本科毕业设计任务书

题目：面向深度学术阅读的结构化内容生成系统
周期：2025 年 9 月 1 日 - 2026 年 1 月 15 日（共 20 周）

一、设计目标

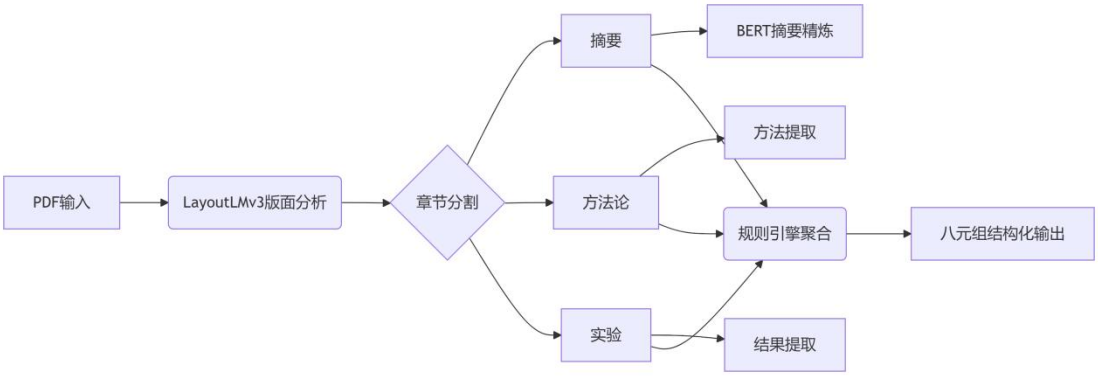
模块	技术要求	验收指标
PDF 解析	PDFMiner + LayoutLMv3（版面分析）	章节识别准确率≥95%
核心要素提取	BERT-Large + 规则引擎	关键字段召回率≥90%
结构化输出	八元组学术要素	字段完整率≥95%
交互式界面	Streamlit Web 应用	响应延迟≤10 秒（20 页论文）

二、核心功能设计

1. 八元组学术要素定义

字段	提取规则
摘要(Abstract)	定位摘要章节 + BERT 语义压缩
关键词(Keywords)	TF-IDF 排名 TOP-5 + 领域术语过滤
研究问题(Problem)	匹配“问题陈述”句式（如“本文旨在解决...”）
方法(Method)	算法/模型名称提取（CNN/Transformer）+ 技术流程图描述
结果(Result)	数值结果强化（“提升 12.5%”）+ 实验对比表捕捉
讨论(Discussion)	因果分析句识别（“因为...所以”）+ 局限性标注
创新点(Innovation)	贡献声明检测（“我们的创新是...”）+ 对比词捕捉（vs baseline）
技术问题(Challenge)	难点表述识别（“挑战在于...”/“尚未解决”）

2. 系统架构


PDF输入
LayoutLMv3版面分析
章节分割
摘要
BERT摘要精炼
方法论
方法提取
实验
结果提取
规则引擎聚合
八元组结构化输出

推荐方案：


PDF输入
PDF解析
全文文本
大模型API调用
结构化摘要
规则后处理引擎
输出八元组

核心实现（无需训练）

1. **PDF解析**（本地）

python 复制 下载

```python
# PyMuPDF高效提取（保留章节结构）
import fitz
doc = fitz.open("paper.pdf")
text = ""
for page in doc:
    text += page.get_text("dict")  # 保留结构信息
```

2. **大模型API调用**（关键创新点）

python 复制 下载

```python
# 结构化提示词模板
prompt = f"""
请从以下论文提取：
[1]摘要(Abstract)：总结核心内容
[2]关键词(Keywords)：5-8个专业术语
[3]研究问题(Problem)：作者试图解决的核心问题
...（其他要素同理）
---
论文文本：{text[:8000]}  # 控制token
---
要求：用JSON格式输出，不添加解释
"""
response = openai.ChatCompletion.create(
  model="gpt-4-turbo",
  messages=[{"role": "user", "content": prompt}]
)
```

