

基于 RAG 与工具调用的税务多智能体系统

文颖¹, 黎栋梁¹, 徐丰磊², 张颖²

1. 华东理工大学 信息科学与工程学院, 上海 200237

2. 大象慧云信息技术有限公司, 贵州 贵阳 518100

DOI: 10.61369/TACS.2026050008

摘 要 : 针对税务政策具备繁杂内容、多样规则, 以及问答系统常会出现知识缺口、推理偏差的现状, 设计一套融合检索增强生成与工具调用能力的税务多智能体架构; 搭建起用户、智能体、工具和数据四层完整体系, 依托多智能体协同运作的方式, 完成知识检索、规则推理和税额计算的一体化流程, 进而拉高税务问答具备的精准度、可解释程度与自动化层级。

关 键 词 : 多智能体系统; 检索增强生成; 工具调用; 税务智能服务; 大语言模型

Multi-Agent Tax System based on RAG and Tool Use

Wen Ying¹, Li Dongliang¹, Xu Fenglei², Zhang Ying²

1. School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237

2. Ele-Cloud, Guiyang, Guizhou 518100

Abstract : To address the complexity of tax policies, the diversity of business rules, and the issues of knowledge gaps and reasoning bias in intelligent question-answering systems, this paper proposes a tax-oriented multi-agent architecture integrating retrieval-augmented generation (RAG) and tool invocation. The system adopts a four-layer framework comprising the user layer, agent layer, tool layer, and data layer, and enables integrated processing of knowledge retrieval, rule reasoning, and tax calculation through multi-agent collaboration. Experimental results demonstrate that the proposed approach improves the accuracy, interpretability, and automation level of tax-related question answering.

Keywords : multi-agent systems; retrieval-augmented generation; tool invocation; intelligent tax services; large language models

引言

随着数字税务的持续发展, 市场有了不断增多的税务咨询和申报辅助相关需求, 一方面税务政策有着繁杂的条文与频繁的更新节奏, 传统人工服务很难兼顾理想的服务效率与服务标准一致性^[1]; 另一方面现有主流技术方案里, 大语言模型容易生成脱离实际的虚假内容, 也无法同步完成信息检索、逻辑推理、数值计算这类多环节复杂任务^{[2][3]}; 因此本文设计出结合 RAG 与工具调用模式的税务多智能体架构, 依托智能体之间的协同运行机制有效增强系统处理复杂业务的综合能力, 也拉高了整套架构实际落地应用的实用价值^[4]。

一、税务多智能体系统总体架构设计

为适配税务场景自带的知识密集属性与规则繁杂特征, 本文搭建出包含用户层、智能体层、工具层、数据层的四层整体架构; 用户层可以承担问题录入和业务交互的相关工作, 智能体层能够完成语义理解和任务拆解处理的相关流程, 工具层可以提供数据检索和数值计算的基础能力, 数据层能够给到政策法规和实际案例的内容支撑^{[5][6]}; 整套架构可以形成用户提问、智能体处理、工具执行、数据支撑的完整运行闭环, 还可以兼顾架构本身

的专业属性与后续拓展延伸的适配空间^[7]。

系统依托任务链条的逻辑, 设计出多种具备不同功能的智能体, 涵盖用户交互、政策检索、规则推理、税务计算与结果生成等相关模块, 通过各模块的分工协作, 有效提升复杂任务的处理效率^{[8][9]}; 在执行机制方面, 采用“任务分解—协同执行—结果融合”的具体模式, 先把复杂问题拆解成信息检索、逻辑推理、数值计算等多个子任务, 再由各类智能体配合完成这些子任务, 最后统一整理并输出结果, 进而拉高系统的拓展适配能力与运行可靠程度^[10]。

项目信息: 2024年度观山湖区科技计划项目“基于大模型的税务知识中台及智能问答关键技术研究”观科合同(2024)1号。

作者简介: 文颖(2002.11—), 女, 汉族, 湖南省长沙市, 硕士研究生, 研究方向为电子信息工程。

二、检索增强的税务知识获取方法

（一）税务知识库构建

税务知识库是系统实现专业问答和规则分析的基础；一方面税务信息存有海量零散的政策文字内容，另一方面也涵盖税率标准、适用准入条件、申报执行标准等规范化业务条款，在数据来源方面，主要包括税法法规、税务政策文件、官方解读材料、典型税务案例、税率表和业务办理规范等。

