

廈門大學

本科毕业设计

基于 RAG 技术的智能标书撰写系统

Intelligent Bid Document Writing System Based on RAG Technology

姓名：朱纪玉

学号：22920212204333

学院：信息学院

专业：网络空间安全

年级：2021 级

校内指导教师：罗晔 副教授

二〇二五年四月三十日

厦门大学本科学位论文诚信承诺书

本人呈交的学位论文是在导师指导下独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果，均在文中以适当方式明确标明，并符合相关法律法规及《厦门大学本科毕业论文（设计）规范》。

该学位论文为（ ）课题（组）的研究成果，获得（ ）课题（组）经费或实验室的资助，在（ ）实验室完成（请在以上括号内填写课题或课题组负责人或实验室名称，未有此项声明内容的，可以不作特别声明）。

本人承诺辅修专业毕业论文（设计）（如有）的内容与主修专业不存在相同与相近情况。

学生声明（签名）：

年 月 日

摘 要

随着招投标行业的数字化转型加速，传统的标书撰写方式大量依赖人工，存在较多重复性工作。本文提出了一种基于 RAG（Retrieval-Augmented Generation，检索增强生成）技术的智能标书撰写系统，旨在通过结合大语言模型（LLM）和外部知识库，实现高效、高质量的标书自动生成。系统通过解析招标文件，提取关键信息，并利用知识图谱和向量检索技术，为标书撰写提供精准的知识支持。此外，系统还具备风格迁移和决策参考功能，能够满足不同用户的个性化需求。未来，我们将进一步优化知识图谱的构建和检索机制，提升系统的智能化水平，以更好地服务于招投标行业。

关键词：智能标书撰写；RAG 技术；大语言模型；知识图谱；招投标；决策参考

Abstract

With the accelerated digital transformation of the bidding and tendering industry, traditional methods of bid document writing rely heavily on manual labor, resulting in a significant amount of repetitive work. This paper proposes an Intelligent Bid Document Writing System based on RAG (Retrieval-Augmented Generation) technology. The system aims to achieve efficient and high-quality automatic bid document generation by integrating Large Language Models (LLMs) with external knowledge bases. By parsing tender documents and extracting key information, the system leverages knowledge graphs and vector-based retrieval techniques to provide precise knowledge support for bid writing. Additionally, the system features style transfer and decision-making reference capabilities, which can meet the personalized needs of different users. In the future, we will further optimize the construction and retrieval mechanisms of knowledge graphs to enhance the intelligence level of the system and better serve the bidding and tendering industry.

Key Words: Intelligent bid document writing; RAG technology; Large Language Models; Knowledge graphs; Bidding and tendering; Decision-making reference

目 录

第一章 绪论	1
1.1 招投标行业数字化发展现状与挑战	1
1.2 大语言模型	2
1.3 RAG 技术	2
1.3.1 RAG 简介	2
1.3.2 RAG 技术在智能标书撰写工具中的应用潜力	2
1.4 本文工作	3
1.5 论文结构	3
第二章 相关工作	5
2.1 招投标领域的研究现状	5
2.1.1 传统投标方法	5
2.1.2 现有智能标书撰写工具	6
2.2 大语言模型的研究现状	6
2.2.1 大语言模型的发展历程	6
2.2.2 提示词工程	7
2.3 RAG 技术研究现状	8
2.3.1 RAG 技术原理	8
2.3.2 RAG 技术发展历程	10
第三章 智能标书撰写系统设计	11
3.1 系统架构设计	11
3.2 企业信息管理	11
3.3 方案库（知识图谱）	11
3.4 资历库和素材库	12
3.5 标书内容生成	14
3.5.1 识别招标需求	14
3.5.2 生成标书目录	14
3.5.3 生成标书正文	15
3.6 风格迁移	15
3.7 决策参考	15

3.8 后期处理	15
第四章 系统实现与实验	16
4.1 系统开发环境与工具	16
4.1.1 系统环境	16
4.1.2 系统框架	16
4.1.3 开发工具	16
4.2 系统功能实现	16
4.2.1 方案库 (知识图谱)	16
4.2.2 资历库和素材库	17
4.2.3 招标需求识别模块	17
4.2.4 标书目录生成模块	18
4.2.5 标书正文生成模块	19
4.2.6 决策参考模块	19
4.2.7 后期处理模块	20
第五章 总结与展望	21
参考文献	22
附录	23
致谢	24

Contents

Chapter 1 Introduction	1
1.1 Development Status and Challenges of Digitalization in the Bidding and Tendering Industry	1
1.2 Large Language Model	2
1.3 RAG Technology	2
1.3.1 Introduction to RAG	2
1.3.2 Application Potential of RAG in Intelligent Bid Document Writing Tools	2
1.4 Work of This Thesis	3
1.5 Structure of the Thesis	3
Chapter 2 Related Work	5
2.1 Research Status in the Bidding and Tendering Field	5
2.1.1 Traditional Bidding Methods	5
2.1.2 Existing Intelligent Bid Document Writing Tools	6
2.2 Research Status of Large Language Models	6
2.2.1 Development History of Large Language Models	6
2.2.2 Prompt Engineering	7
2.3 Research Status of RAG Technology	8
2.3.1 Introduction to RAG Technology	8
2.3.2 Development History of RAG Technology	10
Chapter 3 Design of the Intelligent Bid Document Writing System	11
3.1 System Architecture Design	11
3.2 Enterprise Information Management	11
3.3 Solution Library (Knowledge Graph)	11
3.4 Qualification Library and Material Library	12
3.5 Bid Document Content Generation	14
3.5.1 Identifying Tender Requirements	14
3.5.2 Generating Bid Document Table of Contents	14
3.5.3 Generating Bid Document Body	15

3.6 Style Transfer	15
3.7 Decision-Making Reference	15
3.8 Post-Processing	15
Chapter 4 System Implementation and Experiments	16
4.1 System Development Environment and Tools	16
4.1.1 System Environment	16
4.1.2 System Framework	16
4.1.3 Development Tools	16
4.2 Implementation of System Functions	16
4.2.1 Solution Library (Knowledge Graph)	16
4.2.2 Qualification Library and Material Library	17
4.2.3 Tender Requirement Identification Module	17
4.2.4 Bid Document Table of Contents Generation Module	18
4.2.5 Bid Document Body Generation Module	19
4.2.6 Decision-Making Reference Module	19
4.2.7 Post-Processing Module	20
Chapter 5 Conclusion and Future Work	21
References	22
Appendix	23
Acknowledgements	24