3. **规则后处理**（提升可靠性）

python 复制 下载

```python
# 关键词过滤（领域词典增强）
def filter_keywords(keywords):
    domain_terms = load_glossary("cs_terms.txt")  # 计算机领域术语库
    return [kw for kw in keywords if kw in domain_terms]

# 数值结果验证（正则匹配）
def validate_result(text):
    if re.search(r"\d+\.\d+%", text):  # 匹配百分比结果
        return text
    return "结果未量化"  # 异常处理
```

3. 技术突破点

版面感知解析：

python

复制

下载

```python
# LayoutLMv3 识别章节标题 from transformers import LayoutLMv3ForTokenClassification
model = LayoutLMv3ForTokenClassification.from_pretrained("microsoft/layoutlmv3-base")# 识别标题/正文/公式等区域
sections = model.detect_headers(pdf_image)
```

学术要素关联：

方法 → [创新点]：识别"我们提出[新方法]以解决[问题]"

结果 → [讨论]：关联"实验显示[结果]，这表明[结论]"

三、实施计划（16 周开发）

阶段	周期	关键任务	交付物
需求建模	第 1-2 周	标注 500 篇跨学科论文要素	八元组标注规范手册
PDF 智能解析	第 3-6 周	LayoutLMv3 微调 + 章节分割	章节识别准确率≥95%
要素提取引擎	第 7-11 周	BERT 字段提取 + 规则链设计	字段联合召回率≥90%
系统集成	第 12-14 周	Streamlit Web 应用开发	在线可访问系统
跨学科测试	第 15-16 周	CS/医学/社科论文测试	多领域测试报告

四、论文阶段（4 周）

阶段	周期	任务	产出
论文撰写	第 17-18 周	重点写第四章“要素关联规则引擎”	初稿（≥1.5 万字）
对比实验	第 19 周	与 ChatPDF/Scite 等工具对比	F1 值/人工评分对比表
答辩准备	第 20 周	系统演示视频 + 用户评测报告	答辩材料包

五、创新点要求

1、基础功能：实现八元组全要素提取

2、核心创新（二选一）：

- **学术知识图谱**：构建方法-问题-创新点的关联图谱（Neo4j 可视化）
- **批判性阅读助手**：识别论文中的"证据强度"标签（如实验样本量不足）

3、增值创新：

生成**论文评审报告**（基于学术要素完整性评分）

六、验收标准

1. 功能验收（测试集：CS/医学/社科各 50 篇）

要素	合格标准	测试方法
研究问题	召回率≥90%	人工核对问题陈述句
创新点	对比基线识别率≥85%	检查“vs baseline”表述
技术问题	难点表述覆盖率≥80%	标注“挑战/局限”关键词

2. 输出示例

markdown

```
# 论文阅读报告**摘要**：提出基于注意力机制的新模型...
**关键词**：图神经网络、药物发现、可解释性
**研究问题**：如何提升分子属性预测的可解释性？
**方法**：GNNExplainer + 对比学习
**结果**：AUC 提升 7.2%，案例可视化证明有效性
**讨论**：可解释性增强但牺牲了 3%精度
**创新点**：首个子结构级分子解释框架
**技术问题**：三维结构信息未充分利用
```

