软件学报 ISSN 1000-9825, CODEN RUXUEW
2026,37(2):543−562 [doi: 10.13328/j.cnki.jos.007421] [CSTR: 32375.14.jos.007421]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

智能问答系统逻辑推理测试*

沈庆超, 李行健, 姜佳君, 陈俊洁, 齐一先, 王 赞

(天津大学 智能与计算学部, 天津 300350)

通信作者: 姜佳君, E-mail: jiangjiajun@tju.edu.cn

摘 要: 智能问答系统利用信息检索和自然语言处理技术, 实现对问题的自动化回复. 然而, 与其他人工智能软件相似, 智能问答系统同样存在缺陷. 存在缺陷的智能问答系统会降低用户体验, 造成企业的经济损失, 甚至引发社会层面的恐慌. 因此, 及时检测并修复智能问答系统中的缺陷至关重要. 目前, 智能问答系统自动测试方法主要分为两类. 其一, 基于问题与预测答案合成假定事实, 并基于假定事实生成新问题和预期答案, 以此揭示问答系统中的缺陷. 其二, 从现有数据集中提取不影响原问题答案的知识片段并融入原始测试输入中生成答案一致的新测试输入, 实现对问答系统的缺陷检测任务. 然而, 这两类方法均着重于测试模型的语义理解能力, 未能充分测试模型的逻辑推理能力. 此外, 这两类方法分别依赖于问答系统的回答范式和模型自带的数据集来生成新的测试用例, 限制了其在基于大规模语言模型的问答系统中的测试效能. 针对上述挑战, 提出一种逻辑引导的蜕变测试技术 QALT. QALT 设计了 3 种逻辑相关的蜕变关系, 并使用了语义相似度度量和依存句法分析等技术指导生成高质量的测试用例, 实现对智能问答系统的精准测试. 实验结果表明, QALT 在两类智能问答系统上一共检测 9247 个缺陷, 分别比当前两种最先进的技术 (即 QAQA 和 QAAskeR) 多检测 3150 和 3897 个缺陷. 基于人工采样标注结果的统计分析, QALT 在两个智能问答系统上检测到真阳性缺陷的期望数量总和为 8073, 预期比 QAQA 和 QAAskeR 分别多检测 2142 和 4867 个真阳性缺陷. 此外, 使用 QALT 生成的测试输入通过模型微调对被测软件中的缺陷进行修复. 微调后模型的错误率成功地从 22.33% 降至 14.37%.

关键词: 智能问答系统; 测试用例生成; 蜕变测试; 大型语言模型

中图法分类号: TP311

中文引用格式: 沈庆超, 李行健, 姜佳君, 陈俊洁, 齐一先, 王赞. 智能问答系统逻辑推理测试. 软件学报, 2026, 37(2): 543–562. http://www.jos.org.cn/1000-9825/7421.htm

英文引用格式: Shen QC, Li XJ, Jiang JJ, Chen JJ, Qi YX, Wang Z. Logical Reasoning Testing of Intelligent Question Answering System. Ruan Jian Xue Bao/Journal of Software, 2026, 37(2): 543–562 (in Chinese). http://www.jos.org.cn/1000-9825/7421.htm

Logical Reasoning Testing of Intelligent Question Answering System

SHEN Qing-Chao, LI Xing-Jian, JIANG Jia-Jun, CHEN Jun-Jie, QI Yi-Xian, WANG Zan

(College of Intelligence and Computing, Tianjin University, Tianjin 300350, China)

Abstract: Intelligent question answering (QA) system utilizes information retrieval and natural language processing techniques to deliver automated responses to user inquiries. Like other artificial intelligence software, intelligent QA system is prone to bugs. These bugs can degrade user experience, cause financial losses, or even trigger social panic. Therefore, it is crucial to detect and fix bugs in intelligent QA system promptly. Automated testing approaches fall into two categories. The first approach synthesizes hypothetical facts based on questions and predicted answers, then generates new questions and expected answers to detect bugs. The second approach generates semantically equivalent test inputs by injecting knowledge from existing datasets, ensuring the answer to the question remains unchanged.

* 基金项目: 国家自然科学基金 (62202324, 62322208, 62472310)
沈庆超、李行健有相同贡献.
收稿时间: 2024-01-30; 修改时间: 2024-06-30, 2025-02-20; 采用时间: 2025-03-06; jos 在线出版时间: 2025-07-23
CNKI 网络首发时间: 2025-07-23

However, both methods have limitations in practical use. They rely heavily on the intelligent QA system's output or training set, which results in poor testing effectiveness and generalization, especially for large-language-model-based intelligent QA systems. Moreover, these methods primarily assess semantic understanding while neglecting the logical reasoning capabilities of intelligent QA system. To address this gap, a logic-guided testing technique named QALT is proposed. It designs three logically related metamorphic relations and uses semantic similarity measurement and dependency parsing to generate high-quality test cases. The experimental results show that QALT detected a total of 9247 bugs in two different intelligent QA systems, which is 3150 and 3897 more bugs than the two current state-of-the-art techniques (i.e., QAQA and QAAskeR), respectively. Based on the statistical analysis of manually labeled results, QALT detects approximately 8073 true bugs, which is 2142 more than QAQA and 4867 more than QAAskeR. Moreover, the test inputs generated by QALT successfully reduce the MR violation rate from 22.33% to 14.37% when used for fine-tuning the intelligent QA system under test.

Key words: intelligent question answering (QA) system; test case generation; metamorphic testing; large language model (LLM)

1 引 言

智能问答系统, 运用自然语言处理 (natural language processing, NLP) 技术, 回应人们提出的各类问题. 它能够通过检索知识库或上下文信息推理出问题的答案[1]. 随着 NLP 技术的迅速发展, 智能问答系统得到了广泛的关注和研究[2−4]. 在智能客户服务系统 (如 AlphaChat[5]) 和虚拟助手 (如 Siri[6]) 等多个领域中已被广泛应用, 大幅提升了人们的工作效率和生活质量. 当前大型语言模型 (large language model, LLM) 的出现进一步增强了问答系统的效能和准确性[7,8].

与机器翻译系统[9,10]和自动驾驶系统[11]类似, 智能问答系统也存在缺陷. 具体而言, 智能问答系统可能提供不正确的答案, 从而降低用户的使用体验, 给企业带来经济损失. 更严重的是, 智能问答系统对于敏感问题的错误回答可能引发用户的强烈不满, 甚至带来社会恐慌. 因此, 及时地揭露并修复智能问答系统中的缺陷至关重要. 由于智能问答系统与机器翻译软件等其他类型的智能软件在核心功能和使用方式上存在巨大差异, 因此需要针对智能问答系统的具体功能特征设计相应的测试方法. 近年来, 针对智能问答系统的测试技术已经陆续被提出[12−15]. 它们可被分为基于参考答案的测试技术和基于蜕变关系 (metamorphic relations, MR) 的自动化测试技术. 基于参考答案的测试技术要求开发者为每个测试输入手动标注正确答案, 即根据知识库或上下文为每个问题提供参考答案[13,15]. 这种技术存在人工标注工作量大、成本高、测试充分性不足等局限性[16]. 为了克服人工标注答案的局限性, Chen 等人[16]首次提出了一种基于蜕变关系的自动化问答系统测试技术 QAAskeR. 紧接着, 为了解决已有方法生成测试技术存在较高误报的问题, Shen 等人[17]进一步提出一种基于句子级别变异的自动化智能问答系统测试技术 QAQA.

尽管 QAAskeR 和 QAQA 属于前沿的智能问答系统测试技术, 但它们在测试基于 LLM 的智能问答系统时存在一定的局限性. 这些局限性主要源于已有测试方法对特定数据或条件的依赖. 具体而言, QAAskeR 依赖于问答模型返回的答案进行规则匹配以合成所需信息, 导致复杂的答案可能阻碍有效测试输入的生成. 然而, LLM 在生成答案时通常遵循最大化用户辅助的原则[8], 其回答相较传统问答系统更为详尽, 这一特性在一定程度上影响了 QAAskeR 的泛用性. 例如, 对于给定上下文, 存在特定问题"How are the certain costs which are difficult to avoid shared?" QAAskeR 期望获得尽可能简洁的答案, 例如"by everyone", 以便于规则匹配. 但 LLM 给出了细致的回答, 例如"The certain costs which are difficult to avoid are shared by everyone." 这使得 QAAskeR 难以通过规则匹配的方法从回答中提取所需信息, 从而影响了生成测试用例的合法性. 同样地, 虽然 QAQA 提高了测试输入的揭错率, 也规避了部分不合理测试输入的生成, 但使用该技术时需要获取待测智能问答系统的自带的数据集. 然而, 现有的 LLM 的数据集通常会被公司视为商业机密而不公开. 此外, 目前 LLM 使用的数据集庞大且多样化, 包含了广泛的主题和语境, 即使我们能够获取部分项目的数据集, 从数据集中提取与特定测试问题高度相关的语料仍然存在挑战. 此外, QAAskeR 和 QAQA 的测试内容主要集中在评估智能问答系统的语义理解能力, 忽视了对问答系统逻辑推理能力的测试. 尽管语义理解是智能问答系统的核心组成部分, 能使智能问答系统根据问题或上下文中的一处知识点直接推理出问题的答案, 但缺少对逻辑推理能力方面的测试可能导致系统在遇到复杂问题时无法给出准确或合理的答案. 逻辑推理能力的展现在于智能问答系统是否可以根据问题或上下文中的多处知识点来综合回答问

题. 然而, 在引入逻辑关系的过程中通常会遇到两类问题, 分别为: (1) 引入的推理关系数量不可控: 在改变原始问题逻辑时, 控制引入问题的推理关系数量是有必要的. 这样可以自主控制问题的难度, 从多个方面对智能问答系统的逻辑推理能力进行测试; (2) 缺乏上下文: 某些复杂推理问题需要依赖于更广泛的背景知识和更全面的上下文信息, 如何构造包含推理关系的合法测试用例极具挑战. 为此, 开发一种可以控制引入推理关系数量并能补充上下文信息的方法用于测试智能问答系统的逻辑推理能力变得十分重要. 在提升测试方法泛用性和高效性的同时, 我们也需要关注测试方法本身的自然性、误报率等关键特性. 其中, 自然性反映了检测到的缺陷在现实场景中出现的可能性. 自然性越高表明测试方法发现的缺陷在实际应用中越常见, 其潜在危害也越大. 真阳性率则体现了测试方法报告缺陷的准确性. 真阳性率越高说明测试方法的可信度越高.

为了弥补现有智能问答系统测试技术的不足, 本文提出了一种逻辑关系引导的蜕变测试方法 QALT. 与已有研究方法类似[16,17], QALT 依托蜕变测试技术避免手动标注数据的局限性. 针对现有方法在测试智能问答系统逻辑推理能力和测试场景覆盖方面的局限性, QALT 设计了一组逻辑推理相关的蜕变关系. 具体而言, QALT 借助语言模型在原始测试输入中插入一条与问题相关的逻辑推理关系, 从而生成与原始测试输入语义等价的新测试输入. 这种方式避免了新生成的测试用例依托于上下文和回答范式的局限性, 使得测试方法具有广泛适用性. 同时, 新引入的逻辑推理关系可以使得新问题需要综合上下文中的多处知识点来回答, 从而可以深度测试智能问答系统的推理能力. 此外, 为了保证生成测试输入的自然性, QALT 分别提出了关键变异位置和关键名词词组的选择策略. 其中, 关键变异位置选择策略通过使用语义相似度度量方法在上下文中找到与问题直接相关的语句作为变异插入位点, 从而使得变异生成的上下文更加流畅自然. 同理, QALT 选择与上下文相关度高的名词词组进行变异能够保证新生成推理句子与当前上下文所讨论话题直接相关, 从而使得最终生成的新测试输入更加自然真实. 最后, 对于所有的测试技术, 保证其拥有较低的误报率将会提升测试的自动化程度. 为此, QALT 采用基于依存句法分析的选择策略对不合法的测试输入进行自动化过滤以降低测试的误报率.