第一章 绪论

1.1 招标投标行业数字化发展现状与挑战

随着我国经济社会的快速发展，招标投标行业作为市场经济的重要组成部分，其作用日益凸显。招标投标行业的健康发展对于促进经济增长、提高市场竞争力、维护社会公平正义具有重要意义。招标投标行业具有广泛的市场覆盖面，包括基础设施建设、公共服务、工业生产等多个领域。据统计，2024 年，我国招标采购规模约 34.74 万亿元，其中，政府公开采购招标规模 2.49 万亿元，工程招标采购 26.82 万亿元，其他采购招标 5.43 万亿元；^[2] 目前，在国家进一步“放管服”、加速推进“优化营商环境”的大背景下，招标采购加速向信息化、数字化发展。^[1] 一方面，招标采购交易全流程逐渐实现网络信息化、数字化：2024 年全国电子招标投标平台日均处理量突破 100 万单，AI 评标系统在中部六省试点准确率达 92%^[3]。另一方面，招标投标的服务模式从单一、程序性服务向复合型、综合性咨询服务转型。



图 1-1: 中国招标投标公共服务平台已经支持 CA 互认和远程异地评标和评标专家库共享

目前，地方政府公共资源交易平台功能日臻完善，各大央企集团大多已建成使用招标采购电子交易平台，行业领域内招标采购业务活动基本实现全流程在线化，目前正在加快向数字化和智能化方向转型，AI 应用基础良好。总体而言，与招标人相比，投标人对人工智能技术应用驱动力不足，在功能开发和应用方面均不如招标人及资源交易中心广泛和深入。投标人也已经逐步意识到大语言模型对招标采购行业的深远影响，目前已经有投标人使用 GPT 编制投标文件部分章节，如施工组织方案。^[4]

从开发者的角度来看，开发为投标人服务的智能工具存在以下难点：1. 招标、投标文件往往包含大量的专业术语，需要遵循特定的逻辑或行业规则，目前的自然语言处理技术理解和生成这些内容时存在一定局限性。2. 在投标过程中，

投标人需要综合市场、企业、客户、竞争对手等多方面信息进行决策，仅仅依靠自然语言处理技术或简单的检索技术难以提供有参考价值的决策建议。3. 招标采购行业涉及大量的商业机密和敏感信息，尤其是对于提供 SAAS（Software as a Service，软件即服务）的服务商而言，在技术上和社会信任上都需要长期努力。

1.2 大语言模型

大语言模型（英语：Large Language Model，简称 LLM）是一种基于深度学习的人工智能技术，也是自然语言处理的核心研究内容之一。其核心是使用大规模数据集对模型进行训练，从而使其能够生成自然语言文本或理解语言文本的含义。这些模型通过层叠的神经网络结构，学习并模拟人类语言的复杂规律，达到接近人类水平的文本生成能力。大语言模型采用与小模型类似的 Transformer 架构和预训练目标（如 Language Modeling），与小模型的主要区别在于增加模型大小、训练数据和计算资源。相比传统的自然语言处理（Natural Language Processing, NLP）模型，大语言模型能够更好地理解和生成自然文本，同时表现出一定的逻辑思维和推理能力。

1.3 RAG 技术

1.3.1 RAG 简介

RAG（Retrieval-Augmented Generation，检索增强生成）是一种连接外部数据源以增强大语言模型（LLM）输出质量的技术。这种技术帮助 LLM 访问私有数据或特定领域的数据，并解决幻觉问题。因此，RAG 已被广泛用于许多通用的生成式 AI（GenAI）应用中，如 AI 聊天机器人和推荐系统。RAG 技术一般可以划分为信息检索和生成两个阶段。

在信息检索阶段，系统解析用户输入的文本，从构建的专业知识库中，通过向量检索、图检索等方式筛选相关程度最高的文本片段。在生成阶段，模型将检索到的文档片段作为上下文与用户输入的文本一起输入到文本模型中，得到输出。

1.3.2 RAG 技术在智能标书撰写工具中的应用潜力

标书撰写的过程可以分解为招标要求理解和标书内容生成两个任务，传统的大语言模型能够满足一般的文本理解和内容生成任务，但受限于模型的训练数据，对于标书撰写这样涉及专业知识和企业信息以及多种综合因素的复杂任

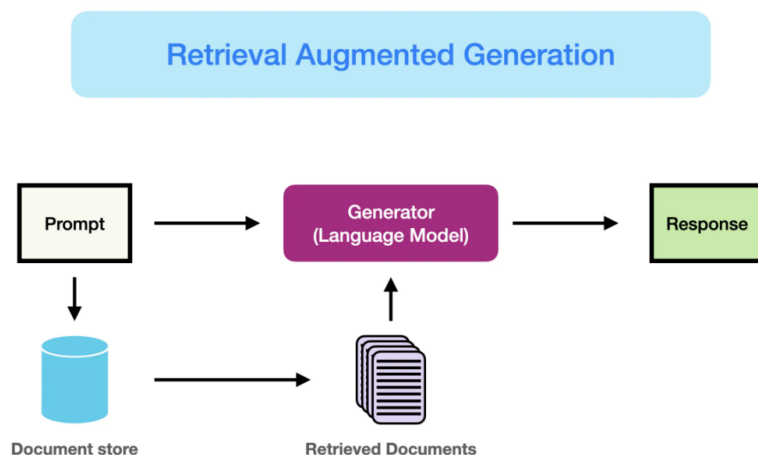


图 1-2: RAG 技术基本原理

务，存在一定局限性。而 RAG 技术通过检索和利用外部知识库中的相关信息，能够为标书撰写提供更丰富的知识支持和更准确的生成内容。

RAG 技术预计可以实现以下功能：1、理解招标要求：通过检索与招标文件相关的行业标准、法规和历史案例，RAG 模型能够更精准地理解招标文件的核心要求，并利用相关知识指导内容生成。2、生成标书内容：结合检索到的外部知识和输入的招标文件要求，RAG 模型可以生成可靠的标书正文内容，并且依靠检索提供的证据链，为人工复核提供便利。3、提供决策参考：RAG 模型可以检索市场数据、竞争对手信息等外部知识，结合预设的竞价、博弈模型，为标书撰写提供有价值的决策参考。

