低代码平台中智能问答组件的研究与实现

曾 超[1,2] 于徐红[1,2*]
ZENG Chao YU Xuhong

摘 要 随着大语言模型在企业智能化场景中的广泛应用，低代码平台已成为集成智能问答能力的重要载体。然而，现有低代码平台在智能交互功能方面仍存在较大局限，缺乏对大语言模型的有效支持，限制了企业在低代码环境下实现智能问答的能力。文章以低代码平台 Appsmith 为基础，设计并实现了基于大语言模型的智能问答组件。该组件结合语义嵌入与向量检索等技术，有效提升了问答系统的准确性与可控性。通过结构化的数据处理流程与模块化的系统架构，组件在多个应用场景中实现了快速部署与高效响应，进一步推动了低代码平台在企业数字化转型过程中智能服务能力的扩展。

关键词 低代码平台；Appsmith；大语言模型；智能问答；语义嵌入

doi：10.3969/j.issn.1672-9528.2025.12.003

0 引言

近年来，大语言模型迅速发展，推动了人工智能在各个领域的应用。在智能客服[1]、知识管理[2]、自动化办公[3]以及个性化推荐[4]等场景中，大语言模型展现出了强大的能力，逐步渗透到企业的核心业务流程中。借助其强大的问答能力，企业能够全天候、高效地处理客户的常见问题和需求，减轻人工客服的负担。通过自然语言交互，减少了传统界面和指令繁琐带来的障碍，使用户体验更加流畅，显著提升用户满意度。与此同时，大语言模型的种类日益丰富，包括国外 OpenAI 的 GPT 系列、Google 的 Gemini、Meta 的 Llama，以及国内的多种开源和商用大模型。这些模型在不同的业务场景下各具优势[5]，企业需要根据自身需求选择最适合的模型。

低代码平台作为快速构建应用的工具，提供了一种高效的解决方案。低代码平台降低了应用开发门槛，使企业在无需深厚技术背景的情况下也能搭建智能应用[6]。然而，要实现高效的智能问答功能，低代码平台必须具备良好的扩展性和兼容性，以便与各类大语言模型无缝集成。在众多低代码平台中，Appsmith 凭借其开源特性、高度可定制化能力以及强大的 API 集成功能，成为开发智能问答组件的理想选择。与 OutSystems 和 Mendix 等商业化低代码平台相比，Appsmith 提供免费的开源生态，降低了企业的技术和成本门槛。相比宜搭和简道云等国产低代码平台，Appsmith 在 UI 组件自定义能力方面更具优势，使其在构建智能应用时更加灵活。此外，Appsmith 允许开发者通过 API 和数据库连接轻松集成外部服务，为智能问答组件的开发提供了良好的基础。

本文基于 Appsmith 低代码平台，结合大语言模型开发智能问答组件。该组件充分发挥低代码开发的高效性，融合大语言模型的智能能力，为企业提供高效、可定制的智能问答解决方案。同时，组件具备广泛的适配能力，能够灵活对接不同的大语言模型，支持在不同应用场景中进行模型切换。此外，为满足企业的定制化需求，组件通过接入私有知识库以及添加回答内容的人工干预，增强了问答的准确性与可控性。综上所述，在 Appsmith 低代码平台中构建智能问答组件，扩展了低代码平台的应用场景，助力企业加速智能化转型，增强企业在数字化竞争中的核心优势。

1 智能问答组件的设计与架构

低代码平台将不同的功能封装成独立的组件，开发人员通过组件之间的组合来实现业务需求。因此，在低代码平台中开发组件时，应充分考虑组件功能的边界性、可配置性、与平台的兼容性，以及其在可视化编辑器中的表现形式，以确保组件在实际使用中具备良好的可复用性与交互性。

1.1 系统整体架构设计

智能问答作为典型的人机交互模式，已广泛应用于各种业务系统中，成为提升服务效率和用户体验的基础功能之一。常见的应用场景包括智能客服、员工自助服务、在线教育、知识管理平台等。尽管这些场景的业务内容和知识结构各不

1. 贵州师范大学网络空间安全学院 贵州贵阳 550001
2. 贵州师范大学贵州省先进计算省重点实验室 贵州贵阳 550001

[基金项目] 贵州省科技计划项目（黔科合支撑〔2023〕重点 001）

相同，但它们有一个共同的需求：需要高效、智能、响应迅速的能力，能够理解用户的提问并提供准确的回答，从而增强人机交互的自然性和流畅性。

为了在不同系统中高效部署，并提高回答内容的准确性，问答组件的设计需兼顾两个方面：一方面，必须支持开发者以低门槛方式快速集成，使组件能够通过简单配置直接嵌入到现有应用中；另一方面，组件需要具备高度的通用性与适配性，确保其在多样化的业务环境中稳定、高效地运行。