为了验证 QALT 的有效性, 本文选择了 ChatGPT[7]和 ChatGLM-6B[18]两款前沿的大型语言模型作为实验待测软件. 3 种不同格式且被广泛使用的问答数据集 (即 BoolQ[4], SQuAD2[19]和 NarrativeQA[20]) 被用作 QALT 的原始测试输入. 作为目前最先进的智能问答系统测试技术, QAQA 与 QAAskeR 被选为本文的对比方法. 我们分别评估了 QALT 缺陷检测能力, 不同组件的有效性和 QALT 生成的测试输入对修复智能问答系统缺陷修复的有效性. 实验结果表明, QALT 在两类待测问答系统上分别比 QAQA 和 QAAskeR 多检测出 3 150 和 3 897 个缺陷. 同时, QALT 的不同组件能够有效地保证生成测试输入的高质量 (高自然性和低误报率). 此外, 本文还观察到基于高阶变异的蜕变关系可以检测到最多的缺陷, 证明基于高阶变异的蜕变关系有着更强的揭错能力. 最后, 使用 QALT 生成的数据对智能问答模型进行微调, 成功地将开源的智能问答系统 ChatGLM 的错误率从 22.33% 减少到 14.37%.

本文第 2 节介绍智能问答系统测试的研究现状. 第 3 节介绍智能问答系统的背景知识. 第 4 节介绍本文构建的逻辑引导的智能问答系统蜕变测试技术. 第 5 节通过对比实验验证所提方法的有效性和高效性. 第 6 节讨论影响方法效度的因素. 最后总结全文.

2 相关工作

目前, 关于智能问答系统质量与安全问题的测试研究正受到越来越多研究者的关注. 根据质量评估的手段不同, 已有的研究工作可以分为对抗样本攻击和测试技术.

对抗样本攻击方法通过对模型输入添加微小的扰动来试图引起系统输出的变化, 旨在探测系统的脆弱性和潜在的安全漏洞. 一个经典的针对 NLP 模型的对抗样本攻击方法是 Eger 等人[12]提出的字符级别的对抗攻击方法. 该方法设计了单词内部字符顺序打乱 (inner-shuffle)、特殊符号插入 (intruders)、元音字符移除 (disemvoweling) 等 10 种字符级别变异算子, 用于测试 NLP 模型的鲁棒性. 该技术能够很好地评估 NLP 模型的单词识别鲁棒性. 除此之外, Sharma 等人[21]、Tang 等人[22]、Sheng 等人[23]和 Walmer 等人[24]通过对图片的像素进行微调实现针对视觉问答系统的鲁棒性评测.

与对抗样本攻击的目的不同, 测试技术旨在生成强揭错能力的测试用例并构造相应的测试预言实现对智能问

答系统的缺陷检测. 根据测试用例的来源, 现有的研究工作可分为基于参考答案的测试技术和基于蜕变关系的测试技术. 基于参考答案的测试技术依赖于人工标记的测试输入来构建数据集. 典型的数据集包括 BoolQ[4]、BoolQ-NP[25]、MultiRC[26]、SQuAD1.1[19]、SQuAD2[27]、NewsQA[28]、Quoref[29]、NarQA[20]、NatQA[30]和 DROP[31]等. 然而, 这种技术的局限性包括高昂的人工标注成本、无法支持智能问答系统上线后的实时缺陷检测以及测试充分性的不足. 此外, 每个数据集通常只针对特定格式的问答数据, 如判断型、提取型、抽象型等. 根据 Chen 等人[16]的研究, 仅在这些数据集上进行基于参考答案的测试扩展性差且测试结果存在偏差.

基于蜕变关系的测试技术[27]是一种高效的自动测试方法, 它根据待测程序的领域知识来构造蜕变关系, 并通过判断测试输入组是否满足这些蜕变关系来检测程序中的缺陷. Chen 等人[16]首次提出了针对智能问答的蜕变测试技术 QAAskeR, 该方法摆脱了传统方法需要人工标注问题答案的限制. 具体而言, QAAskeR 首先根据一个给定问题及其由被测智能问答系统产生的答案, 合成一个假设的事实. 接着, 基于这个假设的事实, QAAskeR 会生成一个全新的问题及其对应的预期答案. 如果这个新问题的实际预测答案与预期答案不符, 则认为检测到智能问答系统中的一个缺陷. 此外, Tu 等人[32]针对智能问答系统提出了 4 种与问题本身直接相关的蜕变关系和 5 种与上下文相关的蜕变关系. 前者包括对问题中字母大小写转换、语义等价转述、反义词替换以及更换问题的主语. 而后者则涉及对上下文进行大小写转换、句子顺序重新排列、插入或删除不影响答案的句子以及替换与答案相关的单词. 这些蜕变关系的设计旨在评估智能问答系统的准确性. Shen 等人[17]提出了一种先进的自动化智能问答系统测试技术 QAQA, 该方法通过在数据集中检索与测试输入最相似的语料, 将其作为冗余信息插入原始问题或上下文中, 从而生成语义等价的高保真测试用例. 这一方法避免了直接修改原始内容, 实现了对系统语义理解能力的精准测试. 目前, QAAskeR 和 QAQA 是智能问答系统测试领域中的前沿技术.

本文提出了一种逻辑引导的蜕变测试技术 QALT. 与已有方法不同, QALT 设计了 3 种逻辑推理相关的蜕变关系, 实现对智能问答系统逻辑推理能力的测试. 相比之下, QAQA 和 QAAskeR 则主要关注测试智能问答系统的语义理解能力, 且迁移到基于大型语言模型的智能问答系统时存在一定的局限性, 威胁生成测试用例的合法性或自然性. QALT 摆脱了对回答范式和数据集的依赖, 具备更强的泛化能力.

3 背景知识: 智能问答系统

智能问答系统是一种基于人工智能技术的应用, 致力于理解和回答用户提出的自然语言问题. 该系统不仅能够分析和理解问题, 还能从信息库或者上下文中提取有用信息并给出答案[33]. 根据其信息源的不同, 智能问答系统可以划分为封闭世界 (closed-world) 和开放世界 (open-world) 两种类型[16]. 封闭世界智能问答系统以问题及其上下文作为输入, 专注于特定领域或主题, 限定了问答的范围. 这种系统通常通过深入理解特定领域的语境来提供更准确的答案. 而开放世界智能问答系统则仅以问题为输入, 并依赖开放知识库作为其知识源. 开放世界问答系统旨在处理各种主题和领域的问题, 不受特定领域的限制. 这种系统通常需要具备更广泛的知识储备和理解能力, 以便回答用户提出的任何问题. 开放世界问答系统的挑战在于涉及多领域知识、语义理解和推理, 因此其设计和实现更为复杂. 相较于开放世界问答系统, 封闭世界问答系统更容易实现和优化, 并有着更高的可用性. 同时, 封闭世界智能问答系统是开放世界智能问答系统的核心组成部分. 此外, 封闭世界智能问答系统有众多公开可用的模型和数据集, 便于进行研究. 因此, 本文与对比方法 QAAskeR 以及 QAQA 的研究对象一致, 专注于对封闭世界智能问答系统的测试.

近年来, 多种 QA 数据集的构建促进了 QA 算法的发展. 这些数据集覆盖了多种问答类型, 如布尔型 QA (答案为“Yes”或“No”)[4]、提取型 QA (答案为上下文中的子字符串)[27]、抽象型 QA (答案为基于上下文推理的自由格式文本)[30]. 针对这些数据集, 多种 QA 算法已被提出, 例如 MultiQA[34]和 DOCQA[35], 这些算法通常针对特定类型的 QA 任务设计. Khashabi 等人[36]首次提出在单一模型上处理不同类型的封闭世界 QA 任务. 此后, 随着大型语言模型的崛起, 研究人员在人工智能领域取得了显著进展. 他们不再局限于训练专门针对单一问题的模型, 而是转向开发能够处理多种不同类型问答任务的通用模型. 这种方法的优势在于, 一个模型可以灵活应对多种情况, 从而提高效率和泛用性.

ChatGPT[7]和 ChatGLM[37]就是这种新趋势的代表. 其中 ChatGPT 是由 OpenAI 公司开发的一种人工智能技术, 它基于 GPT-3.5 模型, 使用指令微调和基于人类反馈的强化学习进行训练, 对人工智能领域产生了深远影响. ChatGPT 具有强大的语言生成能力, 能够生成连贯、多样化的文本, 适用于智能问答、机器翻译等领域. ChatGPT 的优点在于语言生成能力强, 适用于多样化任务, 但该模型规模较大, 部署和运行成本高. ChatGLM 则是由清华大学 KEG 实验室和智谱 AI 公司共同训练的语言模型, 这种模型架构基于 GLM (general language model) 架构, 具有 62 亿参数. 它结合了模型训练和有监督微调技术, 优化了中文问答和对话能力, 生成符合人类偏好的回答. ChatGLM 适用于对话系统、垂直领域知识问答等场景. ChatGLM 的优点在于模型已经开源, 并且支持多语言问答和对话能力优, 部署门槛低.

4 逻辑引导的蜕变测试技术

4.1 整体架构

本文采用蜕变测试技术[38]解决基于参考答案的测试技术依赖人工标注的局限性. 蜕变测试是一种基于特定领域知识构建蜕变关系来检测软件缺陷的方法. 首先定义一个包含问题和上下文的测试输入, 表示为 $t=\{q,c\}$. 然后, 由被测智能问答系统对这个输入生成一个预测答案, 表示为 $P(t)$. 接下来, QALT 基于 t 构造一个新的测试输入 $t'=\{q',c'\}$, 其中 t' 语义等价于 t. 因此, $P(t')$ 应在语义上与 $P(t)$ 等价. 如果 $P(t)$ 与 $P(t')$ 语义不等价, 则表明该测试用例检测到智能问答系统中的一个缺陷.

相较于已有的测试方法, QALT 最显著的特点在于避免了测试方法对问答系统回答范式和数据集本身的依赖, 并且能够测试智能问答系统的逻辑推理能力. 具体来说, QAAskeR 在生成测试用例时需要获取问答系统在原始测试输入上的回答, 生成测试用例的有效性依赖于模型的回答范式. 然而, 基于大型语言模型的问答模型的回答范式并不固定限制了该方法的泛用性. 与此同时, QAQA 在生成测试输入的过程中需要将待测模型的数据集作为搜索空间, 在无法访问待测模型的数据集时, 该方法不再适用. 与 QAQA 和 QAAskeR 不同, 为了摆脱测试方法对模型回答范式和数据集本身的依赖性, QALT 通过自然语言生成模型 GPT-Neo 实现了测试输入中关键名词词组的变异, 确保变异前后测试输入语义和逻辑的一致性. 这一做法有效地解决了现有方法测试场景受限的问题, 同时实现了对智能问答系统逻辑推理能力的测试目的.