为了提升知识质量，系统会先对原始数据做清洗处理，删掉重复的文本、格式杂乱的内容和无关信息，再依据政策条款的语义拆分模块，确保每个知识单元的内容都相对完整；在向量化处理环节，系统会借助文本嵌入模型，把税务文本转换成高维语义向量，再存到向量数据库里，方便后续开展相似度检索，而税率表、税种分类、优惠条件阈值这类内容，会采用结构化方式存储，以此提高精确查询和工具调用的效率。

（二）基于向量检索的 RAG 机制

为了减少大语言模型在税务场景中的知识偏差，系统里引入 RAG 机制，这种机制的核心思路是，在模型生成回答之前，先从外部知识库中检索出和用户问题最相关的政策文本或案例内容，再把这些信息作为上下文输入模型，进而提升回答的专业性和依据性。

在检索过程中，先把用户问题编码成查询向量，再和知识库中各文本分块的向量表示做相似度计算，系统会按照相似度排序选出 Top-k 知识片段，再把这些片段和用户问题结合起来组成提示词，输入到大语言模型中进行生成，比起直接依靠模型参数回答，RAG 能明显提高回答与现行政策的贴合度，减少没有依据的生成内容，同时也能方便后续提供出处说明和结果追踪。

（三）税务知识检索流程

税务知识检索流程包括四个环节，一是系统接收用户问题后，会对问题做实体识别和关键词提取，二是把问题转换成向量表示，再到税务知识库中做相似度检索，三是对检索到的候选结果做筛选和重排序，四是把最相关的政策片段拼接成上下文，提供给大语言模型生成结果^[5]。

这一流程本质上形成了“问题输入—向量检索—上下文增强—模型生成”的知识获取链路，对于税务政策咨询类问题，这种检索增强生成机制能有效缓解模型知识滞后和政策遗忘的问题，对于复杂业务问题，则可和规则推理、工具调用进一步结合，形成更完整的智能处理框架。基于 RAG 的税务知识检索流程如图 1 所示。

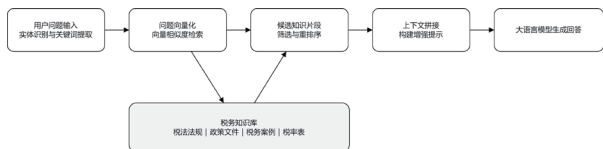


图 1 基于 RAG 的税务知识检索流程图

三、工具调用驱动的税务任务执行机制

（一）工具调用框架设计

税务服务中存在很多需要精确执行的任务，比如税率匹配、

税额计算、政策精确定位和风险校验等，这类任务要是单纯依靠大语言模型生成，很容易受到数值推理能力不足和业务约束不完整的影响，为此这篇内容会引入工具调用机制，把大语言模型和外部专业工具连接起来。

大语言模型负责理解用户意图、识别任务类型并生成工具调用参数，外部工具则负责完成可执行的结构化操作，等工具返回结果后，结果生成模块会把执行结果和政策依据整合起来，形成最终的回答。从功能上看，税务工具主要包括税率计算工具、政策查询工具和税务风险辅助工具。

（二）税务计算工具设计

税务计算是税务智能服务中的典型任务，以企业所得税为例，应纳税额可以表示为：

$$T = I \times r$$

其中 T 为应纳税额，I 为应纳税所得额，r 为适用税率。实际业务中，税率的选择还受到企业规模、行业类型、优惠政策和所得区间等因素影响，因此税额计算是规则判断与公式执行相结合的过程。

税务计算 Agent 根据用户问题提取金额、期间、纳税主体和业务类型等信息，再结合政策检索结果和规则推理结果确定税率与适用口径，最后调用计算工具输出税额结果，提高了税务结果的可验证性。

（三）政策查询与风险辅助工具

除税额计算外，税务服务还需要政策的精确查询和业务风险辅助判断，政策查询工具根据关键词、税种、年份和业务行为快速定位相关法规条款，适用于政策咨询、适用依据说明等场景，系统能在生成回答前先锁定具体政策来源，进而增强答案的规范性。

税务风险辅助工具则用来识别申报过程中的潜在异常情况，比如税率适用不一致、票据信息不完整、业务归类不准确或优惠条件不满足等，可以和规则推理 Agent 配合使用，为用户提供风险提示和补充建议，提高系统服务的实用性。

