软件学报 ISSN 1000-9825, CODEN RUXUEW
2026,37(8):3205−3222 [doi: 10.13328/j.cnki.jos.007598] [CSTR: 32375.14.jos.007598]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于文本表达特征分析的大语言模型协议交互抽取*

张伯洋, 钱 巨, 唐靖然, 卫 依

(南京航空航天大学 计算机科学与技术学院/软件学院, 江苏 南京 210016)

通信作者: 钱巨, E-mail: jqian@nuaa.edu.cn

摘 要: 文本协议交互抽取旨在从自然语言形式的说明文档中识别并提取协议有关的交互信息, 其可用于在协议代码实现前抽取模型验证协议的正确性、从协议规格描述构造测试用例等. 当前从文本中抽取协议主要采用深度学习、大语言模型等技术, 深度学习方法依赖大规模的高质量数据集, 且适用范围受限于训练数据集, 存在迁移困难的问题. 现有的大语言模型方法存在提示模板和示例构造较为简单、处理流程欠优化的局限. 针对以上问题, 提出一种增强的基于文本表达特征分析的大语言模型协议交互抽取方法. 首先, 从真实的协议描述案例出发, 总结出协议描述文本中存在的常见语言表达特征; 然后, 提取能够体现这些特征的典型协议描述案例, 提炼用于协议交互抽取的处理规则; 进一步地, 融合案例与规则, 提出一套规则回溯思维链方法; 最后, 使用多路推理和自我验证技术优化任务处理流程. 在多个协议数据集上的实验结果表明, 所提方法在协议交互的抽取精确率和召回率等方面均优于基线方法, 证实了所提方法的有效性.

关键词: 协议模型抽取; 自然语言理解; 大语言模型; 提示工程

中图法分类号: TP311

中文引用格式: 张伯洋, 钱巨, 唐靖然, 卫依. 基于文本表达特征分析的大语言模型协议交互抽取. 软件学报, 2026, 37(8): 3205–3222. http://www.jos.org.cn/1000-9825/7598.htm

英文引用格式: Zhang BY, Qian J, Tang JR, Wei Y. Extracting Protocol Interactions via LLM Based on Linguistic Expression Pattern Analysis. Ruan Jian Xue Bao/Journal of Software, 2026, 37(8): 3205–3222 (in Chinese). http://www.jos.org.cn/1000-9825/7598.htm

Extracting Protocol Interactions via LLM Based on Linguistic Expression Pattern Analysis

ZHANG Bo-Yang, QIAN Ju, TANG Jing-Ran, WEI Yi

(College of Computer Science and Technology/College of Software, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: Extracting protocol interactions from textual specification documents written in natural language is useful, especially when to verify the correctness of a protocol before its implementation and application, or when to generate test cases for protocol-connected systems directly from specification documents. Existing approaches for this purpose rely on deep learning or large language models (LLMs). The deep learning approaches require large-scale and high-quality annotated datasets. They may not work well across protocols in different domains due to limitations imposed by the training datasets, and suffer from difficulties in transfer. The LLM-based approaches offer better generalizability, but existing work only uses simple prompt templates. It does not carefully utilize extraction examples in LLM prompting, and the information extraction process lacks optimization, which affects the effectiveness of the proposed approaches. To address these challenges, this study proposes an enhanced LLM-based method for extracting protocol interactions from protocol texts, based on linguistic expression pattern analysis. Specifically, real-world protocol description texts are first analyzed to summarize common linguistic expression patterns in such texts. Then, representative protocol description examples exhibiting these patterns are selected, and

* 基金项目: 国防基础科研项目 (JCKY2022605C006)
本文由“大模型与软件工程”专题特约编辑聂长海教授、王璐副教授、王莹教授、姜艳杰副教授、张敏灵教授推荐.
收稿时间: 2025-09-08; 修改时间: 2025-10-28; 采用时间: 2025-12-08; jos 在线出版时间: 2025-12-24
CNKI 网络首发时间: 2026-03-04

© 中国科学院软件研究所 http://www.jos.org.cn

corresponding extraction rules are distilled. Further, these examples and rules are integrated to design a rule retrospection chain-of-thought method for LLM-based protocol interaction extraction. Finally, multi-path inference and self-verification techniques are used to optimize the task execution process. Experimental results on multiple protocol datasets show that the proposed method outperforms the baseline methods in terms of precision and recall of protocol interaction extraction, which confirms the effectiveness of the proposed method.

Key words: protocol model extraction; natural language understanding; large language model (LLM); prompt engineering

协议交互抽取从文本、网络报文或流量等非结构化或半结构化数据中自动提取表达协议业务逻辑的事件交互、状态转移等信息[1−15]. 当前协议交互抽取多依赖协议的程序代码、通信报文序列或网络流量等实现和应用层信息[1−8], 尽管如此, 一类单纯从自然语言文本中理解协议的工作[9,10,12,15]也非常重要. 这类技术视协议描述文本为其规格说明, 可用于在协议尚未得到实现和应用前获得协议的基本模型, 从而验证协议的正确性, 或从协议的规格描述出发为协议设计测试用例等.

例如, 对于由分布式组件构成的系统, 如舰船指控、情报、武器等模块通过 DDS 报文发布订阅构成的任务系统, 汽车发动机、变速箱、电池等通过 CAN 总线衔接的电控系统等, 其各个组件通过自定义的协议进行功能串联. 在集成测试的过程中, 往往需要模拟被测集成部件之外的组件, 按协议规定的内容和顺序发送激励消息, 才能驱动被测子系统的工作. 传统上, 为支持分布式系统的自动测试, 需要人工来建立协议交互的模型, 如事件流图、消息序列图 (message sequence chart) 等[16]. 此过程依赖专家经验, 开销巨大. 而在分布式系统的规格文档中, 往往包含组件间如何交互的自然语言文本描述. 如果能够从中自动抽取组件之间的协议交互模型, 明确交互主体、内容、顺序等, 可以自动构造消息序列测试用例, 从而对组件集成展开高效测试.

本文关注上述从自然语言文本提取协议交互信息的问题, 现有的文本协议交互抽取工作主要包括基于规则的方法[15]、基于学习的方法[9,10]以及基于大语言模型的方法[12]. 基于规则的方法定义面向领域的语法规则, 把句子拆成句法树, 应用启发式规则配合一定的人工辅助等识别协议内的交互关系. 基于学习的方法依靠人工制定或自动学习获得的特征, 通过大规模的训练从输入输出中学习识别协议关键信息的模型, 并将其用于理解新协议. 这两类方法或依赖于人工定义的规则和模板或依赖大量高质量的标签数据集进行模型训练, 通常需要较多的人工操作且迁移性弱, 往往难以直接应用于新的领域.

相较于这两种方法, 基于大语言模型的方法能够有效缓解数据稀缺和泛化能力差的问题, 因此受到越来越多的关注[12−14]. PROSPER[12]采用提示工程技术, 通过 APE 策略[17]和贪婪策略自动或半自动构造面向协议交互抽取的大语言模型提示模板, 来从文本理解协议的交互. 现有基于大语言模型的文本协议交互抽取存在以下问题. 首先, 在提示模板构造方面, 虽然对初始提示词进行了优化, 但主要开展的是单词、语句的同义替换, 这种替换使表达更精准, 但并不能提供关于任务的额外信息, 只能获得有限的提升. 其次, 示例的构造和应用较随意, 任意选择示例来用于引导提示词的优化选择, 没有引入针对性的示例来引导模型推理; 此外, 流程相对粗放, 仅一次问询大语言模型来直接获得结果, 结果随机性较大. 以上问题导致方法对于许多常见复杂协议表述分析结果不理想.

针对上述问题, 本文提出了一种增强的基于文本表达特征分析的大语言模型协议交互抽取方法. 首先, 调研了来自公开专利的 100 个覆盖网络层、应用层等的协议文本描述样本, 深入分析协议文本表达方面的内在逻辑和表达结构, 总结了协议文本描述的常见语言表达特征, 如交互主体层次化、指称混用、描述中存在过程压缩等. 从这些特征出发, 本文一方面提取能够体现语言表达特征的典型案例, 另一方面, 提炼出用于提升协议交互抽取效果的处理规则. 在此基础上, 融合典型案例以及规则推理, 提出了一种规则回溯思维链技术. 该技术以典型案例为上下文示例, 融入对协议文本抽取的规则引导与约束, 来强化模型对不同文本结构的分析能力. 此外, 本文还引入多路推理[18,19]和自我验证[20,21]思想, 提出协议文本抽取的融合与验证规则, 来优化任务的执行流程. 协议文本输入经过多路推理模块处理, 得到 N 个并行执行结果. 随后, 所有的输出结果与任务信息以及原始协议文本被送至自我验证模块进行答案校验与融合, 修改不合理的答案并补充遗漏的信息, 得到模型的最终输出.

为检验所提方法的有效性, 本文选取了两个具有代表性的文本协议交互抽取工作 RFCNLP[9]和 PROSPER[12]作为实验基线, 在 RFC 协议文档数据集和一个新构建的面向应用的中文协议描述数据集 CPP 上对比了本文方法和基线方法的表现. 结果表明, 在两个数据集上, 依托案例和规则理解能力强的大语言模型, 本文方法的协议交互

© 中国科学院软件研究所 http://www.jos.org.cn

抽取精确率和召回率提升均达约 10%, 误报和漏报的总错误率降低约 15%. 在 CPP 数据集上, 最终的精确率和召回率相对基线方法实现了 90% 的突破, 证实了方法的良好效果.

本文的主要贡献如下.

(1) 深入调研了协议文本中的语言表达逻辑和结构, 总结了协议描述文本中的语言表达特征.

(2) 基于文本表达特征提取案例、提炼规则, 构建规则回溯思维链, 以增强提示词引导性, 同时, 结合多路推理和自我验证来优化文本协议交互抽取任务的提示-问答框架, 以提高抽取效果.

(3) 构建了一套 50 个案例的协议文本描述数据集 (https://github.com/AstralMorrow/CPP), 包含关于通信和业务层应用性协议等自包含文本描述, 附带来自原始资源、经核查和文本描述一致的协议交互图作为参考抽取答案.

(4) 在 RFC 和新数据集中的协议上开展实验, 证实了所提方法的有效性.

本文第 1 节介绍相关研究现状. 第 2 节给出一个案例来说明研究动机. 第 3 节分析协议文本描述中的语言表达特征. 第 4 节介绍本文的大语言模型协议交互抽取问答框架. 第 5 节开展实验评估. 最后总结全文.

