Computer Science and Application 计算机科学与应用, 2025, 15(12), 222-227
Published Online December 2025 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2025.1512337

Hans汉斯

LLM在软件测试用例智能生成中的应用研究

赵东升，孟　伟，徐　锋，于铁军

北京京航计算通讯研究所，北京

收稿日期：2025年11月4日；录用日期：2025年12月5日；发布日期：2025年12月12日

摘　要

为了改变传统软件测试用例设计工作中对测试人员个人经验和能力的强依赖，进一步提高测试用例设计的自动化和智能化水平，本文研究提出一种基于大语言模型(Large Language Model, LLM)的软件测试用例智能生成方案，主要包括专业领域测试用例向量知识库构建、检索增强生成、提示词工程、基于人机交互的总结反思优化，增强了大模型的测试用例生成能力，提高了测试用例设计的工作效率和质量。实验结果表明，本文提出的方法能够明显提升大模型生成测试用例的质量和缩短测试用例设计周期，可进一步推广到工程实际中应用。

关键词

软件测试用例，向量知识库，检索增强生成，提示词工程

Research on the Application of LLM in Intelligent Generation of Software Test Cases

Dongsheng Zhao, Wei Meng, Feng Xu, Tiejun Yu

Beijing Jinghang Research Institute of Computing and Communications, Beijing

Received: November 4, 2025; accepted: December 5, 2025; published: December 12, 2025

Abstract

In order to overcome the strong reliance on individual experience and ability of testers in traditional software test case design work, and to further enhance the automation and intelligence level of test case design, this paper proposes an intelligent software test case generation scheme based on a large language model (LLM). It primarily includes the construction of a vector knowledge base for test cases in professional fields, retrieval-augmented generation, prompt engineering, and

文章引用: 赵东升, 孟伟, 徐锋, 于铁军. LLM 在软件测试用例智能生成中的应用研究[J]. 计算机科学与应用, 2025, 15(12): 222-227. DOI: 10.12677/csa.2025.1512337

summary and reflection optimization based on human-computer interaction. This approach enhances the test case generation capabilities of the LLM and improves the efficiency and quality of test case design. The experimental results demonstrate that the method proposed in this paper can significantly enhance the quality of test cases generated by LLM and shorten the test cases design cycle, which can be further promoted for application in engineering practice.

Keywords

Software Test Cases, Vector Knowledge Base, Retrieval-Augmented Generation, Prompt Engineering

1. 引言

随着计算机软件和电子技术的快速发展，软件规模及其复杂度不断提升，特别是在航空航天等高安全高可靠性要求的领域，系统日益庞大，软件架构和算法极其复杂，且对软件质量的要求非常严苛，这给软件测试工作带来了极大的困难和挑战。传统的软件测试工作模式完全依赖测试人员的个人经验，测试人员的数量和个人能力限制了测试用例设计的数量和质量，同时，大量历史积累的典型测试用例复用困难，测试成本高、效率低且存在较大的质量风险[1]。

近年来，随着大数据和人工智能等技术的发展，人类社会迈入了 AI 赋能千行百业的新阶段。2017 年 Google 颠覆性地提出了 Transformer 架构，奠定了大模型算法架构的基础，2020 年 OpenAI 公司推出 1750 亿参数量的 GPT-3，从此大模型迎来了爆发期，2024 年 12 月 DeepSeek 的迅速崛起，震惊全球，使得人工智能进入了“普惠”时代。大模型等人工智能技术在自然语言理解和文本生成等方面的突出能力，为软件测试工作模式的转型升级带来了新的技术途径。

目前，国内外已有学者针对基于大模型的单元测试用例生成开展了相关研究，并取得了一定的研究成果[2]-[6]，展现了大模型在单元测试用例生成中的巨大潜力，但仍有一些问题(例如模型幻觉、测试分支不全、用例质量不高等)导致研究成果无法有效在工程实际中推广应用。鉴于大模型基于大量公域数据训练，对特定专业领域的软件测试专业术语和背景知识缺乏，直接基于通用大模型生成测试用例可能存在生成的测试用例需求覆盖率低、可执行性差、场景覆盖不全面、采纳率低等问题。本文的研究旨在提出一种可行的基于大模型的软件测试用例智能生成方案，降低大模型理解专业术语的幻觉，提高大模型生成测试用例的正确性和采纳率，进一步提升测试用例设计的效率和质量，为软件测评单位测试工作模式的智能化转型升级提供可行的技术路线。