1.4 本文工作

本文工作主要包括以下几个方面：分析现有的智能标书撰写工具，设计基于 RAG 技术的智能标书撰写系统。

实现“招标需求理解-标书框架生成-标书内容生成” workflow；

实现基于知识图谱的方案库，将用户上传的文件，解析为可用于 RAG 的知识图谱；

实现基于 Prompt 的风格迁移功能，为用户提供预设或自定义的风格选项；

实现“数据检索-工具调用-输出决策”的决策参考 workflow；

1.5 论文结构

第一章：本章介绍招投标行业的数字化发展现状，阐述为投标人服务的智能工具的开发难点。随后，简要介绍 RAG 技术并分析标书撰写中的应用潜力。接

着，概述本文的主要工作内容，和论文的整体结构。

第二章：本章介绍传统的招投标方法和现有的智能标书撰写工具，分析其优缺点。详细回顾 RAG 技术的相关研究进展，其中重点介绍基于知识图谱的 RAG 技术。

第三章：本章将对智能标书撰写系统进行需求分析。随后阐述系统架构设计。

第四章：本章将描述系统的开发环境和工具，详细介绍系统功能的具体实现。

第五章：本章将总结本文的主要研究成果，并讨论系统的优势和可能的改进方向。

第二章 相关工作

2.1 招投标领域的研究现状

2.1.1 传统投标方法

一般标书的结构分为商务部分、技术部分和报价部分（但招标文件特殊要求格式除外）。

商务部分：一般包括投标人说明、厂家介绍、业绩、合同、产品授权书、法人授权书、三证、资格证书、交货期、付款方式、售后服务、承诺书、商务偏离表、商务应答、备品备件专用工具清单等，要严格按照标书内容要求及顺序编写。

技术部分：一般包括投标设备技术说明、图纸设计、技术参数、产品配置、技术规格偏离表，技术力量简介、安装施工方案、产品质量、产品简介、产品彩页等等，要严格按照标书内容要求及顺序编写。其中技术偏离表需要标注正偏离、负偏离、无偏离；如投标产品的技术指标优于招标要求即为正偏离，反之为负偏离，符合招标要求即为无偏离；

报价部分，一般包括报价一览表、分项报价表、投标函、投标承诺书、投标单位名称、投标代表签字、法人代表签字等，要严格按照招标文件要求及顺序编写。

标书编写可以分为初步目录编排和后期目录编排两种方式。初次目录编排：根据招标文件的要求，初步编写投标文件目录；为了方便收集投标书的资料。对评分点、控标点、优势应在初步目录中标注，其目的是为了标书制作者重视该部分文档。后期目录编排：按事先拟定好的投标文件目录，对正文内容的标题设置为标题 1、标题 2、标题 3、标题 4 等，然后自动生成投标文件的目录。

在标书电子版制作完成后，参与人员需要进行交叉检查，对错误的地方进行指正修改；需要重点审查的地方有：正本制作正副本内容一致；字体、格式是否统一；审查报价产品明细是否符合招标产品需求明细（包括产品型号和数量），分项（分包）报价是否符合和正确；审查报价表中的分报价和总报价的计算、大小写是否正确，报价表、投标一览表、投标函中的报价大小写是否一致；审查开标文件书写格式是否与招标方的要求一致；

在标书打印完成后，需签字处完成签字盖章，最终审查无误后，进行密封。

传统的标书编写方法依赖人工操作，为确保标书的质量和合规性，投标团队

需要投入大量的时间和精力进行信息收集、内容撰写和格式调整，存在较多重复性、繁琐性的任务。开发智能化的标书撰写工具，将投标工作中的重复性和繁琐性任务自动化，有望提高标书编写的效率和准确性。

2.1.2 现有智能标书撰写工具

目前，市面上已经出现了一批智能标书撰写工具，如链企标书、星火投标等。这些工具的开发背景各不相同，部分由传统招标采购行业的专业公司开发，部分由互联网科技巨头或新兴的互联网创业公司开发。在服务形式方面，大多数工具提供 SaaS（Software as a Service，软件即服务）模式，用户通过网络浏览器登录平台使用标书撰写工具，少数工具提供本地部署服务。

这些智能标书撰写工具的核心功能基本相同，仅在功能完成度和用户体验上存在差异。总体而言，这些工具具备以下几项核心功能：招标文件解析：解析用户上传的投标文件，提取标书撰写的参考信息。投标目录生成：根据自动提取或用户输入的参考信息，生成标书目录。投标正文生成：根据标书目录，生成标书内容。除此以外，部分工具有在线编辑、风格迁移、文件云端存储引用、封面模板等附加功能。



图 2-1: 现有 AI 工具的核心功能，图中以链企 AI 为例

目前市面上的智能标书撰写工具在实际应用中仍然存在如下问题。

- 1、生成的内容不够专业。标书撰写涉及复杂的细分领域，不同行业的招标要求和标书撰写规范存在较大差异。现有工具仅仅能区分服务类、货物类、工程类、其它类四大类别。
- 2、无法提供决策参考。标书中包含必不可少的报价部分，现有工具几乎不能给出有可靠依据的报价。

2.2 大语言模型的研究现状

2.2.1 大语言模型的发展历程

2017 年，Google 提出了基于自注意力机制（Self-Attention）的特征提取器——Transformer，将传统的循环神经网络（Recurrent Neural Network, RNN）彻底

取代，并成为后续所有大型语言模型（LLMs）的基础构件。Transformer 最核心的自注意力计算可表示为：

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

其中 Q, K, V 分别是查询（Query）、键（Key）和值（Value）矩阵， d_k 是键的维度。

2018 年，OpenAI 发布了 GPT-1，首次提出“通用大模型预训练 + 任务特定微调”的范式——先在海量无标注语料上无监督地学习语言结构，再在小规模有标注数据上进行有针对性的微调。这种方法大幅缩短了新任务的开发周期，也降低了对标注数据量的依赖。

随着应用场景的多样化，LLMs 在训练策略、模型架构和下游用例上不断演进。以预训练目标为例，既有面向填空的掩码语言模型（Masked Language Modeling, MLM），也有面向连续文本生成的自回归语言模型（Autoregressive Language Modeling），其概率分解可写为：

$$P(x_1, x_2, \dots, x_n) = \prod_{i=1}^n P(x_i | x_{<i})$$

目前主流的 Transformer 框架大致可分为三类：

Encoder-Decoder 架构：典型代表如 T5 和 BART，编码器负责理解输入语义，解码器生成目标文本，适用于机器翻译、摘要等序列到序列任务。

Encoder-Only 架构：典型代表 BERT 系列，通过双向上下文编码，擅长填空、文本分类和检索排序等理解型任务。

Decoder-Only 架构：典型代表 GPT 系列，采用左到右的自回归生成方式，擅长长文本创作、对话系统及代码生成等生成型任务。

2.2.2 提示词工程

为了充分发挥大语言模型的优势、避免对全量参数的昂贵微调，近年来学界将目光聚焦于提示学习（Prompt learning）。提示，作为附加于模型输入的辅助信息，其核心在于让下游输入在形式上更贴近预训练阶段，从而缩小两者在分布和表达方式上的差异，以更高效地承载分类、生成及推理等任务。提示形式可大致分为离散提示（Discrete prompt）与连续提示（Continuous prompt）两大类。

离散提示研究一方面致力于自动化搜索与构建最优提示，例如通过梯度引导或进化算法，从海量候选短语中挑选最优模板；另一方面依靠人工设计来激

发模型潜能，典型代表包括**上下文学习 (In-context learning)**——模型在少样本示例的上下文中直接给出答案，无需参数更新；以及**思维链 (Chain-of-thought)**——在推理题上先生成逐步思路再得出最终结论，实现零/少样本强推理能力。