智能问答系统旨在用自然语言回答人类提出的问题, 因此保证 QALT 生成测试用例的自然性非常重要. 更自然的测试输入所揭示的错误更有可能在实践中对用户体验产生负面影响. 为了实现这一目标, QALT 针对关键变异位置和关键名词词组选择分别设计相应的选择策略以提升变异测试用例的自然性. 与此同时, Huang 等人[39]研究发现, 在对 4 种不同类型的 NLP 软件进行测试时, 采用基于单词级别变异的测试输入方法会产生高达 44% 的误报率. 误报率高的测试技术会降低测试方法的可靠性, 同时增加研究人员因误报数据分析而产生的人工成本. 因此, 确保测试结果的低误报率同样重要. 为此, 本文根据变异方法生成测试用例的特点进一步提出了基于依存分析的选择策略来提高测试结果的准确性.

图 1 展示了 QALT 的技术概览. 如图所示, QALT 包含 3 个主要阶段: 测试用例生成阶段、测试输入选择阶段和测试输入执行阶段. 其中, QALT 的前两个阶段负责生成高质量的包含逻辑推理的测试用例 (包括高揭错能力、低误报率、高自然率), 最后一个阶段为测试预言的构造. 这 3 个阶段共同组成了 QA 系统的逻辑测试的完整流程. 具体来说, 在测试输入生成阶段, QALT 设计了 3 种逻辑推理相关的蜕变关系 (第 4.2 节), 对于每个原始测试输入 (包括问题和上下文), QALT 随机从本文设定的蜕变关系库中选取一种具体蜕变关系. 接着, 为了生成自然性高的测试输入, QALT 根据所选取的蜕变关系及问题和上下文的特点, 确定原始测试输入中最适合进行变异的位置和待变异的名词词组, 基于选定的名词词组生成相应的描述性语句, 并与原始测试输入相结合生成候选测试用例 (第 4.3 节). 接着, 在测试输入选择阶段, QALT 基于依存分析方法从多个候选测试输入中选择出最佳的测试用例用于后续的测试任务 (第 4.4 节). 最后, 在测试输入执行阶段, QALT 通过文本相似度计算来判断测试输入的预测答案与预期答案是否在语义上一致. 如果两者存在显著的语义差异, 则认为 QALT 检测到了智能问答系统中的一个缺陷.

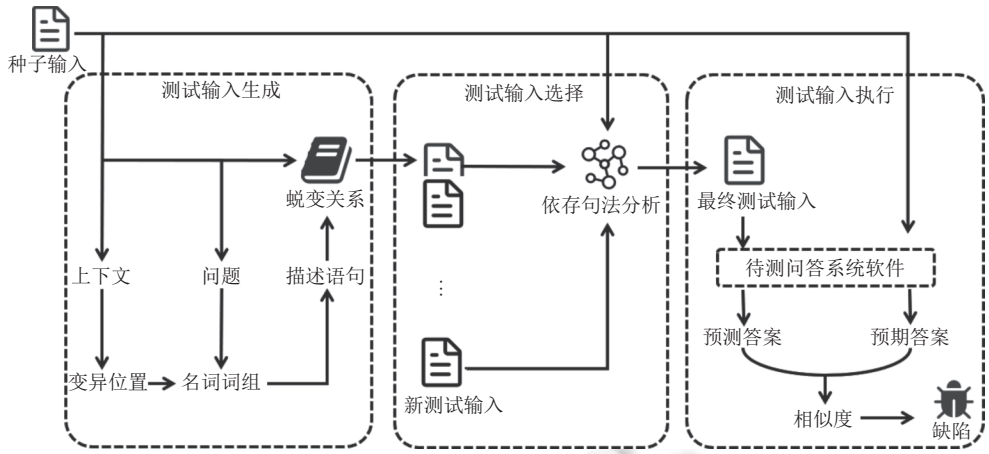


图 1 QALT 架构图

4.2 蜕变关系定义

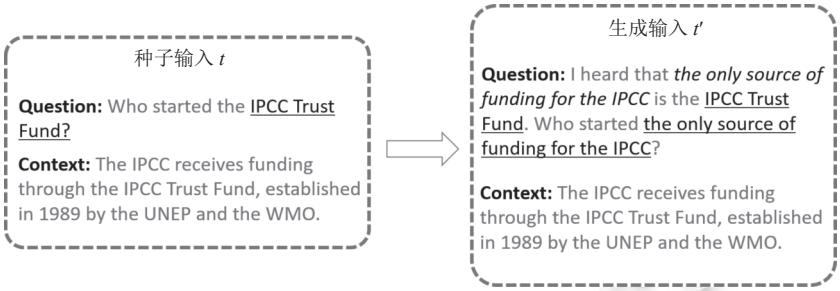
在实施蜕变测试过程中, 需要建立一系列蜕变关系来支持测试过程. 鉴于智能问答系统应具备问题推理和上下文推理两项核心能力, 因此, 本文分别设计了两种相应的蜕变关系. QALT 对原始测试输入 t 进行变异的关键在于不破坏 t 原始语义的前提下, 引入额外的逻辑推理关系. 具体来说, QALT 提出了两种变异规则, 分别适用于支持问题推理和上下文推理的蜕变关系构建. 为了确保原始测试输入 t 与变异测试输入 t'之间保持语义等价性, QALT 变异规则的核心思想是用等价的描述性短语替换选定的目标名词词组, 并将他们之间的等价关系作为背景知识嵌入测试输入中. 这种变异方法不会改变答案之间的等价关系.

● MR1: 等价推理问题 (equivalence reasoning problem, ERQ). 关于智能问答系统的问题推理能力, MR1 构建了一种等价推理问题的蜕变关系, 用于生成与原始测试输入问题等价的替代问题. 具体来说, 给定测试输入 $t=\{q,c\}$, 如果 QALT 构造出另一个输入 $t'=\{q',c\}$, 其中 q 与 q' 在逻辑上语义等价, 那么被测智能问答系统应给出语义上一致的答案 $P(t)$ 和 $P(t')$. 为了构造 q', MR1 从问题 q 中选择一个名词词组 n, 然后, 将选定的名词词组 n 输入到生成模型 GTP-Neo 中生成一个描述性语句 d. d 的格式为“It is acknowledged that A is B”. 其中 A 为选定的名词词组 n, B 为与 A 语义等价的描述性短语. 最后, MR1 用短语 B 替换问题 q 中的名词词组 A, 并在 q 中额外添加描述性语句 d 作为推理知识源形成一个新的问题 q'. 至此, 变异后的测试输入便在原始测试输入的问题上新增了一层推理关系 $d \Rightarrow n$ (其中 $\Rightarrow$ 表示推理关系). 这样的变异方法确保了 q' 在经过逻辑推理后可以还原为 q, 同时保留了原始上下文的完整语义, 不改变答案的相关性.

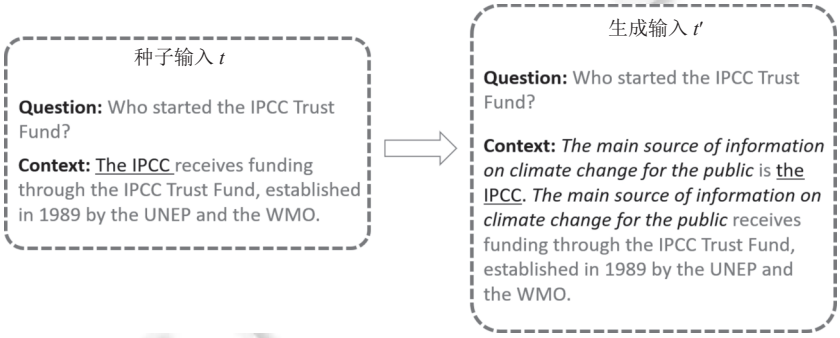
如图 2(a) 所示, QALT 将原问题“Who started the IPCC Trust Fund?”转换为“I heard that the only source of funding for the IPCC is the IPCC Trust Fund. Who started the only source of funding for the IPCC?”, 通过逻辑推理, 可以将问题中替换后的描述性语句还原为替换前的名词词组, 而不改变原问题的语义. 在此例中, 变异用的描述性语句是“the only source of funding for the IPCC”, 被替换的名词词组是“the IPCC Trust Fund”. 为了使句子与原问题更自然地融合, 本文复用了 QAQA 方法中的模板来组装句子, 例如“I heard that …”“It is acknowledged that …”.

● MR2: 等价推理上下文 (equivalence reasoning context, ERC). 关于智能问答系统的上下文推理能力, MR2 构建了一种等价推理上下文的蜕变关系, 用于生成与原始测试输入问题等价的替代上下文. 具体来说, 给定测试输入 $t=\{q,c\}$, 若构造出 $t'=\{q,c'\}$, 其中 c 与 c' 在逻辑上等价, 那么在问题 q 下, $P(t)$ 与 $P(t')$ 的答案语义一致. 为了构造 c', MR2 选择 c 中的一个句子作为变异位置, 并从该句中选择一个名词词组 n. 然后, 将 n 输入生成模型中得到一个描述性语句 d. d 的格式为“It is acknowledged that A is B”, 其中 A 为选定的名词词组 n, B 为与 A 语义等价的描述性短语. 最后, MR2 用短语 B 替换上下文 c 中的名词词组 A, 并在 c 中额外添加描述性语句 d 作为推理知识源生成一个新的上下文 c'. 至此, 变异后的测试输入便在原始测试输入的上下文上新增了一层推理关系 $d \Rightarrow n$. 这样

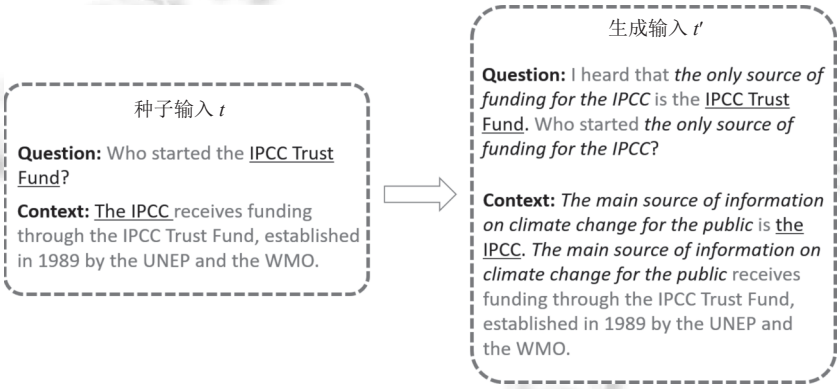
的变异方法确保了 c' 在经过逻辑推理后可以还原为 c, 从而保留了原始上下文的完整语义, 新问题的答案与原始问题的答案仍然相同.



(a) MR1: 等价推理问题



(b) MR2: 等价推理上下文



(c) MR3: 等价推理问题及上下文

图 2　3 种蜕变关系的例子

例如, 图 2(b) 展示了一个例子, 阐释了如何在给定测试输入的上下文中应用变异构建 ERC 蜕变关系. 在此案例中, QALT 将原始上下文“The IPCC receives funding through the IPCC Trust Fund, established in 1989 by the UNEP and the WMO.” 转换为“The main source of information on climate change for the public is the IPCC. The main source of information on climate change for the public receives funding through the IPCC Trust Fund, established in 1989 by the UNEP and the WMO.” 通过逻辑推理, 可以将上下文中替换后的描述性语句还原为替换前的名词词组, 而不改变原始上下文的语义. 此例中的描述性句子为“the main source of information on climate change”, 目标名词词组为“The IPCC”. 新上下文虽然添加了额外的推理层级, 但并未改变与答案相关的原始事实, 因此, 对于变异后等价测试输入的答案在语义上应是一致的.