四、面向税务业务的多智能体任务处理流程

（一）任务分解机制

税务问题通常有着明显的复合特征，比如一个关于“某企业研发费用加计扣除是否适用以及如何计算”的问题，既涉及政策适用条件的判断，也包含金额计算和结果说明，针对这类复杂问题，系统会先通过用户交互 Agent 拆分问题，把它划分为知识检索子任务、规则判断子任务、工具执行子任务和答案生成子任务。

任务分解的关键在于识别问题中的业务目标、输入条件和输出要求。只有将复杂问题拆解为多个清晰的处理单元，系统才能充分发挥多智能体协同的优势。

（二）任务协同与状态传递

在任务拆解完成后，系统会通过调度模块控制各智能体协同执行，政策检索 Agent 负责返回相关政策依据，规则推理 Agent

依据检索内容判断适用性，税务计算 Agent 根据推理结果执行公式计算，最终由结果生成 Agent 组织回答内容；在这一过程中，中间状态的传递十分关键，比如规则推理 Agent 输出的“满足优惠适用条件”，会直接决定税务计算 Agent 采用何种税率或扣除比例，因此系统需要设计统一的中间结果表示形式，确保不同智能体之间的信息衔接准确无误。

多智能体协同与工具调用执行流程如图2所示。

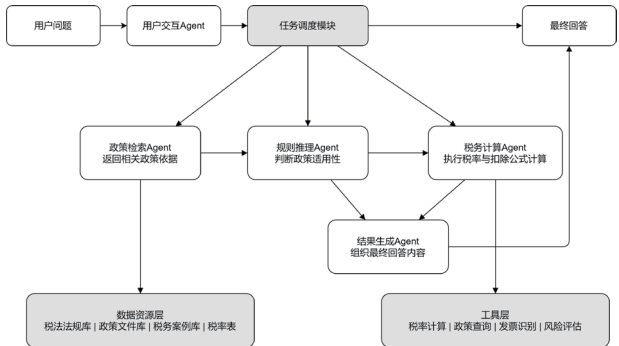


图2 多智能体协同与工具调用执行流程图

（三）结果融合与回答生成

结果融合是多智能体系统面向最终用户输出的关键环节，系统在生成回答时，不仅需要给出结论，还需要尽可能补充政策依据、适用前提和计算逻辑。对于税务场景而言，单纯给出结果往往不足以满足业务需求，用户更关注“为什么适用”“依据是什么”“计算过程如何”，所以结果生成 Agent 需要对前面各智能体结果进行总结，将检索到的政策内容、规则判断过程和工具返回结果整合为结构清晰、逻辑完整的回答。

五、实验设计与结果分析

（一）实验数据与评测任务

为验证所提出方法的有效性，本文构建了面向税务场景的实验数据集与知识库，在知识库构建方面，会基于2022 - 2024年全国税收政策文件、法规条文及相关解读材料，经过数据清洗、文本分块与向量化处理后，最终形成包含2691条知识单元的税务知识库，每条知识单元都会以政策条款或语义完整片段为基本粒度，用来支持后续的向量检索与上下文增强生成。

在模型配置方面，本文选用通义系列大语言模型 Tongyi-Xiaomi-Analysis-Pro 作为基础生成模型，用于完成问题理解、上下文融合及结果生成任务，在此基础上，通过引入检索增强、工具调用以及多智能体协同机制，对模型能力进行逐步扩展。

在评测任务设计上，结合税务业务实际需求，构建测试集，涵盖三类典型任务：（1）政策问答任务，主要考察模型对税收政策条文的理解与检索能力；（2）规则判断任务，侧重评估模型对税务适用条件的逻辑推理能力；（3）税额计算任务，用于验证模型在结合规则与数值计算场景下的执行能力。

在实验对比设置上，选取四种方法进行评估：纯大语言模型（Baseline 1）、RAG 增强方法（Baseline 2）、RAG+ 工具调用方法（Baseline 3）以及本文提出的 RAG+Tool+Multi-Agent 方

法。通过统一评测框架，对不同方法在各类任务中的表现进行系统对比分析。

（二）实验结果与对比分析

（1）整体性能对比

在统一评测框架下，对四种系统配置进行了对比实验。实验结果如表1所示。在30条样本的小规模实验中，本文方法取得最高总体任务分数（overall task score=0.4333），优于 Baseline 3（0.4000）、Baseline 2（0.0667）和 Baseline 1（0.0333）。在扩展至300条样本后，本文方法仍保持最优（0.4000），表明该方法在更大规模数据下具有良好的稳定性。