基于这一目标，组件整体架构包括四个部分：前端交互、后端服务、模型调用和数据提供。如图 1 所示，该架构设计兼顾低代码平台的快速集成特性，确保组件能够在多个业务场景中实现快速配置和灵活应用。

前端交互：基于低代码平台为开发者提供可视化的组件操作界面，支持通过拖拽方式将问答组件添加至应用页面，并在属性面板中配置模型类型、知识源等参数，同时提供智能问答组件与其他组件的数据交互。

后端服务：负责进行身份验证，确保数据的准确性与合规性；保障应用的性能与稳定性，包括负载均衡、缓存管理、数据库优化等功能。

模型调用：封装对不同大语言模型提供商服务的接口调用逻辑，支持灵活切换提供商，提供多场景、多内容下的语义应答能力。

数据提供：负责为系统和组件提供相关的知识库与数据源，保存系统数据，并为大语言模型提供知识支持，确保问答组件能够根据用户需求提供精准、及时的回答。

为满足组件跨应用复用的需求，采用标准化的封装机制。问答组件作为平台组件库中的独立单元，具备完整的输入输出接口，并支持多项参数化配置。其核心逻辑不依赖具体业务数据，而是通过属性传参确定所使用的内容资源，从而使问答能力能够灵活嵌入多类系统中而无需重新开发，体现出良好的工程抽象性与平台适配性。同时，组件还支持与低代码平台其他模块的事件联动。当用户在问答组件中完成提问并获取回复后，系统可通过事件驱动机制调用外部接口、跳转页面或更新全局变量，实现问答能力在更大范围内的业务协同与流程自动化。

该架构设计使得组件使用者能够更加便捷地进行开发操作。清晰的模块划分与技术边界，为后端功能的持续扩展与性能优化提供了有力支持，确保了组件的长期发展潜力。在实际业务中，智能问答组件展现出了强大的适应能力，能够根据不同需求灵活配置，满足各类场景对交互形式、响应行为与功能集成的差异化要求，从而在实际应用中发挥出最大的效用。

1.2 数据处理流程设计

面对不同业务，开发人员通过配置组件、调用接口和设计逻辑流，快速构建各类应用以满足业务需求。由于这些应用面向的内容类型各异，若问答模块无法准确理解各场景中的语义信息，将严重影响交互质量与系统实用性。

为满足响应内容的准确性，系统设计并实现了一套基于语义嵌入的结构化数据处理流程，如图 2 所示。

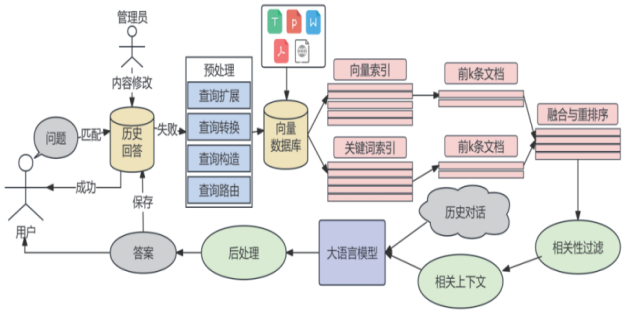

图 2 数据处理流程图

该流程主要包括以下几个核心步骤：对用户上传的原始文档进行数据清洗、格式标准化以及语义分段操作，以提高文本的一致性与可处理性。利用嵌入模型将处理后的文本转换为高维语义向量，存入向量数据库中，形成一个包含大量相关文档的语义数据库。