● MR3: 等价推理问题及上下文 (equivalence reasoning question & context, ERQC). 基于上述两种变异规则, 本文还进一步构造了一个高阶的变异规则, 用于生成更复杂的蜕变关系. MR3 构建了一种综合性的蜕变关系, 该蜕变关系结合了 MR1 和 MR2 的概念, 通过同时生成等价的问题和上下文, 以测试智能问答系统对于问题和上下文的复合推理能力. 具体来说, 给定一个测试输入 $t=\{q,c\}$, 如果构造出一个新的测试输入 $t'=\{q',c'\}$, 其中 q 与 q', c 与 c' 均在逻辑上等价, 那么智能问答系统应当能够基于其问题推理能力和上下文推理能力, 生成语义上一致的答案 $P(t)$ 和 $P(t')$. 在构造 t'的过程中, QALT 分别在问题 q 和上下文 c 中插入描述性语句 d_q 和 d_c, 并用这些语句中等价描述替换目标名词词组 n_q 和 n_c. 这种方法分别在原始测试输入的问题和上下文中新增了一层推理关系, 但模型的预测答案不应改变.

图 2(c) 使用一个例子展示了如何使用变异构建 ERQC 蜕变关系. 在该例中, 描述性句子“the only source of funding for the IPCC”和“the main source of information on climate change”分别被插入到原始问题和上下文中, 形成新的问题以及新的上下文. 这些新的问题和上下文仅添加了额外的补充事实, 并建立了一层额外的推理关系, 而不会破坏其他有助于获得答案的原始事实. 因此, 新生成的测试输入的答案在语义上应与原始问题的答案相同.

4.3 测试输入自然性增强策略

应用上述 3 种蜕变关系生成测试输入面临的一个重要性挑战是如何选择变异组件, 包括变异发生的句子位置和施加变异的具体名词词组, 变异位置的选择对生成测试输入的自然性有显著影响. 具体来说, 由于上下文通常由多个句子组成, 选取恰当的句子作为变异位置可以使变异发生在与问题直接相关的语句上, 从而增加变异后测试输入与原始测试输入的相关性. 同理, 选择与上下文相关度越高的名词词组越有可能生成与当前上下文所讨论话题有关的句子. 因此, 本文分别提出了关键变异位置和关键名词词组选择策略来保证生成测试输入的自然性.

4.3.1 关键变异位置选择策略

在应用 MR2 和 MR3 时, QALT 需要从上下文中选择变异发生的位置. 上下文中存在多个潜在变异位置可供选择, 虽然各个位置都可能揭示系统的缺陷, 但不是所有位置的变异都会迫使模型进行必要的逻辑推理. 由于推理问题答案的知识来源通常只包含在上下文部分的几个句子中. 这些句子包含了回答问题所需的关键要素, 与问题具有强相关性. 为了提升测试模型的逻辑推理能力且增强生成测试输入的自然性, QALT 需要在上下文中找到对回答问题最关键的句子并对其进行变异. 为此, QALT 分别计算问题与上下文中每个句子的语义相似度, 并选择语义相似度最高的句子作为关键变异位置.

关键变异位置的选择方法见算法 1. 由于问题是驱动智能问答系统预测的核心, QALT 提取测试输入中的问题作为目标语义向量 (第 3 行), 然后通过测量其语义与搜索空间 (上下文) 中每个句子的语义之间的相似度来指导变异位置的选择. 为了便于计算相似度, QALT 使用了先进的句子嵌入模型 SBERT[40]来将问题和上下文进行向量化, 该模型已被证明在句子语义表示方面是高效的. SBERT 有效地将问题和上下文中各个位置的句子转化为语义向量 (第 7 行), 通过计算问题向量与每个句子向量之间的余弦相似度来识别出与问题最相关的句子 (第 8–12 行), 并将其选作为最优变异位置, 用于后续测试输入的生成. 在进行变异前, QALT 需要对上下文中的句子进行预处理. 具体来说, QALT 使用共指消除模型 NeuralCoref[41]将指示代词替换为对应的实体名词 (第 4 行), 这是因为指示代词只有在特定语境下才有实际意义.

算法 1. 关键变异位置选择方法.

输入: q: 原始测试输入的问题; c: 原始测试输入的上下文;
输出: 上下文 c 中最适合变异的位置 p^*.

1. $max := -\infty$;
2. $p^* := \varnothing$;
3. $q\_vector = sentence\_embedding(q)$;
4. $c = eliminate\_coreference(c)$;
5. $p\_list = segment\_sentences(c)$; //$p\_list$ 为包含了上下文中所有句子的列表

```
6. foreach p in p_list do
7.    p_vector = sentence_embedding(p);
8.    sim_score = sentence_similarity(q_vector, p_vector);
9.    if sim_score > max then
10.         max = sim_score;
11.       p* = p;
12.   end
13. end
14. return p*
```

4.3.2 关键名词词组选择策略

除了选取关键变异位置之外, 名词词组的选择对测试模型的逻辑推理能力以及提升测试输入的自然性同样重要. 当变异过程中新增的语义信息与上下文中所讨论的主题不相关时, 这种测试输入被认为是不自然的. 尽管不自然的测试输入同样具有揭错能力, 但这类输入在现实生活中较少见, 因而相关缺陷的实际危害程度通常较低. 因此, QALT 在变异过程中使用关键名词词组选择方法选择与上下文所讨论主题最相关的名词词组.

具体来说, 在应用第 4.2 节所提出的蜕变关系时, 如果仅是随机选择一个名词词组进行变异, 往往难以生成符合自然语言习惯的测试输入. 如图 3 所示, 在原始测试输入“Where have oxygen bars been since 1990?”中存在“oxygen bars”与“1990”两个名词词组. 如果对于“1990”应用蜕变关系会产生如“It is acknowledged that 1990 is the year when the Hubble Space Telescope was launched. Where have oxygen bars been since the year when the Hubble Space Telescope was launched?” 这样的测试输入. 然而, “the Hubble Space Telescope”并不符合原测试输入所讨论的主题, 这样的问题在现实中可能并不存在. 为了解决这一问题, 本文提出了一种名词词组的选择策略, 以减少随机选择带来的测试输入不自然的问题. 该方法首先使用基于词图模型的关键词提取方法提取上下文中的核心名词, 随后通过计算相似度的方式选择待变异测试输入中的名词词组. 这种选择方法保证了所选名词词组与原始测试输入所讨论主题的相关程度, 以此提高测试输入的自然性.

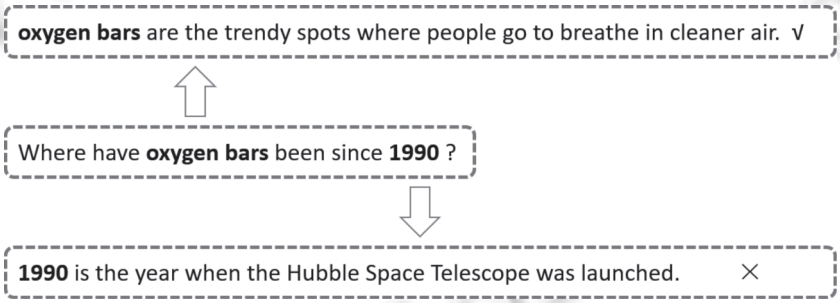


图 3 名词词组选择实例

在自然语言处理领域, 无监督关键词提取主要分为 3 类方法: 基于统计特征、基于主题模型和基于词图模型的方法. 基于统计特征的方法依赖于文档中词语的统计信息来抽取关键词; 而基于主题模型的方法则利用主题分布的特性进行关键词提取, 这两种方法通常在输入为多文档数据集时才可生效. 然而, 封闭世界问答系统通常处理的是单一文档, 所以这两种方法并不适用. 因此, QALT 选用基于词图模型的关键词提取方法作为基本方法进行关键词提取. 基于词图模型的关键词提取方法通过构建和分析文档的语言网络图, 并在这个图上寻找具有重要作用的词或者短语作为文档的关键词. 该方法的优点在于只依赖文档本身, 以此摆脱对于测试过程中对于智能问答系统数据集 (即多文档数据集) 的依赖性. 为此, 本文使用了 TextRank[42]算法提取上下文中的核心名词. TextRank 算法将上下文视为一个词的网络, 每个词是该算法中的一个节点. 该方法通过将词与词的语义关系表示为网络中的

链接, 然后根据词之间的共现关系构造网络, 得到无向有权边. 最后, TextRank 对节点权重进行倒序排序, 并按照重要程度依次获取核心名词.

关键变异位置的选择方法见算法 2. 在对测试输入进行变异时, 本文首先使用 TextRank 算法从原始测试输入中提取核心名词 (第 3 行), 然后运用 Phrase-BERT (简称 PBERT) 模型[43]将提取的关键词转化为向量表示 (第 4 行). PBERT 模型通过对比学习方法[44]对 BERT 模型[45]进行微调, 使其更适合于短语嵌入任务. 相比于原始的 BERT 模型, PBERT 模型在短语层面的语义表示方面表现更为准确[40,46]. QALT 使用 PBERT 将候选名词词组转换为向量形式 (第 8 行), 并计算与核心名词集合的平均余弦相似度 (第 9 行). 最后, QALT 选择其中相似度最高的名词词组 (第 10–13 行) 作为最终的变异名词词组. 基于该名词词组, QALT 将其输入生成模型并生成描述性语句, 进而对原始测试输入进行变异构造全新的测试输入. 需要注意的是, 在提取候选名词词组时, 需要注意当前选择的蜕变关系: 蜕变关系为 ERQ 时, 从测试输入的问题中提取名词词组 (第 5 行); 蜕变关系为 ERC 时, 从测试输入所选择的上下文变异位置中提取名词词组 (第 6 行); 蜕变关系为 ERQC 时, 分别执行上述操作.

算法 2. 关键名词词组选择方法.

输入: p: 上下文中选定的变异位置; q: 原始测试输入的问题; c: 原始测试输入的上下文; MR: 选定的蜕变关系;
输出: p 或 q 中最适合变异的名词词组 n*.

1. $max := -\infty$;
2. $n^* := \varnothing$;
3. *keywords_list* = *TextRank*(*c*); //*keywords_list* 为包含所有核心名词的列表
4. *keywords_vector* = *phrase_embedding*(*keywords_list*);
5. **if** *MR* == *ERO* **then** *noun_phrase_list* = *segment_noun_phrase*(*q*);
6. **else if** *MR* == *ERC* **then** *noun_phrase_list* = *segment_noun_phrase*(*p*);
7. **foreach** *n* **in** *noun_phrase_list* **do**
8. *n_vector* = *phrase_embedding*(*n*);
9. *sim_score* = *phrase_similarity*(*keywords_vector*, *n_vector*);
10. **if** *sim_score* > *max* **then**
11. *max* = *sim_score*;
12. *n** = *n*
13. **end**
14. **end**
15. **return** *n**

4.4 基于依存句法分析的选择策略

由于 QALT 合成的测试输入和原始测试输入的语法结构可能存在不一致, 从而进一步导致生成句子不合法. 具体来说, QALT 在生成测试输入的过程中会用描述性语句替换原始测试输入中的名词词组, 这种做法在某些情况下会破坏原有句子的语法结构, 从而引起变异后的测试输入的语义发生变化.

如图 4 所示, 原始问题“As a euphoric how is oxygen used in bars?”中名词词组“oxygen”被描述性语句“the most abundant component of air” 代替. 该变异使得在原始问题中“used in bars”的修饰对象则从“component” (替换前为“oxygen”) 变成了“air”. 此时句子语法结构发生变化, 使得变异后的测试输入与原始测试输入语义无法保持一致从而导致误报. 为了提升生成测试输入的质量, QALT 利用依存句法分析技术对句子的语法结构进行解析, 结合选择策略进一步降低测试输入的误报率, 从而提高所检测到缺陷的准确性.