与此同时，连续提示不受离散词汇的束缚，而是将提示信息表征为一组可训练向量。设提示向量为 P ，模型原参数固定为 θ ，则下游任务的目标可表述为

$$\min_P \mathbb{E}_{(x,y) \sim \mathcal{D}} [\mathcal{L}(f_{\theta}([P; x]), y)]$$

其中 $[P; x]$ 表示将提示拼接到输入 x 前部或特定层中。通过只优化 P 而冻结 θ ，训练所需资源仅为全量参数的一个极小子集，极大降低了显存与计算开销。为进一步压缩参数量，研究者们还提出了**前缀微调 (Prefix-tuning)**、**P-tuning** 及 **Adapter-tuning** 等变体，它们分别在不同网络层插入可调前缀或轻量级模块。

新增方法还能融入提示位置与训练目标的设计：如在中间层注入“软模板”，或采用对比学习与逆向生成等多种监督信号，提升模型对语义、情感或结构化信息的敏感度。此外，为了缓解提示偏差 (prompt bias)，研究者引入**提示校准 (Prompt calibration)** 技术，通过在验证集上统计校正分布，实现更稳健的 zero-shot/few-shot 性能。

2.3 RAG 技术研究现状

2.3.1 RAG 技术原理

RAG 技术可以分为检索和生成两个阶段。

给定用户输入 q ，检索过程可表示为： $z = \text{Search}(q)$ ，其中 z 为检索到的文档集合， Search 为检索函数。

生成过程可表示为： $y = G(q, z)$ ，其中 y 为生成的文本序列， $G()$ 为生成函数。

RAG 过程可以被完整表示为：

$$y = G(q, \text{Search}(q))$$

具体的检索方法可以是基于向量的检索方法，也可以使用基于知识图谱的检索方法。

向量检索方法的基本原理如下：

首先将文本原文向量化存储。

给定文档集合 $D = \{d_1, d_2, \dots, d_n\}$ 得到文档向量：

$$v_i = E(d_i) \in \mathbb{R}^d$$

其中 $E()$ 为嵌入函数（如 BERT、text-embedding 模型）， d 为嵌入维度
给定用户查询 q ，得到查询向量：

$$v_q = E(q) \in \mathbb{R}^d$$

计算文档向量与查询向量的相似度，此处以余弦相似度为例：

$$Sim(q, d_i) = (v_q v_i) / (||v_q|| ||v_i||)$$

将相似度进行排序得到检索结果 z ，此处以 top-k 排序为例：

$$z = \operatorname{argmax}_{d_i \in D} Sim(q, d_i)$$

图检索方法的基本原理如下：

对于知识图谱 $G = (V, E)$ ，其中 V 为实体节点， E 为关系边：

给定用户查询 q ，基于知识图谱的检索方法如下：

首先，识别图中的入口点（初始节点）：

$$V_0 = f(q, G) \subset V$$

其中 f 是任意一个查询函数，将查询映射到相关节点，可以是基于文本的匹配、基于语义的匹配等。

然后，遍历图中的入口点，使用图检索算法来扩展节点：

$$V_r = \operatorname{Expand}(V_0, G, r)$$

其中， Expand 是图检索函数， r 是遍历半径。

检索到的子图 G_q 为：

$$G_q = (V_q, E_q)$$

其中， V_q 为检索到的子图节点集合， E_q 为检索到的子图边集合。

最后，检索到的信息可以使用相关性评分进行排序：

$$Score(v, q) = \alpha \cdot Sim(v, q) + \beta \cdot Centrality(v, G_q)$$

其中， α 和 β 为权重参数， Sim 为相似度函数， $Centrality$ 为子图中节点的重要性度量。

2.3.2 RAG 技术发展历程

RAG 技术最早由 Facebook AI Research 提出，随后在任务规划、推理检索、多模态等研究方向进一步发展。

Rewrite-Retrieve-Read 模型利用 LLM 的能力，通过重写模块和 LM 反馈机制来优化检索查询，并通过更新重写模型来提高任务性能^[7]。HyDE 通过关注生成答案与真实文档之间的嵌入相似性，旨在提高检索的相关性^[8]。2024 年 7 月 2 日，微软开源 GraphRAG^[6]。GraphRAG 利用基于知识图谱的方法，从原文档导出知识图谱，聚类后生成社区摘要。将社区摘要对用户问题的响应汇总，作为对用户的最终响应。这一方法在生成答案的全面性、多样性都相较于传统 RAG 的基线有显著提升。

经过多年发展，RAG 技术的范式已经从朴素 RAG 发展到模块化 RAG，RAG 过程已经被抽象出各种专用子组件，如搜索模块、记忆模块、预测模块、任务适配器模块。^[5]

RAG 技术栈也越加丰富。LangChain 和 LLamaIndex 等大模型框架已经支持 RAG 技术的快速实现。目前也已经出现多个成熟的专用 RAG 框架，如 Haystack、RAGFlow。

第三章 智能标书撰写系统设计

3.1 系统架构设计

系统架构主要分为三个层次：

1. 数据访问层：负责存储和管理企业的历史标书、招标文档、用户参考文段等数据资源。这一层包括方案库（知识图谱）、资历库和素材库（半结构化数据），为企业信息管理和标书内容生成提供数据支持。

2. 业务逻辑层：是系统的核心，负责标书内容的生成和处理。这一层包括 RAG（Retrieval-Augmented Generation）模块、标书内容生成模块（招标理解、目录生成、正文生成）、风格迁移模块（基于 prompt 和基于 LoRA）以及后期处理模块。

3. 表示层：提供用户界面，使用户能够方便地输入招标文件、管理模板、查看生成的标书内容等。

系统包含以下模块

1. 企业信息管理模块：该模块负责存储管理企业的历史标书、方案库（知识图谱）和资历库素材库（半结构化数据），提供可视化管理页面。

2. 标书内容生成模块：该模块是系统的核心，负责根据招标文件的要求生成标书的框架和正文内容，包括招标理解、目录生成和正文生成三个子模块。

3. 风格迁移模块：该模块实现基于 prompt 的风格迁移技术，提供预定义的风格和用户自定义风格。

4. 决策参考模块：该模块提供成本定价模型等决策，为用户提供决策参考。

5. 后期处理模块：该模块对生成的标书内容进行校对、格式调整和优化，以文件的格式输出。

3.2 企业信息管理

3.3 方案库（知识图谱）

基于知识图谱的方案库，作为将用户上传的文件解析为可以用于 RAG 的结构化数据的模块，需要实现文件解析和检索支持两方面的功能。我们需要从文件中提取两部分内容，首先将文本原文分块并向量化存储，向量将作为全局索引用于检索，其次从文本中提取实体和关系的语义信息这两部分内容都将存储在知