用户原始的自然语言查询通常存

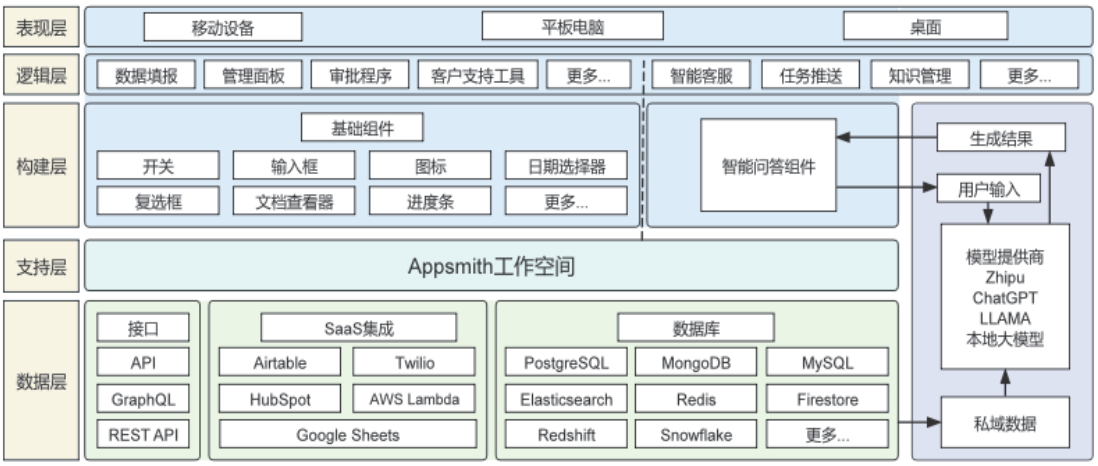

图 1 系统整体架构图

在语义表述不规范、关键词缺失或信息不充分等问题，通过预处理生成伪文档，如表 1 所示，将大语言模型生成的伪文档与用户原始问题进行拼接，不仅保留了用户的意图表达，还引入了具有高度相关性的背景信息，从而有效提升了后续检索与生成模块的语义匹配准确性。

表 1 用户查询预处理表

原始问题	Appsmith 低代码平台的优势是什么
伪文档	Appsmith 低代码平台的优势是什么，Appsmith 是一款开源的低代码开发平台，旨在帮助开发者和企业快速构建内部工具、仪表盘和业务应用程序。该平台支持通过可视化界面拖拽组件，无需复杂编码即可实现功能开发。此外，Appsmith 支持多种主流数据源集成，具有灵活的权限管理和团队协作能力，帮助企业降低开发成本、提升开发效率、加速应用上线速度。

经过处理的查询将被送入两个主要的检索模块——向量索引 [7] 和 BM25 索引。基于语义相似度的向量索引，利用高维嵌入空间中的距离度量实现深层语义匹配。基于词项匹配的 BM25 稀疏索引，强调词频与文本覆盖度。融合这两类检索结果之后，系统能够以更全面的视角定位与查询最为相关的文档，并将其与用户原始问题一并输入大语言模型中，最终生成具备上下文一致性和语义充分性的问答结果 [8]。

该流程的核心优势在于语义一致性与场景适配性。通过对每个应用内容建立统一的语义索引结构，问答组件在不同应用场景中均可通过同一处理机制实现内容感知与智能回答，避免了大模型幻觉的产生和回答内容的无关性。同时，该流程可在 Appsmith 平台中通过标准化接口集成，无需开发者手动重写处理逻辑，显著降低了跨应用场景问答能力构建的复杂度。

此外，该机制具备良好的扩展性和模型兼容性。由于语义嵌入与检索过程独立于具体的大语言模型，系统可根据实际业务场景灵活选择大语言模型如 GPT-4、Gemini、Llama 等不同模型。特别在内容更新频繁的场景下，开发者仅需重新处理新增文本，无需重构组件逻辑，即可实现内容的动态适配。

基于语义嵌入的数据结构化流程，有效支撑了低代码平台下智能问答组件在多应用场景中的快速接入与内容适配，既提升了生成回答的相关性与准确性，又保证了平台开发的高效性与通用性，是构建灵活智能问答能力的关键技术基础。

2 智能问答组件的功能与实现

组件主要实现功能包括上传文件以提升问答相关性、管理员手动修改回答内容以增强可控性、实际系统中的应用效果。这些功能从用户体验、内容管理、问答优化三个维度，全面体现了组件的业务价值与技术落地能力。

2.1 文件上传构建知识库

为了扩展组件的应用场景，并满足企业对定制化内容回答的需求，系统支持管理员上传自定义文档作为问答知识源。通过向量化处理与语义检索技术，系统在回答问题时能够优先参考管理员上传的材料，从而生成更具上下文准确性的回答结果。

管理员登录系统后台后，可通过配置界面上传常见的文件类型（如 PDF、DOCX、TXT 等）。上传完成后，系统会自动对文档内容进行分段、清洗，并生成语义嵌入向量，存入向量数据库中。问答组件会实时更新知识索引，确保系统能够基于最新资料生成准确的回答。

该机制适用于内容更新频繁的企业场景。在这种场景下，管理员可定期更新文档，无需修改问答组件的逻辑结构，即可实现知识的动态更新，显著降低维护成本，并确保问答系统始终使用最新信息响应用户需求。

2.2 问答内容管理

尽管智能问答组件依托大语言模型与知识嵌入机制，已具备生成高质量回答的能力，但在某些高敏感度或要求绝对准确性的场景中，仍需人工审核与干预。为此，系统设计了回答审核与修改机制，允许管理员对历史问答进行人工修订并设定优先答案。

在组件管理界面中，管理员可以浏览用户的历史问答记录，并对系统生成的回答进行评价与编辑。对于不准确或不满意的回答，管理员可以直接修改为标准答案，并确保在其他用户提出类似问题时优先返回管理员设定的结果。该机制不仅保障了答复的一致性与规范性，还显著提升了问答内容的准确性与可控性。