如图 5 所示, 在应用选择策略时, QALT 首先生成 N 个 (本文中为 3) 候选的测试输入, 接着, QALT 将每个候选测试输入拆分成变异部分结构与未变异部分结构两个不同部分, 并分别输入筛除模块与排序模块. 在每个模块

中, 候选测试输入的对应部分分别与原始测试输入的对应结构进行对比, 并基于对比结果从 N 个候选测试输入中选出一条最优测试输入作为最终的测试任务. 其中, 筛除模块用于剔除不符合规范的测试输入, 排序模块用于筛选出最优的测试输入.

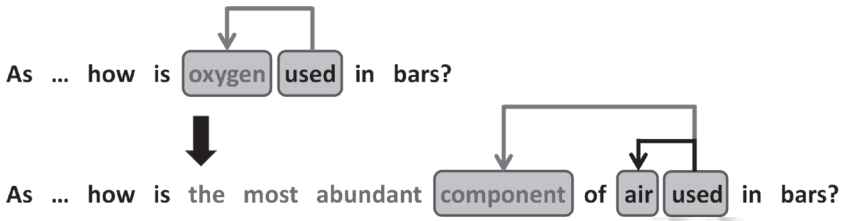


图 4 QALT 误报实例

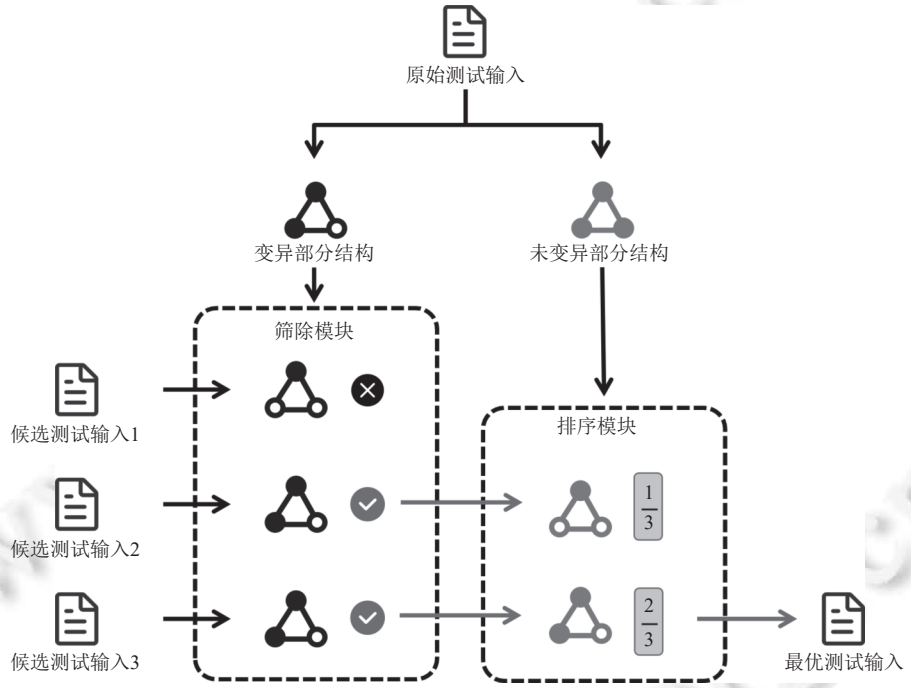


图 5 基于依存句法分析的选择策略

在筛除模块, QALT 主要关心测试输入中发生替换部分的结构 (即变异部分结构), 这部分结构由依靠生成模型得到的描述性语句所构成. 在替换名词词组的过程中, 由于描述性语句是作为一个整体嵌入在候选测试输入里, 所以描述性语句本身的语法结构不可变化, 如果发生变化则认为该候选测试输入不可用. 所以, 筛除模块专注于对比变异部分结构中各个词语的依存对象和依存关系, 以此严格保证变异前后变异部分结构的一致性. 如果任何一个词语的依存对象或者依存关系与原始测试输入不一致, 则认为当前测试输入的语义发生了变化, 从而排除这条测试输入.

在排序模块中, QALT 则主要关心测试输入中未发生替换的部分, 即除去测试输入中描述性语句部分外 (即未变异部分结构) 的各个词语, 使用该部分对应的依存对象和依存关系对测试输入进行排序. 具体来说, 本文使用依存句法分析直接获取句子中不同词语之间的依存关系. 依存关系可以使用三元组表示为<obj1, obj2, relation>. 其中 obj1 和 obj2 代表句子中的两个词语, 且 obj1 的依存对象为 obj2. relation 表示这两个词语之间的依存关系. 这些关系包括主谓关系, 动宾关系, 定语关系, 状语关系等. 排序过程中, QALT 会遍历未变异部分结构中的每个词语, 如果当前词语的依存关系或者依存对象出现不一致, 则认为出现一次错位, 随后按照公式 (1) 计算相似度得分, 并选出候选测试输入中相似度得分最高的测试输入作为最优测试输入.

$$\text{语法结构相似度} = 1 - \frac{\text{错位数量}}{\text{未替换部分词语总数}} \tag{1}$$

排序模块只对测试输入打分而不用于排除不合理的测试输入的原因是在一些句子结构中, 对名词词组的替换会导致句子中其他部分的依存关系类型或者依存关系出现变化, 但这种变化有一部分实际上是无害的, 并不导致句子语义发生变化. 所以 QALT 为了尽可能地保障句子语法和语义的一致性, 在所有候选测试输入中选择一致性得分最高的测试输入作为最优测试输入.

算法 3 展示了基于依存句法分析的选择策略. QALT 使用原始测试输入的语法结构作为模板, 将变异后的测试输入作为候选测试输入. 首先, QALT 使用描述性语句生成候选测试输入 (第 4 行). 接着, QALT 分别提取原始测试输入的句子结构和描述性语句结构 (第 5 行) 以及候选测试输入的变异部分结构和未变异部分结构 (第 6 行). 然后, QALT 通过对比二者的变异部分结构是否一致, 如果不一致则排除当前测试输入 (第 7 行). 排除不符合规范的测试输入之后, QALT 计算变异前后测试输入句子结构的相似度, 并选出其中未变异部分结构变化最小的测试输入 (第 8–12 行). 需要注意的是, 未变异部分结构的相似度通过计算依存关系类型和依存关系指向的差异作为其一致性得分结果, 计算过程中并不包括任何的变异部分结构. 筛除模块已经对变异部分做过检查, 排序模块的再次检查只会增加时间开销且不会带来任何额外收益.

算法 3. 基于依存句法分析的选择策略.

输入: *t*: 给定的测试输入; *statements_list*: 候选的描述性句子列表;
输出: *statements_list* 中最合适的句子 *s**.

```
1. max := −∞;
2. s* := ∅;
3. foreach s in statements_list do
4.     t' = apply_mutation(t, s);
5.     unmutated_structure, mutated_structure = dependency_parsing(t, s);
6.     unmutated_structure', mutated_structure' = dependency_parsing(t', s);
7.     if mutated_structure == mutated_structure' then
8.         sim_score = structure_sim(unmutated_structure, unmutated_structure');
9.         if sim_score > max then
10.            max = sim_score;
11.            s* = s;
12.        end
13.    end
14. end
15. return s*
```

5 实 验

为了验证本文提出的逻辑引导蜕变测试技术 QALT 的有效性, 我们提出了以下 3 个关键研究问题.

问题 1: QALT 对基于大型语言模型的智能问答系统的揭错能力如何?

问题 2: QALT 中每个关键组件是否有助于其整体有效性?

问题 3: QALT 生成的测试输入是否有助于改进智能问答系统和修复缺陷?

研究问题 1 的目的是评估 QALT 在面对当前基于大型语言模型的智能问答系统的揭错能力如何, 并给出量化的评估结果. 研究问题 2 的目的是验证 QALT 中每个组件的有效性, 需要分别验证变异组件选择策略、3 种不

同蜕变关系、基于依存分析的选择策略对 QALT 整体方法的贡献. 研究问题 3 的目的是验证使用 QALT 生成测试用例对问答系统模型进行修复的效果, 需要使用 QALT 生成的测试用例通过模型微调手段对错误进行修复, 并评估效果.

5.1 实验设计

5.1.1 待测系统

本文测试的智能问答系统是 ChatGPT 与 ChatGLM. 具体来说, 本文选用基于 GPT (generative pre-trained Transformer) 框架的语言模型 ChatGPT 与 ChatGLM-6B 进行测试. 选择这两个模型的原因如下: 1) 在模型性能方面, ChatGPT 是由 OpenAI 研发的强大的大型语言模型, 它在自然语言理解方面取得了显著进展. ChatGLM-6B 是一个开源的、支持中英双语的对话语言模型, 具有 62 亿参数. ChatGLM-6B 使用了和 ChatGPT 类似的技术进行优化, 包括使用了 1T 标识符 (tokens) 的中英文双语语料库进行训练, 辅以指令微调和人类反馈强化学习等方法, 使其能生成符合人类偏好的回答. 2) 在问答任务方面, ChatGPT 和 ChatGLM-6B 已成为传统基于知识的问答模型的替代品. 这两种模型在多种不同格式的数据集上均有较高准确度, 包括布尔型 QA 任务、提取型 QA 任务和抽象型 QA 任务. 3) 在用户数量方面, ChatGPT 目前拥有超过 1 亿用户, 其官网每月产生 18 亿次访问量[47]. ChatGLM-6B 是开源平台 HuggingFace[48]上被广泛使用的项目, 其通过结合模型量化技术, 使得用户可以在消费级的显卡上直接部署.

5.1.2 实验数据

本文采用在相关研究中被广泛使用的 3 个 QA 数据集 (BoolQ、SQuAD2 和 NarQA) 的测试集数据作为 QALT 及对比方法的种子输入. 这些数据集具有不同的格式、数据来源和规模. BoolQ[4]是一个布尔型 QA 数据集, 其中问题的答案均为“yes”或“no”. SQuAD2[27]是一个提取型 QA 数据集, 其中问题的答案均可用上下文中的文本子串表示. NarQA[20]是一个抽象型 QA 数据集, 其中问题的答案超出了简单的上下文子串, 采用自由形式文本. 3 种数据集的详细信息如表 1.

表 1 实验数据集

数据集	类型	训练集大小	测试集大小
BoolQ	布尔型	9427	3270
SQuAD2	提取型	130319	11873
NarQA	抽象型	32747	3461

5.1.3 评价指标

首先, 本文使用最前沿的智能问答系统测试技术 QAQA 与 QAAskeR 作为对比方法衡量 QALT 揭错能力. 然后, 本文评估了 QALT 不同组件对整体方法的贡献, 包括对提升测试输入的自然性效果和减少测试结果误报率的效果. 最后, 使用微调手段对智能问答系统中的缺陷进行修复. 实验涉及 3 个评价指标, 分别是揭错率、自然性和误报率.