1 相关工作

协议 (protocol) 指双方或多方在交互过程中所共同遵守的一套规则和约定. 这些规则和约定涵盖交互顺序、消息格式、错误处理等方面, 确保不同方能够互联互通、正确并可靠地交换信息. 协议交互抽取从源代码、网络通信、流程描述等内容中自动识别和抽取通信协议中的交互流程及详细信息, 例如谁发起了请求、发送了什么命令、期望什么响应等, 从而形成结构化的描述. 当前的协议交互抽取工作可分为基于规则、基于学习和基于大语言模型这 3 种类型.

基于规则的方法多基于文本语法结构、上下文、报文内容、程序流程等方面的规则来理解协议内涵. 代表技术有程序分析和网络流量分析. 程序分析技术利用静态的数据流/控制流分析[1]和符号执行[2]、动态的运行时监控[3]和污点追踪[4]等提取程序在执行流程和数据访问方面的特征, 从而识别协议交互顺序和消息格式等. 网络流量分析[5–8]方法通过分析通信过程中的网络数据包或流量来还原协议的交互顺序、消息格式等. 基于学习的方法[9–11]依靠人工制定或自动学习获得的特征, 从训练数据中利用机器学习或深度学习发现隐含的规律, 并用所学得的模型甄别文本、代码、报文、流量等来理解协议. 基于大语言模型的方法[12–14]依靠大语言模型的语义理解和指令遵循能力, 将问题封装成 Prompt 的形式, 由模型分析并直接给出回答. 这种方法不依赖于人为制定的规则, 往往也无需大量数据集进行模型训练.

本文关注从纯文本角度出发的协议交互抽取, 使用自然语言作为输入 (如 RFC[9]等规格文档), 通过文本分析与处理提取协议交互信息. 这类方法不依赖协议的程序代码、通信报文序列等具体实现和运行信息, 可用于协议相关系统早期设计阶段的分析验证或从规格说明生成测试用例等.

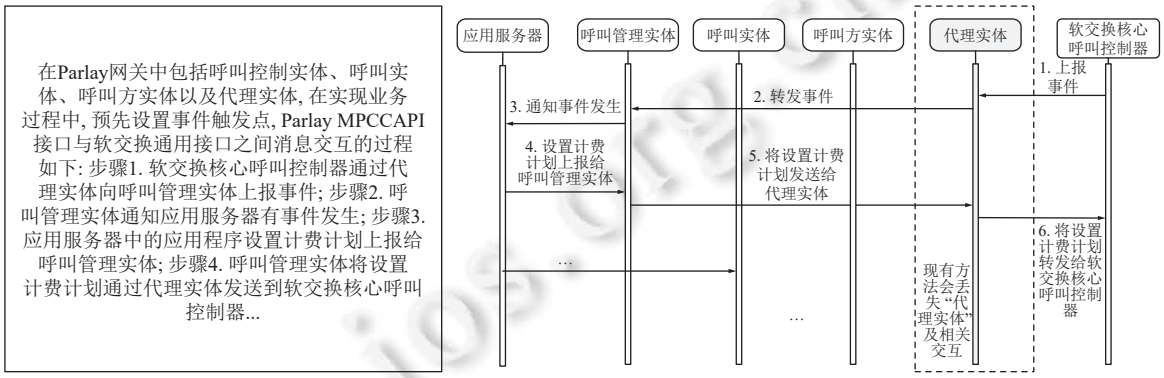
在此领域, Yen 等人[15]采用基于规则的方法研制了从自然语言半自动生成协议实现代码的工具 Sage. 其引入领域相关的特定词汇和语法规则, 借助 CCG 等语义解析器来解读 RFC 协议描述文档. 在清洗并人工澄清模糊句子后将协议转换为逻辑形式来理解其内涵. RFCNLP[9]提出了一种利用深度学习从自然语言编写的协议规范中自动提取协议交互信息以支持攻击合成和协议安全分析的方法. 其首先在大量协议相关的技术文档上微调 BERT 模型来学习协议技术术语的词嵌入表示, 捕捉协议规范中的专业术语和结构. 然后将协议文本映射到特定中间表示, 提取协议要素, 将中间表示转换为具体的协议自动机. Hermes[10]采用基于深度学习的方法从自然语言编写的蜂窝网络协议规范中自动生成有限状态机, 以支持协议的形式化建模和安全分析. 该方法首先使用训练得到的神经句法分析器 NEUTREX 处理协议规范文本, 提取协议交互信息, 然后根据设计的领域特定语言 (DSL) 利用依存句法树将提取的信息转换为逻辑公式, 最后将逻辑公式转变为有限状态机. PROSPER[12]采用基于大语言模型的方法, 通过 APE 策略[17]和贪婪策略来获得优化的提示词, 以从文本提取协议交互. 在从文本段落提取协议信息的同时, PROSPER 还设计资源提取器从 RFC 等文档中特有的以文本字符绘制的图表提取协议信息. 上述方法中, 基于规则或学习的

© 中国科学院软件研究所 http://www.jos.org.cn

方法[9,10,15]或依赖于人工定义的规则, 或依赖于高质量的训练数据集, 可扩展性和迁移性较差. 大语言模型方法[12]提示模板和示例的构造没有针对特定问题进行深入的分析与总结, 问答方式、流程相对简单、缺乏优化, 在复杂任务场景下常表现不佳.

2 动机示例

图 1(a) 展示了一个针对 Parlay 网关交互的协议描述示例. 该描述构成了对电信业务中软交换核心呼叫控制器、代理实体、呼叫管理实体等组件之间交互机制的规格说明. 由此出发, 可以理解协议实现需求, 构建如图 1(b) 的协议交互模型. 依据该交互模型, 可以进一步通过名称匹配等绑定消息报文, 产生触发系统内组件交互的消息序列, 用于对待测系统开展集成测试等.



(a) 协议文本描述 (b) 协议交互模型

图 1 一个示例性的协议描述及其交互模型

传统上, 协议模型的构建主要由专家根据自身理解人工完成, 繁琐复杂. 文献 [16] 给出了一种利用扩展的正则表达式来快速建模交互活动场景以开展测试的方法, 减轻了模型构建的繁琐程度, 但依然依赖人工编写模型. 如能够自动构建协议交互模型, 将能够降低测试等过程的难度, 提高测试流程效率.

大语言模型为从自然语言描述出发来理解协议、抽取其交互提供了有效途径. 提供合适的提示 (Prompt), 大语言模型能够从自然语言文本中抽取出协议中的关键交互事件. 尽管如此, 用户, 尤其是普通软件开发者, 所描述的协议可能表达方式多样, 其中的一些表达结构可能为大语言模型的理解带来挑战. 以图 1(a) 的协议描述为例, 采用 DeepSeek-R1 按下述提示来抽取协议交互, 由于协议描述中存在“X 通过代理实体向 Y…”这类先发往代理, 再由代理传递的情形, 多交互步骤被压缩为一步表达, 会导致中间交互过程丢失, 即图 1(b) 中虚线框部分所有“代理实体”相关的事件被遗漏, 无法得到正确的结果.

对于用户输入的协议描述文本, 请准确识别其中存在的交互事件及其相关信息, 如交互双方、事件执行前提、事件中传递的消息, 如果没有则用无表示, 并进一步分析事件间的流程关系.

事件表示格式

…

输出模板

…

PROSPER 方法[12]采用贪婪策略和 APE 策略来优化提示表达, 主要方式是调整提示的表述词句, 例如将“执行前提”变为“执行条件”, 根据在样本协议文本上开展的验证, 选用效果最优的提示表述. 但当前大语言模型已经具有相当强的理解能力, 在含义不变的情况下, 简单地改变提示表述对于协议交互抽取效果影响不大. 在给定案例上开展验证和学习, 可以提升大语言模型表现, 但 PROSPER 对案例没有恰当的选择机制, 而是在简单的案例上进行

© 中国科学院软件研究所 http://www.jos.org.cn

验证, 并不能帮助产生能更好应对复杂场景的提示. 对于图 1(a) 的例子, 按 PROSPER 的策略来优化提示, 难以解决经过“代理实体”的中间过程丢失问题.

本文从真实的协议描述文本出发, 总结常见的挑战性协议语言表达特征, 然后针对这些特征强调相关应对规则来提示给大语言模型, 从而增强大语言模型分析所需的协议交互抽取知识. 例如, 对图 1 中的案例, 本文会提示大语言模型“在进行协议交互抽取时, 对于连续的顺序过程被压缩表达的描述, 如 A 通过 B 转发消息 M 给 C, 应该完整提取整个过程, 即 A 先发送 M 给 B, 然后 B 再发送 M 给 C”. 同时, 还会配以相关上下文示例和思维链解读. 在此技术下, 协议抽取效果将能得到改进, 图 1(b) 中的错误协议抽取结果可得到纠正. 进一步地, 融入多路推理和自我验证的思想, 可以加强大语言模型处理协议文本的稳健性, 提高其综合表现.

3 协议文本表达特征分析

普通的应用性协议多凭经验定义, 其表达往往没有统一的标准和规范, 由此带来的灵活性为协议的理解带来极大挑战. 本文首先广泛调研协议的文本表达特征来指引协议交互的抽取. 考虑到普通应用性协议描述来源的公开易获取性, 本文从专利数据库中抽取了 100 个覆盖通信和业务流程的协议描述文本作为调研样本. 专利中的协议描述详实具体, 来自大量不同作者, 具有代表性, 且多针对应用性系统. 与之相似, 公开的论文中也常包含协议描述, 但其侧重创新点的表达, 协议描述往往不完整. RFC 文档也含协议描述, 但其多面向基础性协议, 由较高水平的专家编撰, 常不代表普通研发人员为各类系统所编制的应用性协议描述.

本文的协议描述从国家知识产权系统 (CNIPA) 通过检索关键词“协议”“交互”“流程”“时序”获取. 在检索过程中采用关键词组合的方式, 以“协议”为主体, 分别构造“协议+交互”“协议+流程”以及“协议+时序”的关键词组进行检索, 按照检索结果从前到后的顺序, 从专利文本中提取协议相关的内容作为样本.

考虑到大语言模型在用词、句法等方面的天然优势, 在进行语言表达特征分析时, 重点关注语句在表达逻辑和结构上的特点. 表达逻辑关注语句之间的因果、时序等联系, 表达结构关注对象、事件等关键实体在语句中的组织方式.