2. 基于 LLM 的测试用例智能生成方案

大语言模型(Large Language Model, LLM)即大模型以其强大的自然语言理解与生成能力，在代码生成、代码优化、程序分析、代码注释、软件测试等软件工程任务中展现出强大的潜力与广泛的应用前景[7]，然而直接应用通用大模型的基础能力进行测试用例生成通常难以取得理想的效果，无法在工业生产中实现应用落地。为了有效增强大模型在软件测试领域的专业背景知识和测试用例生成能力，本文通过研究提出一种基于 LLM 的软件测试用例智能生成方案，有效解决历史知识资产复用困难、测试用例设计耗时费力、生成测试用例采纳率低等问题。基于 LLM 的测试用例智能生成方案整体架构与工作流程如图 1 所示，主要

包括专业领域测试用例向量知识库构建、检索增强生成、提示词工程、基于人机交互的总结反思优化。

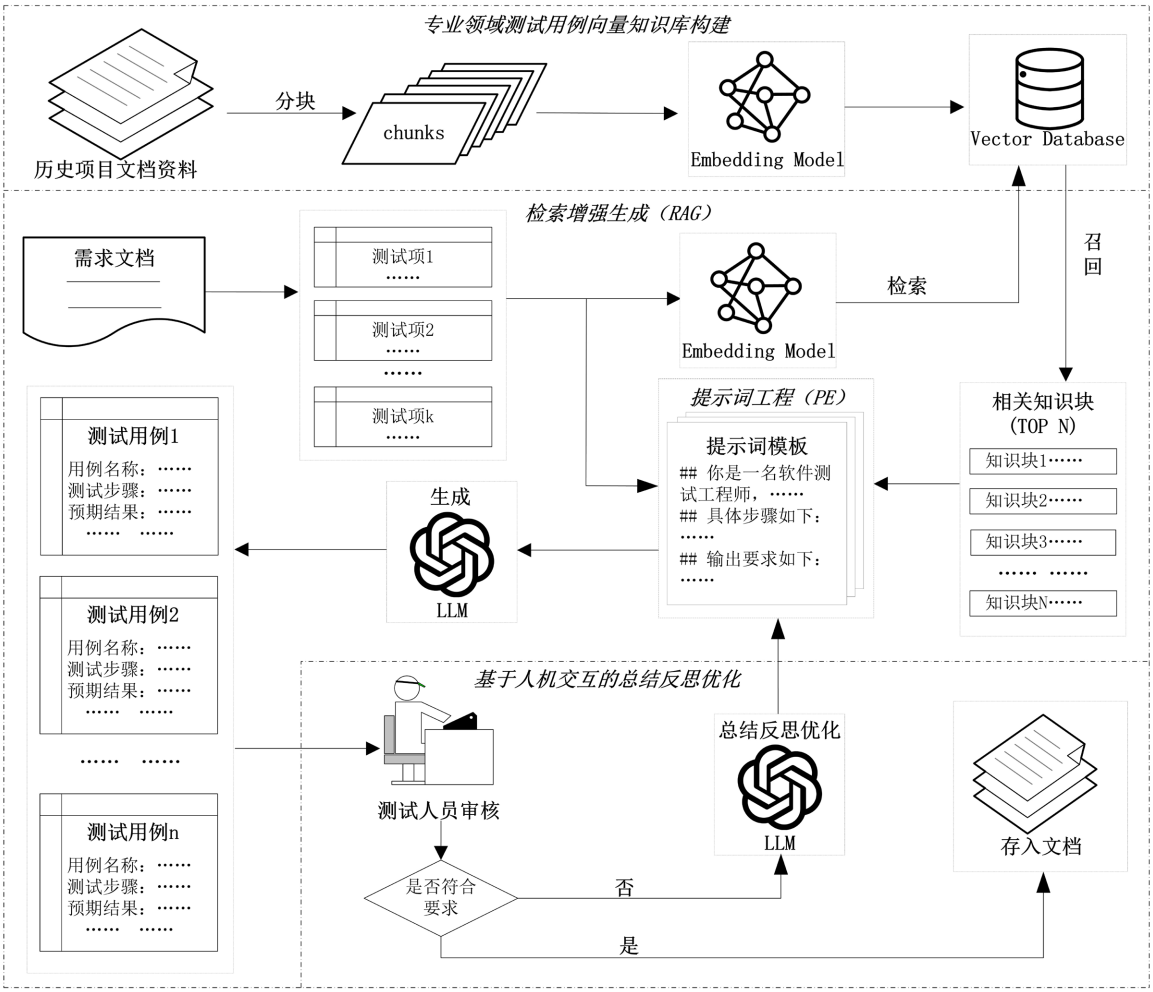

Figure 1. The overall architecture and workflow diagram of the intelligent test case generation solution based on LLM
图 1. 基于 LLM 的测试用例智能生成方案整体架构与工作流程图

2.1. 专业领域测试用例向量知识库构建

(1) 文本切分

构建向量知识库需要先对文档进行切分，即将测试用例相关数据长文本切分成多个文本块(chunks)，而文档切分的质量直接关系到后续检索的精度。目前，文本切分有多种策略，主要包括固定大小切分、语义切分、递归切分、特殊格式切分等，在工程实践中，需要根据应用场景和实验来选择最合适的策略。