识图谱中。

文本解析 workflows 如图：

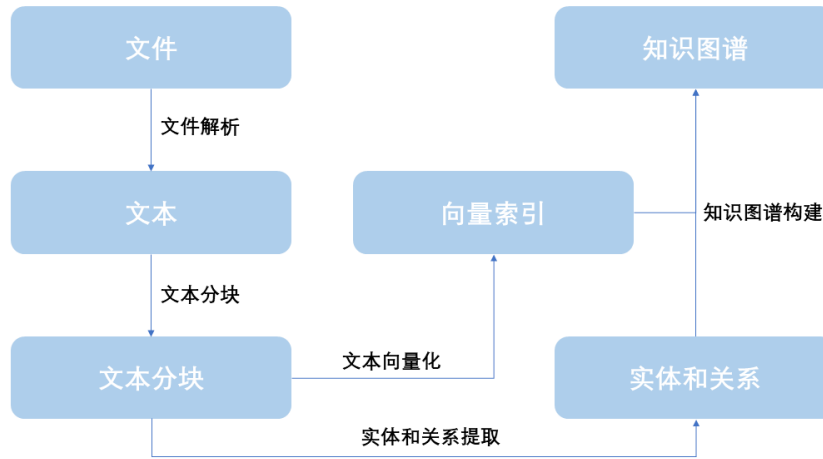


图 3-1: 上传的文件解析为文本后，分块进行实体和关系提取，合并相同实体后存储为知识图谱

文档解析：将用户上传的文件解析为文本。文本分块：过长的文本会降低文本中早期信息的召回率^[9]，因此需要将文件提取的文本进一步分块。实体和关系提取：利用 LLM 提取文本块中的实体和关系信息。文本向量化：将文本块语义信息转化为向量信息，以便于后续检索。知识图谱构建：将文本块、实体作为节点，关系作为边存储在知识图谱中。实体通过“FROM_CHUNK”关系连接到所属的文本块。文本块节点之间通过“NEXT_CHUNK”关系作为边连接。同一文件中不同文本块提取的实体和关系直接视为同一实体进行合并。

检索 workflows 如图：

文本向量化：将用户输入转化为向量信息。向量检索：从全局索引（即向量信息）中检索找到初始文本块。图检索：根据初始文本块在相应的节点中查询邻居节点路径。返回结果：将检索到的图输出。

知识图谱参数定义如下：

3.4 资历库和素材库

资历库存储企业的合同、授权书、三证等资质信息。素材库存储其它撰写标书涉及的信息。资历库需要能够存储和检索企业的资质信息。一般来说，资历的文件格式为 PDF 或图片，仅依靠文件名无法有效检索，因此需要将文件的信息

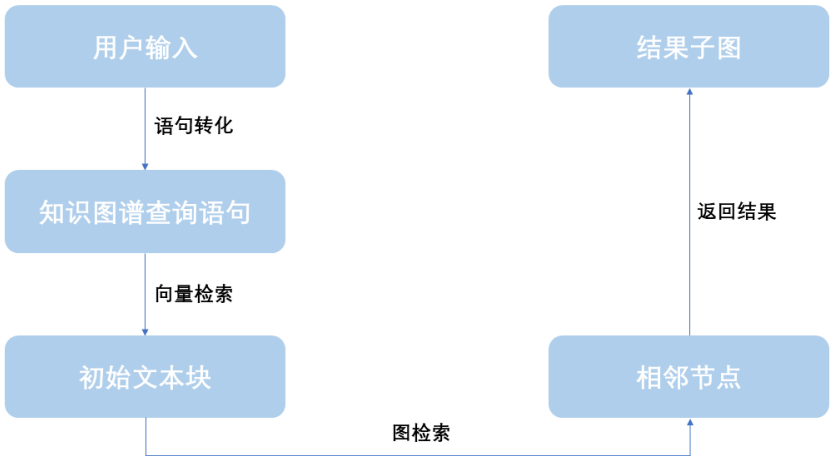


图 3-2: 将用户输入转化为查询语句，先通过向量检索得到初始文本块，再搜寻向量节点形成完整图

表 3-1: 节点定义

参数	参数类型	参数描述	备注
id	str		
label	str		
properties	dict[str,Any]		
embedding_properties	dict[str, list[float]] None		

表 3-2: 关系定义

参数	参数类型	参数描述	备注
start_node_id	str		
end_node_id	str		
type	str		
properties	dict[str,Any]		
embedding_properties	dict[str, list[float]] None		

提取并关联存储。

各个行业涉及的证书信息不同，需要灵活的存储方式。将资历库设计为半结构化的数据库。对于常见的资质文件，可以预设存储的字段。对于不在预设中的资质文件，可以由用户自行增加存储字段。

资历库存储以 JSON 格式，结构如下：

```
{
  "id": "string",
  "name": "string",
  "type": "string",
  "file\_path": "string",
  "properties": {
    "key1": "value1",
    "key2": "value2"
  }
}
```

3.5 标书内容生成

3.5.1 识别招标需求

该模块负责提取招标文件中的关键信息，用于指导后续的标书撰写。该模块分为两步骤：文件解析和文本解析。一般而言，招标文件为 PDF 或 word 文档，在少数情况下为图片格式（纸质文档拍照）。文件解析需要将文件转化为文本格式，文本解析需要将文本分块并提取关键信息。

文件解析过程在后端处理。文本解析由 LLM 完成。

为了方便处理，我们将提取到的信息分为“项目概况”和“评分标准”两类（实际的招标文件中可能涉及更复杂的关键信息）。项目概况包括招标单位、招标项目、招标范围、投标人资格要求、投标文件递交时间和地点、开标时间和地点等信息。评分标准包括商务评分、技术评分、报价评分等信息。

3.5.2 生成标书目录

该模块负责生成标书的目录。根据“项目概况”和“评分标准”两类信息，生成标书目录（尚未引入 RAG）。实际生成过程中，用户可能需要调整目录的顺序和内容，为保证灵活性，我们采用逐章节生成的方式。具体生成方案如下：

初始章节生成：LLM 仅根据全局信息（即“项目概况”和“评分标准”和“风格信息”）生成第一章目录。后续章节生成：将全局信息、已经生成的全部章名、上一章完整目录、下一章完整目录（如果存在）作为输入，生成当前章的目录。

为方便处理，我们将标书的框架固定为“章-节-小节”三级标题形式。每个章可对应多个节，每个节可对应多个小节。在不进行任何生成时，用户可以在前端对目录进行任意调整修改。