经归纳总结, 除一般性地按顺序依次描述协议交互动作外, 本文发现 8 个协议描述中常见的挑战性语言表达特征 (详见表 1), 这些特征在许多协议样本中单独或同时出现, 需要一定的理解能力才能够正确应对.

1) 流程描述中夹杂非操作性的解释说明 (33% 的样本存在此问题). 协议描述中介绍操作性步骤时常会夹杂大量解释性信息, 这些信息与交互事件的提取无关, 可能会被错误地提取为交互. 在进行协议交互抽取时应区分交互事件相关与无关信息, 避免从无关内容中提取出交互步骤.

2) 流程存在不同的处理分支 (43% 的样本存在此问题). 协议描述中有时流程会出现分支, 极端情况下, 某些分支不进行任何操作或仅包含少量操作. 这些分支在提取时容易被遗漏, 且各分支的准入条件也可能被错误提取. 为了确保流程的完整性, 所有的分支都应被显式提取, 且应该特别关注不同分支的准入条件.

3) 交互主体层次化 (15% 的样本存在此问题). 协议描述中某些交互主体内部可分, 简单地提取顶层主交互实体会导致协议交互中的细节信息被遗漏. 在进行协议交互抽取时, 如果交互主体内部可分, 则应提取具体的子交互主体, 同时使用父交互主体作为子交互主体的标识.

4) 指称混用 (18% 的样本存在此问题). 协议描述中同一交互主体可能使用不同的命名, 被误认为不同的交互实体, 影响对协议交互流程的理解. 在进行协议交互抽取时, 应根据协议描述的上下文信息统一各交互主体的命名.

5) 描述中存在过程压缩 (36% 的样本存在此问题). 协议描述中部分事件间存在紧密的顺序关系, 有时为了简洁表达会压缩完整的交互过程, 导致中间过程因没有被明确表达而易被遗漏. 在进行协议交互抽取时, 应根据协议的上下文信息拆分压缩过程, 将其当作独立的事件进行抽取.

6) 一对多事件被合并描述 (13% 的样本存在此问题). 协议描述中当同一实体连续与多个不同的实体进行交互时, 经常会以并列的形式合并多个针对不同实体的事件, 这种表达方式常导致部分交互事件丢失. 在进行协议交互抽取时, 应对并列表达的不同事件进行拆分, 确保所有事件都得到准确识别.

7) 事件前提条件和交互动作描述接近 (31% 的样本存在此问题). 协议描述中事件的执行可能依赖于某些前提

© 中国科学院软件研究所 http://www.jos.org.cn

条件, 这些条件的表达方式有时和事件交互动作的描述方式接近, 常被错误地认为是协议流程中的事件. 在进行协议交互抽取时, 应认真区分交互事件和执行前提, 前提条件应作为事件的一个属性被事件吸收, 而不是当作独立的事件进行提取.

表 1 协议文本表达特征总结与应对规则

特征	典型案例	案例解释	应对规则	规则的Prompt表达
F1: 流程描述中夹杂非操作性的解释说明	识别码生成终端接收用户输入的识别码生成指令. **识别码生成终端可以生成任意终端的号码对应的识别码. 可选的, 终端的号码通常由阿拉伯数字构成, 当终端为移动终端时**,…	加粗文字是对“识别码生成终端接收用户输入的识别码生成指令”中操作性步骤的解释说明, 从这类非操作性说明中易错误地提取出交互事件	R1: 流程无关信息过滤规则	在进行协议交互抽取时, 需要判断目标内容是否与协议流程密切相关, 不要提取流程无关信息, 如一些初始化信息、说明信息等, 这些内容可能出现在具体的协议交互流程之外, 也可能出现在核心流程中某一事件之后, 充当事件的具体解释
F2: 流程存在不同的处理分支	…, 由结果判定单元进行分析结果判定, **判定通过则进行下一环节检测, 不通过则表示检测异常, 检测结束, 输出对应的传输层协议检测异常信息**,…	加粗文字会根据判断结果触发不同的分支路径, 在提取流程时, 应准确提取各分支	R2: 分支处理规则	在进行协议交互抽取时, 当流程中出现分支时, 需要完整地提取所有分支, 不能出现遗漏. 且对于不同的分支, 需要准确判断分支结构中各事件之间的关系, 如不同分支上都有哪些事件、同一分支上各事件间的先后顺序等
F3: 交互主体层次化	以主链向子链进行交易查询为例,…, 步骤1. 当节点需要查询运营商A的子链中的交易信息时, 可通过调用主链中的智能合约发送交易记录查询… 步骤2. 主链的证明模块为该查询分配一个授权标识符,… 步骤3. 在收到信息…后, 运营商A的子链上的证明模块负责…	案例中“主链中的智能合约”“主链的证明模块”等子模块为具体交互主体, 如用“主链”等顶层名称或“智能合约”等无限定子模块名称表达, 会导致交互主体不明确. 应准确抽取具体交互主体, 避免交互主体被混淆	R3: 对象粒度细化规则	在进行协议交互抽取时, 对于交互主体内部存在细分的情况, 如A的子系统B, 在提取时要具体到交互主体的子部分, 同时使用父交互主体作为标识进行区分, 不能只提取A或B, 而应该完整提取A的子系统B
F4: 指称混用	系统包括测试主板、转接板、PC端上位机APP和TV显示设备…, 上位机显示比对的测量结果,…	不同名称“PC端上位机APP”和“上位机”指向同一交互主体, 易错误地提取出多主体	R4: 交互主体命名统一规则	在进行协议交互抽取时, 确保同一个交互主体使用全局唯一的命名, 若命名不唯一, 在提取时需要根据上下文进行统一
F5: 描述中存在过程压缩	步骤1. 软交换核心呼叫控制器通过代理实体向呼叫管理实体上报事件; 步骤2. 呼叫管理实体通知应用服务器有事件发生,…	通过代理转发压缩了发送给代理实体, 然后由代理实体发出两步. 此表达容易导致关键的中间操作被遗漏, 在提取时应拆分为独立事件	R5: 压缩过程拆分规则	在进行协议交互抽取时, 对于连续的顺序过程被压缩表达的描述, 如A通过B转发消息M给C, 应该完整提取整个过程, 即A先发送M给B, 然后B再发送M给C
F6: 一对多事件被合并描述	…, 当相关记录存在时, 根据交易编号获取凭证状态和凭证所有权记录, 并根据记录向出售方和购买方发起验证请求,…	系统根据记录向出售方和购买方发起验证请求, 此一对多表达方式易导致不同交互主体被混在一起提取, 也易遗漏部分交互事件, 在提取时应拆分为多个独立事件	R6: 交互主体合并事件拆分规则	在进行协议交互抽取时, 当一对多的交互信息被合并描述, 需要独立提取各个交互信息, 如A发送消息M给B和C, 提取时应该独立提取A发送消息给B, A发送消息给C
F7: 事件前提条件和交互动作描述接近	**智能设备120在接收到智能控制终端110发出的红外触发信号的情况下**, 可以向智能控制终端110发送配网请求信号,…	加粗文字是智能设备120向智能控制终端110发送请求的前提, 且多在前面已表达, 此处应作为事件的条件属性, 不宜当作独立的事件进行提取	R7: 事件依赖提取规则	在进行协议交互抽取时, 应认真区分流程事件和执行条件, 执行条件应作为事件的一个属性被事件吸收, 而不是当作独立的事件进行提取
F8: 协议描述中存在自交互事件和交互事件	…, 步骤14. 主叫控制面设备接收到寻呼请求后, 向主叫代理返回寻呼请求的回复报文; 步骤15. 主叫控制面设备,…确定被叫控制面设备,…, 对跨运营商业务分配主叫网络资源;…	步骤14中有“主叫控制面设备”和“主叫代理”两个交互实体, 属于多端交互事件, 步骤15中没有涉及多个交互实体, 属于自交互事件, 不进行区分容易导致步骤错漏	R8: 交互类型区分规则	在进行协议交互抽取时, 需要判断各事件的交互类型, 只有一个交互实体时为自交互事件, 存在两个及以上交互实体时为交互事件. 自交互事件在提取时事件的发起方和接收方视为同一交互实体

© 中国科学院软件研究所 http://www.jos.org.cn

8) 协议描述中存在自交互事件和交互事件 (41% 的样本存在此问题). 协议交互流程中存在两类事件, 自交互事件只涉及一个交互实体, 不存在与其他交互实体的联动. 交互事件则侧重描述多个交互实体间的行为. 在进行协议交互抽取时, 应区分这两种不同的事件类型. 对于自交互事件, 事件的主动方和被动方为同一交互实体.

4 基于文本表达特征分析的大语言模型协议交互抽取问答框架

第 3 节调研了协议文本描述中存在的语言表达特征, 这些特征对于准确提取协议交互非常关键. 本节尝试在基于大语言模型[22,23]的协议交互抽取问答框架中有效利用这些特征来改进分析效果. 如图 2 所示, 框架的输入为目标协议的相关文本段落, 输出是协议交互流程的结构化表示. 其从文本表达特征出发, 提取能够体现这些特征的典型案例, 提炼出用于提高协议交互抽取效果的针对性规则. 进一步地, 融合典型案例与规则推理于思维链框架[24], 提出一种规则回溯思维链技术, 在大语言模型的任务推理中融入对协议文本抽取的规则引导与约束. 最后, 融合多路推理与自我验证技术, 优化任务的执行流程, 保障模型输出的质量, 尽可能地避免错误和遗漏的出现.