(2) 文本向量化

切分好的文本块要用嵌入模型(Embedding Model)转为高维数值向量，实现语义的数学表达，以便进行相似度计算和检索。Embedding 模型能够将文本的语义信息编码到高维向量中，使得语义相似的文本在向量空间中的距离更加接近，因此，通过计算向量之间的距离即可判断两个文本之间的相关性。当前主流的 Embedding 模型有 Qwen3-Embedding 系列、BGE-M3、OpenAI text-embedding-3、Jina Embeddings v3 等，维度包括 256、384、512、768、1024、1536、3072 等，支持的语言包括中文、英文、多语言等。向量维度越高语义表达越精细，但同时存储和计算成本也越高，所以在工程实践中，需根据实际应用场

景和条件进行实验后选择合适的 Embedding 模型。

(3) 向量数据库存储

向量数据库的核心作用是高效存储、索引和检索 Embedding 向量，测试用例文本数据通过嵌入模型向量化后需进一步构建索引并写入向量数据库进行存储，构成测试用例向量知识库。选择一款合适的向量数据库对系统的性能至关重要，常用的向量数据库包括 FAISS、Milvus、Weaviate、Chroma、Qdrant 等，通常需要根据具体业务场景、硬件条件、性能需求等综合考虑，选择合适的向量数据库。

2.2. 检索增强生成

通过检索增强生成(Retrieval-Augmented Generation, RAG)技术能够将构建的测试用例知识库作为外挂知识库增强大模型的专业领域背景知识，从而提高大模型生成测试用例的能力。

(1) 检索与召回

测试人员对需求文档进行分析后拆分出多个对应的{测试项}，并以其作为输入，经过 Embedding 模型向量化后输入测试用例向量知识库进行相似度匹配检索。常用的向量相似度计算方法包括余弦相似度、欧氏距离、点积等。本文选用余弦相似度计算方法，对于向量 $A=(a_1,a_2,\cdots,a_n)$ 和 $B=(b_1,b_2,\cdots,b_n)$ 的余弦相似度计算公式如下：

$$Cosine\ Similarity=\frac{A\cdot B}{AB}=\frac{\sum_{i=1}^{n}a_ib_i}{\sqrt{\sum_{i=1}^{n}a_i^2}\times\sqrt{\sum_{i=1}^{n}b_i^2}}$$

检索完成后召回最相关的 Top N 个文本块作为{上下文}背景知识检索结果输出。常用的检索方法主要包括关键词检索、语义检索、混合检索等，组合多种检索方法(例如关键词和语义检索)可以弥补单一方法缺陷，从而提升检索召回率和准确率。实际工程应用中，需结合实验情况选择合适的检索方法和召回的文本块个数 N。

(2) 增强与生成

从测试用例向量知识库中检索召回的{上下文}背景知识与输入的{测试项}一起输入对应的提示词模板，组装构成完整的提示词输入大模型，从而增强大模型软件测试领域背景知识，使其生成的测试用例更符合测试人员设计习惯和特定专业领域软件测试的要求。

2.3. 提示词工程

提示词工程(Prompt Engineering)是大模型应用中的关键技术之一，旨在通过设计和优化输入提示(Prompt)，引导大模型生成更准确、可靠、符合预期的输出。随着大模型能力的增强，提示词工程的重要性愈发凸显，高质量的提示可显著提升大模型解决测试用例生成等垂直领域问题的能力。

在设计大模型生成测试用例的提示词模板时，除了将输入的{测试项}和通过检索增强提供的{上下文}背景知识插入提示词模板对应位置，还需要考虑以下因素：

(1) 角色设定：即设定大模型的角色，告诉大模型以什么身份或者站在什么人的角度去分析和思考。

(2) 步骤提炼：通过提示引导大模型像人类一样分步骤拆解和分析问题，将测试用例生成任务分解为多个中间过程步骤，通过一步步推理得出最终结论。

(3) 输出格式及约束条件：针对大模型最终输出的内容要设定输出格式和约束条件等，例如限定大模型以 JSON、XML、表格等形式输出，并告诉大模型不输出哪些内容等。

2.4. 基于人机交互的总结反思优化

大模型生成的测试用例是否能够满足测试用例设计要求，需要测试人员通过人机交互界面进行审核

确认。如果大模型生成的测试用例符合要求则将其存入文档；如果大模型生成的测试用例不符合要求，则由大模型进行总结反思优化，对提示词进行优化调整，然后重新生成测试用例，如此循环迭代直到测试人员确认生成的测试用例符合要求为止。通过 Human-in-the-loop 的人机交互循环迭代，有效提高了大模型生成的测试用例的质量。