● 揭错率: 本文复用 QAQA 论文中揭错率的定义. 如公式 (2) 所示, 揭错率为能够触发智能问答系统中缺陷的测试输入数量和尝试生成测试输入的总数之比. 需要注意的是, 该指标使用了测试技术尝试生成测试输入的总数而非实际生成的测试输入总数. 这之间的区别在于, 对于一个种子数据, 如果测试技术无法基于该种子生成任何新的测试输入, 即测试技术已经进行了一次尝试, 但生成失败. 这样做的好处在于, 在评估测试方法揭错能力的同时, 还可以分辨方法是否存在无法对种子进行变异的局限性.

$$\text{揭错率} = \frac{\text{能够触发缺陷的测试输入数量}}{\text{尝试生成测试输入的总数}} \tag{2}$$

● 自然性: 在评价测试输入自然性增强方法的有效性时, 本文使用余弦相似度衡量生成模型所生成的描述性语句与原始测试输入的相关程度[17]. 具体而言, 假设上下文 C 包含的各个句子记为 c_i, 本文分别计算 c_i 与描述性语句 d 之间的余弦相似度, 加和后再取平均值. 其中, 余弦相似度的计算方式如公式 (3) 所示. 使用余弦相似度 $Similarity(d, c_i)$ 计算文本相似性的好处在于该方法可以忽略绝对词频和向量长度的影响.

$$Similarity(d, c_i) = \frac{d \cdot c_i}{\|d\| \cdot \|c_i\|} \tag{3}$$

● 误报率: 与已有的研究方法相同[16,17], 本文采用人工检查的方式来验证测试结果是否是误报. 具体来说, 本文从生成的测试用例中随机选择 300 个测试结果进行人工检查. 最终根据人工检查的结果计算每种方法产生的误报率. 误报率等于分析得到的误报个数与分析数量总数的比值. 与之对应的是真阳性率: 真阳性率 = 1 – 误报率.

5.1.4 实验配置

本文的所有实验均在同一台服务器上执行, 该服务器搭载 Ubuntu 18.04 操作系统, 同时配备 Intel Xeon Gold 6240C CPU, NVIDIA GeForce RTX 3090 GPU 和 128 GB 内存. QALT 的测试输入执行模块需要设定一个阈值来判断预测答案和预期答案的语义是否一致, 为此我们复用了 QAQA 方法中的默认阈值 (即 0.76). 对于 SBERT 等工具的其他超参数, 我们使用工具原始论文中推荐的配置.

5.2 实验结果与分析

5.2.1 针对问题 1 的结果分析

研究问题 1 旨在探索 QALT 生成测试输入的揭错能力. 对于种子池 (即测试集) 中的每个种子 (即测试输入), 本文分别使用 QALT、QAQA、QAAskeR 这 3 种技术各生成一个新的测试用例对被测智能问答系统进行测试, 并记录每种测试技术检测到违反蜕变关系的测试输入数量. 违反蜕变关系表明智能问答系统中存在潜在缺陷. 表 2 分别展示了由 QALT、QAQA、QAAskeR 技术检测到缺陷的数量, 括号中的数字是揭错率, 即违反蜕变关系的测试输入所占的比例. 总体而言, 相比 QAQA 与 QAAskeR, QALT 检测到了更多的缺陷, 并产生了更大的揭错率. 对于待测模型 ChatGPT, QALT 报告的缺陷总体上比 QAQA 和 QAAskeR 分别多检测 2 013 和 1 744 个. 同时, 对于待测模型 ChatGLM, QALT 报告的缺陷总体上比 QAQA 和 QAAskeR 分别多检测 1 137 和 2 540 个. QALT 检测出来的缺陷数量和比例在不同的数据集上有不同的表现. 对于不同的待测模型, QALT 均在 SQuAD2 数据集上展现出最高的揭错率 (分别为 25.9% 与 25.7%). 在 SQuAD2 数据集中, 问题的答案均可使用上下文的子串内容表示, 而 BoolQ 与 NarQA 数据集均不使用上下文的子串内容作为答案. 这说明逻辑推理问题更容易影响到智能问答系统定位原文信息的能力. 此外, 由于 QALT 的测试效果并不依赖于待测系统的回答范式, 因此, 在不同数据集上 QALT 均表现稳定且良好.

表 2 检测到缺陷的数量和揭错率

待测软件	测试技术	BoolQ		SQuAD2		NarQA		总和	
		数量	揭错率 (%)	数量	揭错率 (%)	数量	揭错率 (%)	数量	揭错率 (%)
ChatGPT	QALT	**682**	**20.9**	**3 074**	**25.9**	**798**	**23.1**	**4 554**	**24.5**
	QAQA	674	20.6	1 524	12.8	343	9.9	2 541	13.7
	QAAskeR	19	0.6	2 506	21.6	285	8.2	2 810	15.1
ChatGLM	QALT	**868**	**26.5**	**3 053**	**25.7**	772	22.3	**4 693**	**25.2**
	QAQA	840	25.7	1 844	15.5	**872**	**25.2**	3 556	19.1
	QAAskeR	64	2.0	2 194	18.5	282	8.1	2 540	13.7

由于每种方法的测试结果均存在一定量的误报, 所以对比不同方法的误报率同样重要. 为此, 我们分别对 3 种方法在每个数据集上生成的测试用例中随机抽取 100 个测试用例, 并检测报告的缺陷是否是真阳性的. 分析结果发现, QALT 在 BoolQ、SQuAD2 和 NarQA 这 3 个数据集上采样结果的真阳性分别为 87%、88%、85%. QAQA 和 QAAskeR 在这 3 个数据集上采样结果的真阳性率分别为 98%、97%、98% 和 22%、60%、65%. 我们可以发现 QALT 测试方法的真阳性率显著高于 QAAskeR 但略低于 QAQA. 这是因为 QAQA 为了保证较高的测试精准度, 生成的测试输入并不会对原始问题本身进行修改, 这使得 QAQA 有着极高的测试精准度. 同时, 这种测试用例中添加干扰知识的行为无法对智能问答系统的逻辑推理能力进行测试. 真阳性缺陷的期望数量等于缺陷检测技术报告的缺陷数量与真阳性率的乘积. 通过将真阳性率和表 2 中检测到缺陷的数量对应项相乘, 我们可以得到每种测试方法检测到真实缺陷的期望数量. 最终 QALT、QAQA 和 QAAskeR 在这 3 种数据集上检测真实缺陷的总体

期望数量分别为在 ChatGPT 上的 3976、2475、1693 和在 ChatGLM 上的 4097、3456、1513. QALT 在 ChatGPT 和 ChatGLM 上检测到真实缺陷的总体期望分别比 QAQA 多 1501 和 2283 个, 比 QAAskeR 多 641 和 1941 个. 因此, QALT 比 QAQA 和 QAAskeR 有更强的缺陷检测能力, 且能够测试智能问答系统的逻辑推理能力. 此外, 我们将在未来的工作中进一步降低 QALT 的误报率.

5.2.2 针对问题 2 的结果分析

QALT 有 3 个关键的组件: 蜕变关系设计、测试输入自然性增强策略和基于依存句法分析的选择策略. 为了验证每个组件对整体方法的效果, 研究问题 2 分别探索了关键变异位置/选择策略对自然性的影响、不同蜕变关系对揭错能力影响和选择策略对真阳性的影响.

● 选择策略对自然性的影响. 本文使用的关键变异位置选择策略和名词词组选择策略旨在提升 QALT 生成测试用例的自然性. 自然的测试输入应该有正确的语法和流畅的语义[17]. 为了验证选择策略的有效性, 我们使用随机策略替换的选择策略构造一个 QALT 的变体. 通过将二者所生成测试用例中的描述性语句与原始测试输入中各个句子的平均余弦相似度作为衡量指标, 比较测试输入的自然性. 需要注意的是, 本文统计了各个测试数据集中上下文的平均余弦相似度作为基准用于数据的归一化. 表 3 展现了不同选择策略生成测试输入的自然性.

表 3 不同选择策略生成测试输入的自然性

选择方法	BoolQ	SQuAD2	NarQA	平均值
QALT	**0.75**	**0.72**	**0.66**	**0.71**
随机选择	0.65	0.60	0.55	0.60

从表 3 中可以看出, 测试输入自然性增强方法使得 QALT 技术产生了更自然的测试输入. 具体来说, QALT 相较于随机选择的实验, 使得插入句子的余弦相似度平均提升至 0.71, 这意味着 QALT 的测试输入具有更好的可读性, 所检测到的缺陷也更具备研究意义. 尽管自然性较低的测试输入也能够检测到缺陷, 但是由于检测到的缺陷在现实生活中不常见, 所以缺陷的危害等级较低. 因此, 使用自然的测试输入检测危害更强的缺陷更有意义. 此外, 由于余弦相似度衡量了插入句子与测试输入之间的相关程度, 余弦相似度越大, 说明二者之间的相似程度越高. 当余弦相似度取值为 1 时, 说明二者之间完全一致, 没有任何不一样的语义信息, 但在测试输入中显然不存在上述情况. 因此, 0.71 对于文本相似度来说是一个相对较大的值, 这说明二者在余弦空间中相对接近, 共享一定的语义内容或存在相似的上下文.

● 不同蜕变关系的缺陷检测能力. 本研究提出 3 种蜕变关系, 包括两种基于一阶变异和一种基于高阶变异的蜕变关系. 因此, 我们需要对每个蜕变关系执行独立的测试任务, 以评估各个蜕变关系的缺陷检测能力. 具体而言, 针对每种蜕变关系, 本研究为测试集中的每个种子独立生成新的测试输入, 并记录每种蜕变关系检测到的缺陷数量. 由于选择结果会影响 ERQC 测试用例生成的成功率 (需要两次选择), 为了实现不同变异算子之间公平的比较, 我们关闭了 QALT 中的基于依存分析的选择策略, 分别单独执行蜕变关系, 并对比它们的揭错能力. 表 4 给出了每种蜕变关系检测到缺陷的数量和比例.

表 4 每个蜕变关系检测到缺陷的数量与揭错率

待测软件	蜕变关系	BoolQ		SQuAD2		NarQA		总和	
		数量	揭错率 (%)	数量	揭错率 (%)	数量	揭错率 (%)	数量	揭错率 (%)
ChatGPT	ERQ	882	27.0	3395	28.6	936	27.0	5213	28.0
	ERC	238	7.2	2360	19.9	466	13.5	3064	16.5
	ERQC	**1006**	**30.8**	**3634**	**30.6**	**1053**	**30.4**	**5693**	**30.6**
ChatGLM	ERQ	692	21.2	3566	30.0	1116	32.2	5374	28.9
	ERC	249	7.6	2107	17.7	516	14.9	2872	15.4
	ERQC	**713**	**21.8**	**3886**	**32.7**	**1278**	**36.9**	**5877**	**31.6**

分析发现, 对于 ChatGLM 来说, 两种基本蜕变关系 (ERQ 和 ERC) 在 BoolQ 数据集上的缺陷检测率均较低. 这是因为 ChatGLM 模型的表现受测试输入长度的影响. 此外, ERQ 在所有数据集中和所有待测模型上均比 ERC

检测到更多的缺陷. 这表明被测智能问答系统在对问题的推理与理解方面更弱. 对于 BoolQ 数据集, ERC 与 ERQ 之间的相对差距相较于其他数据集更大 (约相差 3–4 倍). 这是由于 BoolQ 包含的测试输入均为一般疑问句, 可用“yes”或“no”进行回复, 而两种基本蜕变关系中的逻辑推理均为同向推理, 导致生成测试输入的答案更容易与原始测试输入答案保持一致. 基于高阶变异的蜕变关系 ERQC 检测到的缺陷数量显示在表 4 的底部一行. 值得注意的是, 在所有的数据集中, 高阶变异均检测到了最多的缺陷, 具备最大的揭错率, 且在不同的智能问答系统中高阶变异的平均效果最好.