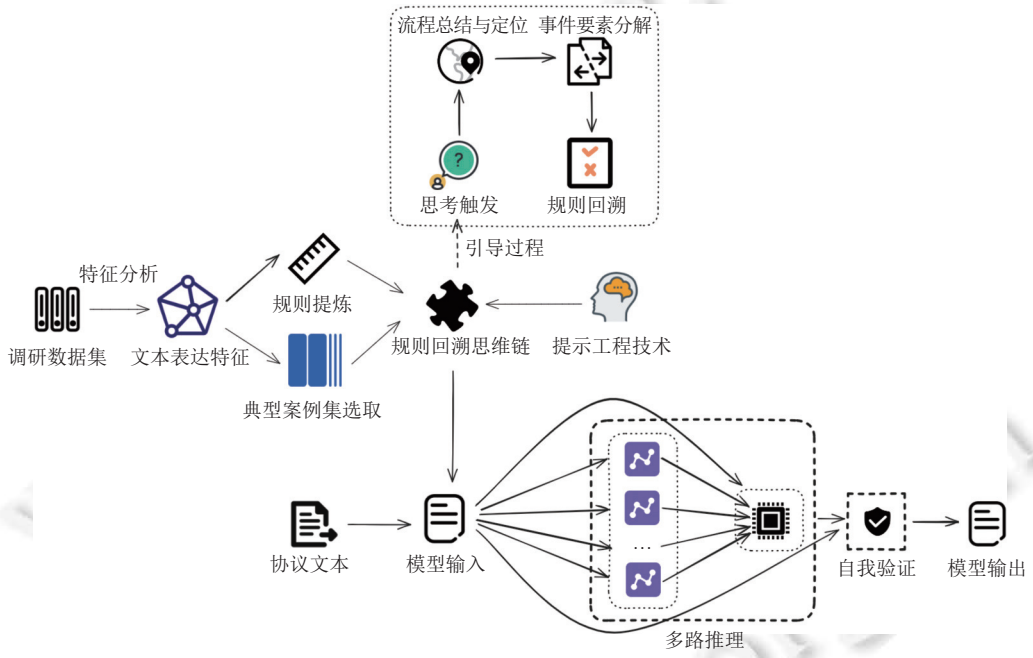


图 2 基于文本表达特征分析的大语言模型协议交互抽取问答框架

本文提示构成框架包括提示模板和规则回溯思维链两大部分. 提示模板定义了协议交互抽取的任务要求, 如表 2 所示, 由角色、任务定义、协议文本、事件表示格式、规则约束、任务步骤和输出格式组成. 表 2 中“组成部分”和“语言模式”列的内容以 Markdown 标题和内容的形式组合成一个完整提示文档, 输入大语言模型来进行处理. 各部分中, 角色和任务定义用来奠定任务基调, 使模型输出任务领域相关的专业回答; 协议文本给出了待理解的协议文本描述. 在采用 LangChain 聊天模型这样易于批处理的提示问答方式时, 可由其他部分预先定义协议抽取任务, 该部分放在提示词外单独输入; 事件表示格式定义了协议交互抽取的事件表示形式; 规则约束定义了用于改善模型提取效果的处理规则, 规则表述由表 1 最右两列内容构成; 任务步骤以主动引导的方式指导模型按给定步骤完成任务; 输出格式定义了模型最终的输出形式.

事件表示格式定义了协议交互中各事件的表示形式, 用于规范化提取协议流程中的事件信息, 其主要由 6 部分组成, 如表 2 的第 4 行所示. 形式化地, 一个事件可以表示为 $\text{idx} : \text{condition} \vdash \text{a} \xrightarrow[\text{event}]{\text{message}} \text{b}$, 其中, a 和 b 表示事件的交互主体; idx 为标识流程中事件的序号; event 对应交互主体 a 和 b 之间发生的事件; condition 表示事件执行的前

© 中国科学院软件研究所 http://www.jos.org.cn

提条件; message 代表事件 event 中传递的消息或数据, 若不存在数据传递, 则用“无”表示. 根据该事件格式要求, 大语言模型会输出如“UE->1$$发送初始注册请求$$无$$用户 ID->AP1”的事件信息, 表示第 1 号事件中 UE 会在“发送初始注册请求”这一事件中发送“用户 ID”至“AP1”.

表 2　本文提示模板的语言模式

序号	组成部分	语言模式
1	角色	你是一个善于处理文本协议交互抽取的专家
2	任务定义	文本协议交互抽取的目标是从用户输入的自然语言形式的文本描述中提取有关协议交互的信息…
3	协议文本	{protocol_text}
4	事件表示格式	事件表示格式G: a->idx$$event$$condition$$message->b a: 交互主体 b: 交互主体 idx: 事件序号, 默认从1开始, 顺序递增 event: 交互主体a和b之间发生的事件 condition: 事件发生的前提条件 message: 事件中发送的消息或数据
5	规则约束	{表1中的R1–R8规则的Prompt表达}
6	任务步骤	{任务步骤描述}

7	输出模板	{
		“event-flow”: {
		“events”: [
		“符合事件表示格式G的提取结果”,
		“符合事件表示格式G的提取结果”,
		…
		],
		“flow”: “按照如下形式输出协议流程中各事件的顺序关系, 不要有多余的解释和说明: 事件间顺序关系如下【1->2->3->4, 4->5->6, 4->7->8】”
		},
		“rule_validation”: “检查event-flow/events中的事件提取结果是否满足协议交互抽取任务中定义的所有规则约束”
		}

需要说明的是, 上述事件格式侧重协议流程中的外延性的事件交互, 可用于构建事件流图或消息序列图等, 描述协议中各主体之间的互动, 服务于测试用例设计等任务. 图 3 展示了大语言模型抽取协议交互的图形化表达, 其内容和文本性的输出结果等价.

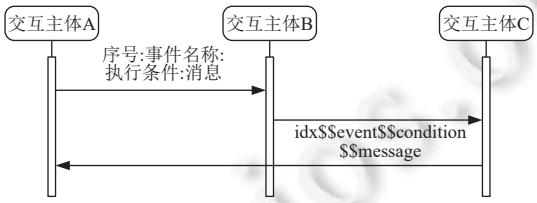


图 3　协议交互抽取结果的图形化表达示例

此外, 本文也支持提取协议内涵性的状态转移信息, 构建协议状态机, 用于协议相关性质的验证等. 尽管侧重点不同, 但两者的表示形式极为相似, 都可以用类似“A-M-B”的形式进行表示. 对于事件交互信息, 其中 A、B 为交互主体, M 为 A 和 B 之间发生的事件; 而对于状态转移信息, 其中 A、B 为状态, M 可理解为 A 向 B 转换的条件或需要执行的操作. 限于篇幅, 本文重点阐述事件流的抽取, 仅需适当调整提示模板即可用于提取协议流程中的状态转移信息.

在任务步骤部分, 为了增强模型的可控性, 明确列出模型需要按照哪些步骤进行处理, 这对于确保模型严格遵循给定的规则或标准十分重要, 避免模型自行思考处理流程时偏离正确的路径. 具体地, 本文在提示中制定了以下

© 中国科学院软件研究所 http://www.jos.org.cn

操作步骤来指引大语言模型分析: ① 详细分析协议描述文本, 理解协议交互流程; ② 按照给定的事件表示格式以及规则约束提取协议描述中的所有事件信息; ③ 分析协议交互流程中各事件间的时序和逻辑关系; ④ 按照给定的输出格式返回最终的结果.

输出格式部分主要定义了模型在文本协议交互抽取阶段的预期输出形式, 为便于后续的分析与处理, 本文采用 JSON 形式的结构化输出, 协议提取结果存放在“event-flow”中. 其中, “events”存放协议交互中具体的事件信息, “flow”存放事件之间的顺序关系. 模型会按照提示中描述的格式要求进行输出. 本文的思维链中也会给出输出样本示例, 进一步加强对输出格式的引导.

规则回溯思维链部分结合上下文示例与规则推理来引导模型进行文本协议交互抽取, 尤其是处理具有不同文本表达特征的协议文本时, 其通过融合典型案例与规则引导, 展示了在处理具有特定文本表达特征的协议文本时, 如何根据相应的规则及协议上下文, 实现对协议事件提取的约束与校正, 为模型提供了不同文本表达特征下的任务处理范式.

4.1 典型示例抽取与规则提炼

为了有效利用协议描述中的语言表达特征, 本文在特征分析的基础上, 分别进行典型案例的选取和处理规则的提炼. 首先, 从协议文本表达特征出发, 提炼出用于提高文本协议交互抽取效果的处理规则, 并凝练为适用于大语言模型理解的 Prompt 表达, 从而将协议描述中的语言表达特征融入基于大语言模型的协议交互抽取框架. 具体规则如前文表 1 的第 4 列所示.

典型案例来源于调研数据集, 以原始文本中目标协议相关的文本段落的形式存在, 并需要满足以下条件: (1) 典型案例的协议流程不能过短或过长, 协议交互过程的数量控制在 5–10 之内, 目的是确保案例的有效性, 过短的案例包含的信息过少, 过长的案例可能包含较多复杂信息, 影响模型学习的效果; (2) 典型案例应完全自包含, 不涉及对外部的引用, 且描述过程清晰、无歧义, 目的是确保案例的质量和可用性; (3) 典型案例应来源于相对陌生的语料, 避免使用特别常见的协议作为示例, 以减少模型在学习过程中对先验知识的依赖, 突出对案例本身语义知识的学习.

4.2 规则回溯思维链

基于提取到的典型案例和处理规则, 本文提出了一种规则回溯思维链技术来优化协议交互的抽取, 其通过融合典型案例与规则推理, 对文本协议交互抽取过程进行拆解, 分解协议交互流程中各事件的要素, 并根据处理规则逐个检查各候选答案的可信度, 优化表达欠佳的结果, 实现对流程中事件的精确提取.

如图 4 所示, 规则回溯思维链由关于一系列典型案例的解析构成. 每个典型案例的解析包括案例协议文本描述、任务推理与规则回溯、事件关系总结和模型最终输出这 4 部分. 图 4 中, 1 和*表示数量对应关系, 1-1 表示一对一, 1-*为一对多.

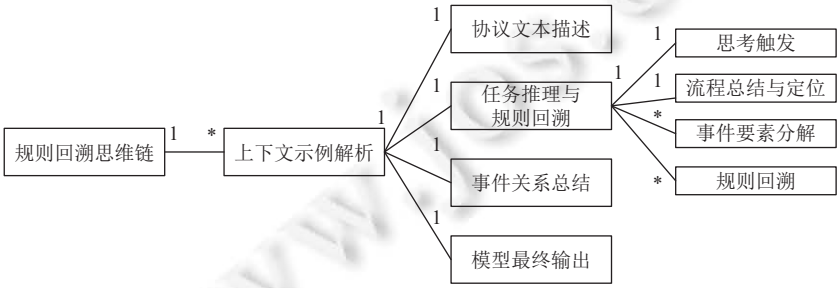


图 4 规则回溯思维链的构成

任务推理与规则回溯部分的核心目标是通过明确的推理过程和规则的回溯确认, 逐步从输入的协议文本中分解各事件的要素并完成纠正. 在推理过程中, 首先要求模型根据事件自身信息对其进行拆解, 获取各要素的候选答案. 在此基础上, 进一步要求模型根据定义的处理规则并结合协议上下文去回溯推理过程, 检查推理过程是否严谨、

© 中国科学院软件研究所 http://www.jos.org.cn

解析获取的各要素候选答案是否存在命名不一致等冲突. 对于违背规则定义的推理过程和候选答案, 要求模型参照给定的规则重新推导并解析事件, 修正事件各要素的候选答案. 通过这些规则, 系统在提取协议事件时能够根据协议上下文优化提取结果, 以实现对不同文本表达特征的针对性处理, 准确识别协议流程中各事件的关键要素.