3. 实验分析

3.1. 实验环境与评估指标

本文的实验环境是在局域网中部署的开源的 Qwen2.5-7B、Qwen2.5-32B、Qwen3-32B、DeepSeek R1-32B 大模型，选择某中等规模的测试项目用大模型直接生成测试用例和本文方法生成测试用例进行对比实验，并采用专家法从测试专家角度对生成的测试用例质量进行综合评估和打分。专家法评估指标主要包括：

(1) 覆盖性：测试用例对需求、边界、正常、异常、错误处理等覆盖性。

(2) 正确性：测试用例逻辑是否合理，是否符合测试习惯，生成的测试用例是否符合相关要求，测试步骤描述是否正确，预期结果是否符合系统行为等。

(3) 可执行性：测试用例是否具备明确操作步骤和可验证的执行结果。

(4) 完整性：测试用例要素的完整程度，即是否按要求生成了测试用例名称、测试步骤、预期结果等全部内容。

(5) 规范性：测试用例是否符合指定的输出格式和约束条件等。

此外，根据大模型生成的测试用例直接被测试人员采纳的比例计算测试用例采纳率，可从测试人员角度对测试用例质量进行评价。

3.2. 实验结果

项目实验结果显示，直接用大模型生成测试用例时，生成的测试用例质量普遍不高，采纳率不足 20%，而用本文方法能将大模型生成测试用例的采纳率大幅提升，测试用例质量也获得了领域专家较高的评价与认可。此外，参数量为 32B 的大模型在测试用例生成方面的整体能力优于 7B 大模型，参数量同为 32B 的大模型在测试用例生成方面的能力相差不大，在工程应用中可根据实际情况选择相应的大模型。同时，利用本文方法生成测试用例能够缩短测试用例设计周期，减少人员投入，提高软件测试工作效率。

4. 局限性分析

本文提出的基于 LLM 的软件测试用例智能生成方案实际应用效果和知识库、LLM 的能力等因素密切相关。为保证本方案的工程实际应用效果，需构建高质量的专业领域测试用例知识库；同时，通用 LLM 的基础能力有限，经过微调的 LLM 可进一步增强其专业领域测试用例生成的能力。

5. 结语与展望

本文提出一种基于 LLM 的软件测试用例智能生成方案，主要包括专业领域测试用例向量知识库构建、检索增强生成、提示词工程、基于人机交互的总结反思优化，改变了传统全人工的软件测试用例设计工作模式，有效提高了历史项目积累的典型知识的复用率，增强了大模型专业领域背景知识和测试用例生成能力，提高了测试用例设计的工作效率，通过人机交互进一步确保了测试用例的质量。通过实验可知，用本文的方法能够明显提升大模型生成测试用例的能力、缩短测试用例设计周期，可进一步推广到工程实践中应用。后续，将进一步研究基于知识图谱、多模态、多智能体等相关技术的 LLM 测试用例智能生成方法，进一步提升大模型生成测试用例的能力。

参考文献

[1] 张清睿, 黄松, 孙乐乐. Web 功能自动化测试综述[J]. 软件导刊, 2023, 22(3): 227-236.

[2] Chen, Y.H., Hu, Z.H., Zhi, C., *et al.* (2023) ChatUniTest: A Framework for LLM-Based Test Generation. arXiv: 2305.04764.

[3] Tang, Y., Liu, Z., Zhou, Z. and Luo, X. (2024) ChatGPT vs SBST: A Comparative Assessment of Unit Test Suite Generation. *IEEE Transactions on Software Engineering*, **50**, 1340-1359. https://doi.org/10.1109/tse.2024.3382365

[4] Yang, L., Yang, C., Gao, S., Wang, W., Wang, B., Zhu, Q., *et al.* (2024) On the Evaluation of Large Language Models in Unit Test Generation. *Proceedings of the* 39*th IEEE/ACM International Conference on Automated Software Engineering*, Sacramento, 27 October-1 November 2024, 1607-1619. https://doi.org/10.1145/3691620.3695529

[5] Roy Chowdhury, S., Sridhara, G., Raghavan, A.K., Bose, J., Mazumdar, S., Singh, H., *et al.* (2024) Static Program Analysis Guided LLM Based Unit Test Generation. *Proceedings of the* 8*th International Conference on Data Science and Management of Data* (12*th ACM IKDD CODS and* 30*th COMAD*), Jodhpur, 18-21 December 2024, 279-283. https://doi.org/10.1145/3703323.3703742

[6] 杨浠, 熊盼, 郑旭飞, 等. AIGC 辅助软件单元测试的研究[J]. 人工智能科学与工程, 2024(4): 31-41.

[7] 王赞, 王莹, 陈碧欢, 姚远, 张敏灵. 大模型下的软件质量保障专题前言[J]. 软件学报, 2025, 36(6): 2401-2403.