2.3 组件在系统中的应用与交互

智能问答组件在实际系统中的应用效果如图 3 所示，用户可以通过问答组件直接构建独立页面，上传系统的文件资料，帮助用户解答系统信息和使用教学等各类问题。通过这种方式，问答组件作为一个独立的查询入口，可以为用户提供一个直观的界面，轻松获取各种系统相关的信息，提升了信息查询的效率。

此外，组件还可集成至弹窗模式：用户在需要时点击触发弹窗，在其中输入问题并获得即时反馈。该模式特别适用于界面空间有限或用户需求需快速响应的场景。问答组件嵌入弹窗后，可与当前页面内容紧密联动，依据页面上下文提供更具针对性的回答。同时，通过配合业务逻辑实现页面跳转等功能，进一步增强交互的相关性与用户体验。这种集成方式不仅有效节省页面空间，避免界面拥挤，还确保整体界面布局的整洁与操作的流畅性。

图 3 组件应用效果图

3 发展展望

随着人工智能技术的持续演进与低代码平台的广泛应用，智能问答组件的功能边界和应用场景仍在不断拓展。尽管本文提出的智能问答组件已实现对多种大语言模型的适配、私有知识库接入以及人机交互能力的融合，在实际推广与部署过程中，仍面临一些值得深入研究的问题。

当前组件主要聚焦于静态知识问答和语义检索，尚未充分融合生成式多模态能力。未来，可考虑将图像识别、语音输入等技术集成到问答组件中，进一步实现“图文问答”和“语音交互”等更自然、丰富的交互形式，从而提高用户整体体验。

在问答质量控制和安全性保障方面，传统的人工审查具有延时性，可以引入更完善的审查机制与可解释性策略。例如通过链式调用外部验证模块、设置内容过滤规则等手段，确保回答的准确性、规范性和安全性，从而满足高合规性场景下的应用需求。

随着国产大模型的逐步成熟及其生态体系的完善，未来组件可进一步支持多种国产模型的灵活接入，提升数据合规性与本地部署能力，满足各行业在数据主权与隐私保护方面的多样化需求。

4 结语

本文在 Appsmith 低代码平台中设计并实现了智能问答组件，重点介绍了其通过集成大语言模型和私有知识库接入，以提升问答精度和用户体验。展示了智能问答组件在企业中的应用潜力，通过分析当前存在的技术瓶颈和未来的发展趋势，为企业在实际应用中如何更好地部署和优化智能问答系统提供了实践参考。随着技术的不断进步，智能问答组件将被运用于更多的场景，在应用快速开发和推动企业数字化转型和智能化发展方面发挥更加积极的作用。

参考文献：

[1] 刘家帅，檀雨涵．科技赋能客户服务：智能客服替代人工客服的可能性研究 [J]. 科技与创新，2025(7):64-67.

[2] CHEN L, DENG Y, BIAN Y T, et al. Beyond factuality: a comprehensive evaluation of large language models as knowledge generators[EB/OL].(2023-10-11)[2025-11-01].https://doi.org/10.48550/arXiv.2310.07289 arXiv, 2023.

[3] WANG Z L, CUI Y D, ZHONG L, et al. OfficeBench: benchmarking language agents across multiple applications for office automation[EB/OL].(2024-07-26)[2025-10-25].https://doi.org/10.48550/arXiv.2407.19056.

[4] 谢浩然，陈协玲，郑国城，等．人工智能赋能个性化学习：E-Learning 推荐系统研究热点与展望 [J]. 现代远程教育研究，2022, 34(3): 15-23.

[5] LASKAR M T R, ALQAHTANI S, BARI M S,et al. A systematic survey and critical review on evaluating large language models: challenges, limitations, and recommendations[C]//Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. Brussels:ACL, 2024: 13785-13816.

[6] AL ALAMIN M A, MALAKAR S, UDDIN G, et al. An empirical study of developer discussions on low-code software development challenges[C]//2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR). Piscataway: IEEE, 2021: 46-57.

[7] KARPUKHIN V, OĞUZ B, MIN S, et al. Dense passage retrieval for open-domain question answering[EB/OL].(2020-09-30)[2025-10-25].https://doi.org/10.48550/arXiv.2004.04906.

[8] LEWIS P, PEREZ E, PIKTUS A, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks[EB/OL].(2021-04-12)[2025-10-22].https://doi.org/10.48550/arXiv.2005.11401.

【作者简介】

曾超（2001—），男，四川绵阳人，硕士研究生，研究方向：人工智能。

于徐红（1974—），通信作者（email:yuxuhong@gznu.edu.cn），男，山西昔阳人，硕士，高级工程师，研究方向：并行计算。

（收稿日期：2025-06-25 修回日期：2025-12-04）