整个任务推理与规则回溯部分包含<思考触发><流程总结与定位><事件要素分解>与<规则回溯>这 4 大核心环节. 如图 5 中第①部分所示, 思考触发常以“让我们逐步分析这个问题”等引导式术语触发模型对任务流程的思考.

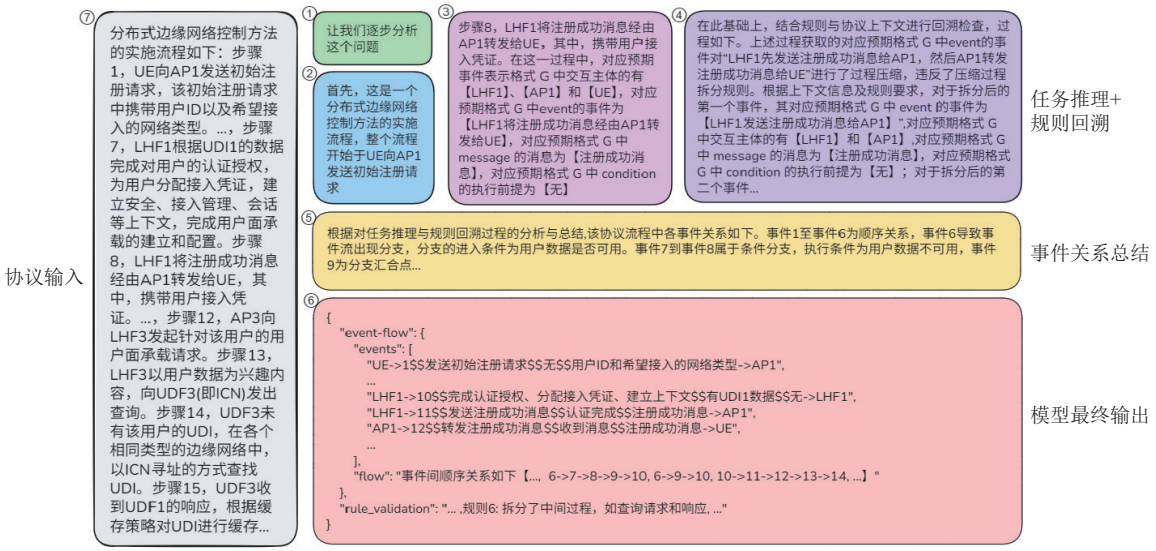


图 5 规则回溯思维链中一个典型上下文示例的解构内容

流程总结与定位负责概括流程的目标, 以作为协议流程解析的基本背景, 并尝试定位流程的起始. 如图 5 中第②部分“这是一个分布式边缘网络控制方法的实施流程, 整个流程开始于 UE 向 AP1 发送初始注册请求”, 介绍了协议的目标功能, 并说明其流程从“UE 向 AP1 发送初始注册请求”开始. 这一内容提示模型从自身知识库中获取有关该协议业务背景的先验知识, 并能够跳过协议开头部分可能存在的流程无关的说明性信息, 减少对提取结果的干扰.

事件要素分解负责逐个剖析协议流程中的事件, 识别事件的交互主体、事件前提等关键因素并施加标记, 通过【事件要素】的标记方式明确事件中各要素的候选答案. 例如, 在图 5 中第③部分“LHF1 将注册成功消息经由 AP1 转发给 UE, 其中, 携带用户接入凭证”这段描述中, 经初步分析, 对应预期事件表示格式 G 中交互主体的有【LHF1】、【AP1】和【UE】, 对应预期格式 G 中 event、message 和 condition 的事件、消息和执行前提分别为【LHF1 将注册成功消息经由 AP1 转发给 UE】、【注册成功消息】和【无】. 事件要素分解主要基于单一事件本身的信息开展, 未考虑上下文中整个事件流, 也不应用本文总结的处理规则, 答案可能存在不合理之处, 这些问题将在规则回溯中加以解决.

规则回溯是本文针对普通思维链对模型推理缺乏强有力指导的改进. 其在解析事件、获得各事件要素的基础上, 尝试应用从文本描述特征总结的处理规则, 直观展示在进行协议事件提取时如何根据规则及协议上下文来修正和优化提取结果, 为大语言模型提供针对不同文本表达特征的处理范式. 以图 5 中所示例的协议为例, 在事件要素分解中, 对应预期格式 G 中 event 的事件将“LHF1 先发送注册成功消息给 AP1, 然后 AP1 转发注册成功消息给 UE”进行了压缩表达. 思维链中首先判断该提取结果违反了压缩过程拆分规则, 然后按照规则要求对这一事件进行拆分, 进一步的, 为拆分后的各个事件确定其各要素. 例如, 对于拆分后的第 1 个事件, 其对应预期格式 G 中 event 的事件为【LHF1 发送注册成功消息给 AP1】, 交互主体有【LHF1】和【AP1】, message 的消息为【注册成功消息】. 限于篇幅, 图 5 案例中第④部分仅展示了一条规则的回溯引导过程, 实际过程中同一事件可能会触发更多的规则回溯, 此处不再一一赘述.

© 中国科学院软件研究所 http://www.jos.org.cn

事件关系总结部分负责对任务推理与规则引导生成的内容进行总结分析, 主要目标是根据推理过程梳理协议流程中各事件之间的顺序关系, 尤其是当流程中出现分支时. 其主要包含顺序事件陈述与流程分支解析两大内容. “顺序事件陈述”串联协议交互流程中连续发生的顺序事件, 对于顺序的连续过程, 要求模型使用“事件 X 至事件 Y 为顺序关系”的形式进行表示. “流程分支解析”则是对流程中可能出现的分支进行结构解析, 对于分支信息, 要求模型详细说明分支出现的位置、各分支上包含的事件、分支的准入条件、分支结构的总结以及可能的分支合并位置. 如“事件 Z 导致流程出现分支”“事件 W 属于第 1 分支, 分支的准入条件是…”等. 这些信息直观地阐述了分支中不同路径的信息, 最终经过汇总得到输出中的格式化的顺序关系表达, 如【…, 6->7->8->9->10, 6->9->10, 10->11->12->13->14,…】等, 方便后续算法读取.

模型最终输出部分主要展示协议交互抽取任务的预期输出格式, 其格式与表 2 中第 7 行的输出格式一致, 采用 JSON 形式表达, 用于在示例中强化模型的结构化输出能力.

4.3 多路推理与自我验证

多路推理引导大语言模型沿多个独立的路径进行思考, 最后对这些路径的结果进行整合, 以此来提升复杂任务的推理准确性和鲁棒性. 自我验证在大语言模型得到初步结论后, 反向验证其答案的合理性, 从而确保模型输出的质量. 本文借鉴了这两种思想, 并提出了一套针对协议文本抽取的融合验证规则. 其主要思想是并行执行同一协议交互抽取任务得到多个不同路径的输出, 根据不同路径的输出结果、协议交互抽取任务描述以及原始协议文本, 引导大语言模型遵循给定的融合规则实现对多路输出的统一, 从结构合理性、语义完整性以及时序一致性等角度验证融合后的结果, 修改不合理的答案并补充遗漏的信息, 得到最终的输出.

这一过程可抽象为:

$$result = FV([R_1, R_2, \ldots, R_n], T, I),$$

其中, *result* 代表融合验证后的输出, *FV* 代表融合验证操作, 通过大语言模型实现, R_1, R_2,…, R_n 表示多路推理中不同执行路径的输出, *T* 表示协议交互抽取任务的详细信息, 包括任务的定义、描述、要求等, *I* 表示用于协议交互抽取的原始文本. *FV* 以零样本学习方式实现, 其提示模板如表 3 所示.

表 3 融合验证提示模板

序号	组成部分	语言模式
1	角色	你是协议交互抽取领域的融合验证专家
2	任务定义	根据协议交互抽取任务的相关描述以及原始协议文本, 严格按照给定的融合验证规则, 对同一任务的多个不同输出进行融合和校验
3	协议交互抽取任务	{task}
4	原始协议文本	{protocol_text}
5	多路协议抽取结果	{results}
6	验证规则	1. 事件交互主体, 即提取结果中/event-flow/events下各事件中的交互发起方和目标方, 必须表达协议中真实存在且具体到子对象的实体, 不可存在同一主体具有不同名称的情况, 且实体提取无遗漏 2. 事件执行前提, 即提取结果中/event-flow/events下各事件中的*condition*部分, 必须为协议描述中事件发生的充分、必要条件, 不能遗漏, 若不存在则用“无”表示 3. 事件传递的消息, 即提取结果中/event-flow/events下各事件中的*message*部分, 对于交互型事件, 必须准确提取不同交互主体间所传递的消息, 不能遗漏, 对于自交互型事件默认用“无”表示, 不涉及任何消息传递 4. 协议交互抽取结果中必须包含协议文本中提及的所有事件, 各事件必须能够从初始事件可达, 不应该存在与流程无关的孤立事件 5. 事件交互顺序, 即提取结果中/event-flow/flow下的流程描述, 必须有正确的时序和因果顺序, 不能出现后发生的事件出现在先发生事件之前的情况 6. 规则满足性, 即提取结果/event-flow/events下各事件的要素信息必须满足协议交互抽取任务中定义的处理规则, 不能出现事件压缩或合并表达, 不能将事件的执行条件错误地提取为独立事件
7	融合规则	综合分析不同路径的协议交互抽取结果, 根据协议交互抽取任务要求与原始协议文本, 对不同路径的输出结果进行统一, 融合成一个符合任务要求和预期输出的协议交互抽取结果

© 中国科学院软件研究所 http://www.jos.org.cn

融合验证提示模板由角色、任务定义、协议交互抽取任务、原始协议文本、多路协议抽取结果、验证规则和融合规则组成. 其中, 角色和任务定义部分为模型提供基础的任务信息. 协议交互抽取任务{task}和原始协议文本{protocol_text}以及多路协议抽取结果{results}为提示模板的动态参数, 由问答框架在运行过程中动态填充, 用来驱动整个融合验证模块的运行.

验证规则主要从结构合理性、语义完整性和时序一致性角度出发, 定义了协议交互抽取结果验证的一些关键要求.