● 基于依存句法分析的选择策略对真阳性的贡献. QALT 旨在通过选择策略得到语法正确、语义流畅的自然测试输入. 本文提出的基于依存句法分析的精准测试输入选择策略从一定程度上保证了所生成测试输入的语法正确性, 从而提高了测试输入的精准度. 为了评估 QALT 的测试精准度, 本文设计了一个 QALT 的变体 QALT–. QALT–关闭了 QALT 中的基于依存分析的选择策略. 接着, 我们分别为 QALT 和 QALT–基于每个数据集均随机选择固定数量报告出来的缺陷, 并通过人工检查来判断是否是真正的缺陷. 然后, 两位研究人员独立对采样数据中每一个报告出来的缺陷进行标记. 对于每个测试输入, 如果两位研究人员 1) 能够很好地理解问题的含义; 2) 可以根据上下文找到问题的正确答案; 3) 认定模型预测答案与预期答案有不同含义, 那么他们会将检测到的缺陷标记为真阳性. 对于标记结果, 两位研究人员之间的 Cohen’s Kappa 一致性得分 [48]为 0.86. 对于标记不一致的数据, 研究人员邀请了第 3 位专家共同讨论, 最终消除了所有的分歧. 表 5 展示了采样数据的真阳性标注结果. 表中每个数字代表每个数据集和测试技术产生的 100 个缺陷样本中真阳性比率. 例如, 单元格内数字为 87% 表示 QALT 基于 BoolQ 数据集检测到缺陷中的 100 个样本里有 87 是真阳性缺陷和 13 个误报数据. 从表 5 中可以看出, 与 QALT–相比, QALT 在所有 3 个数据集上均提升了检测到缺陷的真阳性率, 平均达到了 86.67%. 其中, 真阳性率 = 1 – 误报率.

表 5　检测到缺陷的真阳性率 (%)

测试技术	BoolQ	SQuAD2	NarQA	平均值
QALT	**87**	**88**	**85**	**86.67**
QALT–	63	71	66	66.67

由于 QALT 结果仍然存在部分误报, 因此, 我们进一步分析这些误报产生的原因. QALT 中发现的剩余误报大致可以分为两类. 第 1 类是由于在提取名词的过程中, 测试集中的某些测试输入会包含“you”“him”等指示代词. 这些词语本身指代上下文中的某些事物, 放入生成模型后没有实际语义, 变异后所产生的新测试输入并不能保证与变异之前是语义一致的. 例如, 在原始问题“can a widower marry his sister in law?”中提取到名词词组“his sister”后, 不论生成的描述性语句如何, 如“his sister is a 16 year old girl, is can a widower marry a 16 year old girl in law?”都无法在不阐述“widower”的前提下对“his sister”产生实际的语义. 第 2 类是在生成描述性语句的过程中, 存在一词多义的情况. 例如在“What type of musical instruments did the Yuan bring to China?”中, 提取出名词词组“the Yuan”. 此处的“Yuan”指的是 “元朝、元代”, 但在生成模型生成的描述性语句“the Yuan”中, 其指代“人民币”. 所以, 由于该描述性语句与原文的实际含义存在出入, 造成替换后的测试输入与原始测试输入语义不一致, 从而引发了误报. 经分析发现, QALT 的测试结果中由语法结构变化引起的误报已经基本消失, 而上述的两类误报是由方法所使用的自然语言处理工具和生成模型存在差错而引起的, 与测试输入的语法结构无关. 该发现体现了基于依存分析的选择策略在降低测试输入误报率的有效性.

5.2.3　针对问题 3 的结果分析

研究问题 3 旨在研究 QALT 是否可以产生有助于修复缺陷的新数据集. 本文使用新数据集对模型进行微调, 这种基于数据增强的修复方法已被广泛应用于智能问答系统 [37]和其他深度学习软件 [49,50]中. 与传统软件修复不同, 基于大型语言模型开发的软件训练代价十分昂贵. 为了减少对待测问答系统的训练耗时, 本文使用了最前沿的低秩适配器 LoRA[51]作为修复方法. LoRA 利用对应修复任务的数据, 只通过训练新加部分参数来适配下游任务. 当训练好新的参数后, 利用重参的方式, 将新参数和原参数合并, 这样既能在新任务上达到修复整个模型的效果, 又不会增加推断的耗时, 从而极大程度地节省修复过程的开销. 由于待测模型中只有 ChatGLM 模型为开源模型,

所以研究问题 3 使用 ChatGLM-6B 作为修复方法的验证对象. 为了避免模型过拟合, 本文使用训练数据作为种子池, 并为每个数据集生成 1 500 个新的问答对. 本实验总共使用 4 500 个新的问答对, 同时使用最先进的基于 LoRA 的微调方法, 实现了对 ChatGLM-6B 模型的微调. 然后, 本文在第 5.2.1 节中生成的测试输入上比较原始模型和微调模型分别产生的错误率. 与揭错率不同, 错误率是对模型性能的评估指标, 然而揭错率则是对测试技术的评价. 错误率等于模型在给定数据集上预测结果错误的比例. 本文使用 20% 的测试集作为验证集来指导模型训练, 剩下的测试集用来比较缺陷修复结果. 模型微调前后的错误率结果见表 6.

表 6 原始模型的错误率和使用 QALT 生成数据微调后模型的错误率 (%)

指标	BoolQ	SQuAD2	NarQA	平均值
原始模型的错误率	17.64	24.31	25.03	22.33
微调后模型的错误率	12.17	16.07	14.87	14.37
缺陷修复的比率	22.19	27.97	35.43	29.32

从表 6 中可以看出, 微调后的模型相比于原始模型在错误率上有着显著的下降. 错误率平均下降了 29.32% (从 22.33% 下降至 14.37%). 同时, 本文使用公式 (4) 计算了缺陷修复比率. 计算结果显示, QALT 在 NarQA 数据集上的修复能力最为显著, 修复了其中 35.43% 的缺陷. 模型微调的平均修复率为 29.32%, 这也证明了 QALT 生成测试用例对模型修复的有效性.

$$\text{缺陷修复比率} = \frac{(\text{原始模型错误率} - \text{微调后模型错误率})}{\text{原始模型错误率}} \tag{4}$$

6 方法的效度威胁

在研究过程中, 效度威胁是指那些可能对研究结果的准确性、可靠性和泛化性构成潜在威胁的因素. 效度威胁通常可以分为外部效度威胁和内部效度威胁两大类.

外部效度威胁主要涉及评估方法揭错能力的指标设计. 与传统软件中的缺陷不同, 由于智能问答系统的黑盒特性, 多个测试用例的回答错误可能源于同一个缺陷, 也可能源于多个缺陷. 由于修复智能问答系统的缺陷需要调整整个模型的参数, 因此无法对检测到的缺陷进行去重操作并统计去重后缺陷数量. 这对评估缺陷检测技术的效度造成了一定影响. 为了最大程度地降低这一影响, 本文遵循已有的研究方法[16,17], 对每个原始的种子输入仅生成一个新的测试变异体作为新的测试输入. 通过计算所有测试用例中能够揭露出智能问答系统缺陷的比例 (即, 测试用例的揭错率) 来衡量缺陷检测技术的揭错能力. 这种方法可以最大程度地避免因同一个种子测试输入生成的多个新的测试用例揭露同一个缺陷而被重复计算的情况.

内部效度威胁主要挑战在于对测试输入的自然性与是否为误报数据的人工标注过程可能存在的主观性和误差. 为了减轻这些潜在威胁, 本研究采用了双重核查机制. 两位具有较高英文阅读理解水平的研究者分别独立对每个随机筛选的数据进行标记. 当标记结果不一致时, 第 3 位研究人员会被邀请参与讨论, 直至所有标记结果达成一致. 这种方法在最大程度上保证了数据标注结果的可靠性和准确性.

7 总 结

近年来, 智能问答系统的测试技术取得了显著进展, 其中 QAAskeR 与 QAQA 通过采用蜕变测试技术成功摆脱了对人工标注数据的依赖. 然而, 这些方法在面对当前主流的基于大型语言模型的问答系统时, 暴露出明显的局限性. 首先, 它们未能充分评估系统的逻辑推理能力. 其次, QAQA 依赖于待测问答系统的数据集, 当数据集不可获取时, 该方法将无法实施. 而 QAAskeR 则受限于规则匹配机制, 其有效性高度依赖于模型输出的标准化程度, 当模型输出格式不统一时, 其测试效果会显著下降. 这些局限性使得现有方法在测试基于大型语言模型的问答系统时表现欠佳. 针对上述问题, 本研究提出了一种基于逻辑关系引导的问答系统蜕变测试技术 QALT. QALT 首先设计了 3 种逻辑蜕变关系来避免测试过程中对问答系统的数据集或回答范式产生依赖性, 从而提升了方法的泛用性, 同时弥补了现有方法测试智能问答系统逻辑推理能力的不足. 接着, QALT 通过计算文本相似度的方式选择变

异过程中需要的变异位置和名词词组来提升生成测试输入的自然性. 最后, QALT 通过基于依存句法分析的选择策略提升测试方法的精准度. 实验结果表明, 与最前沿的测试方法相比, QALT 在两个问答系统中预期比 QAQA 和 QAAskeR 分别多检测 2 141 和 4 867 个真阳性缺陷. 消融实验证明了 QALT 的 3 个核心组件均对整体方法起到正向作用. 此外, 本文使用 QALT 生成的测试输入对待测模型进行微调以修复缺陷. 微调后模型实现了 29.32% 的缺陷修复比率.

References

[1] Zhang ZS, Zhao H, Wang R. Machine reading comprehension: The role of contextualized language models and beyond. arXiv:2005.06249, 2020.

[2] Hirschman L, Gaizauskas R. Natural language question answering: The view from here. Natural Language Engineering, 2001, 7(4): 275–300. [doi: 10.1017/S1351324901002807]

[3] Xiong CM, Zhong V, Socher R. Dynamic coattention networks for question answering. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: OpenReview.net, 2017.

[4] Clark C, Lee K, Chang MW, Kwiatkowski T, Collins M, Toutanova K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In: Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis: ACL, 2019. 2924–2936. [doi: 10.18653/v1/N19-1300]

[5] DeLong B. AlphaChat: Underappreciated moments in economic history. 2016. https://equitablegrowth.org/alphachat-underappreciated-moments-in-economic-history/

[6] Kaplan A, Haenlein M. Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence. Business Horizons, 2019, 62(1): 15–25. [doi: 10.1016/j.bushor.2018.08.004]

[7] Roumeliotis KI, Tselikas ND. ChatGPT and Open-AI models: A preliminary review. Future Internet, 2023, 15(6): 192. [doi: 10.3390/fi15060192]

[8] Ouyang L, Wu J, Jiang X, Almeida D, Wainwright C L, Mishkin P, Zhang C, Agarwal S, Slama K, Ray A, Schulman J, Hilton J, Kelton F, Miller L, Simens M, Askell A, Welinder P, Christiano P, Leike J, Lowe R. Training language models to follow instructions with human feedback. In: Proc. of the 36th Int'l Conf. on Neural Information Processing Systems. New Orleans: ACM, 2022. 2011.

[9] Zhong WK, Ge JD, Chen X, Li CY, Tang Z, Luo B. Multi-granularity metamorphic testing for neural machine translation system. Ruan Jian Xue Bao/Journal of Software, 2021, 32(4): 1051–1066 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6221.htm [doi: 10.13328/j.cnki.jos.006221]

[10] Sun ZY, Zhang JM, Harman M, Papadakis M, Zhang L. Automatic testing and improvement of machine translation. In: Proc. of the 42nd ACM/IEEE Int'l Conf. on Software Engineering. Seoul: ACM, 2020. 974–985. [doi: 10.1145/3377811.3380420]

[11] Li Z, Pan MX, Zhang T, Li XD. Testing DNN-based autonomous driving systems under critical environmental conditions. In: Proc. of the 38th Int'l Conf. on Machine Learning. PMLR, 2021. 6471–6482.