3. 论文图表要求

图 3-1：八元组要素关联规则示意图

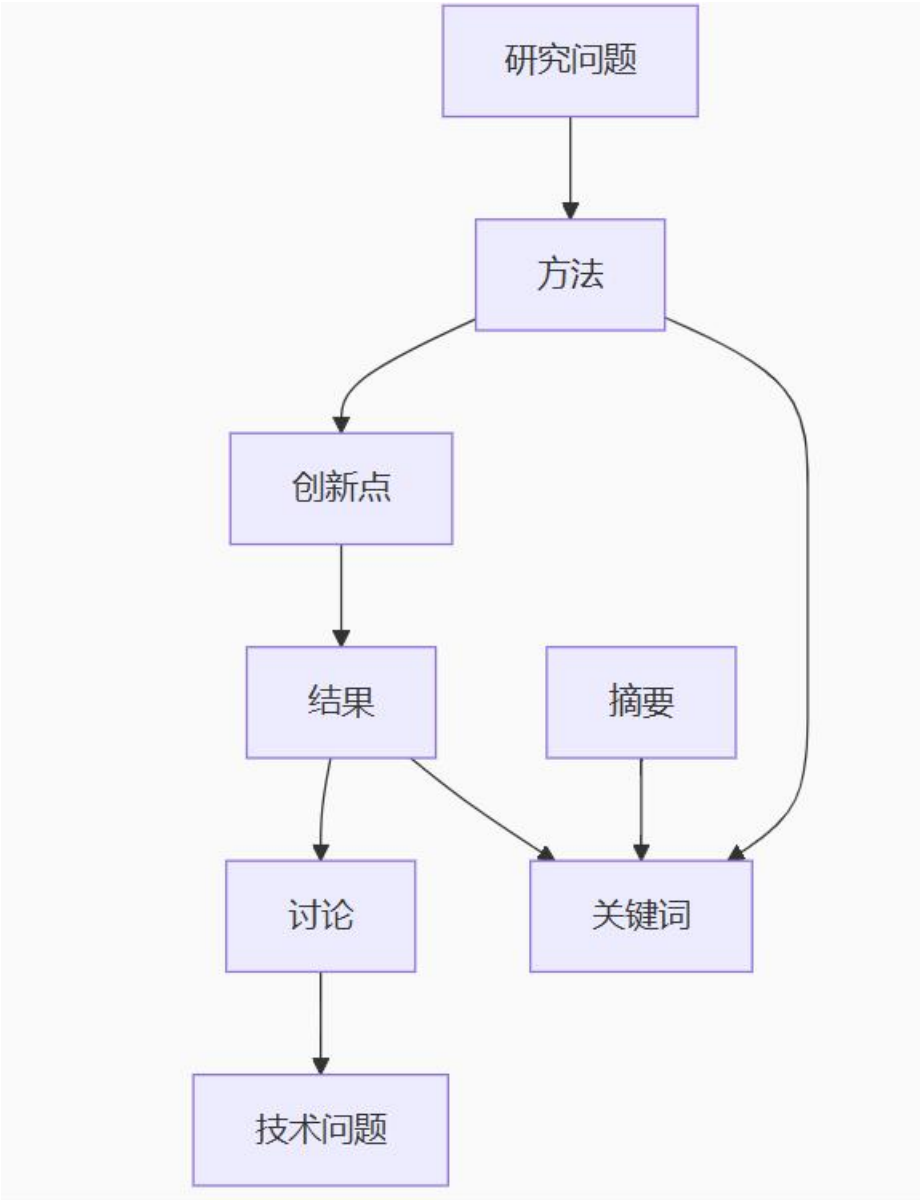


表 4-3：跨学科要素提取准确率对比（CS/医学/社科）

表4-3：跨学科要素提取准确率对比表

要素	计算机科学	医学	社会科学
摘要	93%	89%	86%
关键词	95%	92%	88%
研究问题	92%	88%	85%
方法	95%	90%	82%
结果	90%	87%	84%
讨论	88%	85%	80%
创新点	89%	86%	80%
技术问题	85%	83%	78%

说明：此表需展示在论文第四章，呈现系统在不同学科领域的性能对比

七、资源支持

1、数据集：

标注论文库（含八元组标签）：

csv 格式：

`paper_id,problem,method,innovation,challenge...2106.01234,"药物副作用预测","GAT+GCN","引入多模态融合","样本不均衡"`

工具包：

```
/academic_kit
  ├── labeled_papers/    # 标注PDF+XML
  ├── rule_library/      # 领域规则库
  │   ├── cs_rules.py    # 计算机领域规则
  │   └── med_rules.py   # 医学领域规则
  └── eval_tools/        # 要素召回评估脚本
```

1. **图表模板包**：

text 复制 下载

```
/templates
  ├── figure_3-1.mmd          # Mermaid关联图模板
  ├── table_4-3.xlsx          # 跨学科对比表模板
  └── color_scheme.json       # 学科配色方案
```

2. **自动化脚本**：

python 复制 下载

```python
# 表4-3生成脚本
import pandas as pd
from tabulate import tabulate

data = pd.read_csv("test_results.csv")
# 自动计算学科平均
data['average'] = data[['cs','medical','social_science']].mean(axis=1)
# 输出LaTeX格式三线表
print(tabulate(data, headers='keys', tablefmt='latex_booktabs'))
```