● 结构合理性要求主要体现在规则 1−3、6 中, 关注提取结果中各事件节点 (交互主体) 和边 (事件), 这些要素是组成事件流的关键部分. 对于每一个节点, 都需要确认其是否在协议中真实存在, 是否应该作为事件的交互主体. 对于每一个边, 都需要确认其是否是协议流程中的关键过程, 是否在协议描述中真实存在, 事件包含的子要素信息, 如事件执行条件、事件传递消息等是否正确提取.

● 语义完整性要求体现在规则 1−4 中, 关注协议交互过程是否完整体现. 具体的, 需要检查协议描述中是否存在部分事件 (边) 被遗漏, 以及是否存在事件中交互主体 (节点)、执行条件等关键要素被遗漏.

● 时序一致性重点关注协议流程中各事件的发生顺序, 确保不会出现顺序颠倒和因果逆转的情况, 主要体现在规则 5 中. 协议流程中各事件之间存在时序或因果层面上的先后关系, 这种先后关系决定了事件的执行顺序, 关乎协议流程的正确表示, 必须保证时序一致.

融合规则侧重于为模型提供具体的融合步骤, 即先统一各路径结果, 再生成最终的答案.

在多路推理与自我验证模块中, 尽管一些路径的推理结果可能部分偏离了正确答案, 但本文依然认为不同的回答对模型获取高质量的正确输出和避免可能的错误存在一定的参考价值. 自我验证机制在多路推理的基础上进一步增强了对模型输出的质量把控, 以此来尽最大努力规避不合理的输出结果, 减少遗漏和误判.

5 实验评估

本文尝试回答以下实验问题来评估所提协议交互抽取方法的有效性.

RQ1: 本文方法从自然语言文本描述中提取协议中交互事件的准确性如何?

RQ2: 本文方法从自然语言文本描述中提取协议中事件间顺序关系的准确性如何?

RQ3: 本文方法中规则回溯思维链、多路推理和自我验证对协议交互抽取效果的影响如何?

5.1 数据集和评价指标

(1) 数据集

实验选择 RFC 数据集和一个新构建的面向应用的中文协议描述数据集 CPP 作为协议交互抽取对象. RFC 文档描述了众多协议的规格, 其中的协议是协议交互抽取任务常采用的实验对象[5,7,9,12,14,15]. 在文本协议交互抽取领域, 多个工作也以 RFC 协议文档为数据集[9,12,15], 因此本文选其作为实验数据集之一. 在文献 [9,12] 中公开了针对 TCP 和 DCCP 两个协议从文本抽取的结果, 为便于比较, 本文主要以 RFC 中的这两个协议为实验对象.

本文方法的主要应用对象是那些面向领域应用的自定义协议, 抽取其交互以支持验证和测试等活动. 因此, 也选择此类协议作为实验对象. 如第 3 节所述, 本文从国家知识产权系统 (CNIPA) 检索来自专利的协议描述文本作为样本构建 CPP 数据集, 专利中的协议描述除详实具体外, 常附带相应的协议交互图等, 可作为供校验的协议交互抽取正确结果. 具体地, 本文在第 3 节抽取调研样本的基础上, 按检索标准继续向后筛选出 50 个不同样本用于检验文本协议交互抽取的效果. 相对于仅用于调研, 制作 CPP 数据集的样本必须进一步满足以下几点: (1) 案例中包含交互过程, 且交互对象不少于 2 个、交互事件不少于 3 个; (2) 每个案例必须配有相应的交互图作为标准答案, 且案例的文本描述与图形表达语义基本一致, 图形中错漏的事件数量少于 3 个, 且可由实验人员在评估前修正; (3) 案例必须保证自包含, 不存在以“见××部分”等外部引用形式省略表达的事件.

最终所用的 RFC 数据集中包含 23 个状态和 54 个状态迁移, 平均样本长度约 3 800 字. CPP 数据集中包含 574 个交互事件, 平均样本长度约 475 字.

© 中国科学院软件研究所 http://www.jos.org.cn

(2) 评估指标

在案例数据集上, 如表 4 所示, 本文选择精确率、召回率、错误率以及正确率作为实验的评价指标. 其中, 精确率、召回率和错误率用来评估 RQ1 交互事件 (或状态转移) 的抽取准确性, 错误率为误报和漏报交互事件数量比上预期交互事件数, 可直观衡量总的协议交互提取错误; 正确率用来评估 RQ2 交互顺序的抽取准确性, 以协议文本中表达的所有先后和分支顺序为标准答案, 如果交互事件 A 和 B 之间存在一个先后或分支顺序, 而抽取结果中 A 和 B 得到提取, 且 A、B 之间的关系被准确构建, 则视为获得一个正确顺序. 所有指标均从整个案例集的角度计算, 即考察所有案例中总事件 (或状态转移) 集的处理准确性, 这能够考察总体上对每一个交互事件的成功解析概率. 在计算精确率和召回率时, 对一个事件, 除完全正确地被提取, 本文还放宽条件, 考察其部分正确提取条件下各方法的表现, 因为部分正确对于许多后续应用也有价值. 完全正确指提取的交互主体/状态及事件中的执行前提、传递的消息完全正确, 部分正确表示提取的交互主体完全正确, 但允许事件中的部分要素错误, 其余情况均视为提取错误.

表 4 评价指标

类型	计算方式
精确率	(所有案例中抽取的正确协议交互事件的数量/所有案例中提取为协议交互事件的数量)×100%
召回率	(所有案例中抽取的正确协议交互事件的数量/所有案例标准答案中协议交互事件的数量)×100%
错误率	(所有案例中误报和漏报交互事件的数量/所有案例标准答案中协议交互事件的数量)×100%
正确率	(所有案例中正确的顺序数/所有案例标准答案中的顺序数)×100%

在评估某一事件处理正确性的过程中, 由于对事件中实体名称等的抽取并非只有唯一答案, 语义等价的描述均可视为正确, 因此本文在交互主体、事件执行前提和事件传递消息的方面, 采用人工来评判结果, 安排 3 名不同的实验人员分别为每个协议统计提取结果中完全正确、部分正确或错误事件的数量. 在此基础上, 以完全正确的事件数量为排序标准, 从 3 份统计结果中选择排序居中的一份结果, 用来计算表 4 中的指标. 此外, 本文实验重点关注协议流程中交互事件 (或状态转移) 的抽取效果, 交互主体 (或状态) 直接影响上述因素的正确与否, 能够在最终的指标中得到体现, 因此实验不再单独评估交互主体 (或状态) 的提取情况.

5.2 基准方法

本文选取两个具有代表性的、最新的、提供 RFC 数据集上文本协议交互抽取结果的工作 RFCNLP[9]和 PROSPER[12]作为实验的基线方法. 其中, RFCNLP 是基于深度学习的文本协议交互抽取的代表性工作. PROSPER 使用大语言模型进行文本协议交互抽取. Hermes[10]方法也进行文本协议交互抽取, 但其主要针对 4G、5G 蜂窝协议, 部分实现依赖于蜂窝协议相关特性. 尽管其在个别模块的验证中使用了 RFC 文档作为测试对象, 但未进行完整的协议交互抽取, 也没有提供 RFC 相关的模型和标记数据来支撑抽取, 无法用于 RFC 文档上的文本协议交互抽取比较, 因此本文未将其作为基线工作进行对比.

在具体的实验中, 本文选择 RFCNLP 文献中最好的实验配置, 用于测试 TCP 和 DCCP 协议的提取效果. 对于 PROSPER, 本文参照论文描述对其进行了复现, 选择论文中表现更优的贪婪策略作为提示构造方法, 以此作为实验基线工作. PROSPER 还提出 Artifact Miner 工具用来从协议文档中文本字符绘制的图表中提取协议交互信息, 但该类型的图表仅限于 RFC 等纯文本格式文档, 许多应用性协议的流程描述也缺乏对应的图表. 本文主要关注从纯自然语言文本中提取协议交互信息的问题, 暂不考虑直接解析时序图、事件流图等现成图表来理解协议流程, 在实验中不启用 Artifact Miner.

5.3 实验方法

对于 RQ1, 本文首先在 RFC 数据集上比较了本文方法和两个基线方法的效果. RFCNLP、PROSPER 主要支持抽取有限状态机, 本文方法如第 4 节所述, 也可用于抽取状态机, 因此主要比较抽取状态机的效果. 实验的输入均为自然语言形式的协议描述文本, 比较输出状态机的状态迁移来评估效果. 考虑到基于大语言模型的方法输出

© 中国科学院软件研究所 http://www.jos.org.cn

存在不稳定性, 对此类方法, 本文分别运行 3 次, 由一位实验人员按照完全正确事件的数量, 选择一个表现居中的结果作为最终的实验结果.

本文同时在 CPP 数据集上开展了比较, RFCNLP 在 RFC 协议上训练, 主要针对 RFC 协议, 可扩展性和迁移性受限, 很难用到 CPP 协议数据集上. 因此实验仅与基于大语言模型的 PROSPER 方法比较效果. 该方法具有良好的扩展性, 本文按其论文中的提示构造策略, 将 PROSPER 迁移并应用到了 CPP 数据集上, 并要求输出与本文相同格式的协议事件流来进行结果比较.

对于 RQ2, 实验在 CPP 数据集上比较本文方法与 PROSPER 方法在正确提取协议交互顺序上的表现. 对于 RFC 数据集上的协议状态机抽取, 状态机的转移顺序本身已经体现在状态迁移边中, 并无额外结果来描述状态转换顺序, 其转换顺序本质上已在 RQ1 中考虑, 因此在 RQ2 中不再重复评估.

对于 RQ3, 本文在 CPP 数据集上对所提方法开展消融实验来评估方法中各关键环节对文本协议交互抽取效果的影响. 消融实验分别选择激活和不激活相关功能来获得协议解析结果. 如果不激活规则回溯思维链, 则使用简单的协议文本-预期答案组合作为上下文示例; 如果不激活多路推理, 自我验证模块直接处理协议提取结果, 得到最终输出; 如果不激活自我验证, 则直接以多路推理模块融合后的输出作为最终答案.

为考察本文方法在不同大语言模型下的表现, 实验在 DeepSeek-R1 和 ChatGPT-4o 上开展了分析. 这两个模型未必最强, 但基本能够代表当前大语言模型的前沿水平. 实验基于 Python 和 LangChain 框架实现, 并选择聊天模型作为提示问答方式, 相较于传统大语言模型问答, 聊天模型实现上更加灵活. 实验中多路推理采用 3 路方案. 在消融实验中, 为便于实验, 本文将融合验证模块拆解为融合和自我验证两个独立模块, 其功能上等价于融合验证模块. 最后, 通过 LangChain 框架实现工作流编排, 根据不同的实验需求灵活组合不同的模块, 实现协议交互抽取.