3.5.3 生成标书正文

该模块负责生成标书正文。每个小节根据全局信息、当前“章-节-小节”三级标题进行基于 RAG 的生成。初始生成时，仅通过方案库进行 RAG 检索。LLM 的输出为“内容”“引用列表”固定格式。

用户可以对生成的内容进行如下修改：编辑：用户可以直接在前端编辑生成的内容。重写生成：用户可以选择某部分内容重新生成。全局信息、当前“章-节-小节”三级标题、重写部分上下文将作为输入。引用生成：用户可以手动添加来自方案库、素材库、资历库的引用，手动添加引用生成将不再进行 RAG 检索。报价生成：用户可以通过“决策参考”模块生成报价部分的内容。风格化生成：用户可以覆盖全局的风格化设置，选择当前小节的风格化选项。

3.6 风格迁移

该模块负责调整生成标书风格。风格迁移采用基于 Prompt 的方式，将预设文字或用户自定义的文字作为提示词，指导 LLM 生成符合特定风格的内容。

3.7 决策参考

此处的决策参考主要指报价部分的生成。招标文件的报价部分较为复杂，《中国国家标准 GB50500-2013：建设工程工程量清单计价规范》（后称“规范”）中规定了“工程量清单计价”和其它计价规范。“规范”指出，工程量清单应由招标人依据国家标准、招标文件、设计文件以及施工现场实际情况编制，因此决策参考模块主要功能为工程量清单标价。

根据用户输入的额外提示信息（工程量清单的项目）进行方案库 RAG 检索，如果得到符合预设估价公式中的数值，则返回数值和来源，如果没有预设公式中的数值，由 LLM 给出数值并标注来自 AI 生成。

价格预估使用成本定价模型：

$$P = C(1 + r) \quad (3-1)$$

其中 C 为成本，r 为利润率，P 为报价。

3.8 后期处理

后期处理模块负责对生成的标书进行格式调整并生成文件。将生成的标书目录和正文内容合并为一个完整的标书文档，给定格式后，生成为文件。

第四章 系统实现与实验

4.1 系统开发环境与工具

4.1.1 系统环境

- 操作系统: Ubuntu 22.04
- Python 版本: 3.12.3

4.1.2 系统框架

- 前端框架: Vue.js
- 后端框架: FastAPI
- 数据存储: MongoDB、neo4j

4.1.3 开发工具

- 前端开发工具: IDEA、vite
- 后端开发工具: visual studio code

4.2 系统功能实现

4.2.1 方案库 (知识图谱)

使用 neo4j 存储知识图谱。neo4j 是一个图数据库, 使用 `cipher` 语句进行数据存储和查询。后端使用 `neo4j-graphrag` 库和 `neo4j` 库进行知识图谱操作。`neo4j` 库用于连接到 neo4j 进行基本数据操作。`neo4j-graphrag` 库是 neo4j 提供的知识图谱框架库, 用于实现文件解析和检索支持。文件解析过程中, 文件转文本使用 Langchian 的 `Unstructured` 库文本向量化使用阿里百炼 `text-embedding-v3` 模型, 向量维度为 1024 维, 实体和关系提取使用 `deepseek-chat` 模型。检索支持过程中, 文本向量化同样使用阿里百炼 `text-embedding-v3` 模型

前端使用阿里 `g6` 图可视化引擎。该引擎提供了图的绘制、布局、分析、交互、动画等基础的图可视化能力。

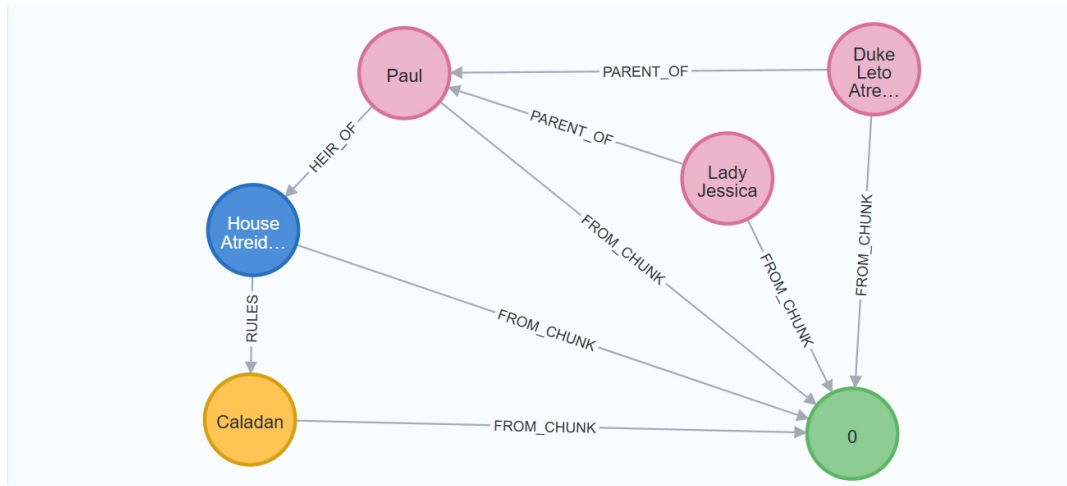


图 4-1: ”The son of Duke Leto Atreides and the Lady Jessica, Paul is the heir of House ”
”Atreides, an aristocratic family that rules the planet Caladan.” 文段生成的知识图谱，0 节点为文本块

4.2.2 资历库和素材库

资历库和素材库使用 MongoDB 存储。MongoDB 是一个 NoSQL 数据库，契合资历库与素材库存储需求。资历库和素材库的常规增删改查在后端使用 pymongo 进行数据操作。资历库和素材库 RAG 由 like 语句模糊查询实现。将 mongoDB 的模糊查询封装为 langchain 工具类，由 LLM 提取若干关键词进行查询。

