表年份、出版顺序等信息。时间轴上标注作品名称和时间，用户可以点击查看作品详情。生平年谱模块展示鲁迅本人重要生平事件的时间线，如出生、留学、成名、逝世等。用户可以通过时间轴了解作家生平节点。

2.4 各模块关键代码

AppRoot

```
// src/main.js
import { createApp } from 'vue'
import App from './App.vue'
createApp(App).mount('#app')
```

```
// src/App.vue（片段：仅展示已实现接口名的调用关系）
onMounted(() => {
  applyTheme()              // 读取已保存主题并应用
})
watch([themeColor, gradStart, gradEnd, gradAngle, cornerMode, cornerRadius], () => {
  applyCssVars()            // 实时写入 :root
  persistTheme()            // localStorage('lx_theme')
})
function switchModule(id)
function toggleThemePanel()
function resetTheme(){ applyTheme() }
```

数据概览

```
// 词云（仅展示已实现接口/参数）
function loadWordCloud(){
  // 构造 /wordcloud?href_key=... 的图片 URL 并交给 <img> 显示
}

// 统计：词/字
function switchStatsTab(tab){
  // tab in {'word','char'} -> loadStats()
}
async function loadStats(){
  // GET /freq/word 或 /freq/char -> renderBarChart(pairs, title)
}
```

序号：14

```
// 人物关系
async function loadRelations(){
  const res = await fetchRelationships(true)
  const norm = normalizeRelationshipsPayload(res)
  const { nodes, links } = toGraph(norm)
  await ensureContainerSized(chartEl)
  renderRelGraph(nodes, links)
}
```

EPUB 阅读器

```
onMounted(async () => {
  await fetchTOC()
  await ensureContainerSized(hostEl)
  await initEpubReader(epubName)    // 动态导入 epubjs 并创建 rendition
})

function toggleTheme(){
  // 切换 light/dark 并 applyTheme()
}

function nextPage(){ /* rendition.next() 封装 */ }
function prevPage(){ /* rendition.prev() 封装 */ }
function changeFontSize(delta){ /* 同步到 rendition.themes.fontSize */ }
```

后端服务

```
@app.get("/toc")
def get_toc(): ...
@app.get("/wordcloud")
def wordcloud(href_key: str, width: int = 800, height: int = 600): ...
@app.get("/freq/word")
def freq_word(href_key: str, topk: int = 50): ...
@app.get("/freq/char")
def freq_char(href_key: str, topk: int = 50): ...
@app.get("/character/relationships")
def character_relationships(href_key: str, model: str = "glm-4.5", prefer_cache: bool = True, refresh: bool =
False): ...
@app.post("/chat/ask")
def chat_ask(body: ChatAskBody): ...
```

源代码存放地址为：

通过网盘分享的文件：luxunOverviewWebsite.zip
链接: https://pan.baidu.com/s/1k9VP7pBLfsiPS0_lP8nqCw?pwd=0000 提取码: 0000