5.4 实验结果与分析

5.4.1 RQ1: 交互事件抽取准确性

表 5 展示了不同方法在 RFC 协议数据集上的抽取效果对比. 表 5 及后续表格中加粗数据表示最优方法配置的结果. 对于 DeepSeek-R1 模型, 在完全正确和部分正确的评价标准下, 本文方法精确率比 RFCNLP 分别高 48.2% 和 36.8%, 比 PROSPER 高 11.1% 和 10.0%. 其召回率比 RFCNLP 分别高 20.4% 和 5.5%, 比 PROSPER 高 9.2% 和 9.3%. 表明本文方法在精确率方面相对现有工作提升明显. 对于 ChatGPT-4o 模型, 在完全正确和部分正确的评价标准下, 本文方法精确率比 RFCNLP 分别高 25.7% 和 33.5%, 比 PROSPER 高 12.5% 和 12.5%; 召回率比 RFCNLP 分别高 5.5% 和 1.8%, 比 PROSPER 高 7.4% 和 7.4%. 同 DeepSeek-R1 模型, 精确率和召回率依旧有明显提升. 两个模型下, 总错误率相较基线方法分别降低达 35.1% 和 14.8%, 说明本文方法有效消减了错误的交互识别结果.

表 5 不同方法在 RFC 协议数据集上的抽取效果对比

方法	正确		误报	遗漏	按完全正确标准 (%)			按部分正确标准 (%)		
	完全正确	部分正确			精确率	召回率	错误率	精确率	召回率	错误率
RFCNLP	15	13	21	26	30.6	27.8	111.1	57.1	51.9	87.0
PROSPER+DeepSeek-R1	21	5	5	28	67.7	38.9	70.4	83.9	48.1	61.1
PROSPER+ChatGPT-4o	14	11	7	29	43.8	25.9	87.0	78.1	46.3	66.7
Ours+DeepSeek-R1	26	5	2	23	**78.8**	**48.1**	**55.6**	**93.9**	**57.4**	**46.3**
Ours+ChatGPT-4o	18	11	3	25	56.3	33.3	72.2	90.6	53.7	51.9

表 6 展示了本文方法与 PROSPER 在 CPP 协议数据集上的抽取效果对比. 对于 DeepSeek-R1 模型, 本文方法在完全和部分正确标准下, 精确率分别比 PROSPER 高 9.0% 和 9.7%, 召回率高 9.6% 和 10.3%, 总错误率降低达 19.0%. 而对于 ChatGPT-4o 模型, 本文方法精确率分别比 PROSPER 高 4.6%、2.8%, 召回率高 5.4%、3.7%, 总错误率降低达 6.1%. 从结果可见, 本文方法在性能更强的模型上有着较好的表现, 各项指标提高近或超 10%, 对于一般模型, 提升幅度略小, 但依然存在 5% 左右优势.

© 中国科学院软件研究所 http://www.jos.org.cn

表 6 本文方法与 PROSPER 在 CPP 数据集上的抽取效果对比

模型	方法	正确		误报	遗漏	按完全正确标准 (%)			按部分正确标准 (%)		
		完全正确	部分正确			精确率	召回率	错误率	精确率	召回率	错误率
DeepSeek-R1	PROSPER	466	8	86	100	83.2	81.2	33.8	84.6	82.6	32.4
	Ours	521	12	32	41	**92.2**	**90.8**	**14.8**	**94.3**	**92.9**	**12.7**
ChatGPT-4o	PROSPER	435	14	105	125	78.5	75.8	42.5	81.0	78.2	40.1
	Ours	466	4	91	104	**83.1**	**81.2**	**34.7**	**83.8**	**81.9**	**34.0**

以上在 RFC 和 CPP 两个实验数据集上的结果表明, 本文所提出的方法在协议状态机抽取和协议交互流程抽取两个方面的准确性均优于实验的基线方法, 尤其是在 DeepSeek-R1 模型下. 相对传统基于深度学习的 RFCNLP 方法, 本文方法精确率和召回率均取得 20% 以上提升, 相对现有基于大语言模型的 PROSPER 方法, 本文方法精确率和召回率提升约 10%, 综合减少超 15% 的误报加漏报. 上述结果证实提炼协议描述的文本表达特征并用其加强对大语言模型问题分析的引导有助于改善协议交互事件的抽取效果.

本文方法的改进主要源于基于规则对协议交互事件及相关参数提取过程的回溯检查, 尤其是针对一些典型的文本表达特征, 依靠更明确的提取标准和处理策略来提高模型抽取的准确性. 其中, 分支处理规则、交互主体命名统一规则、压缩过程拆分规则以及交互主体合并事件拆分规则对实验表现影响较大, 其他的一些规则, 如事件依赖提取规则、流程无关信息过滤规则等在案例中也有体现, 但影响相对较小, 主要用于辅助核心规则做进一步的优化.

虽然本文方法在 DeepSeek-R1 模型上相对基线方法取得了较为明显的改进, 但仍未能实现完全正确和无遗漏的协议交互抽取, 原因包括多个方面. 首先, 文中总结的协议文本表达特征和处理规则, 尽管覆盖了相当一部分协议表述, 但仍可能有遗漏. 另外, 一些难以解决的问题源于自然语言在表达方面的模糊性, 协议编写者限于写作技巧等, 有时出现词不达意的情况, 而这类问题很难通过简单分析文本解决. 针对上述问题, 一方面, 未来需要开展更多协议表达特征的总结, 凝练处理规则或案例等, 来引导大语言模型分析; 另一方面, 可能需要借助其他方面的知识, 或更强大的模型推理能力来解决.

实验中, DeepSeek-R1 模型在各方面的表现都优于 ChatGPT-4o, 推测可能的原因是模型的性质和用途不同. ChatGPT-4o 能够解决日常中的众多问题, 是一款面向通用任务的常规模型. 而 DeepSeek-R1 是当下热门的推理类模型, 尤其善于处理推理性质的任务. 本文提出的方法需要在分析过程中进行基于规则的过程回溯, 这一过程要求模型必须对任务需求和自身输出有清晰的认知. 在此基础上, 还需要能够根据外部信息灵活地做出改进, 对于这一过程, 推理类模型相对常规模型似乎更具优势.

实验结果在 RFC 和 CPP 数据集上存在一定差异. 主要的原因是, 首先, 这两个数据集上的任务存在细微差异, 前者关注协议交互过程中内涵性的状态迁移关系, 而后者侧重于协议流程中外延性的交互事件. 此外, 这两个数据集在规模和语言方面也存在差异, 本文所用的 RFC 数据集规模相对小, 协议描述文本为英文, 而 CPP 数据集包含更多的协议和交互关系, 且采用中文描述. 尽管存在差异, 两个数据集均证实了本文方法的效用.

5.4.2 RQ2: 交互顺序抽取准确性

表 7 展示了本文方法和 PROSPER 方法在 CPP 数据集上交互顺序的抽取准确性对比. 其结果表明, 本文方法在不同模型上的提取表现均高于基线方法. 具体地, 在 DeepSeek-R1 上的表现超出基线工作 12.8%, 而在 ChatGPT-4o 上超出 3.5%. 即从交互事件间顺序的角度, 本文方法也有效提升了协议交互抽取的效果. ChatGPT-4o 上的提升不如 DeepSeek-R1, 其可能的原因已在第 5.4.1 节中讨论.

表 7 CPP 数据集上的交互顺序抽取准确性

模型	方法	正确顺序关系数	总顺序关系数	准确率 (%)
DeepSeek-R1	PROSPER	390	523	74.6
	Ours	457		**87.4**
ChatGPT-4o	PROSPER	370		70.7
	Ours	388		**74.2**

© 中国科学院软件研究所 http://www.jos.org.cn

实验中, 基线方法在准确提取协议中事件方面并不理想, 存在事件误报或遗漏, 这导致协议上下文中大量的前后衔接出现错误, 最终降低了事件间顺序关系的提取准确度. 而本文方法有效地减少了协议抽取过程中的漏报和误报, 尤其是对于含不同分支的情况, 这使得提取结果中上下文的衔接更加完整, 因此顺序关系提取效果得到有效提升.

5.4.3 RQ3: 关键环节影响分析

表 8 展示了本文方法在 CPP 数据集上的消融实验结果. 结果显示, 对于 DeepSeek-R1 模型, 在不使用本文提出的规则回溯思维链条件下, 精确率和召回率指标下滑超过 8.2%, 表明规则回溯思维链效用明显. 而在不激活多路推理和自我验证时各自造成实验的精确率和召回率出现 1.1%–3.0% 的下降, 完全正确标准下总错误率分别降低 3.3%–4.9%, 表明方法虽然影响不如思维链, 但也对最终的算法性能形成贡献. 对于 ChatGPT-4o 模型, 虽然受制于模型性能, 本文方法各环节的影响不如 DeepSeek-R1 模型下大, 但不激活也造成了微弱的协议交互抽取指标退化. 总体来看, 本文提出的规则回溯思维链在整个方法中起主要作用, 多路推理与自我验证亦能带来方法效果的提升, 它们都是支撑协议交互抽取的必要技术.