表1 不同方法在税务任务上的性能对比

方法	总体得分	问答得分	计算得分
仅 LLM	0.0333	0.02	0.0
RAG	0.0667	0.03	0.0
RAG + Tool	0.35	0.05	1.0
RAG+Tool+Multi-Agent	0.4	0.15	1.0

（2）子任务分析

从子任务角度分析，在税额计算任务中，包含工具调用的方法（Baseline 3与本文方法）均达到 calculation score=1.0，显著优于未引入工具的 Baseline 1与 Baseline 2（均为0.0），表明工具调用是提升数值计算能力的关键因素。

在问答与规则推理任务中，本文方法在300条实验中 qa score 达到0.15，高于 Baseline 3（0.05），说明多智能体协同机制在复杂语义理解与推理场景中具有额外增益。

（3）消融实验分析

进一步开展消融实验，分别去除 RAG、工具调用与多智能体协同模块。实验结果如图3所示，工具调用对计算类任务影响最为显著，而多智能体机制对复杂问答与推理任务具有关键作用，验证了系统模块设计的合理性。

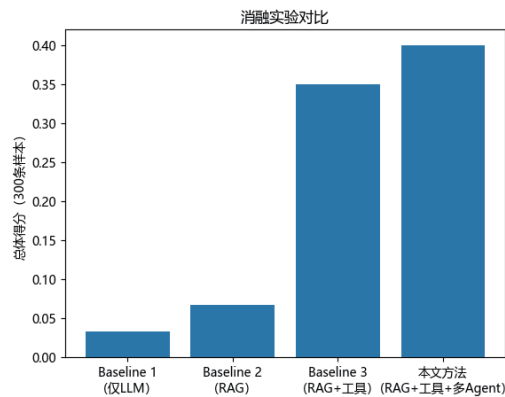


图3 不同方法在消融实验中的性能对比

综合实验结果可以看出，单独引入 RAG 带来的性能提升相对有限，而工具增强和多智能体协同能有效的提升系统性能，工具调用主要提升数值计算的准确性，多智能体协同增强了复杂语义任务的稳定性与结果可解释性。

六、结语

本文针对税务场景中知识获取困难、规则复杂及任务执行要求高的问题，提出了一种结合 RAG 与工具调用的多智能体系统架构，实现了政策检索、规则推理与税务计算的协同处理。实验结

果表明，该方法在问答与计算任务中具有较好的性能与稳定性，相比传统单模型方法，本架构在知识支撑与执行能力方面更具优势，能够提供更具依据性的服务，未来可结合知识图谱与规则链推理，进一步拓展系统能力与应用范围。

参考文献

-
- [1]Aslett J, Hamilton S, Gonzalez I, et al. Understanding artificial intelligence in tax and customs administration[M]. Washington, DC: International Monetary Fund, 2024.
 - [2]Nay J J, Karamardian D, Lawsky S B, et al. Large language models as tax attorneys: a case study in legal capabilities emergence[J]. Philosophical Transactions of the Royal Society A, 2024, 382(2270): 20230159.
 - [3]Fei Z, Shen X, Zhu D, et al. Lawbench: Benchmarking legal knowledge of large language models[C]//Proceedings of the 2024 conference on empirical methods in natural language processing. 2024: 7933–7962.
 - [4]Pipitone N, Alami G H. Legalbench-rag: A benchmark for retrieval-augmented generation in the legal domain[J]. arXiv preprint arXiv:2408.10343, 2024.
 - [5]Gao Y, Xiong Y, Gao X, et al. Retrieval-augmented generation for large language models: A survey[J]. arXiv preprint arXiv:2312.10997, 2023, 2(1): 32.
 - [6]Huang Y, Huang J. A Survey on Retrieval-Augmented Text Generation for Large Language Models[J]. arXiv e-prints, 2024: arXiv: 2404.10981.
 - [7]Fan W, Ding Y, Ning L, et al. A survey on rag meeting llms: Towards retrieval-augmented large language models[C]//Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining. 2024: 6491–6501.
 - [8]Schick T, Dwivedi-Yu J, Dessì R, et al. Toolformer: Language models can teach themselves to use tools[J]. Advances in neural information processing systems, 2023, 36: 68539–68551.
 - [9]Yao S, Zhao J, Yu D, et al. React: Synergizing reasoning and acting in language models[C]//The eleventh international conference on learning representations. 2022.
 - [10]Qin Y, Hu S, Lin Y, et al. Tool learning with foundation models[J]. ACM Computing Surveys, 2024, 57(4): 1–40.