前端使用 vuetify 进行渲染。

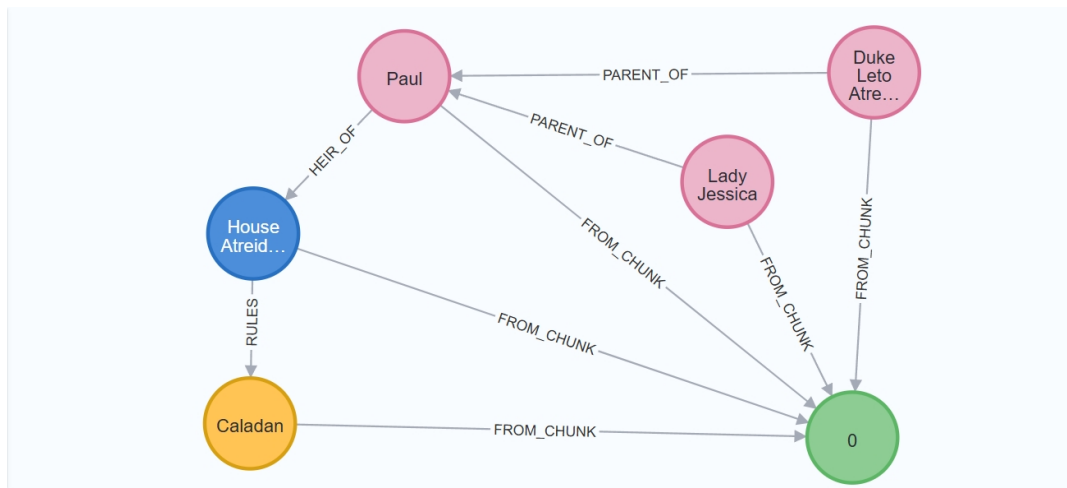


图 4-2: 资历库和素材库前端展示

4.2.3 招标需求识别模块

使用 deepseek-chat 模型进行招标需求识别。结构化返回值定义如下：

```
from pydantic import BaseModel, Field
class Requirements(BaseModel):
    project_requirements: str = Field(
        description="project_requirements")
    scoring_criteria: str = Field(
        description="scoring_criteria")
```

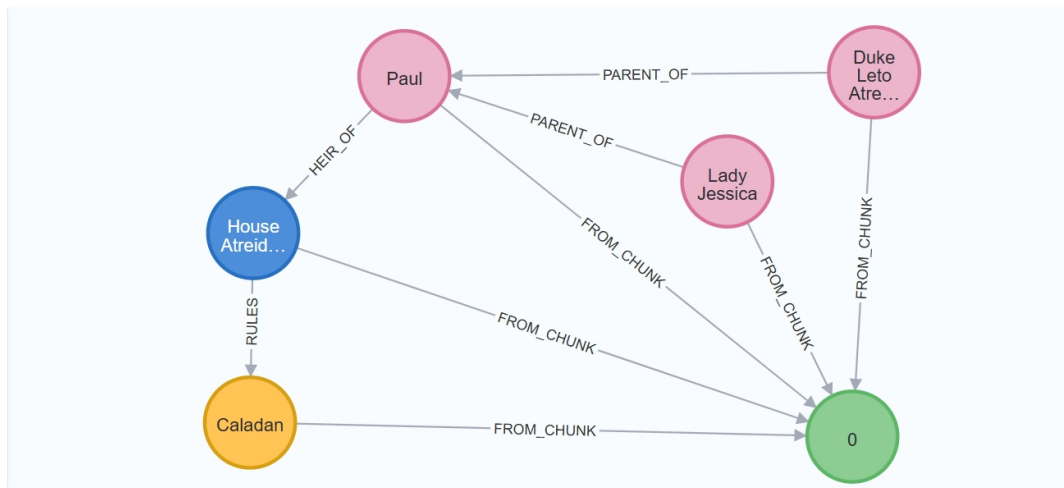


图 4-3: 招标需求识别模块前端

4.2.4 标书目录生成模块

使用 deepseek-chat 模型进行标书目录生成。结构化返回值定义如下：

```
from pydantic import BaseModel, Field
class Chapter(BaseModel):
    title: str = Field(description="章节标题")
    sections: Optional[list["Section"]] = Field(
        default_factory=list,description="一级子目录列表")

class Section(BaseModel):
    title: str = Field(description="一级子目录标题")
    subsections: Optional[list["Subsection"]] = Field(
        default_factory=list,description="二级子目录列表")

class Subsection(BaseModel):
    title: str = Field(description="二级子目录标题")
    content: Optional[list["text"]] = Field(
        default_factory=str,description="正文列表")
```

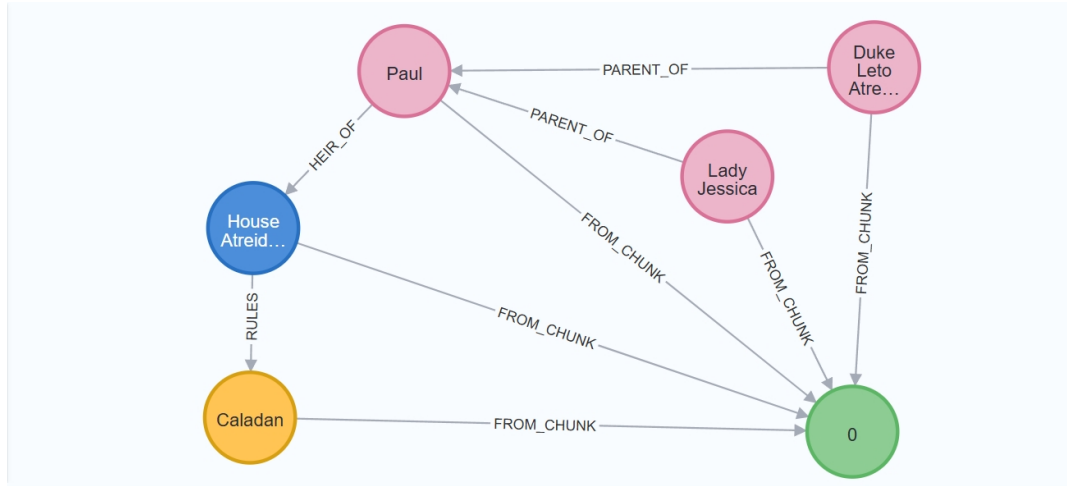


图 4-4: 标书目录生成模块前端

4.2.5 标书正文生成模块

使用方案库后端作检索增强，使用 deepseek-chat 模型生成正文。生成的正文内容结构化返回值定义如下：

```
class text(BaseModel):
    content: str = Field(description="正文内容")
    source: Optional[str] = Field(description="内容来源elementId")
```