表 8 消融实验

模型	Type	正确		误报	遗漏	按完全正确标准 (%)			按部分正确标准 (%)		
		完全正确	部分正确			精确率	召回率	错误率	精确率	召回率	错误率
DeepSeek-R1	无规则回溯思维链	474	11	81	89	83.7	82.6	31.5	85.7	84.5	29.6
	无多路推理	504	16	43	54	89.5	87.8	19.7	92.4	90.6	16.9
	无自我验证	511	16	39	47	90.3	89.0	18.1	93.1	91.8	15.3
	应用全部技术	521	12	32	41	**92.2**	**90.8**	**14.8**	**94.3**	**92.9**	**12.7**
ChatGPT-4o	无规则回溯思维链	447	12	99	115	80.1	77.9	39.4	82.3	80.0	37.3
	无多路推理	453	9	97	112	81.0	78.9	38.0	82.6	80.5	36.4
	无自我验证	461	7	95	106	81.9	80.3	36.2	83.1	81.5	35.0
	应用全部技术	466	4	91	104	**83.1**	**81.2**	**34.7**	**83.8**	**81.9**	**34.0**

5.5 实验有效性风险

影响实验结论有效性的风险主要包括 4 个方面. 首先, 内部有效性方面, 实验在 CPP 数据集标准答案的设置、各协议交互抽取结果的准确性统计等方面涉及一些人工因素, 可能有偏差. 尽管如此, 本文每一个协议交互的抽取结果都至少有 3 位实验人员检查, 因此有理由相信结果是恰当的, 实验数据能够表达本文方法的实际表现. 其次, 外部有效性方面, 实验论证的协议仍相对有限, 可能影响方法泛化能力的评估. 尽管如此, 本文不仅在 RFC 协议集上开展实验, 也在 CPP 协议集上开展实验. 这些协议的文本描述编撰自不同作者, 覆盖通信、安全、业务流程等不同领域, 其上的结果能够反映方法的通用性. 其三, 结构有效性方面, 实验仅在 DeepSeek-R1 和 ChatGPT-4o 上论证了方法效果, 在其他大语言模型上, 本文方法的效用可能下降. 尽管如此, DeepSeek-R1 和 ChatGPT-4o 已是当前较为先进的模型, 在其上的效用表明所提方法具有实际的应用价值. 此外, 在统计结论有效性方面, 大语言模型的输出结果可能不稳定, 产生不同的实验表现. 如第 5.3 节所述, 本文通过在一个样本上多次运行协议交互抽取, 取一个居中表现的结果作为指标计算基础, 尽力规避了上述风险. 实验中, 本文方法由于采用了规则的详细指引和多路推理与自我验证, 波动范围相对小, 多轮次运行中, 相对居中表现, 差和好的情况在 DeepSeek-R1 上分别有+1.8% 和−1.0% 的总体错误率波动, 在 ChatGPT-4o 上有+3.1% 和−1.7% 的波动, DeepSeek-R1 的规则理解能力更强, 波动比 ChatGPT-4o 小. PROSPER 方法缺少规则指引和多路推理与自我验证支持, 结果稳定性更差, 在 DeepSeek-R1 上有+3.3% 和−1.2% 的错误率波动, 在 ChatGPT-4o 上有+4.7% 和−1.3% 的波动. 实验中选择了各方法多次运行的居中表现, 能够代表其常规效果. 即使从较差的情况看, 本文方法在 DeepSeek-R1 模型下的表现也好于基线方法较优的表现.

6 总 结

本文针对从自然语言文本来理解协议的问题, 调研并总结了协议文本表述中的常见语言表达特征, 基于对特

© 中国科学院软件研究所 http://www.jos.org.cn

征的分析, 提出了一种增强的基于大语言模型的协议交互抽取方法, 该方法提出一种规则回溯思维链来引导大语言模型推理过程, 同时引入多路推理和自我验证技术来优化基于大语言模型的分析流程. 实验结果表明, 所提出的方法能够提高协议交互的抽取精确率、召回率等多方面指标.

本文所提出的方法可用于协议的验证、测试等多种任务. 尽管实验确认了其有效性, 但该方法仍存在提升空间. 未来将进一步总结语言表达特征和案例、在目前人工定义提示的基础上应用先进自动提示优化技术以及挖掘优化模型推理的外部协议知识等来改进其效果. 同时, 应用方法到具体领域, 来解决协议相关的终端问题.

References

[1] Shi QK, Xu XZ, Zhang XY. Extracting protocol format as state machine via controlled static loop analysis. In: Proc. of the 32nd USENIX Security Symp. (USENIX Security). Anaheim: USENIX, 2023. 7019–7036.

[2] Kothari N, Millstein T, Govindan R. Deriving state machines from TinyOS programs using symbolic execution. In: Proc. of the 2008 Int'l Conf. on Information Processing in Sensor Networks. St. Louis: IEEE, 2008. 271–282. [doi: 10.1109/IPSN.2008.62]

[3] Gopinath R, Mathis B, Zeller A. Mining input grammars from dynamic control flow. In: Proc. of the 28th ACM Joint Meeting on European Software Engineering Conf. and Symp. on the Foundations of Software Engineering. ACM, 2020. 172–183. [doi: 10.1145/3368089.3409679]

[4] Höschele M, Zeller A. Mining input grammars from dynamic taints. In: Proc. of the 31st IEEE/ACM Int'l Conf. on Automated Software Engineering. Singapore: ACM, 2016. 720–725. [doi: 10.1145/2970276.2970321]

[5] Ye YP, Zhang Z, Wang F, Zhang XY, Xu DY. NetPlier: Probabilistic network protocol reverse engineering from message traces. In: Proc. of the 2021 Network and Distributed Systems Security (NDSS) Symposium. NDSS, 2021. [doi: 10.14722/ndss.2021.24531]

[6] Kleber S, van der Heijden RW, Kargl F. Message type identification of binary network protocols using continuous segment similarity. In: Proc. of the 2020 IEEE Conf. on Computer Communications. Toronto: IEEE, 2020. 2243–2252.] [doi: 10.1109/INFOCOM41043.2020.9155275]

[7] Comparetti PM, Wondracek G, Kruegel C, Kirda E. Prospex: Protocol specification extraction. In: Proc. of the 30th IEEE Symp. on Security and Privacy. Oakland: IEEE, 2009. 110–125. [doi: 10.1109/SP.2009.14]

[8] Cui WD, Kannan J, Wang HJ. Discoverer: Automatic protocol reverse engineering from network traces. In: Proc. of the 16th USENIX Security Symp. USENIX Association, 2007. 199–212.

[9] Pacheco ML, Von Hippel M, Weintraub B, Goldwasser D, Nita-Rotaru C. Automated attack synthesis by extracting finite state machines from protocol specification documents. In: Proc. of the 2022 IEEE Symp. on Security and Privacy (SP). San Francisco: IEEE, 2022. 51–68. [doi: 10.1109/SP46214.2022.9833673]

[10] Ishtiaq AA, Sarathi Das SS, Rashid SMM, Ranjbar A, Tu K, Wu TW, Song ZZ, Wang WX, Akon M, Zhang R, Hussain SR. Hermes: Unlocking security analysis of cellular network protocols by synthesizing finite state machines from natural language specifications. In: Proc. of the 33rd USENIX Conf. on Security Symp. Philadelphia: USENIX Association, 2024. 249.

[11] Jiang JY, Zhang XY, Wan CC, Chen HY, Sun HY, Su T. BinPRE: Enhancing field inference in binary analysis based protocol reverse engineering. In: Proc. of the 2024 ACM SIGSAC Conf. on Computer and Communications Security. Salt Lake City: ACM, 2024. 3689–3703. [doi: 10.1145/3658644.3690299]

[12] Sharma P, Yegneswaran V. PROSPER: Extracting protocol specifications using large language models. In: Proc. of the 22nd ACM Workshop on Hot Topics in Networks. Cambridge: ACM, 2023. 41–47. [doi: 10.1145/3626111.3628205]

[13] Wei HY, Chen LG, Du ZJ, Wu YH, Huang HH, Liu Y, Cheng G, Xu FY, Wang LZ, Mao B. Unleashing the power of LLM to infer state machine from the protocol implementation. arXiv:2405.00393v4, 2025.

[14] Zheng MW, Xie DN, Zhang XY. Large language models for validating network protocol parsers. In: Proc. of the 2025 IEEE Security and Privacy Workshops (SPW). San Francisco: IEEE, 2025. 56–64. [doi: 10.1109/SPW67851.2025.00009]

[15] Yen J, Lévai T, Ye QY, Ren X, Govindan R, Raghavan B. Semi-automated protocol disambiguation and code generation. In: Proc. of the 2021 ACM SIGCOMM Conf. ACM, 2021. 272–286. [doi: 10.1145/3452296.3472910]

[16] Qian J, Wang Y, Cheng H, Wei ZX. Scenario model based testing of integrated DDS-based naval mission systems. Ruan Jian Xue Bao/Journal of Software, 2022, 33(5): 1711–1735 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6557.htm [doi: 10.13328/j.cnki.jos.006557]

[17] Zhou YC, Muresanu AI, Han ZW, Paster K, Pitis S, Chan H, Ba J. Large language models are human-level prompt engineers. arXiv: 2211.01910, 2022.

[18] Chen P, Zhang S, Han BR. CoMM: Collaborative multi-agent, multi-reasoning-path prompting for complex problem solving. In: Findings

© 中国科学院软件研究所 http://www.jos.org.cn

of the Association for Computational Linguistics: NAACL 2024. Mexico City: ACL, 2024. 1720–1738. [doi: 10.18653/v1/2024.findings-naacl.112]

[19] He CB, Zou BC, Li X, Chen JS, Xing JL, Ma HM. Enhancing LLM reasoning with multi-path collaborative reactive and reflection agents. arXiv:2501.00430, 2024.

[20] Weng YX, Zhu MJ, Xia F, Li B, He SZ, Liu SP, Sun B, Liu K, Zhao J. Large language models are better reasoners with self-verification. In: Findings of the Association for Computational Linguistics: EMNLP 2023. Singapore: ACL, 2023. 2550–2575. [doi: 10.18653/v1/2023.findings-emnlp.167]

[21] Gero Z, Singh C, Cheng H, Naumann T, Galley M, Gao JF, Poon H. Self-verification improves few-shot clinical information extraction. arXiv:2306.00024, 2023.

[22] Brown TB, Mann B, Ryder N, *et al*. Language models are few-shot learners. arXiv:2005.14165, 2020.

[23] Guo D, Yang D, Zhang H, *et al*. DeepSeek-R1: Reasoning Incentivizing in capability via LLMs learning reinforcement. arXiv: 2501.12948, 2025.

[24] Wei J, Wang XZ, Schuurmans D, Bosma M, Ichter B, Xia F, Chi EH, Le QV, Zhou D. Chain-of-thought prompting elicits reasoning in large language models. In: Proc. of the 36th Int'l Conf. on Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2022. 1800.

附中文参考文献

[16] 钱巨, 王寅, 程浩, 韦正现. 基于场景模型的 DDS 架构一体化舰船任务系统测试. 软件学报, 2022, 33(5): 1711–1735. http://www.jos.org.cn/1000-9825/6557.htm [doi: 10.13328/j.cnki.jos.006557]

作者简介

张伯洋, 硕士生, 主要研究领域为软件分析与测试.

钱巨, 博士, 副教授, CCF 专业会员, 主要研究领域为软件分析与测试.

唐靖然, 硕士生, 主要研究领域为软件分析与测试.

卫依, 硕士生, 主要研究领域为软件分析与测试.

© 中国科学院软件研究所 http://www.jos.org.cn