[12] Eger S, Benz Y. From hero to zéroe: A benchmark of low-level adversarial attacks. In: Proc. of the 1st Conf. of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th Int'l Joint Conf. on Natural Language Processing. Suzhou: ACL, 2020. 786–803. [doi: 10.18653/v1/2020.aacl-main.79]

[13] Longpre S, Perisetla K, Chen A, Ramesh N, DuBois C, Singh S. Entity-based knowledge conflicts in question answering. In: Proc. of the 2021 Conf. on Empirical Methods in Natural Language Processing. Punta Cana: ACL, 2021. 7052–7063. [doi: 10.18653/v1/2021.emnlp-main.565]

[14] Jia R, Liang P. Adversarial examples for evaluating reading comprehension systems. In: Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing. Copenhagen: ACL, 2017. 2021–2031. [doi: 10.18653/v1/D17-1215]

[15] Ribeiro M T, Wu TS, Guestrin C, Sameer Singh S. Beyond accuracy: Behavioral testing of NLP models with CheckList. In: Proc. of the 58th Annual Meeting of the Association for Computational Linguistics. ACL, 2020. 4902–4912. [doi: 10.18653/v1/2020.acl-main.442]

[16] Chen SQ, Jin S, Xie XY. Testing your question answering software via asking recursively. In: Proc. of the 36th IEEE/ACM Int'l Conf. on Automated Software Engineering (ASE). Melbourne: IEEE, 2021. 104–116. [doi: 10.1109/ASE51524.2021.9678670]

[17] Shen QC, Chen JJ, Zhang JM, Wang HY, Liu S, Tian MH. Natural test generation for precise testing of question answering software. In: Proc. of the 37th IEEE/ACM Int'l Conf. on Automated Software Engineering. Rochester: ACM, 2022. 71. [doi: 10.1145/3551349.3556953]

[18] Team GLM. ChatGLM: A family of large language models from GLM-130B to GLM-4 all tools. arXiv:2406.12793, 2024.

[19] Rajpurkar P, Zhang J, Lopyrev K, Liang P. SQuAD: 100,000+ questions for machine comprehension of text. In: Proc. of the 2016 Conf.

on Empirical Methods in Natural Language Processing. Austin: ACL, 2016. 2383–2392. [doi: 10.18653/v1/D16-1264]

[20] Kočiský T, Schwarz J, Blunsom P, Dyer C, Hermann K M, Melis G, Grefenstette E. The NarrativeQA reading comprehension challenge. Trans. of the Association for Computational Linguistics, 2018, 6: 317–328. [doi: 10.1162/tacl_a_00023]

[21] Sharma V, Kalra A, Vaibhav, Chaudhary S, Patel L, Morency LP. Attend and attack: Attention guided adversarial attacks on visual question answering models. In: Proc. of the 32nd Conf. on Neural Information Processing Systems. Montreal: NeurIPS, 2018.

[22] Tang RX, Ma C, Zhang WE, Wu Q, Yang XK. Semantic equivalent adversarial data augmentation for visual question answering. In: Proc. of the 16th European Conf. on Computer Vision. Glasgow: Springer, 2020. 437–453. [doi: 10.1007/978-3-030-58529-7_26]

[23] Sheng SS, Singh A, Goswami V, Magana JAL, Thrush T, Galuba W, Parikh D, Kiela D. Human-adversarial visual question answering. In: Proc. of the 35th Int'l Conf. on Neural Information Processing Systems. ACM, 2021. 1556.

[24] Walmer M, Sikka K, Sur I, Shrivastava A, Jha S. Dual-key multimodal backdoors for visual question answering. In: Proc. of the 2022 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. New Orleans: IEEE, 2022. 15354–15364. [doi: 10.1109/CVPR52688.2022.01494]

[25] Khashabi D, Khot T, Sabharwal A. More bang for your buck: Natural perturbation for robust question answering. In: Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing. ACL, 2020. 163–170. [doi: 10.18653/v1/2020.emnlp-main.12]

[26] Khashabi D, Chaturvedi S, Roth M, Upadhyay S, Roth D. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In: Proc. of the 2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. New Orleans: ACL, 2018. 252–262. [doi: 10.18653/v1/N18-1023]

[27] Rajpurkar P, Jia R, Liang P. Know what you don't know: Unanswerable questions for SQuAD. In: Proc. of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne: ACL, 2018. 784–789. [doi: 10.18653/v1/P18-2124]

[28] Trischler A, Wang T, Yuan XD, Harris J, Sordoni A, Bachman P, Suleman K. NewsQA: A machine comprehension dataset. In: Proc. of the 2nd Workshop on Representation Learning for NLP. Vancouver: ACL, 2017. 191–200. [doi: 10.18653/v1/W17-2623]

[29] Dasigi P, Liu NF, Marasović A, Smith NA, Gardner M. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In: Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int'l Joint Conf. on Natural Language Processing. Hong Kong: ACL, 2019. 5925–5932. [doi: 10.18653/v1/D19-1606]

[30] Kwiatkowski T, Palomaki J, Redfield O, Collins M, Parikh A, Alberti C, Epstein D, Polosukhin I, Devlin J, Lee K, Toutanova K, Jones L, Kelcey M, Chang MW, Dai AM, Uszkoreit J, Le Q, Petrov S. Natural questions: A benchmark for question answering research. Trans. of the Association for Computational Linguistics, 2019, 7: 452–466. [doi: 10.1162/tacl_a_00276]

[31] Dua D, Wang YZ, Dasigi P, Stanovsky G, Singh S, Gardner M. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In: Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis: ACL, 2019. 2368–2378. [doi: 10.18653/v1/N19-1246]

[32] Tu KY, Jiang MY, Ding ZH. A metamorphic testing approach for assessing question answering systems. Mathematics, 2021, 9(7): 726. [doi: 10.3390/math9070726]

[33] Gupta M, Kulkarni N, Chanda R, Rayasam A, Lipton ZC. AmazonQA: A review-based question answering task. In: Proc. of the 28th Int'l Joint Conf. on Artificial Intelligence. Macao: IJCAI, 2019. 4996–5002. [doi: 10.24963/ijcai.2019/694]

[34] Talmor A, Berant J. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: ACL, 2019. 4911–4921. [doi: 10.18653/v1/P19-1485]

[35] Clark C, Gardner M. Simple and effective multi-paragraph reading comprehension. In: Proc. of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne: ACL, 2018. 845–855. [doi: 10.18653/v1/P18-1078]

[36] Khashabi D, Min S, Khot T, Sabharwal A, Tafjord O, Clark P, Hajishirzi H. UnifiedQA: Crossing format boundaries with a single QA system. In: Proc. of the 2020 Findings of the Association for Computational Linguistics. ACL, 2020. 1896–1907. [doi: 10.18653/v1/2020.findings-emnlp.171]

[37] Du ZX, Qian YJ, Liu X, Ding M, Qiu JZ, Yang ZL, Tang J. GLM: General language model pretraining with autoregressive blank infilling. In: Proc. of the 60th Annual Meeting of the Association for Computational Linguistics. Dublin: ACL, 2022. 320–335. [doi: 10.18653/v1/2022.acl-long.26]

[38] Chen TY, Cheung SC, Yiu SM. Metamorphic testing: A new approach for generating next test cases. arXiv:2002.12543, 2020.

[39] Huang JT, Zhang JP, Wang WX, He PJ, Su YX, Lyu MR. AEON: A method for automatic evaluation of NLP test cases. In: Proc. of the 31st ACM SIGSOFT Int'l Symp. on Software Testing and Analysis. ACM, 2020. 202–214. [doi: 10.1145/3533767.3534394]

[40] Reimers N, Gurevych I. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In: Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int'l Joint Conf. on Natural Language Processing. Hong Kong: ACL, 2019. 3982–3992. [doi: 10.18653/v1/D19-1410]

[41] Tsvetkova A. Anaphora resolution in Chinese for analysis of medical Q&A platforms. In: Proc. of the 9th CCF Int'l Conf. on Natural Language Processing and Chinese Computing. Zhengzhou: Springer, 2020. 490–497. [doi: 10.1007/978-3-030-60457-8_40]

[42] Mihalcea R, Tarau P. TextRank: Bringing order into text. In: Proc. of the 2004 Conf. on Empirical Methods in Natural Language Processing. Barcelona: ACL, 2004. 404–411.

[43] Wang SF, Thompson L, Iyyer M. Phrase-BERT: Improved phrase embeddings from BERT with an application to corpus exploration. In: Proc. of the 2021 Conf. on Empirical Methods in Natural Language Processing. Punta Cana: ACL, 2021. 10837–10851. [doi: 10.18653/v1/2021.emnlp-main.846]

[44] Khosla P, Teterwak P, Wang C, Sarna A, Tian YL, Isola P, Maschinot A, Liu C, Krishnan D. Supervised contrastive learning. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: ACM, 2020. 1567.

[45] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis: ACL, 2019. 4171–4186. [doi: 10.18653/v1/N19-1423]

[46] Joshi M, Chen DQ, Liu YH, Weld DS, Zettlemoyer L, Levy O. SpanBERT: Improving pre-training by representing and predicting spans. Trans. of the Association for Computational Linguistics, 2020, 8: 64–77. [doi: 10.1162/tacl_a_00300]

[47] Strzelecki A. Is ChatGPT-like technology going to replace commercial search engines? Library Hi Tech News, 2024, 41(6): 18–21. [doi: 10.1108/LHTN-02-2024-0026]

[48] Jiang WX, Synovic N, Hyatt M, Schorlemmer TR, Sethi R, Lu YH, Thiruvathukal GK, Davis JC. An empirical study of pre-trained model reuse in the hugging face deep learning model registry. In: Proc. of the 45th IEEE/ACM Int'l Conf. on Software Engineering (ICSE). Melbourne: IEEE, 2023. 2463–2475. [doi: 10.1109/ICSE48619.2023.00206]

[49] He PJ, Meister C, Su ZD. Structure-invariant testing for machine translation. In: Proc. of the 42nd IEEE/ACM Int'l Conf. on Software Engineering. Seoul: ACM, 2020. 961–973. [doi: 10.1145/3377811.3380339]

[50] Gupta S, He PJ, Meister C, Su ZD. Machine translation testing via pathological invariance. In: Proc. of the 28th ACM Joint Meeting on European Software Engineering Conf. and Symp. on the Foundations of Software Engineering. ACM, 2020. 863–875. [doi: 10.1145/3368089.3409756]

[51] Hu EJ, Shen YL, Wallis P, Allen-Zhu Z, Li YZ, Wang SA, Wang L, Chen WZ. LoRA: Low-rank adaptation of large language models. In: Proc. of the 10th Int'l Conf. on Learning Representations. OpenReview.net, 2022. 2463–2475.

附中文参考文献

[9] 钟文康, 葛季栋, 陈翔, 李传艺, 唐泽, 骆斌. 面向神经机器翻译系统的多粒度蜕变测试. 软件学报, 2021, 32(4): 1051–1066. http://www.jos.org.cn/1000-9825/6221.htm [doi: 10.13328/j.cnki.jos.006221]

作者简介

沈庆超, 博士生, CCF 学生会员, 主要研究领域为软件测试, 深度学习编译器测试.

李行健, 硕士, 主要研究领域为深度学习, 智能测试, 多模态大模型.

姜佳君, 博士, 副教授, CCF 专业会员, 主要研究领域为软件工程, 代码调试, 程序变换.

陈俊洁, 博士, 教授, CCF 高级会员, 主要研究领域为软件分析与测试.

齐一先, 硕士生, 主要研究领域为软件分析与测试.

王赞, 博士, 教授, CCF 专业会员, 主要研究领域为基础软件测试, 智能系统测试.