3 《鲁迅全集》导览数据集和可视化的展示

下文将由顶部导航栏-左侧导航栏-右侧顶部导航栏的顺序依次展示《鲁迅全集》导览项目。
顶部导航栏具有三个选项卡：鲁迅全集概览/书籍导览/数据概览

3.1 顶部导航栏

顶部导航栏内置主题色的选取与样式选取。

序号：14

上图为更改主题色后的情况

3.2 鲁迅全集概览

上图为鲁迅生平，使用年谱直接列出。

序号：14

上图为使用时间线列出。

上图为内置网页直接导向鲁迅生平百科。

序号：14

上图为《鲁迅全集》背景

上图为鲁迅主要作品时间线展示

上图为鲁迅主要笔名

3.3 书籍导览

上图为默认打开第一章。

序号：14

上图为从左侧选择章节右侧刷新状态。

上图为使用右侧顶部导航栏的情况，使用了增大字号以及深色模式。

序号：14

上图为 AI 对话模块的使用。

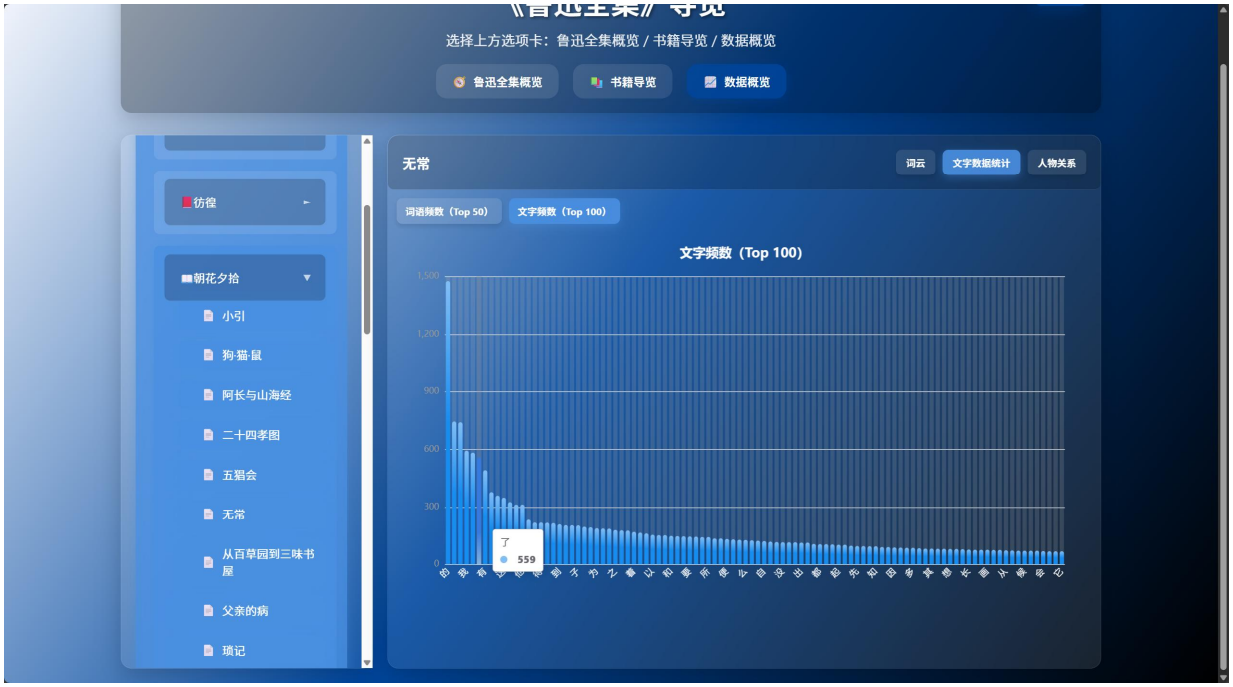
上图为对话期间切换章节同样能流畅对话的使用。

3.4 数据概览

上图为词云显示。

上图为词频统计。

序号：14



上图为字频统计。

上图为加载人物关系，未加载章节会调用 LLMapi 生成 json 文件渲染图片。

上图为加载成功的渲染图。

上图为人物关系图可选的样式。

4 本设计的特点

（1）真实性和可用性：本系统以《鲁迅全集》数字文本为基础，通过前端 Epub.js 库加载并渲染书籍内容 github.com。原始数据经过清洗（例如去除特殊字符、统一编码）后，利用中文分词工具处理

文本。我们采用了广泛使用的 Python 分词库 jieba 来切分句子 github.com，并结合 HanziNLP 等中文 NLP 库支持更多分析功能 github.com。这一方案依赖成熟的开源组件，具有良好的可行性：采用标准数据集和规范处理流程，保证了信息来源的真实性；同时系统面向教育和科研场景，具备较高的实用价值，如辅助教学和文本定量分析等。

（2）感知性和可行性：可视化展示方面，本系统使用 Apache ECharts 绘制多种图表，将数据映射到不同视觉通道以增强感知效果。例如，在时间序列或章节分布中使用坐标轴展示顺序，采用颜色深浅编码词频或情感强度，利用节点大小表达词汇出现频率，并在网络图中用连线突出词汇间关联关系。借助 ECharts 的 VisualMap 组件，可以将数值映射到颜色、大小等视觉属性 apache.github.io，使不同量级的数据呈现明显差异。此外，系统设计了对比视图，可同时展示不同篇章或主题的数据曲线，便于用户直观比较。上述设计遵循感知原则，使用户更容易捕捉数据规律和差异。

（3）艺术性和设计性：界面设计方面，我们采用简洁清晰的布局，遵循可视化设计原则 echarts.apache.org，保证易用性和美观度。默认主题支持响应式布局，可适应不同屏幕尺寸 echarts.apache.org。在中文显示方面，系统调用 HanziNLP 提供的中文字体库，无需额外导入字体即可渲染高质量中文 github.com，这使图表中的中文标签和词云等元素清晰可见。交互设计上，图表支持鼠标悬停提示、点击筛选和下拉菜单切换等操作流程直观合理。系统采用模块化前后端分离架构，前端负责界面渲染，后端提供数据接口（基于 FastAPI），使得功能易于迭代扩展，后续可根据需要增加新的可视化组件或分析功能。

（4）原创性和有价值：可视化过程揭示了文本中的多层次信息和隐含规律。例如，构建的词共现网络图将词汇作为节点、共现关系作为连线，通过可视化揭示了词汇之间的关联结构 blog.csdn.net。系统还通过词频统计、词云和情感趋势图表，让使用者快速识别鲁迅作品中的高频词汇和主题分布，从而反映作者写作风格和关注点。这些分析成果对于语文研究和教学具有参考价值。总体而言，本系统结合多种可视化手段，深入挖掘《鲁迅全集》的文本特征，具有较高的原创性和实用意义。

5 结论与展望

本项目实现了多项功能：电子书阅读模块（基于 Epub.js）支持全文浏览和检索，后端分析模块（基于 FastAPI）对文本进行分词、词频统计等处理，前端可视化模块（基于 ECharts）呈现词云、柱状图、网络图等多维度视图。系统采用前后端分离架构：前端负责页面交互和图形渲染，后端通过 FastAPI 提供 RESTful 数据接口 fastapi.tiangolo.com，整个系统易于部署和扩展。使用场景方面，该系统可作为语文教学的辅助工具，帮助学生直观理解文本特点；也适用于数字人文和文本挖掘研究，为鲁迅文学作品的定量分析提供支持。未来可进一步丰富图表类型和交互功能，以满足更多教学和研究需求。

6 参考文献

[1] FuturePress. *epub.js: Enhanced eBooks in the browser*. https://github.com/futurepress/epub.js.
[2] fxsjy. *jieba: 结巴中文分词*. https://github.com/fxsjy/jieba.
[3] samzshi0529. *HanziNLP: A NLP package for Chinese text*. https://github.com/samzshi0529/HanziNLP.

[4] Sebastián Ramírez. *FastAPI: a modern, fast (high-performance) Python web framework.* https://fastapi.tiangolo.com/.
[5] Apache Software Foundation. *Apache ECharts: An Open Source JavaScript Visualization Library*. https://echarts.apache.org/.
[6] FastAPI. *Full Stack FastAPI Template – Technology Stack and Features*. https://fastapi.tiangolo.com/project-generation/.