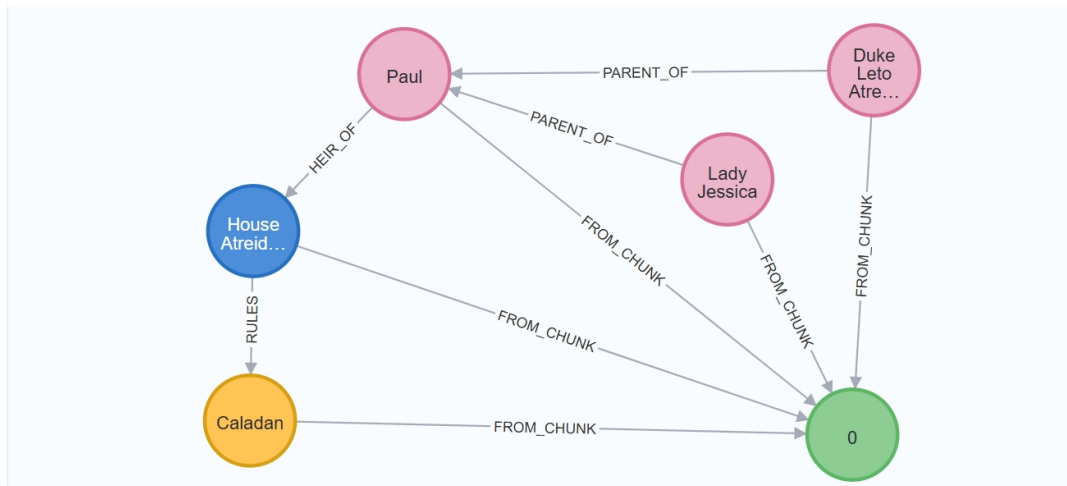


图 4-5: 标书正文生成模块前端

4.2.6 决策参考模块

在标书生成报价部分，使用成本定价模型辅助用户进行投标决策。生成的决策结构化返回值定义如下：

```
class price(BaseModel):
    cost: Optional[int] = Field(
        description="预估的成本")
    rate: Optional[int] = Field(
        description="预估的利润率")
    cost_source: Optional[str] = Field(
        description="成本来源elementId")
    rate_source: Optional[str] = Field(
        description="利润率来源elementId")
```

4.2.7 后期处理模块

使用 docx 库进行后期处理。使用预定义 word 格式的模板文档，利用占位符替换实现文件生成。

模板文档来自中华人民共和国标准施工招标文件（2007 年版）

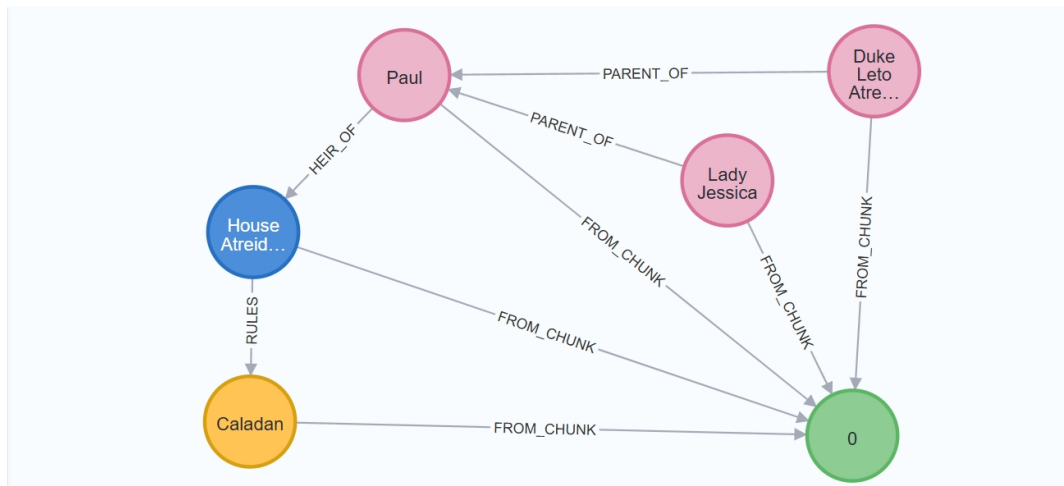


图 4-6: 生成文档效果

第五章 总结与展望

本次设计实现的智能标书撰写系统，虽然功能上能够满足标书撰写的基本需求，但对于标书生成的质量还有很大提示空间。此外，作为一个 web 应用，在性能优化上还有许多中间件可以采用。

未来优化点：一是方案库构建部分，本次知识图谱的构建并没有充分考虑知识融合的问题，不同文件生成的知识图谱之间完全孤立，而单个文件不同文本块的知识图谱实体和关系则直接合并。对于知识融合的方式可以学习更多无监督的方法

二是方案库检索部分，本次 RAG 部分使用向量检索实现节点的首次检索，规避了 cipher 语句的转化问题，在检索到的首个节点孤立时，会退化为传统向量检索。可以学习实现微软 graphRAG 的全局-社区-摘要模式，实现理论上更加有效的检索。

三是识别招标需求可以更细致，需要结合更多实际生产中的招标文档，总结开发更有灵活性和普适性的招标需求识别方法。

四是资历库和素材库查询使用关键词模糊检索进行了简化，mongodb 的向量化查询可以实现传统意义上的 RAG 检索。

最后，决策参考只实现了最简单的成本定价模型，该方案只适用于给定工程量清单进行标价，实际投标定价情况更加复杂，可能存在多轮报价、竞争报价，涉及到复杂的经济学、博弈论知识。要做到高可用的决策参考需要复杂的设计开发。

参考文献

- [1] 芮丽梅, 石国虎. 新形势下招标投标现状与发展趋势 [J]. 招标采购管理, 2021, (07): 11-12.
- [2] 智研咨询. 2025-2031 年中国招标采购行业发展模式分析及未来前景规划报告 [EB/OL]. <https://www.chyxx.com/research/202110/980866.html>, 2024.
- [3] 中研普华产业研究院. 2025-2030 年招投标市场深度分析及发展趋势研究咨询报告 [EB/OL]. <https://www.chinairn.com/news/20250220/103457473.shtml>, 2024.
- [4] 程建宁. AI 技术在招标投标活动中的应用及展望 [J]. 招标采购管理, 2023, (07): 61-63.
- [5] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, Haofen Wang. Retrieval-Augmented Generation for Large Language Models: A Survey [J/OL]. arXiv:2312.10997 [cs.CL]
- [6] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, Jonathan Larson. From Local to Global: A Graph RAG Approach to Query-Focused Summarization [J/OL]. arXiv:2404.16130 [cs.CL]
- [7] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, Nan Duan. Query Rewriting for Retrieval-Augmented Large Language Models [J/OL]. arXiv:2305.14283 [cs.CL]
- [8] Luyu Gao, Xueguang Ma, Jimmy Lin, Jamie Callan. Precise Zero-Shot Dense Retrieval without Relevance Labels [J/OL]. arXiv:2212.10496 [cs.IR]
- [9] Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Dmitry Sorokin, Artyom Sorokin, Mikhail Burtsev. In Search of Needles in a 11M Haystack: Recurrent Memory Finds What LLMs Miss [J/OL]. arXiv:2402.10790 [cs.CL]

附 录

致 谢

本次毕业设计首先感谢我的导师罗晔老师，以及他组建的“《深度学习》学习小组”微信群。罗老师在每个阶段都能给出适时的指导，也不会给我很大压力。在论文选题开始时，罗老师指定了标书撰写这个领域，并且给出了大致研究方法。寒假期间，学习小组中给我们分享了深度学习的大量资料，这为我们后续的设计开发奠定了良好的基础。老师推荐的 langchian 框架，节约了预期大量 LLM 交互的代码量，极大地减轻了开发压力。

其次感谢我的同学们，尤其是曾经在学生会共事的同学们。在我感到压力时，是他们给我鼓励和支持。